



Article

A Field-Tested Harvesting Robot for Oyster Mushroom in Greenhouse

Jiacheng Rong, Pengbo Wang *, Qian Yang and Feng Huang

Jiangsu Provincial Key Laboratory of Advanced Robotics, School of Mechanical and Electric Engineering, Soochow University, Suzhou 215123, China; jcrong@stu.suda.edu.cn (J.R.); yangqiansz@outlook.com (Q.Y.); gary_huangf@outlook.com (F.H.)

* Correspondence: pbwang@suda.edu.cn

Abstract: The fully autonomous harvesting of oyster mushrooms in the greenhouse requires the development of a reliable and robust harvesting robot. In this paper, we propose an oyster-mushroom-harvesting robot, which can realize harvesting operations in the entire greenhouse. The two crucial components of the harvesting robot are the perception module and the end-effector. Intel RealSense D435i is adopted to collect RGB images and point cloud images in real time; an improved SSD algorithm is proposed to detect mushrooms, and finally, the existing soft gripper is manipulated to grasp oyster mushrooms. Field experiments exhibit the feasibility and robustness of the proposed robot system, in which the success rate of the mushroom recognition success rate reaches 95%, the harvesting success rate reaches 86.8% (without considering mushroom damage), and the harvesting time for a single mushroom is 8.85 s.



Citation: Rong, J.; Wang, P.; Yang, Q.; Huang, F. A Field-Tested Harvesting Robot for Oyster Mushroom in Greenhouse. *Agronomy* **2021**, *11*, 1210. <https://doi.org/10.3390/agronomy11061210>

Academic Editors: Gert Kootstra and Asher Bender

Received: 15 March 2021

Accepted: 9 June 2021

Published: 15 June 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 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/>).

1. Introduction

The agricultural industry has always been a labor-intensive industry. The reason may be that the crops produced in agriculture usually do not have uniform characteristics, and the agricultural production environment is more complex and diverse than the industrial production environment. Therefore, it is not possible to replicate the factory automation model in the agricultural production environment and use general automated equipment to replace part of the labor.

With the development of IoT, AI, and robotics, the agricultural industry has been driven to shift from labor-intensive to technology-intensive. Applications and solutions that integrate artificial intelligence and robotics are expected to gradually replace manual work in agricultural production activities. These robots, which require less human intervention, can help farmers manage their crops in the future, and they will be able to address the shortage of food and agricultural labor while significantly increasing productivity. However, the robots for selective harvesting of mushrooms, tomatoes, and other crops instead of manual labor also faces several challenges: (1) diverse growing patterns and unstructured environments; (2) the efficiency of using robots for harvesting is much lower than that of manual harvesting; (3) crops have different growth patterns with anisotropy, uneven sparseness, and inconsistency in size; (4) crops such as mushrooms are high in moisture, brittle, and easily damaged [1,2]. The aforementioned challenges make it difficult to deploy the selective automatic harvesting in practical production.

In this paper, we developed a greenhouse mushroom-harvesting robot that aims to automate oyster mushroom harvesting instead of humans. Compared with manual harvesting, robots can pick around the clock and are not affected by the high temperature and humidity of the greenhouse, which improves the economic efficiency of the farm. Considering the specific planting agronomy and greenhouse environment of the mushrooms, we focus on solving the oyster mushroom recognition and harvesting efficiency

problem. In order to solve the problem of low picking efficiency, four harvesting units are connected in parallel on the mobile platform, and these four harvesting units work at the same time, thereby effectively improving the overall picking efficiency of the robot. In terms of the accuracy of recognition and localization, we have achieved some advantages over traditional machine learning methods using a deep learning-based approach [3,4].

The rest of this paper is organized as follows: In Section 2, we introduce related work. In Section 3, we describe the working scene of the harvesting robot in the oyster mushroom greenhouse. In Section 4, we show the structure design and system control of a mushroom-harvesting robot. In Section 5, we present the mushroom recognition and localization methods. In Section 6, we conduct relevant experiments to verify the feasibility of the mushroom-harvesting robot. In Section 7, results and discussion are presented. In Section 8, we show the conclusions of the paper.

2. Related Work

Currently, many scholars have conducted some research related to agricultural harvesting robots. Agricultural harvesting work requires robots to work in an unstructured environment, which poses great challenges to the robotic visual recognition system and end-effector. The vision system gives the harvesting robot the perception, and dedicated end-effectors enable the robot to pick specific fruits. Existing work has conducted automated harvesting studies on asparagus, lettuce, peppers, litchi, tomatoes, mushrooms, etc.

An autonomous asparagus-picking robot was developed for the selective harvesting of green asparagus [5]. The robot can travel along the asparagus dam in the field to detect asparagus stems and identify those that can be harvested. An RGB-D camera was used to simultaneously acquire RGB images and point cloud maps, cluster the asparagus stem points, and determine the asparagus location based on the size of the clustered generated stems, and a specialized end-effector was designed for harvesting.

To automate the picking of mature lettuce in a natural environment, an automated lettuce-picking robot has been developed [6]. To distinguish between immature lettuce, mature lettuce, and lettuce affected by pests and diseases, the images collected by the top-mounted camera are transmitted to the YOLOv3 network for coarse classification and localization of the target in the entire image, and the border frames detected in the first network are cropped and transmitted to the darknet network for fine classification. The camera located on the end-effector, on the other hand, can fine-tune the position of the end-effector during the capture. In the field trial, the vision system detected 69 lettuces, 60 of which were in the working range of the robot arm, and successfully picked 31 lettuces with an average time of 31.7 s for a single pick.

A bell pepper-picking robot was developed for autonomous working in a greenhouse [7,8]. The robotic end-effector includes an RGB-D camera, lighting, a vibrating knife, and metal fingers with a soft plastic cover. The vision system uses a detection algorithm based on shape and color adaptive thresholds, and when the fruit is detected, the end-effector approaches the fruit and determines the position of the stem relative to the bell pepper, and a vibrating knife on the end-effector cuts the stem. The bell pepper is grasped by a finger mounted on the end-effector, and finally, the picked bell pepper is placed in the fruit basket. The average time for a single pick was 24s and the success rate was 61% for a given variety of bell pepper.

A nocturnal lychee fruit and pedicel detection method was proposed for the detection of lychee fruit in the night environment [9]. The method is based on YOLOv3 for the detection of litchi fruit in the natural environment at night, and the region of interest where litchi fruit pedicels are present is determined based on the predicted litchi fruit boundary frame. Finally, based on the U-Net semantic segmentation network, the pixel semantic segmentation of litchi fruit pedicels is performed to determine the pixels belonging to the category of fruit pedicels, to achieve the detection of litchi fruit and fruit pedicels at night.

Huang et al. proposed a picking robot for picking most stemmed crops [10]. The proposed vision system of the picking robot can locate the appropriate cut points on the

crop stalk. The crop pixel area is segmented using the deep learning instance segmentation network Mask RCNN to obtain the minimum external rectangle of the crop edge. Finally, the region above the crop is selected as the region of interest, and a geometric model is built to obtain the shear point.

By improving the YOLO object detection algorithm for real-time detection of mangoes in orchards [11], Koirala argued that the convolution and pooling layers used in YOLO gradually decrease in spatial size as the network depth increases, making it difficult to detect small target objects such as mangoes at lower resolutions. Therefore, deep layers of semantic information were connected to shallower layers of fine-grained information for better detection of small targets, and the model depth was reduced from 107 to 33 layers. When detecting individual tomato fruits, Liu et al. [12] were able to prevent the non-maximal suppression algorithm from suppressing blocked fruits by improving YOLOv3 and replacing the output rectangular bounding box with a circular bounding box when detecting blocked fruit. Li et al. [13] presented an improved YOLO-V3 real-time apple detection algorithm for detecting apple growth status and yield estimation in orchards. The DenseNet feature extraction network was used to replace ResNet53 to enhance image feature propagation and feature reuse and improve network performance. The model can effectively detect apples that are partially overlapping and shaded by branches and leaves under different lighting conditions.

Some scholars have also researched autonomous mushroom harvesting and developed mushroom-harvesting robots for indoor mushroom harvesting. Reed et al. [14] developed a mushroom-harvesting robot experimental platform including vision system, camera driver, end-effector, picking robot, conveyor, trimming device, and frame. When the camera was above the target mushroom area, the image was acquired and the target mushroom was positioned. The end-effector of the picking robot performed the harvesting operation and placed the mushrooms on the clamping conveyor, while the trimming device cut the mushroom handles, and the conveyor put the mushrooms in the container. In trials on a commercial farm, the picking success rate was over 80%. Masoudian et al. [15] studied the vision system of a mushroom-harvesting robot, which was finally able to automatically distinguish between healthy and unhealthy mushrooms. The method used support vector machines and SIFT feature extraction with a classification accuracy of 90%. Lu et al. [16] carried out a study on image recognition of dicotyledonous mushrooms at different growth stages in the greenhouse. The mushrooms were first selected using the YOLOv3 object detection algorithm, and then the circle diameter and center coordinates of the cap were calculated in conjunction with a designed scoring penalty algorithm. The scoring penalty algorithm consists of five steps: color quantization, edge object removal, contour detection, center point search, and fine-tuning.

Compared to traditional machine learning methods, object detection algorithms based on deep learning have greatly improved the accuracy and robustness of recognition. The detection of oyster mushrooms in the greenhouse also faces problems such as mushroom adhesion, size variation, and light changes. At the same time, the vision system of a picking robot should not only focus on the accuracy of detection but also fully consider the detection speed to improve the overall efficiency of the robot. Therefore, this paper focuses on the study of object detection algorithms based on deep learning. Generally, object detection models based on convolutional neural networks are roughly divided into two types: one-stage models and two-stage models. The one-stage model refers to a model that does not independently and explicitly extract candidate regions and directly outputs the category and location information of the object, such as SSD [17] and YOLO [18–20] series models. The two-stage model first extracts some areas where objects may exist and then judges whether there are objects in each candidate area, such as Faster R-CNN [21] and Mask R-CNN [22]. The one-stage model has advantages in computational efficiency, and the two-stage model has better detection accuracy [23]. Due to the real-time requirements for the recognition of mushroom-harvesting robots, we prefer the one-stage detection model for mushroom identification and localization in the greenhouse.

In addition, mushroom-harvesting robots designed by Reed et al. [14] and Hu et al. [3,4] are both single Cartesian coordinate robots with only one harvesting unit, which are inefficient overall. One way to improve harvesting efficiency is to deploy multiple harvesting robots in a greenhouse, however, it will significantly increase the total cost. Additionally, it is another challenge to control the robot cluster to work together. In order to improve the efficiency of the mushroom-harvesting robot, a mushroom-harvesting robot with multiple harvesting units connected in parallel on a common mobile platform has been designed considering the actual greenhouse scene layout and total cost.

3. Operational Environment

As shown in Figure 1, each greenhouse is about 42 m long and 8 m wide, with a 1.2 m wide walkway in the middle. On both sides of the walkway are standard planting areas, and each planting area is composed of cuboid bags being used for cultivating oyster mushrooms (14 bags in the transverse direction and 4 bags in the longitudinal direction, which constitutes a growing area approximately 3.2 m long and 0.8 m wide). Mushrooms are suitable for growing in a dark environment and only natural light projects from the greenhouse gate, which creates uneven lighting in the greenhouse. Temperature, humidity, and carbon dioxide are controlled to achieve a suitable growth environment for oyster mushrooms in the greenhouse [24]. In order to allow the picking robot to move smoothly in the greenhouse, additional tracks are laid in the middle and on both sides.



Figure 1. A greenhouse for growing oyster mushrooms.

As shown in Figure 2, the bag is used for cultivating mushrooms, with growth holes on the top for fruiting. Through the measurement and statistics of oyster mushrooms that are suitable for picking, the size of oyster mushrooms that can be picked is usually 8–16 cm. During the field inspection, we also found that the mushrooms in multiple cultivation bags may stick to each other and the cultivation bags are aging (see Figure 2b,c).

The mushroom-harvesting robot needs to fully consider the above factors. The robot should not only harvest in different light environments but consider different mushroom sizes. Ultimately, harvesting robots need to achieve autonomous navigation, real-time detection of oyster mushrooms, three-dimensional spatial positioning, and autonomous harvesting.

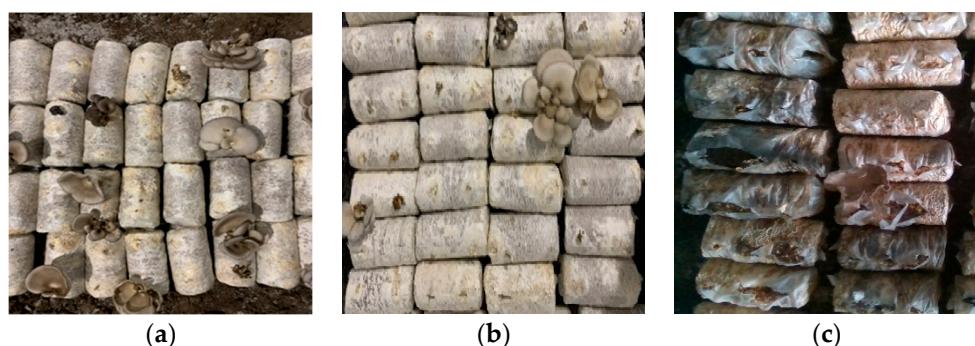


Figure 2. Bacteria bags for mushroom cultivation: (a) normal situation; (b) mushrooms grown in multiple bags adhere to each other; (c) cultivation bag aging.

4. Robot Structure and System Control

4.1. Mobile Robot Platform

The proposed mobile robot platform is illustrated in Figure 3 from a 3D point of view. Taking full account of the greenhouse oyster mushroom planting mode and greenhouse size, three guide rails are laid on the ground and a truss-type mobile platform is designed. The mobile robot platform needs to cover the mushroom growing areas in the left and right areas (see Figure 1), so the length and width of the mobile platform are designed to be about 7.7 and 1.4 m, respectively. The height dimension of the mobile platform takes full account of the field of view of the camera and operational environment, and the final height of the mobile platform is designed to be approximately 1.2 m. The use of aluminum profiles for the construction of the mobile robot platform can reduce the weight of the overall mechanical structure, and simplify the construction of an experimental platform. A control module mounting frame is designed in the middle to install an industrial computer, a motion controller, and batteries. As the host computer, the industrial computer is responsible for overall decision making, receiving various information and statuses sent by the lower computers, processing the received information, coordinating the movement of the harvesting unit, and recognizing the images captured by cameras. The motion controller communicates with the host computer using a network interface and accepts commands from the host computer to control the movement of the actuator and the clamping and rotating action of the end-effector. The electric motors provide power to the two sets of active driving wheels (see Figures 3 and 4), and driven wheels are arranged on both sides (see Figure 3). An active wheelset has two wheels on the top and bottom; the top two wheels are powered by the motor and the bottom two wheels are clamped to the track. The mobile robot platform only needs to move forward along the guide rails to traverse the entire greenhouse. The mobile platform moves forward by 1.1 m each time and harvesting units perform a picking cycle. After completing the traversal of the entire greenhouse, the mobile platform detects the signal of the photoelectric sensor installed at the end of the guide rail and then stops moving forward and returns to the initial position. Based on the above design, this mobile platform of the harvesting robot has a simple structure and can keep the robot moving smoothly. Considering the size of the mobile platform, the harvesting robot is currently only able to work in one greenhouse and cannot switch to another greenhouse autonomously.

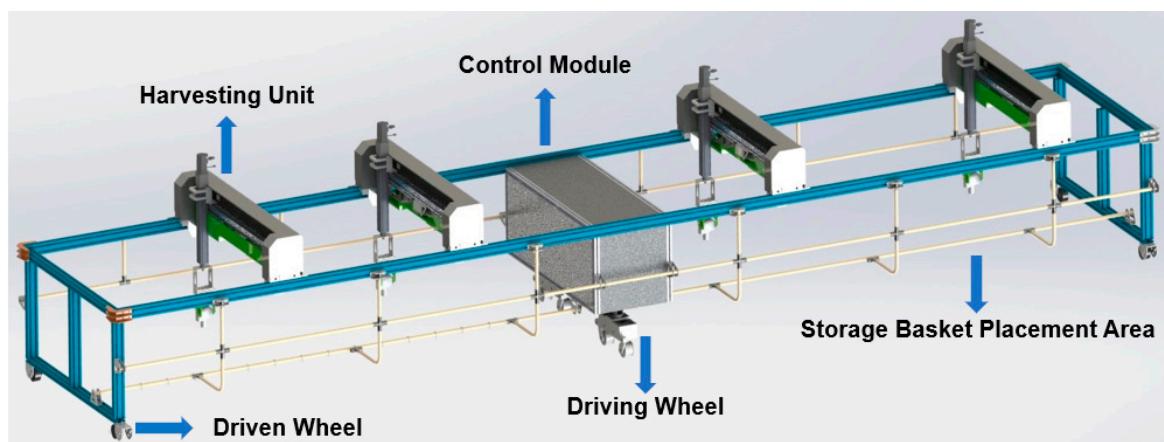


Figure 3. The mobile platform of the proposed harvesting robot.



Figure 4. Active driving wheels for driving mobile platform.

4.2. Harvesting Unit

The harvesting unit is the core component of the mushroom-harvesting robot. Each independent harvesting unit consists of a vision system, an end-effector, and a 3-DOF robotic arm, as shown in Figure 5a.

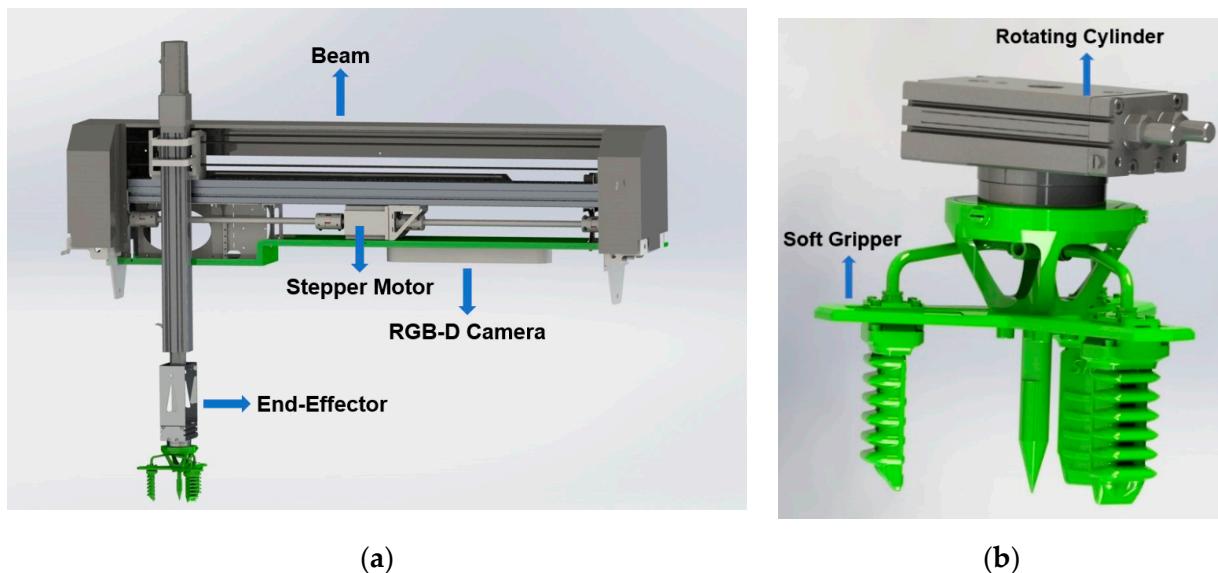


Figure 5. 3D view of the harvesting unit: (a) harvesting unit module; (b) a flexible hand for grasping mushrooms.

The movement mechanism of the harvesting unit is a 3-degree-of-freedom robotic arm based on the Cartesian coordinate system, which can realize movement in three directions

and is responsible for moving the end-effector to the designated space position. Each axis of the robotic arm is driven by a motor, and each axis is equipped with a position sensor to obtain movement information.

The vision system consists of an RGB-D camera (Intel RealSense D435i) and an LED light source. The RGB-D camera is installed under the beam of the harvesting unit (the camera is 1.2 m above the ground) and LED light sources are installed on both sides of the camera to supplement the light. The RGB-D camera provides color images with a resolution of 1920×1080 and depth images with a resolution of 1280×720 . Besides, the field angle of this RGB-D camera is $69^\circ \times 42^\circ$. Thus, the area of view field is about $0.92 \times 1.65 \text{ m}^2$ when the camera is about 1.2 m above the ground. The camera is approximately 1.2 m high from the mushroom and the width of the view field area is 0.92 m. In order not to miss the pickable mushrooms, the mobile platform has to move to the middle of the mushroom-growing area each time. The dimensions of each mushroom-growing area are fixed, each growing area is 0.3 m apart, and the growing area is 0.8 m wide, so the mobile platform moves 1.1 m at a time to allow the camera to capture the complete mushroom area. The end-effector mechanism mainly consists of soft gripping jaws, rotating cylinders, and pneumatic devices (see Figure 5). Considering the brittle nature of the mushroom and its susceptibility to breakage, the surface is easily damaged. The traditional gripper would cause damage to the mushroom, so a soft gripper is used to simulate the gripping-rotating action of manual picking (the robotic arm end-effector grips the oyster mushroom and then rotates it to release it from the cultivation bag) [25]. A four-jaw soft gripper is used for its special airbag structure, and different internal and external differences produce different actions. When positive pressure is input, the gripper tends to grip tightly and adaptively wraps around the outer surface of the object to complete the gripping action; when negative pressure is input, the gripper loosens and releases the object. The rotating cylinder is mounted above the jaws and rotates in both clockwise and counter-clockwise directions by applying and releasing pressure. The pneumatic device is the power source of the end gripping mechanism, mainly including an air hose, air pump, vacuum valve, etc., to achieve the pressure control of the air circuit, so as to control the rotary cylinder and the air jaws to create the corresponding action. Finally, the rotating cylinder rotates 180° after grasping the mushroom tightly so that the mushroom can fall off the bag more easily. In this work, the gripper of the oyster-mushroom-harvesting robot only grasps mushrooms in a fixed pose, and gripper pose optimization has not yet been explored.

4.3. Control Architecture and Control Strategy

4.3.1. Control Architecture

As shown in Figure 6, the motion control of the mobile platform and the harvesting units are controlled by sending commands via the motion controller. As the moving platform moves backward and forwards in only one direction, an active wheelset requires only one motor for driving. In addition, a position sensor is installed on the mobile platform to determine whether the mobile platform traverses the entire greenhouse. The movement of the harvesting unit in three directions is also controlled by the motion controller sending instructions to the motor actuator. The soft gripper is controlled by the gripper controller, and the rotating action of the end-effector is controlled by the host computer, which sends a signal to the solenoid valve to control the rotating cylinder movement. A position sensor is installed on each motion track of a harvesting unit to obtain the position information of the three-axis movement of a harvesting unit.

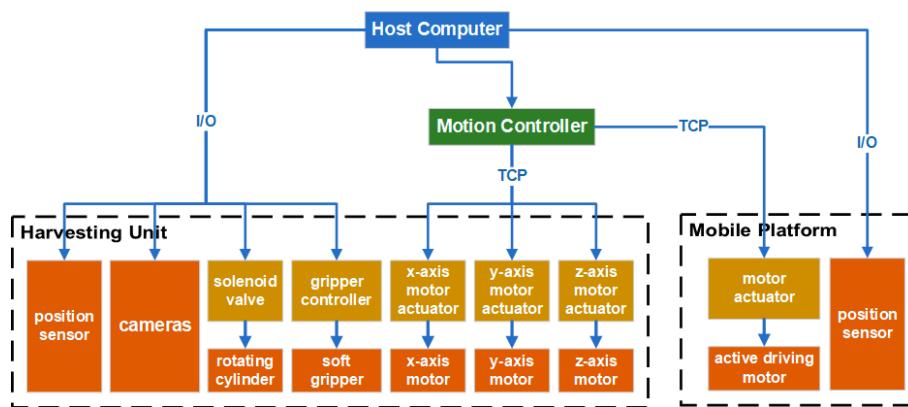


Figure 6. Control diagram of oyster-mushroom-harvesting robot.

4.3.2. Overall Control Strategy

To carry out harvesting operations completely autonomously, it is necessary for the harvesting robot to cruise in the entire greenhouse. Besides, the harvesting unit is responsible for harvesting mushrooms in a local area and we can deploy multiple harvesting units on the mobile platform at the same time. Figure 7 shows the workflow chart of the mushroom-harvesting robot. The robot performs initialization operations and equipment self-checks before performing tasks to guarantee the normal communication of each part, and then the robot performs the harvesting task area by area in the entire greenhouse. The mobile platform moves forward a fixed distance of 1.1 m each time, and harvesting units start a picking cycle in this area. The harvesting units traverse the area where the mobile platform is currently located, capturing images and uploading them to the host computer for mushroom detection. When pickable mushrooms are detected, harvesting units perform the picking action in turn. It should be stated that the picking cycle runs an open loop and there is no feedback determining whether the mushroom is picked successfully. After harvesting units have completed a picking cycle in the current area, the mobile platform moves forward another 1.1 m. Finally, the mobile platform detects a signal from a sensor located at the end of the guide rail and autonomously returns to the initial position, which means that the harvesting robot has completed the harvesting task in the entire greenhouse.

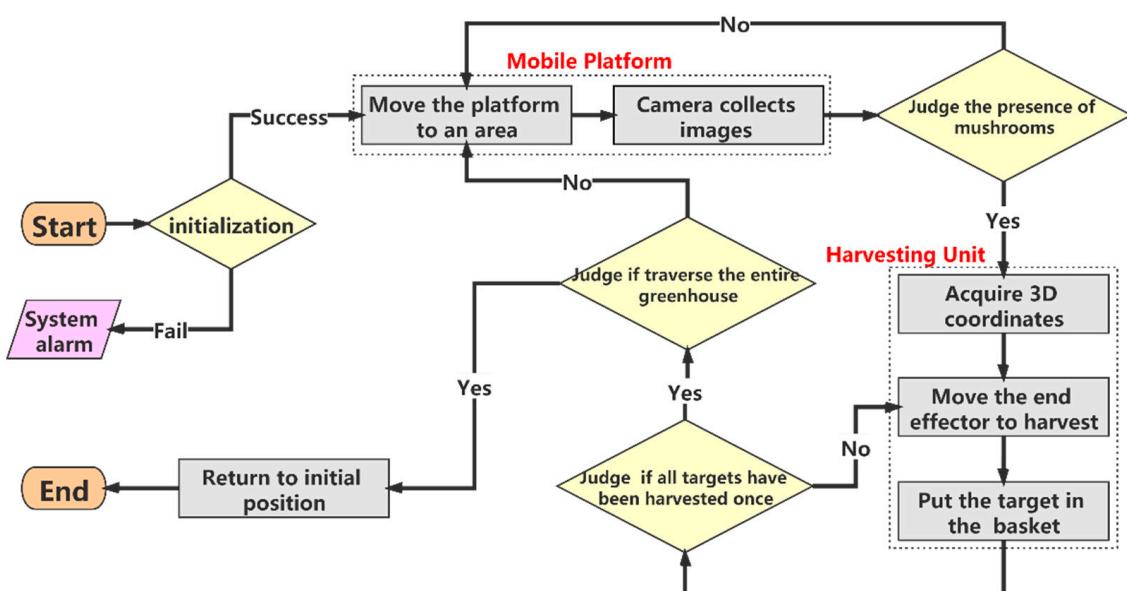


Figure 7. Workflow chart of the mushroom-harvesting robot.

4.3.3. Harvesting Unit Control Strategy

To improve the efficiency of the mushroom-harvesting robot, multiple harvesting units are deployed on the mobile platform to work at the same time, and each harvesting unit is responsible for a sub-area of the area where the current mushroom-harvesting robot is located. Since the planting areas in the greenhouse are aligned left and right on both sides of the walkway, only the working condition of the mushroom-harvesting robot on the left is analyzed.

The harvesting plan of the robot in the left area is illustrated in Figure 8. In the harvesting area on the left, we deploy two harvesting units on the mobile platform, harvesting units A and B. A1 and B1 are the initial positions of harvesting units A and B respectively. The harvesting area is divided into three sub-areas; A is responsible for Area 1 and Area 2, and B is responsible for Area 3. Some of the working areas of the two harvesting units overlap, so it is necessary to plan the working sequence of the two harvesting units to avoid collisions. First, harvesting unit A works in Area 2, and B does not move. After harvesting in Area 2, A continues working in Area 1 and B starts moving in Area 3. Using this strategy, when A is working in Area 2, there is no need to worry that B will move to the overlapping area and collide. In each harvesting area, harvesting units need to determine the picking order of multiple mushrooms in order to reduce the movement distance of picking units to increase efficiency. As shown in Figure 8, there are multiple mushrooms in Area 1 and Area 3. The first mushroom selected for picking is the one closest to the end-effector, the second is closest to the first, and so on in that order.

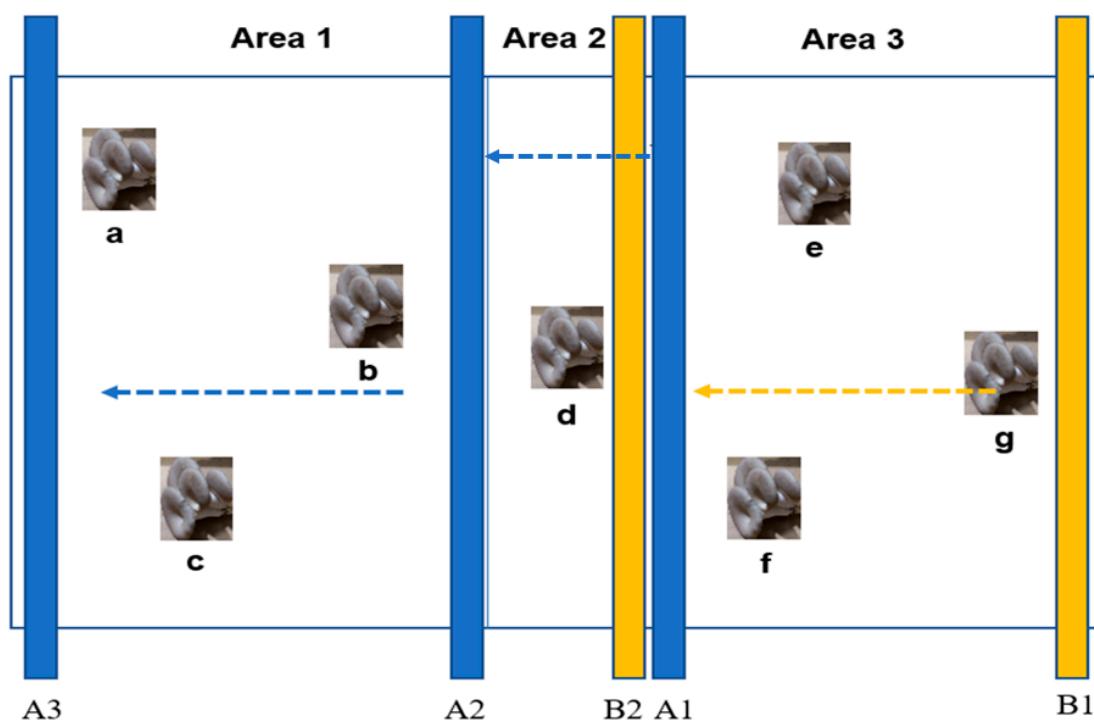


Figure 8. Harvesting plan of the robot in the left area.

The working method of the harvesting unit is as follows:

(1) The harvesting unit traverses the oyster mushroom planting area, and at the same time, the camera located under the beam collects images in real time and transmits the video stream to the industrial computer;

(2) Using the object detection algorithm based on the convolutional neural network to detect the input video stream end-to-end, the oyster mushrooms are identified and located (the coordinate of the mushroom in the image coordinate system is the center of the bounding box);

(3) When the camera collects the RGB image, it obtains the point cloud image at the same time and the three-dimensional coordinates of the oyster mushroom relative to the camera coordinate system is obtained by indexing the mushroom coordinate in the image coordinate system (more details can be seen in Section 5.3). Finally, the three-dimensional coordinates of the oyster mushroom relative to the robotic arm are obtained through the hand-eye conversion matrix;

(4) The soft gripper opens and reaches directly above the oyster mushroom. Finally, the soft gripper grasps the oyster mushroom, rotates it at a certain angle (rotates it 180 degrees), then lifts it and places it in the basket, as shown in Figure 9. Baskets are placed in fixed positions, and each harvesting unit has a basket. In the process of putting the picked mushroom into the basket, the harvesting unit places mushrooms at multiple position points in the basket in turn by adding multiple placement points to the program in advance, which allows the mushrooms to be placed more evenly.

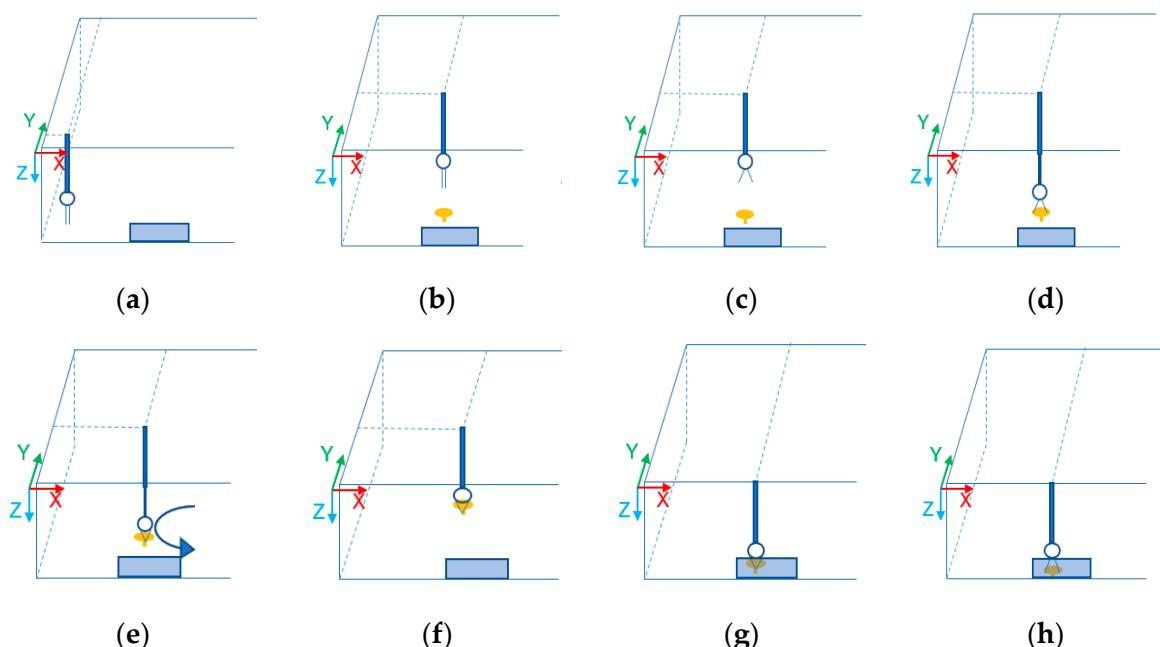


Figure 9. Schematic diagram of mushroom grabbing with soft gripper: (a) initial state; (b) move on the plane; (c) open the gripper; (d) reach the designated height; (e) close the gripper and rotate; (f) raise the gripper; (g) move the mushroom to the basket; (h) loosen the gripper.

5. Mushroom Recognition and Localization

The mushroom detection process involves both identification and localization. Identification requires an algorithm to determine whether the mushroom is present in the image and to select the position of the mushroom in the image by a bounding box. However, the robot grasps the mushroom by calculating the position of the camera relative to the robot coordinate system to obtain the position of the mushroom relative to the end-effector. In principle, any object detection algorithm based on deep learning can perform this function. We use a lightweight main feature network to improve the SSD object detector and improve detection speed without reducing accuracy. We first test it on a laptop, then deploy it on an embedded device, where faster speed on a laptop means it is better suited for deployment on an embedded device.

5.1. Data Collection and Enhancement

The image was collected at the greenhouse oyster mushroom planting base in Yuhu District, Xiangtan City, Hunan Province, China. The collection period was from 21 to 25

June 2019. The camera was an Intel RealSense D435 depth camera with a color resolution of 1920 * 1080 pixels which was installed on the collection device parallel to the ground. Due to the light penetration on both sides of the greenhouse, it was necessary to take full account of the light changes, and images were collected at 8:00–10:00 a.m., 1:00–3:00 p.m., and 5:00–6:00 p.m. (see Figure 10), while light sources were added to illuminate the greenhouse when it was dark. For sample diversity, the oyster mushroom dataset was collected on different days and in different greenhouses. The collected datasets were manually labelled by Rong and Yang using the annotation tool LabelImg, and cross-check labels were performed to ensure the quality of the labeled data. When pickable mushrooms appeared in the image, rectangular boxes were used to enclose a complete cluster of mushrooms. The rules used to label a mushroom were as follows: (1) only mature mushrooms with a size of 8–16 cm were labeled; (2) if the mushrooms were not completely present in the image (obscured by other mushrooms or at the edge of the image) they were not labeled. These annotation rules aimed to make the network learn the characteristics of mature mushrooms and not to recognize incomplete mushrooms on the edge of the image which cause inaccurate positioning. Finally, a total of 4600 labeled images were collected to train and test the models.

For convolutional neural networks, weights of the network learned the data distribution of the data set, so that the convolution kernel parameters formed the cognition of the feature distribution in the dataset. The purpose of data enhancement is to increase the amount of data for training to improve the generalization of the model and to increase the noise data to improve the robustness of the model. The existing image data enhancement methods are divided into three types: space conversion (flip, random rotation, random cropping, and random size), color distortion (brightness conversion, tone conversion), and information discarding (random erasure, Mixup) [26]. One purpose is to enrich the data set, and the other is to improve the generalization ability of the network to detect targets under different light interference and poor imaging effects.



Figure 10. Dataset under different lighting: (a) strong light; (b) normal light; (c) weak light.

5.2. Mushroom Recognition Method

The use of the object detector helps the picking robot to find the harvestable mushrooms in the image and obtain the position of the mushroom in the image coordinate system. Using the point cloud information from the D435i, the 3D coordinates of the mushroom relative to the camera coordinate system are obtained so that the end-effector can obtain the gripping point. First of all, it is necessary to select an object detection algorithm that meets the requirements of picking robots. The algorithm needs to be fast enough while ensuring the detection accuracy.

In this paper, the SSD object detection algorithm is chosen to detect mushrooms. The feature extraction part of the SSD object detection algorithm uses the structure of VGG16, and predictions of SSD are made on each of the last six feature maps. As the SSD algorithm is an end-to-end one-stage object detection algorithm, it is inherently fast. To further increase the speed of detection, we use a lightweight network as the feature extraction network of the SSD, which we call the improved SSD below. The improved SSD replaces the primary feature extraction network from VGG16 with MobileNet-v2 [27]. In the MobileNet-

v2 feature extraction network, the depthwise separable convolution is used instead of the normal 3×3 convolution to reduce the weight parameters and computation. Fewer calculations and parameters make for faster detection. Besides, gradient disappearance is mitigated by using residual connections.

5.2.1. Model Training

The equipment used to train and test models in this paper is a laptop equipped with an Intel Core i7-8750H CPU and NVIDIA GTX1080 GPU. There is a total of 4600 images, which are divided into 4000 in the training set, 300 in the validation set, and 300 in the test set (images in the three data sets are randomly selected). We train these two models (SSD and improved SSD), which are trained for 200 epochs with a batch size of 6, an initial learning rate of 0.002, a momentum of 0.949, and a decay of 0.0005.

5.2.2. Testing Results of Models

After training, we use the weights of models obtained on the test set to verify the performance of improved SSD. We use the F1 value as the evaluation index of the training result, as shown in Formulas (1)–(3); P is the accuracy rate, R is the recall rate, and F1 is an index that weighs the accuracy rate and the recall rate.

$$P = TP / (TP + FP) \quad (1)$$

$$R = TP / (TP + FN) \quad (2)$$

$$F1 = 2 \times P \times R / (P + R) \quad (3)$$

where TP is the number of correctly identified oyster mushrooms, FP is the number of incorrectly identified oyster mushrooms, and FN is the number of missed oyster mushrooms. The Intersection-over-Union (IoU) of the bounding box predicted by the network and the ground-truth bounding box is used to judge the quality of the prediction. When the IoU is greater than 0.5, it is considered to be a correctly identified oyster mushroom, and when less than 0.5, it is considered to be an incorrectly identified oyster mushroom.

Mean Average Precision (mAP) is the standard method for evaluating the performance of the object detectors. Besides, we also report F1 value, which is another evaluation method. The object detector outputs the predicted bounding box information, including the predicted category, location, and confidence. Test results can be seen in Table 1. The SSD has higher F1 values (confidence threshold is set to 0.9 here, which gives a good trade-off between precision and recall), 1.4% higher than the SSD using the MobileNet-v2 feature network. The mAP of the SSD is slightly higher than that of the improved SSD, which means better performance. The difference in the accuracy between SSD and improved SSD is relatively small, and the difference in detection time is relatively large, which means that for simple detection tasks such as detecting mushrooms, the advantages of using a lightweight network are more obvious, and it is more suitable to deploy on embedded devices.

The harvesting unit collects images in real-time while moving in the picking area, and detects the presence of pickable mushrooms in the field of view. When pickable mushrooms are detected, the harvesting unit stops moving and then collects another image for identification and positioning. If the harvesting unit moves too fast, it may cause the harvesting unit to be too far away from the pickable mushrooms detected before the harvesting unit stops. Two solutions to this problem are to increase the detection speed and reduce the moving speed of the harvesting unit. However, reducing the moving speed of the harvesting unit will reduce the efficiency of the robot. Therefore, we hope to increase the detection speed as much as possible while maintaining the detection accuracy, so as to increase the maximum moving speed of the harvesting unit (the maximum moving speed of harvesting units reaches 0.15 m/s currently). The SSD needs 0.058s to process an image. At the same time, it takes 0.032s for the improved SSD to detect an image, and the host computer detects the images uploaded by four cameras at the same time (the frame rate is

7.8 FPS). After several field tests, the moving speed of the harvesting unit is set to 0.15 m/s. The improved SSD model uses a more lightweight backbone, which improves the detection speed a lot while the accuracy is the same. Therefore, we use an improved SSD model as the object detection algorithm for identifying and locating mushrooms.

Table 1. Comparison of identification results with SSD and improved SSD.

Method	Input Size	P (%)	R (%)	mAP (%)	F1 (%)	Speed (s)
SSD	300	95.3	94.9	94.5	95.1	0.058
Improved SSD	300	94.4	93.0	93.2	93.7	0.032

As shown in Figure 11, the experimental results of using the improved SSD object detection algorithm on the test dataset are analyzed. As shown in Figure 11a, the aging of the bacterial bags is likely to cause missed identification (the plastic film can break after prolonged use of bacterial bags, affecting identification); part of the reason may be that the image data of the aging of the bacterial bags in the dataset are less and the characteristics of the broken plastic film interfere with mushroom recognition. By comparing the accuracy and recall rate of mushroom recognition under different lighting conditions (see Figure 11b), we can find that the missed detection rate is higher when the light is weak. The F1 values are 95.0%, 94.2%, and 93.4% in full light, normal light, and very weak light, respectively. It is difficult for the detection algorithm to separate the mushrooms from the background in the dark. In the case of mutual adhesion, the adhesion of the oyster mushroom overlaps with each other, which easily causes multiple objects to be recognized as one or misses others, as shown in Figure 11c. In order to alleviate this effect, images of adhesion mushrooms can be added to allow the network to correctly identify these features for correct recognition.

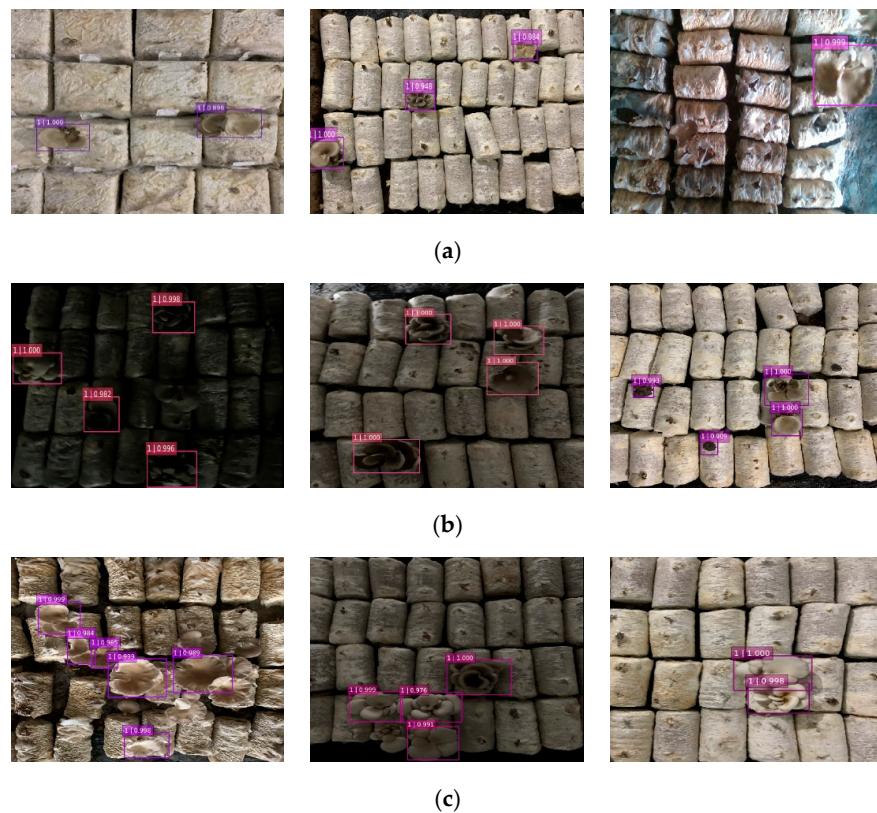


Figure 11. Experimental results on the test dataset: (a) different bacteria package; (b) different light intensities; (c) adhesion.

5.3. Mushroom Localization

The object detection algorithm can predict the harvestable mushrooms in the RGB images and obtain the pixel coordinates of the center point of the mushroom bounding box. If only the center point of the bounding box is selected to index in the corresponding point cloud, a missing data value may be found in the point cloud (a missing data value is equal to 0). As shown in Figure 12, the coordinates of nine pixels in the image coordinate system are selected to avoid a missing data value, and three-dimensional coordinates of nine pixels are obtained by using pixel coordinates in the image coordinate system to index corresponding values in the point cloud. Then, the missing data of nine three-dimensional coordinates are removed and the average value of the remaining coordinates is taken to obtain the final three-dimensional picking point. Finally, the positioning of the mushroom is achieved by converting the coordinates of the picking point relative to the camera to coordinates relative to the origin of the robotic arm via a calibrated hand-eye conversion matrix [28].

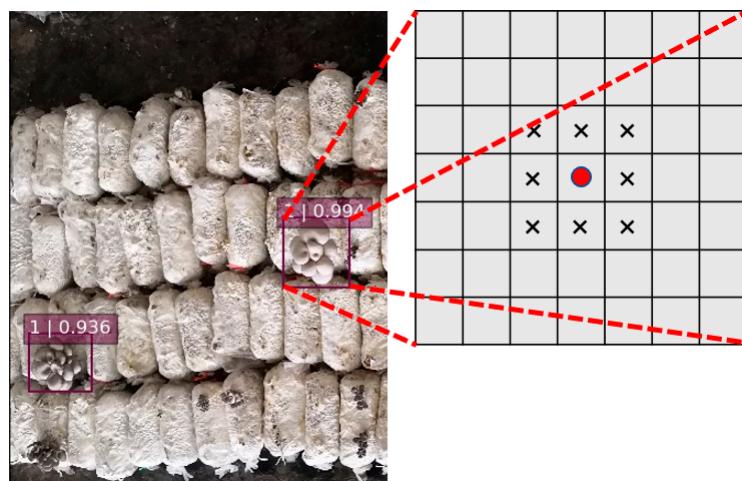


Figure 12. Image labeling diagram.

6. Field Experiment

The built-up oyster-mushroom-harvesting robot test platform is shown in Figure 14. The mobile platform can move along the laid track in the entire greenhouse, and two harvesting units are deployed on the left and right planting areas. The field experiments were all conducted in one afternoon in March 2020 (the weather was clear and there was plenty of light) at the greenhouse oyster mushroom base in Yuhu District, Xiangtan City, Hunan Province, China. The following field experiments aim to evaluate the overall performance of the harvesting robot. We followed the steps below to test the system, as shown in Figure 14:

- (1) Manually count the number of oyster mushrooms that can be picked in the greenhouse;
- (2) System self-check. First, start the machine; then, initialize the system to determine whether each part is working properly, and perform position calibration on the harvesting unit;
- (3) When the system is ready, click the “start picking button”. The mobile platform starts to move, and the four groups of harvesting units work at the same time. The camera recognizes the mushroom and transmits the calculated coordinates to the harvesting unit for grabbing and the harvested mushrooms are placed in the storage area. Manually record the harvesting situation until the robot completes the harvesting task of the entire greenhouse;
- (4) Perform statistical analysis on the recorded work of mushroom-harvesting robots; statistics are shown in Table 2.

Table 2. Results of field experiments.

Metric	Number of the Experiments			Total
	No. 1	No. 2	No. 3	
Harvestable	439	351	394	1184
Successfully identified	424	327	374	1125
Successfully harvested	395	290	343	1028
Harvesting time	63 min	51 min	52 min	166 min
Recognized ratio	96.6%	93.2%	94.9%	95.0%
Harvesting success rate	90.0%	82.6%	87.1%	86.8%
Average time	8.92 s	9.36 s	8.34 s	8.85 s

**Figure 13.** Field test of the mushroom-harvesting robot.**Figure 14.** *Cont.*



Figure 14. The process of robotic harvesting: (a) mobile platform starts to move; (b) camera collects images; (c) move gripper to grasp; (b) place oyster mushrooms.

7. Results and Discussion

The performance results of the field experiment of the oyster-mushroom-harvesting robot are shown in Table 2 and we also provide a video of the experiment in the Supplementary Material. A total of three experiments were carried out. According to manual statistics, there are 1184 mushrooms available for harvesting. This paper uses “Recognized ratio” to evaluate the vision system, which means the proportion of mushrooms recognized by the mushroom-harvesting robot in the total number of harvestable mushrooms, and “Harvesting success rate” is used to evaluate the overall harvesting success rate of mushroom-harvesting robots.

Table 2 shows the details of the field experiments. Among the 1184 harvestable mushrooms, the mushroom-harvesting robot identified 1125 mushrooms and successfully harvested 1028 of them. The total time for harvesting was 166 min (including identification and harvesting time). The overall recognition success rate (the ratio of mushrooms that are successfully recognized to mushrooms that can be picked in the field of vision) was 95%, and the harvesting success rate was 86.8% (the ratio of mushrooms successfully placed in the basket to mushrooms that can be picked in the field of vision). It took 8.85 s on average to harvest a mushroom. There are no statistics on damage to successfully harvested mushrooms, and harvesting success is defined as success when the mushrooms are harvested into the basket by a soft gripper.

The harvesting success rate of the *Agaricus bisporus*-harvesting robot designed by Reed et al. in the greenhouse was about 80%, and the single harvesting time was 6.7 s. The harvesting success rate of the oyster-mushroom-picking robot designed in this paper was 86.8%, which is higher than the method of Reed, and the single harvesting time was 8.85 s. The single harvesting time of our mushroom-harvesting robot is 2.18 s slower, which is also related to the growing density of *Agaricus bisporus* and oyster mushrooms. With the low density of oyster mushrooms, the harvesting robot takes more time to pick individual mushrooms with a larger movement stroke. Moreover, it takes more time to grasp with the soft gripper than with the suction cup. The characteristics of oyster mushrooms make it possible to grasp only with a soft gripper.

We further compare the harvesting efficiency of manual harvesting and the oyster-mushroom-harvesting robot, it takes 3–5 s for a skilled worker to pick a mushroom while the robot takes 8.85 s. Obviously, if we compare the harvesting efficiency per unit time, the efficiency of manual harvesting is much higher than that of robots. The environment of the mushroom greenhouse is dark and humid, and working in this environment for a long time makes people feel uncomfortable. However, the robot can work in this environment for 24 h a day. Therefore, it is feasible to use picking robots instead of manual mushroom picking in terms of efficiency.

In the above experiment, some mushrooms were not successfully identified and harvested. We analyze the reasons for identification failure and harvesting failure. The main reasons for the identification failure are the serious adhesion of oyster mushrooms, the aging of the cultivation bag, and the low light: (1) mushroom adhesion makes the vision system recognize multiple mushrooms as one mushroom, which affects the accuracy of mushroom recognition and positioning accuracy; (2) the aging of the cultivation bag damages the plastic film on the surface, and the damaged plastic film will interfere with identification; and (3) in some poorly illuminated areas, the characteristics of mushrooms are similar to the background, and it is difficult to successfully identify them.

The reasons for the failure of harvesting include serious adhesion of the oyster mushrooms, the inaccurate depth, and mushrooms slipping from the soft gripper: (1) serious adhesion causes the oyster mushroom to recognize parts or too much, making the two-dimensional center point inaccurate, and ultimately leading to inaccurate three-dimensional positioning points; (2) the inaccurate point cloud data returned by D435i are not right, and the gripper does not reach the designated position; and (3) one situation is that the mushroom is too small, and the mushroom can easily slip from the gap in the gripper. Another situation is that the mushrooms are too large to be completely covered by the gripper.

Besides, due to the lack of force feedback in the soft gripper, some mushrooms that are successfully harvested are damaged (see Figure 15). Most of the mushrooms that cause damage are large, and the soft gripper grasps the mushroom completely without adjusting the gripping force according to the size of the mushroom. Therefore, considering the soft texture and inconsistent size of oyster mushrooms, the new end-effector should be equipped with a force feedback sensor, so that the end-effector can apply appropriate force according to different sizes of mushrooms, instead of completely gripping the mushrooms to make larger mushrooms damaged. Additionally, a force feedback sensor can feed back a signal on whether the mushroom has been picked successfully.

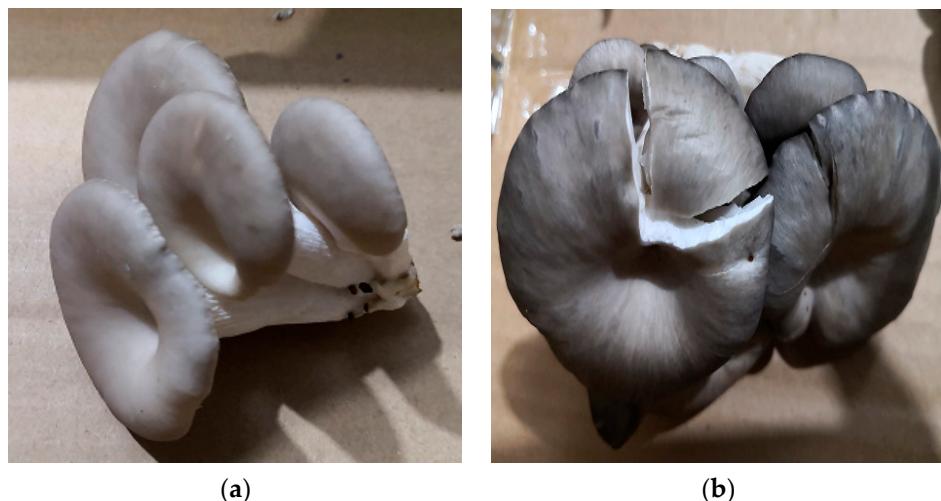


Figure 15. Successfully harvested mushrooms: (a) undamaged mushrooms; (b) damaged mushrooms.

Considering the uneven illumination, robots tend to miss mushrooms under low-light conditions, so we also need to strengthen the detection capability of the object detection network under low-light conditions, as well as collect more images in low-light environments.

8. Conclusions and Future Work

In this paper, we developed a harvesting robot suitable for picking oyster mushrooms in the greenhouse and built a robot experimental platform for field experiments. A mushroom-harvesting robot structure in which multiple harvesting units are connected in parallel on a mobile platform is proposed, which effectively improves the overall picking efficiency. We mainly focused on an overall system construction of a mushroom-harvesting

robot, the mobile robot platform, the harvesting unit, and the vision system of the robot, and introduced how multiple harvesting units work together to improve the efficiency of harvesting in this paper.

We have proved the effectiveness of the mushroom-harvesting robot in the field experiments with an overall recognition accuracy of 95% for the vision system, a harvesting success rate of 86.8% (regardless of damage of the harvested oyster mushrooms), and an average time for picking individual mushroom of 8.85 s.

In future work, we will improve the actual application of the mushroom-harvesting robot and focus on the following key techniques: (1) The integration of mushroom planting agronomy and harvesting robot. Different mushroom varieties have specific biomechanical properties and planting agronomy. It is necessary to make appropriate interventions and improvements in planting agronomy to make mushroom growth more standardized and consistent, which is conducive to improving the accessibility and success rate of robotic picking. We will research the biomechanical properties of mushrooms, and refine the end-effector to adapt to the mushroom characteristics of being high in moisture, brittle, and easily damaged;

(2) Accurate recognition and intelligent control. We only performed preliminary research on the key factors that affect mushroom recognition, such as light, adhesion, and posture. This is essential to improve the recognition effect of mushrooms. We will combine the latest deep learning models to develop algorithms that adapt to complex scene recognition. For the mushroom positioning, obstacle avoidance, and human-machine collaboration, we will develop a smart end-effector control algorithm suitable for selective harvesting, and gradually realize strong adaptability and efficient autonomous harvesting;

(3) Robot effectiveness evaluation. In order to realize the promotion and application of mushroom-harvesting robots in actual production, it is necessary to systematically evaluate the performance of the robots and establish a unified evaluation index, including picking efficiency, damage rate, missed picking rate, economic evaluation, etc. It is also beneficial to the development of other harvesting robots in the future.

Supplementary Materials: The field-tested video is available online at <https://www.mdpi.com/article/10.3390/agronomy11061210/s1>, Video S1: mushroom-harvesting robot working in the greenhouse.

Author Contributions: Conceptualization, P.W.; Funding acquisition, P.W.; Methodology, P.W., Q.Y. and F.H.; Image Annotation, J.R. and Q.Y.; Software, J.R. and Q.Y.; Validation, J.R., Q.Y. and F.H.; Writing—original draft, J.R. and Q.Y.; Writing—review and editing, J.R. and P.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Jiangsu Agriculture Science and Technology Innovation Fund (JASTIF), grant number CX(19)3072 and the National Key Research and Development Program of China, grant number 2017YFD0701502.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Bac, C.W.; van Henten, E.J.; Hemming, J.; Edan, Y. Harvesting Robots for High-value Crops: State-of-the-art Review and Challenges Ahead. *J. Field Robot.* **2014**, *31*, 888–911. [[CrossRef](#)]
2. Charania, I.; Li, X. Smart farming: Agriculture’s shift a labor intensive to technology native industry. *Internet Things* **2020**, *9*, 1–20. [[CrossRef](#)]
3. Hu, X.; Wang, C.; Yu, T. Design and application of visual system in the Agaricus bisporus picking robot. *J. Phys. Conf. Ser.* **2019**, *1187*, 032034.
4. Hu, X.; Pan, Z.; Lv, S. Picking Path Optimization of Agaricus bisporus Picking Robot. *Math. Probl. Eng.* **2019**, *2019*, 1–16. [[CrossRef](#)]

5. Leu, A.; Razavi, M.; Langstädler, L.; Ristić-Durrant, D.; Raffel, H.; Schenck, C.; Gräser, A.; Kuhfuss, B. Robotic Green Asparagus Selective Harvesting. *IEEE/ASME Trans. Mechatron.* **2017**, *22*, 2401–2410. [[CrossRef](#)]
6. Birrell, S.; Hughes, J.; Cai, J.Y.; Iida, F. A field-tested robotic harvesting system for iceberg lettuce. *J. Field Robot.* **2020**, *37*, 225–245. [[CrossRef](#)] [[PubMed](#)]
7. Arad, B.; Kurtser, P.; Barnea, E.; Harel, B.; Edan, Y.; Ben-Shahar, O. Controlled Lighting and Illumination-Independent Target Detection for Real-Time Cost-Efficient Applications. The Case Study of Sweet Pepper Robotic Harvesting. *Sensors* **2019**, *19*, 1390. [[CrossRef](#)]
8. Arad, B.; Balendonck, J.; Barth, R.; Ben-Shahar, O.; Edan, Y.; Hellström, T.; Hemming, J.; Kurtser, P.; Ringdahl, O.; Tielen, T.; et al. Development of a sweet pepper harvesting robot. *J. Field Robot.* **2020**, *37*, 1027–1039. [[CrossRef](#)]
9. Liang, C.; Xiong, J.; Zheng, Z.; Zhong, Z.; Li, Z.; Chen, S.; Yang, Z. A visual detection method for nighttime litchi fruits and fruiting stems. *Comput. Electron. Agric.* **2020**, *169*, 105192. [[CrossRef](#)]
10. Zhang, T.; Huang, Z.; You, W.; Lin, J.; Tang, X.; Huang, H. An Autonomous Fruit and Vegetable Harvester with a Low-Cost Gripper Using a 3D Sensor. *Sensors (Basel)* **2019**, *20*, 93. [[CrossRef](#)]
11. Koirala, A.; Walsh, K.B.; Wang, Z.; McCarthy, C. Deep learning for real-time fruit detection and orchard fruit load estimation: Benchmarking of ‘MangoYOLO’. *Precis. Agric.* **2019**, *20*, 1107–1135. [[CrossRef](#)]
12. Liu, G.; Nouaze, J.C.; Touko Mbouembe, P.L.; Kim, J.H. YOLO-Tomato: A Robust Algorithm for Tomato Detection Based on YOLOv3. *Sensors (Basel)* **2020**, *20*, 2145. [[CrossRef](#)]
13. Tian, Y.; Yang, G.; Wang, Z.; Wang, H.; Li, E.; Liang, Z. Apple detection during different growth stages in orchards using the improved YOLO-V3 model. *Comput. Electron. Agric.* **2019**, *157*, 417–426. [[CrossRef](#)]
14. Reed, J.N.; Miles, S.J.; Butler, J.; Baldwin, M.; Noble, R. AE—Automation and Emerging Technologies: Automatic Mushroom Harvester Development. *J. Agric. Eng. Res.* **2001**, *78*, 15–23. [[CrossRef](#)]
15. Masoudian, A.; Mcisaac, K.A. Application of Support Vector Machine to Detect Microbial Spoilage of Mushrooms. In Proceedings of the 2013 International Conference on Computer and Robot Vision, Regina, SK, Canada, 28–31 May 2013; pp. 281–287.
16. Lu, C.-P.; Liaw, J.-J. A novel image measurement algorithm for common mushroom caps based on convolutional neural network. *Comput. Electron. Agric.* **2020**, *171*, 105336. [[CrossRef](#)]
17. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In Proceedings of the Computer Vision—ECCV 2016, Amsterdam, The Netherlands, 11–14 October 2016; pp. 21–37.
18. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 779–788.
19. Redmon, J.; Farhadi, A. Yolov3: An incremental improvement. *arXiv* **2018**, arXiv:1804.02767.
20. Bochkovskiy, A.; Wang, C.-Y.; Liao, H.-Y.M. Yolov4: Optimal speed and accuracy of object detection. *arXiv* **2020**, arXiv:2004.10934.
21. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv* **2015**, arXiv:1506.01497. [[CrossRef](#)] [[PubMed](#)]
22. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
23. Huang, J.; Rathod, V.; Sun, C.; Zhu, M.; Korattikara, A.; Fathi, A.; Fischer, I.; Wojna, Z.; Song, Y.; Guadarrama, S. Speed/accuracy trade-offs for modern convolutional object detectors. In Proceedings of Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 22–25 July 2017; pp. 7310–7311.
24. Hendrawan, Y.; Anta, D.K.; Ahmad, A.M.; Sutan, S.M. Development of Fuzzy Control Systems in Portable Cultivation Chambers to Improve the Quality of Oyster Mushrooms. In Proceedings of the 9th Annual Basic Science International Conference 2019 (BaSIC 2019), Malang, Indonesia, 20–21 March 2019; p. 032013.
25. Zhang, B.; Xie, Y.; Zhou, J.; Wang, K.; Zhang, Z. State-of-the-art robotic grippers, grasping and control strategies, as well as their applications in agricultural robots: A review. *Comput. Electron. Agric.* **2020**, *177*, 105694. [[CrossRef](#)]
26. Perez, L.; Wang, J. The Effectiveness of Data Augmentation in Image Classification using Deep Learning. *arXiv* **2017**, arXiv:1712.04621.
27. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.-C. Mobilenetv2: Inverted residuals and linear bottlenecks. In Proceedings of Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 19–21 June 2018; pp. 4510–4520.
28. Tsai, R.Y.; Lenz, R.K. A new technique for fully autonomous and efficient 3D robotics hand/eye calibration. *IEEE Trans. Robot. Autom.* **1989**, *5*, 345–358. [[CrossRef](#)]