Detection method for the cucumber robotic grasping pose in clutter scenarios via instance segmentation

Fan Zhang, Zeyu Hou, Jin Gao, Junxiong Zhang*, Xue Deng

(*College of Engineering, China Agricultural University, Beijing 100083, China*)

Abstract: The application of robotic grasping for agricultural products pushes automation in agriculture-related industries. Cucumber, a common vegetable in greenhouses and supermarkets, often needs to be grasped from a cluttered scene. In order to realize efficient grasping in cluttered scenes, a fully automatic cucumber recognition, grasping, and palletizing robot system was constructed in this paper. The system adopted Yolact++ deep learning network to segment cucumber instances. An early fusion method of F-RGBD was proposed, which increases the algorithm's discriminative ability for these appearance-similar cucumbers at different depths, and at different occlusion degrees. The results of the comparative experiment of the F-RGBD dataset and the common RGB dataset on Yolact++ prove the positive effect of the F-RGBD fusion method. Its segmentation masks have higher quality, are more continuous, and are less false positive for prioritizing-grasping prediction. Based on the segmentation result, a 4D grab line prediction method was proposed for cucumber grasping. And the cucumber detection experiment in cluttered scenarios is carried out in the real world. The success rate is 93.67% and the average sorting time is 9.87 s. The effectiveness of the cucumber segmentation and grasping pose acquisition method is verified by experiments.

Keywords: Clutter scenarios, Cucumber grasp, Convolutional neural network, Instance segmentation

DOI: 10.25165/j.ijabe.20231606.7542

Citation: Zhang F, Hou Z Y, Gao J, Zhang J X, Deng X. Detection method for the cucumber robotic grasping pose in clutter scenarios via instance segmentation. Int J Agric & Biol Eng, 2023; 16(6): 215–225.

1 Introduction

Robotic grasping is the critical function of robots. It has been researched for different types of robots in recent years. In manufacturing or logistic industries, robotic grasping serves for goods picking, placement, and assembling. Therefore, the automation of grasping tasks is an important premise to improve efficiency and save costs. While robots in industries grasp boxes or standard workpieces[1], the robots for agriculture grasp agriculture products from nature or in the grocery[2], where products present a significant variety of shapes, sizes, and in a cluttered scenario. Ye et al.[3] proposed an improved GrabCut algorithm to extract the cucumber boundary under a complex background in the greenhouse. Feng et al.[4] designed a tomato harvesting robot in the greenhouse for fresh-eating tomatoes. For robotic grasping, the research of industrial products grasping goes ahead of the agricultural products grasp. At the early stage, grasp detection is commonly based on contour detection[5] and key shape detection[6] from RGB images as visual cues for robotic grasp. The typical drawback of these methods is that they are limited to the detection task in a simple and monochrome background. Template matching is a relatively advanced method for object detection with complex shapes and fixed geometric models by analyzing the corresponding similarity in color, contour, and texture features. Hinterstoisser et al.[7] proposed LineMod algorithm, which is created by combining the contour gradient direction of RGB image and the surface normal direction of depth image. Although the LineMod algorithm achieves high detection accuracy in industrial scenes, this template-based algorithm is not suitable for agricultural products perfectly.

In smart agriculture, the detection and grasp of agricultural products are preliminary for harvesting, sorting, and grading tasks. Clutter scenarios are very common for agricultural products because their shape and size vary, even though they are of the same variety. For this reason, a fixed model is not appropriate for this kind of object. Fortunately, deep learning is fit for object detection in complex backgrounds, and a lot of state-of-the-art algorithms have been proposed and validated in many research fields. Deep learning is widely used in object detection, grasp point detection[8], 2-dimensional grasp pose[9], and even 6-grasp pose detection[10]. Mao et al.[11] proposed a cucumber recognition method in a natural environment based on multi-path convolutional neural networks (MPCNN), color components, and support vector machine (SVM). Liu et al.[12] proposed a cucumber instance segmentation method based on improved Mask RCNN. Therefore, deep learning is applied for cucumber detection in this research.

However, there are fewer applications for cucumber grasping in a cluttered scenario, where objects with similar color and texture are stacked together, bringing challenges to the visual algorithm. Except for object detection, grasp planning plays a significant role in the success of grasping tasks, including grasping pose, and grasping order. According to humans' prior knowledge and reasoning, the best grasping order is the upper layer first, the non-occluded first, and the less obstacle previous. In Reference [13], the reasonable strategy of grasp order is researched, which can avoid the collapse of objects and increase grasping efficiency. For a cucumber with a long strip shape in the cluttered scene, the depth value on its surface varies from one point to another point. And its

Received date: 2022-03-24 **Accepted date:** 2022-11-14

Biographies: Fan Zhang, PhD, research interest: agriculture robot and deep learning, Email: bs20193070612@cau.edu.cn; **Zeyu Hou**, MD, research interest: agriculture robot, Email: cauhzy@163.com; **Jin Gao**, PhD, research interest: agricultural robot and computer vision, Email: jensenko@outlook.com; **Xue Deng**, PhD, research interest: bionic micro robot, Email: DengXue@cau.edu.cn.

***Corresponding author: Junxiong Zhang**, PhD, Associate Professor, research interest: agricultural robot, intelligent agricultural equipment, computer vision. College of Engineering, China Agricultural University, No.17 Qinghua Donglu, Haidian District, Beijing 100083, China. Tel: +86-10-62737726, Email: cau2007@cau.edu.cn.

obstruction degree is hard to numerical descript. The single modal data source, RGB image, or depth image is insufficient to predict the upper object or less-obstacle object.

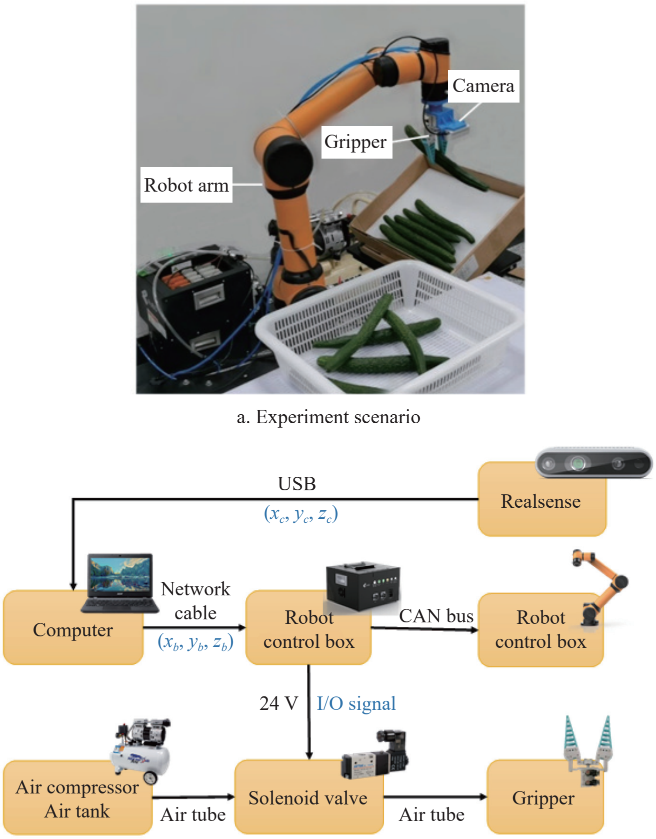
In this case, multimodal information fusion is needed. Tu et al.[14] used RGB and depth information for the last fusion. Sa et al.[15] compared the effect of early fusion and last fusion of different models on Faster R-CNN. The results show that the NIR channel added in early fusion is not compatible with the pre-training parameters from the common public dataset, which consists of only RGB images. Therefore, facing the limitation of the number of training datasets, a fusion method of the color image and depth image suitable for transfer learning needs to be proposed. Furthermore, a method to predict the geometric information and spatial position information of cucumber is necessary for cucumber grasping.

To sum up, the current research mainly focuses on workpiece detection and grasping in manufacturing environments and is more minor in agricultural products. Deep learning has a good performance in recognizing various agricultural products. Still, there is a lack of research on inner-class disambiguation in cluttered scenes, such as upper or lower cucumber, and occluded or non-occluded cucumber. Although some studies have proved that multimodal data fusion is conducive to target recognition, it has not been applied to the cucumber grasping task. In order to realize the cucumber grasping in a cluttered scene, this study constructs a specific dataset using the F-RGBD fusion method, training them based on deep learning algorithm and transfer learning technology, realizing the easy-grasping cucumber detection and cucumber pose estimation. The main contributions of this paper are as follows: 1) A fusion method F-RGBD was proposed, fusing RGB information and depth information, which can enhance the difference between the upper-layer cucumber and the lower-layer cucumber; 2) The prioritizing-grasping cucumber dataset was built and the deep learning instance segmentation algorithm Yolact++ was applied for cucumber segmentation; 3) A 4-dimensional cucumber grab line detection method was proposed, which can guide the robot gripper to hold the bending and non-bending cucumber stably. Finally, a continuous grasping experiment is carried out in the real world.

2 Materials and methods

2.1 Build robotic grasping system

An AUBO i5 6-DOF manipulator, a flexible two-finger gripper, and an Intel RealSense D435 camera are used to build an eye-in-hand system. The calculation equipment is a personal computer (CPU is Intel i5-10300H, GPU is Nvidia gtx1650Ti, memory is 16 G). Image acquisition, preprocessing, training, recognition, and grasping tasks compose the system, as shown in Figure 1. A customized connector was used to fix the camera and the gripper to the end of the manipulator. The flexible gripper was driven by the air circuit. The solenoid valve was connected to the output port of the manipulator controller. The controller was connected to the host computer through the network cable. The host computer sends commands to control the camera to collect images and control the actions of the manipulator and the flexible gripper. For the eye-in-hand system, the translation matrix ***T*** and rotation matrix ***R*** between the camera coordinate system and the manipulator coordinate system were obtained by hand-eye calibration. By the hand-eye system, the grasping position and the grasping line obtained by the recognition algorithm can be converted to the robot base coordinate system, so as to control the manipulator to complete the grasping task.



a. Experiment scenario

b. The pipeline of the control system

Note: The types of connections are written in black, and the information transported in the connection line is written in blue.

Figure 1 Hand-eye system

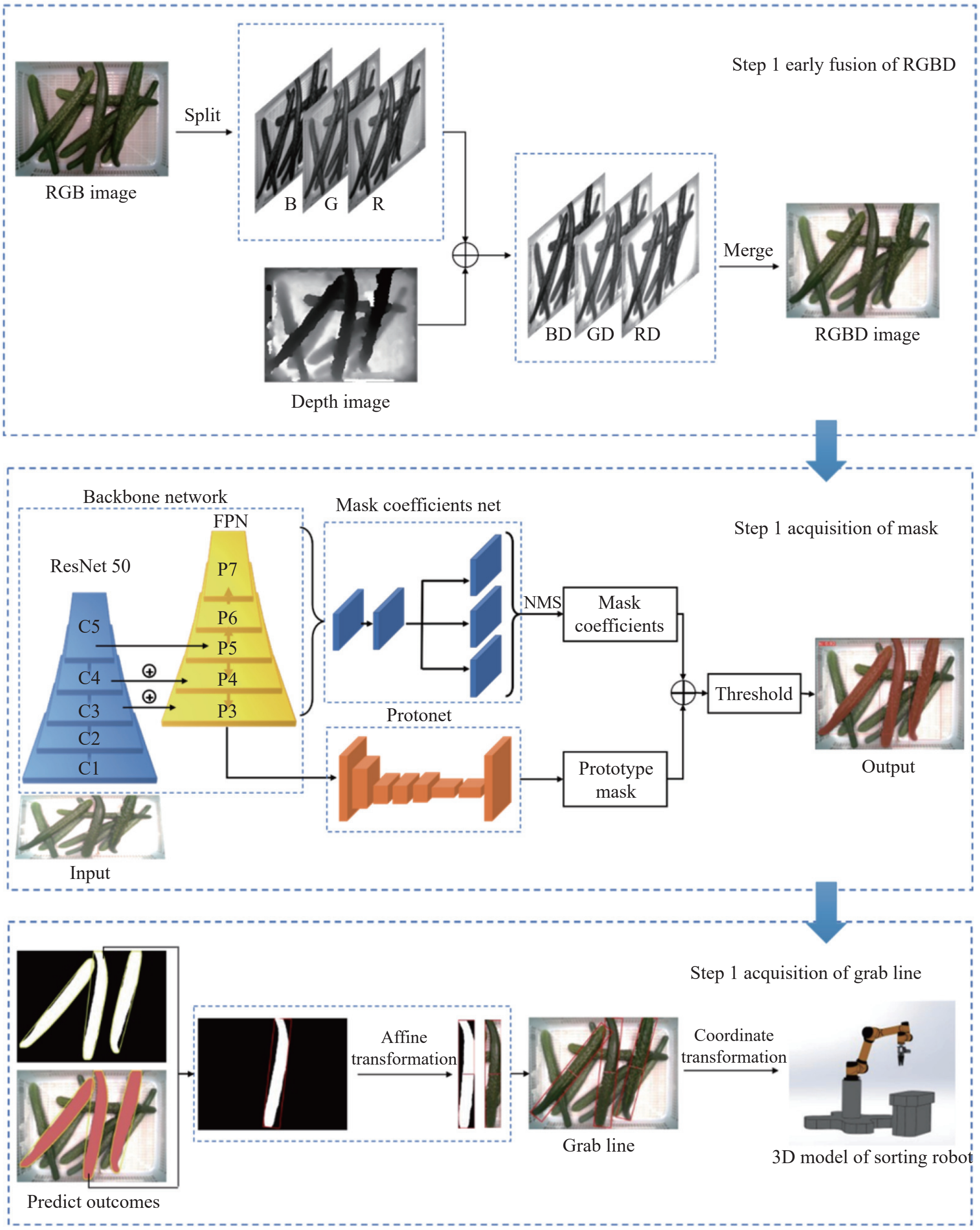
2.2 Image acquisition

Firstly, the appropriate distance from the end of the manipulator to the bottom of the box is controlled to ensure that the image covers the whole range of the box. In the indoor environment, 600 images were captured in different periods (morning, afternoon, evening) and different light intensities (turn on or turn off the light source), with a resolution of 640×480 pixels. The collected image contains cucumbers under different shapes, poses, and illumination conditions, which ensures the diversity of samples. The RealSense D435 camera was used to acquire the depth image corresponding to each image in black-to-white mode while collecting RGB images. Due to the influence of occlusion and uneven illumination, there were many hot pixels and holes in the depth information directly obtained by the camera. Therefore, time filter and hole filling filter are used for post-processing of depth information to remove hot pixels and fill holes. The time filter adjusts the depth value based on the previous frame. The hole-filling filter obtains the farthest or nearest value of the four pixels adjacent to the missing pixel in the depth image data for filling, and it can effectively complete the integrity of depth information.

Then, the 600 frames of aligned depth image and RGB image, namely RGBD image, were obtained from the RealSense and stored for subsequent processing. Each pixel of the RGB image has corresponding depth information.

2.3 Overview of the proposed method

The F-RGBD fusion method and deep learning instance segmentation algorithm were studied to detect the target's masks accurately and quickly. On the basis of the prediction mask, the acquisition method of grab lines in images was studied. The grab line was transformed into 3D space to complete the actual grasping task. Cucumber grasping was divided into three steps, multi-modal data fusion, cucumber instance segmentation, and four-dimensional grab line acquisition. In Step 1, the F-RGBD fusion method was illustrated. In Step 2, Yolact++ was used to detect the mask of the cucumber, and its network structure is shown in detail. Step 3 is the acquisition method of the grab lines in Pixel Coordinate System and in Robot Base Coordinate System. The proposed method in this study is shown in Figure 2, and the specific process is described in this section.



Note: R, G, B, and D refer to red, green, blue, and depth values in each pixel; FPN: Feature Pyramid Network; NMS: Non-Maximum Suppression.

Figure 2 Acquisition method of cucumber mask and the grab pose

2.4 Data fusion

In the common object detection tasks, there were multiple types of objects in one image, which means a large difference between objects. However, in the task of this study, there was a small difference between objects, where the color, and texture features between the cucumbers were similar. But their illumination, deformation, and pose were various. The most important is that the upper cucumbers are hoped to be distinguished from the lower cucumber, which benefits undamaged picking. A four-channel RGBD image could be a good option. However, in Reference [15], it is figured out that a direct-fused four-channel RGBD image has poor performance. The reason lies in the incompatibility between the RGBD data and the weight pre-trained by ImageNet, which consists of only RGB images.

Therefore, a novel method of fusing RGB image and depth image was proposed to generate a new image. The proposed fusion method in this study aimed to enlarge the difference of features between cucumbers, and persist in the good properties for fine-tuning training. A new three-channel image is calculated by Equation (1).

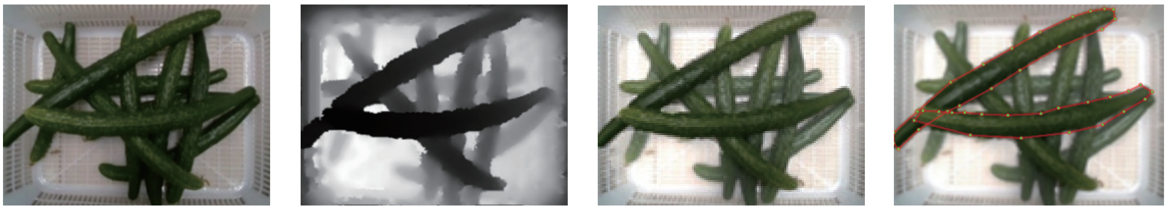
$$\begin{cases} C_1 = R\cdot\alpha + D \\ C_2 = G\cdot\alpha + D \\ C_3 = B\cdot\alpha + D \end{cases} \tag{1}$$

where, R, G, B, and D refer to red, green, blue, and depth values in each pixel; α is a parameter. C_1, C_2, and C_3 are the new values of each pixel at three channels of image. Then, each pixel value in the new image is normalized from 0 to 255 to get the F-RGBD image. This parameter α is figured out as 0.75 through analysis of the pixel of and background and foreground to maximize the g value in Equation (2), with a constraint to avoid excessive color distortions, through visual inspection.

$$g = w_0 \times (\mu_0 - \mu)^2 + w_1 \times (\mu_1 - \mu)^2 \tag{2}$$

$$\mu = w_0 \times \mu_0 + w_1 \times \mu_1 \tag{3}$$

where, g represents the difference between background and foreground. The w_0 represents the ratio of the number of pixels in the background to the number of pixels of the whole image. The w_1 represents the ratio of the number of pixels in the foreground to the number of pixels in the whole image. The μ, μ_0, and μ_1 represent the average pixel value of the pixels in the whole image, the background, and the foreground. Finally, these new three-channel images are named F-RGBD images, in Figure 3c.



a. RGB image b. Depth image c. F-RGBD image d. Labeled image

Figure 3 Cucumber images acquired by Intel RealSense D435 camera

2.5 Data labelling

The dataset is the basis of deep learning application research. Image labeling software LabelMe[16] was used to label images. A prioritizing policy was used during the dataset labeling, in which the mask of the prioritizing-grasping cucumber in each picture was marked. The criterion for prioritizing grasping is that it was located in the upper layer, and less obstacle, less possibility of causing collision and damage to other cucumbers in the execution of grasping. This standard contains human prior knowledge, not only semantic information, but also logical reasoning, decision-making, and knowledge of cucumber fruit characteristics, which was helpful for the algorithm to deal with cluttered scenes. This labelling method kept constant with the strategy of manual grasping significantly improves labeling efficiency and benefits the prediction of the upper-layer, less-obstacle cucumber. Based on the F-RGBD image and label file, the cucumber F-RGBD data set for cucumber grasping was constructed. In addition, the label file of the F-RGBD image was converted to a new label file for the corresponding RGB dataset, which served the comparative experiment. The dataset was divided into the training set, validation set, and testing set at the ratio 8:1:1.

2.6 Instance segmentation network

Both object detection and instance segmentation can be applied for cucumber detection tasks. However, only the rectangular bounding box and the center position of the object are obtained by the object detection algorithm. Therefore, it is impossible to obtain the pose information and accurate contour of the target. Instance segmentation has the advantage of semantic segmentation and object detection, precise masks, and individual object distinguishing. That is, it can classify different instances at the pixel level. For example, the segmentation method Mask RCNN, performs very well on MSCOCO datasets[17]. The mAP is 37.1, but the detection speed is only 5 fps. The algorithm relies on the prediction box of the two-stage target detection algorithm Faster RCNN to generate a mask, which leads to poor real-time performance. In order to solve this problem, fast instance segmentation Yolact and Yolact++ were proposed by Bolya et al.[18,19] based on a single-stage object detection method that mAP can achieve 34.1 at 33.5 fps on the COCO dataset. In this study, the Mask RCNN and Yolact++ are studied comparatively.

2.6.1 Backbone network (ResNet50+FPN)

The main function of the backbone network is to extract the features of the input image and generate the feature map. Although the deeper the network is, the more complex the model it can describe, problems such as gradient disappearance and degradation will appear. For this concern, the shortcut connection was introduced in the residual network, which effectively avoids these problems and improves the whole performance. In addition, the construction of the Feature Pyramid Network (FPN)[20] can achieve a multi-scale fusion of features, so as to better represent the target in a multi-scale range, benefiting the mask quality on the mask border at the pixel level. Therefore, FPN was introduced to extend the backbone network of both Mask RCNN and Yolact++. In terms of residual networks, Resnet101 and Resnet50 are two options. When Resnet101 + FPN is used as the backbone network, the mAP of the models is the highest in the COCO dataset. It is worth noting that compared with Resnet50+FPN, Resnet101+FPN has no significant advantage, but the computing speed is significantly slower. For this reason, Resnet50+FPN was used as the backbone network in this study to balance the performance and speed of detection.

2.6.2 Training

Facing the shortage of data, transfer learning is the most common technic. In this study, transfer learning was achieved through two steps: pre-training on the public dataset and fine-tuning on the cucumber dataset. Before training our small specific data, a pre-trained model trained by the MSCOCO dataset was introduced, so that the model parameters can be fine-tuned to the best results more quickly.

The MSCOCO dataset was a large image dataset designed for target recognition, detection, and image segmentation. There are 91 categories, and about 330 000 images were included in the dataset. During the fine-tuning phase, all layers are set as trainable. During fine-tuning, in order to recognize the cucumbers with the different area sizes, four anchors of area scale are designed: 48×48, 96×96, 192×192, and 384×384.

In addition, cucumbers were mostly long or nearly long in shape, and the ratio of length to width varies widely, from 0.33 to 3.00. Therefore, five length-width ratio anchors were used, including 1:1, 1:2, 1:3, 2:1, and 3:1. The loss function and other parameters are set as Reference [19].

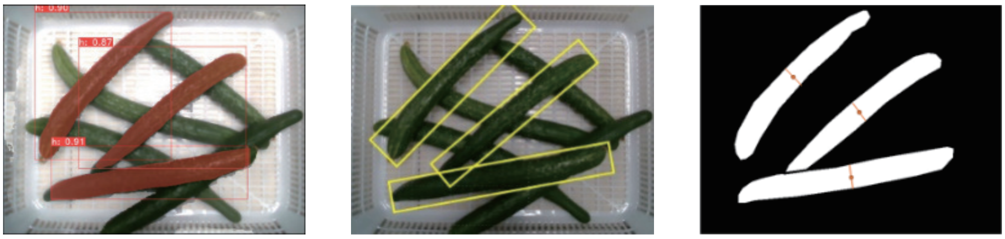
2.7 Acquisition of grab line

Pose information is the key to robot path planning. In this study, the grab line was used to describe the pose of the cucumber. The grab line contains the position information of the midpoint and the direction information of the straight line. These two key pieces of information serve to guide the robot arm's position and pose. Then, the 2D grab lines and positioning points can be converted to the 3D robot coordinate system through depth information and hand-eye conversion, so as to plan the robot path and complete the grasping.

2.7.1 Acquisition of grab line in pixel coordinate system

Using the trained network model to process the image, the cucumber mask and bounding box can be generated. However, the edge of the bounding box of the cucumber is horizontal or vertical to the image line, which can not express the posture of the cucumber, which is shown in Figure 4a. In Figure 4b, the minimum circumscribed rectangle of each cucumber is calculated by analyzing the segmented pixels. In Figure 4c, the desired grab line is marked in orange point and orange line.

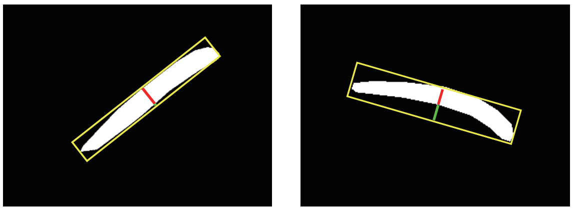
Therefore, it is necessary to generate a minimum bounding rectangle with the rotation angle according to the cucumber mask, as shown by the yellow rectangular box line in Figure 4b, so as to obtain the pose of the cucumber in the image. In the actual cucumber grasping task, considering the grasping stability, the centroid of the object is selected for grabbing. Therefore, the projection line of the cross-section at the half of the cucumber length is usually selected as the grab line, and the method is also used in this study. The convex hull is the smallest convex polygon that contains all the cucumber mask contour points, as shown by the yellow contour line in Figure 5. Cucumber or strip agricultural products similar to cucumber generally can be divided into two cases: non-bending and bending, which is shown in Figure 5.



a. Output of Yolact++ b. Minimum circumscribed rectangle of each mask c. Desired grab line of each cucumber

Note: The red pixels are the result of segmentation. The yellow rectangle is the minimum circumscribed rectangle. The white pixels and the black pixels represent the masks of the target object and background. The orange lines and orange points are the grab lines and their center point.

Figure 4 Obtaining the grab line of cucumber images



a. Non-bending cucumber b. Bending cucumber

Note: The yellow rectangle is the smallest bounding rectangle.

Figure 5 White area is the segmentation of the cucumber

In the first case, the transverse diameter at half of the length of the cucumber mask almost coincides with the symmetric line of the long side of the minimum bounding rectangle, as shown in the upper left corner of Figure 5a. In this case, the symmetric line of the long side of the smallest outer rectangle can be equivalent to the grab line.

However, in the second case, there is a wide gap between the mask and the side of the rectangle, which is marked as a green line in Figure 5b. In this case, the symmetric line of the long side of the rectangle can't represent the grab line. In order to obtain a more accurate grab line and location, the intersection line between the symmetric line of the long side and the cucumber mask is taken as the grab line, as shown in Figure 5b. The processing details are as follows:

Firstly, the small image in the bounding rectangle was cropped and its four vertices were obtained for the affine transformation. The affine transformation matrix was obtained by following Equations (4), and then the bounding rectangle was transformed into a new image by affine transformation.

$$\begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} = \boldsymbol{M} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\gamma) & -\sin(\gamma) & t_u \\ \sin(\gamma) & \cos(\gamma) & t_v \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \tag{4}$$

where, (u, v) and (u', v') represent the pixel coordinate in the original image and the new image. $\boldsymbol{M}$ is the affine transformation matrix. γ, t_u, and t_v refer to the rotation and translation parameters for affine transformation. In the new image, the long symmetric axis is parallel with the row of the image.

Secondly, the pixels in the short symmetric axis of the new image are traversed pixel by pixel, to find the first and last intersection of the short symmetric axis and the cucumber mask. These two points were used as the two ends of the grab line.

Finally, the two endpoints were restored to the original image pixel position through the inverse affine transformation. Finally, according to Equation (5), the middle point of the grab line was taken as the grabbing position.

$$\begin{cases} u_0 = \dfrac{u_1 + u_2}{2} \\ v_0 = \dfrac{v_1 + v_2}{2} \end{cases} \tag{5}$$

where, (u_1, v_1), (u_2, v_2), and (u_0, v_0) refer to pixel coordinates of the two ends of the grab line and the grab position. This method of grab line and grab point detection is suitable for cucumber bending and non-bending and ensures the accuracy of the grab line.

2.7.2 Acquisition of the grab line in robot coordinate system

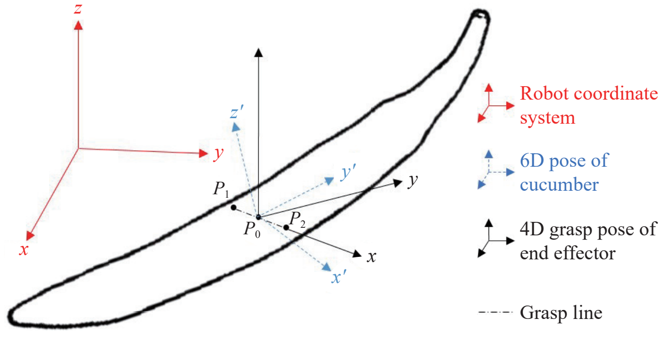
In order to complete the actual grasping task, it is necessary to provide the robot with 3D pose information of the grasping position in the robot base coordinate system. Therefore, the 2D information of the grab line in the image coordinate system was transformed into the robot base coordinate system to obtain the corresponding spatial information in the robot base coordinate system. Firstly, the coordinate value of the grab position in the camera coordinate system (x_c, y_c, z_c) is calculated by Equation (6).

$$\begin{cases} x_c = (u-u_0)z_c/f_x \\ y_c = (v-v_0)z_c/f_y \\ z_c = z_c \end{cases}, \quad \boldsymbol{M}_{\text{in}} = \begin{bmatrix} f_x & 0 & u_0 \\ 0 & f_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{6}$$

where, $\boldsymbol{M}_{\text{in}}$ is the internal parameter matrix of a camera, which is obtained from camera calibration; f_x and f_y are the focal length parameters of the camera. z_c is the depth value of the grab position in the camera coordinate system, which is obtained directly from the depth map. Next, the coordinate value in the camera coordinate system (x_c, y_c, z_c) was converted to the corresponding coordinate value in the robot base coordinate system (x_b, y_b, z_b) through Equation (7).

$$\begin{bmatrix} x_b \\ y_b \\ z_b \\ 1 \end{bmatrix} = \begin{bmatrix} \boldsymbol{R} & \boldsymbol{T} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} = \boldsymbol{M}_b \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} \tag{7}$$

where, $\boldsymbol{M}_b$ is the coordinate transformation matrix between the camera coordinate system and the robot base coordinate system, which was obtained by hand-eye calibration: $\boldsymbol{R}$ is a 3×3 rotation matrix and $\boldsymbol{T}$ is a 3×1 translation matrix. Similarly, the coordinates of the two ends of the grab line (u_1, v_1), (u_2, v_2), and (u_0, v_0) can be used to calculate the coordinates in the robot base coordinate system $\boldsymbol{P}_1$=(x_{b1}, y_{b1}, z_{b1}), $\boldsymbol{P}_2$=(x_{b2}, y_{b2}, z_{b2}), $\boldsymbol{P}_0$=(x_{b0}, y_{b0}, z_{b0}) by Equation (7) and Figure 6.



Note: The transition between the grasp pose and the robot coordinate system consists of a translation and a rotation around the *z*-axis.

Figure 6 Three coordinate systems represent the robot coordinate system, the 6D pose of the cucumber, and the 4D grasp pose of the end-effector

It needs to be mentioned that the point with a greater v value is (u_1, v_1), and the point with a less v value is (u_2, v_2), so the direction of the grab line is always towards the bottom of the image, which can avoid the angle wrap-around problem. Thus, the rotation angle θ of the grab line relative to the z-axis of the robot base coordinate system can be obtained by Equation (8).

$$\theta = \arctan\left(\frac{y_{b2}-y_{b1}}{x_{b2}-x_{b1}}\right) \tag{8}$$

When the center of mass of the object was grabbed by the flexible two-finger gripper, the object was in a stable state. That is, the 4D grasping line can meet the grasping conditions, including a spatial three-dimensional coordinate, the rotation angle around the z-axis. The rotation angle of the grabbing line relative to the z-axis of the base coordinate system was taken as the grab pose. During the execution, the gripper on the end of the robot arm reaches above the grab point first, keeping the x-O-y plane parallel to the ground, and then the gripper rotates angle θ around its z-axis. The opening width of the gripper was 10 mm wider than the width of the grasping line, which can ensure that the fingertips of the gripper are correctly inserted into the gap and separate the target cucumber from other cucumbers. Finally, the robot arm approaches the cucumber in the direction perpendicular to its x-O-y plane, and closes the gripper at the target point, to grasp and transport the cucumber into the target box in order.

3 Results and discussion

In order to test the effect of F-RGBD dataset and RGB dataset used in this paper, based on the same backbone feature extraction network ResNet50, two instance segmentation network models Mask RCNN and Yolact++ were used to carry out comparative experiments, including the following three parts.

1) In order to test the effect of F-RGBD data set compared with RGB data set, different instance segmentation networks were used to measure the predicated cucumber mask. Including the measurement of mask AP and mask quality. Mask AP was measured by changes in the number of training iterations and Intersection over Union (IoU) threshold. The quality of the mask was measured by the continuity of the mask and the Mask IoU.

2) In order to test the performance of different combinations in the grasping scene, the positioning accuracy of the grab positioning point and the attitude angle accuracy of the grab line were evaluated.

3) In order to verify the effect of the cucumber mask and grab line prediction method proposed in this study, the detection and grasping experiment was carried out in real world.

3.1 Accuracy evaluation cucumber target segmentation

3.1.1 Metric of Mask AP

The Average Precision (AP) is an evaluation index to measure network performance. Its value is related to the Precision (P) and Recall rate (R). The higher the AP value, the better the performance of the network model. Precision refers to the proportion of correct prediction results in all prediction results. The Recall represents the proportion of the correct prediction results in the total number of ground truth in the dataset. The calculation processes are shown as the following equations Equation (9), Equation (10), and Equation (11), respectively.

$$\text{AP} = \int_0^1 PR\,\mathrm{d}R \tag{9}$$

$$P = \frac{\text{TP}}{\text{TP}+\text{FP}} \tag{10}$$

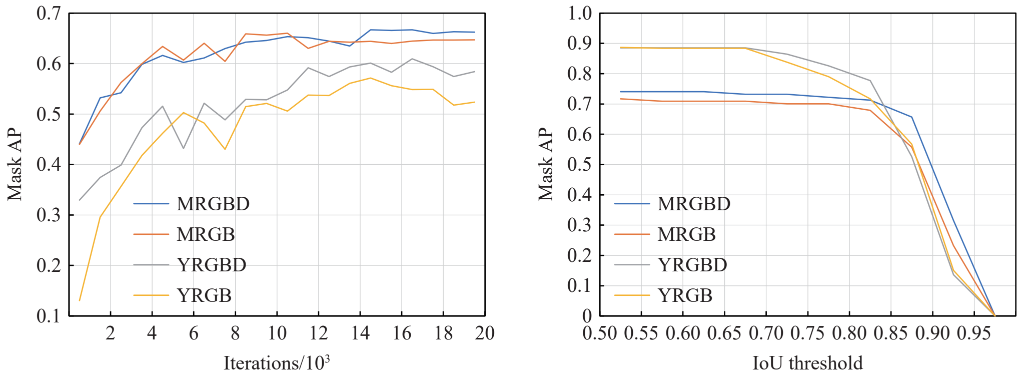
$$R = \frac{\text{TP}}{\text{TP}+\text{FN}} \tag{11}$$

where, True Positive (TP) is the number of cases that are positive and detected as positive, False Positive (FP) is the number of cases that are negative but detected as positive, and False Negative (FN) is the number of cases that are positive but detected as negative. MRGBD, MRGB, YRGBD, and YRGB are symbols of four combinations, which are the combination of Mask RCNN and F-RGBD/RGB datasets and the combination of Yolact++ and F-RGBD/RGB datasets, respectively.

3.1.2 Results of Mask AP

Figure 7a shows the change of Mask AP of four combination validation sets with the increase of training rounds. At the 1000-th

Iteration, Mask AP of YRGBD and YRGB were 0.3294 and 0.1306, respectively. At the peak, the Mask AP of YRGBD reaches 0.6092, which is higher than the optimal value of 0.5712 of YRGB. The Mask AP of MRGBD is 0.6671, and that of MRGB is 0.66. The F-RGBD dataset shows better convergence ability compared with the RGB dataset.



a. Trend of Mask AP versus training iterations

b. Mask AP at different IoU threshold

Note: MRGBD, MRGB, YRGBD, and YRGB are symbols of four combinations, which are the combination of Mask RCNN and F-RGBD/RGB data sets and the combination of Yolact++ and F-RGBD/RGB data sets, respectively. Same below. IoU threshold 0.50-0.95.

Figure 7 Changes of Mask AP of four combination validation sets with the increase of training rounds

Figure 7b and Table 1 show the evaluation results of the four combinations in the validation set at different Mask IoU thresholds. There is a critical IoU threshold is 0.85. Before 0.85, the Mask AP of Mask RCNN is better than Yolact++ in both datasets, but the Mask AP of Yolact++ is higher than Mask RCNN in high-quality mask area. Taking References [21] and [22] as references, the higher the IoU, the higher the quality of the prediction mask. Thus, the Yolact++ has better performance in predicting masks with higher IoU. As listed in Table 1, compared with YRGB, the Mask AP of the proposed YRGBD is 9.95% and 8.31% higher than YRGB, and 13.11% and 17.85% higher than MRGBD, at the threshold of 0.85 and 0.90 respectively.

Table 1 Mask AP at specific IoU threshold

Combination	IoU threshold											
	0.50	0.55	0.60	0.65	0.70	0.75	0.80	0.85	0.90	0.95	Mean@0.50	Mean@0.80
YRGBD	0.7407	0.7407	0.7407	0.7321	0.7321	0.7321	0.7219	**0.6566**	**0.3150**	0.00	0.6092	**0.4621**
YRGB	0.7168	0.7088	0.7088	0.7088	0.7002	0.7002	0.6792	0.5571	0.2319	0.00	0.5712	0.3671
MRGBD	0.8855	**0.8855**	**0.8855**	**0.8855**	**0.8642**	**0.8258**	**0.7772**	0.5255	0.1365	0.00	**0.6671**	0.3598
MRGB	**0.8864**	0.8835	0.8835	0.8835	0.8378	0.7902	0.7172	0.5675	0.1504	0.00	0.6600	0.3588

In addition, compared with MRGB, the segmentation effect of MRGBD is slightly better. In general, the performance of YRGBD is significantly better in high-quality mask prediction.

3.1.3 Metric of Mask quality

In order to accurately obtain the pose of the cucumber, the quality of the predicted mask should first be considered, including the continuity and the Mask IoU of the predicted mask. Mask IoU refers to the proportion of intersection and union between the mask predicted value (PD) and its ground truth (GT) of a target object, which can be calculated by Equation (12).

$$\mathrm{IoU} = \frac{(\mathrm{PD} \cap \mathrm{GT})}{\mathrm{PD} \cup \mathrm{GT}} \tag{12}$$

The continuity of a target prediction mask (Continuity) is determined by the number of connected domains (N_d), as Equation (13). If the number of connected domains is equal to 1, the prediction value of the mask is considered to be continuous; otherwise, it is considered to be discontinuous.

$$\mathrm{Continuity} = \begin{cases} \mathrm{True}, & N_d = 1 \\ \mathrm{False}, & N_d > 1 \end{cases} \tag{13}$$

When the prediction value of the mask is discontinuous, the grab line generated by a certain part of the mask is inaccurate.

3.1.4 Results of Mask quality

The results of the above two indicators are listed in Table 2. Figure 8 shows the box plot and normal distribution curve of Mask IoU on the four combinations. Observing the statistical parameters, the YRGBD's average of Mask IoU is higher than others and has a more concentrated distribution.

Table 2 Key parameters of the predicted Mask's IoU value

Combination	Mean	STD	Ratio of continuity	FPS
YRGBD	0.8895	0.0894	93.62%	11.62
YRGB	0.8677	0.1500	92.63%	12.04
MRGBD	0.8526	0.1449	80.17%	3.06
MRGB	0.8480	0.1634	78.33%	3.83

Note: Mean refers to the mean value of Mask IoU. STD refers to the standard deviation of Mask IoU. The Ratio of Continuity refers to the ratio of the number of continuous cucumber masks to the number of cucumbers. FPS is the speed unit of detection, frames per second (fps).

Table 2 lists the values of three statistical parameters of Mask IoU value and one quality indicator in four combinations, including the mean, the standard deviation, and the ratio of the continuous mask. The Mask IoU predicted by YRGBD is concentrated around 0.8895 with a standard deviation of 0.0894, and the rate of continuity of the predicted mask is 93.62%. In conclusion, the mask effect predicted by YRGBD is the best. It is worth noting that Yolact++ not only has a better effect but also has a faster average detection speed. The detection speed of YRGBD is 11.62 fps, which is more than three times as fast as MRGBD, due to the single-stage algorithm flow of Yolact++.

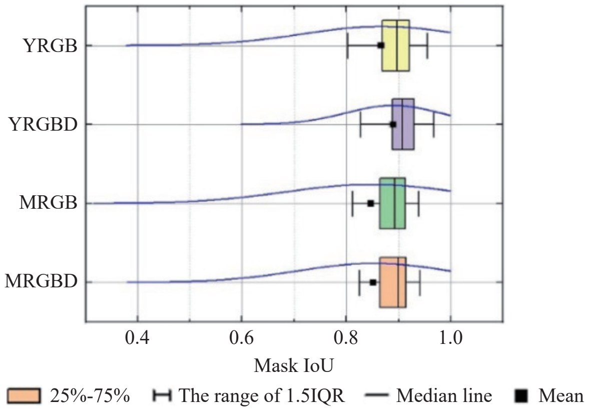


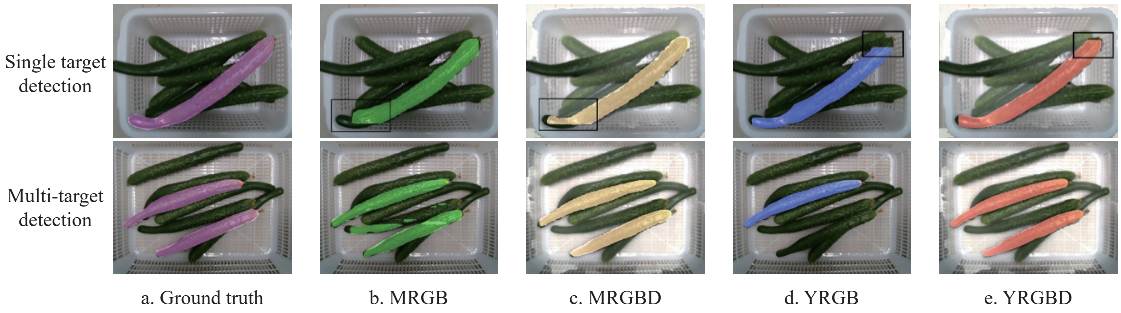
Figure 8 Distribution diagram of predicted Mask's IoU value

3.2 Benefits of data fusion and the network structure

Compared with the RGB dataset, the F-RGBD dataset has advantages in a variety of situations. On the one hand, the mask contour predicted by the model trained by the F-RGBD dataset is closer to the ground truth in most cases, as shown in Figure 9. After the depth information is superimposed on the image, there will be obvious contour around the object due to the difference in depth information, which is helpful to the segmentation of different instances in the cluttered scene. On the other hand, Figure 9 shows the segmentation result of densely stacked cucumber instances with different depths. Due to the difference in depth information, the F-RGBD dataset can better distinguish cucumbers at different depths with complete contour but different depths. The research results show that the early fusion method of RGB and Depth information proposed in this study helped to improve the detection effect.

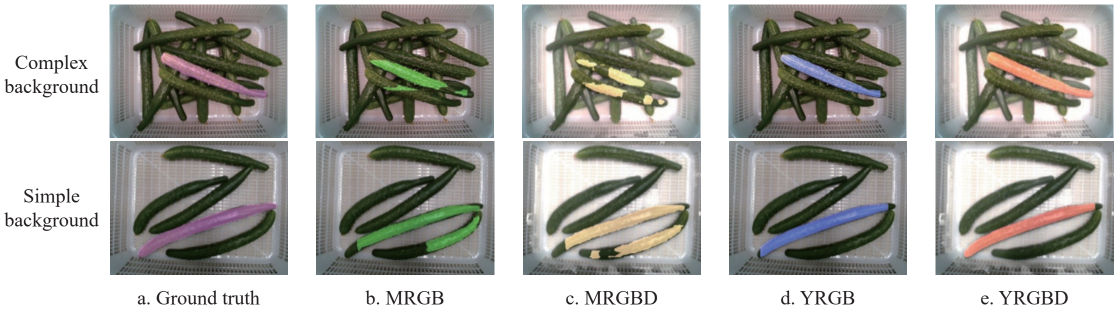
For small datasets, it is necessary to train the network with fine-tuning technic on the model pre-trained by large general data sets. The F-RGBD image obtained by the early fusion proposed by us is still a three-channel image, which can make good use of the pre-training model. This expectation was verified by experimental results. It is foreseeable that the method we propose has a good reference for the grasping of other items in a cluttered environment.

Compared with Mask RCNN, Yolact++ has higher accuracy of mask prediction. Specifically, Yolact++ can be used to distinguish the dense cucumber preferably. Figure 10 shows the benefits of using Yolact++ and shows two cases with different backgrounds.



a. Ground truth b. MRGB c. MRGBD d. YRGB e. YRGBD

Figure 9 Segmentation results of densely stacked cucumber instances with different depths



a. Ground truth b. MRGB c. MRGBD d. YRGB e. YRGBD

Figure 10 Effects of using Yolact++ for two cases with different background

The Mask RCNN, a two-stage instance segmentation algorithm, needs to be based on the proposal region. However, the area of the proposal region generated by the network for the long-strip agricultural products is usually large, containing multiple adjacent instances in one suggestion box, which interferes with the detection of the main target cucumber. Therefore, even if the confidence score is high, segmentation errors can easily be caused by the adjacent cucumber in the proposal region. Error types include misclassification of other cucumbers in the proposal region into a part of the target cucumbers, incomplete segmentation of the target cucumbers, etc. This problem can be effectively avoided by using Yolact++, which is a typical one-stage instance segmentation method based on global images.

3.3 Evaluation of grab line

3.3.1 Metric of grabbing line accuracy

The precision of grab line includes grab positioning precision and grabbing line pose precision. Two indicators, the distance error, and the angle error, are analyzed in the following. The Euclidean distance between the predicted value and the true value of the grasping position (Δd) was calculated by Equation (14) to evaluate the positioning accuracy of the grasping position.

$$\Delta d = \sqrt{\left(u_p - u_g\right)^2 + \left(v_p - v_g\right)^2} \tag{14}$$

where, (u_p, v_p) is the pixel coordinates of the predicted position, (u_g, v_g) is the pixel coordinates of the ground truth value of the grasping position, Δd is the Euclidean distance between them.

The following Equation (15) was used to calculate the absolute value of the angle difference between the predicted value and the ground truth of the grab line, so as to evaluate the pose accuracy of the grab line.

$$\Delta\theta = |\theta_p - \theta_g| \tag{15}$$

where, θ_p is the angle between the predicted value of the grab line and the u-axis in the image coordinate system, θ_g is the angle between the ground truth value of the grab line and the u-axis in the image coordinate system, and $\Delta\theta$ is the absolute value of the difference between them.

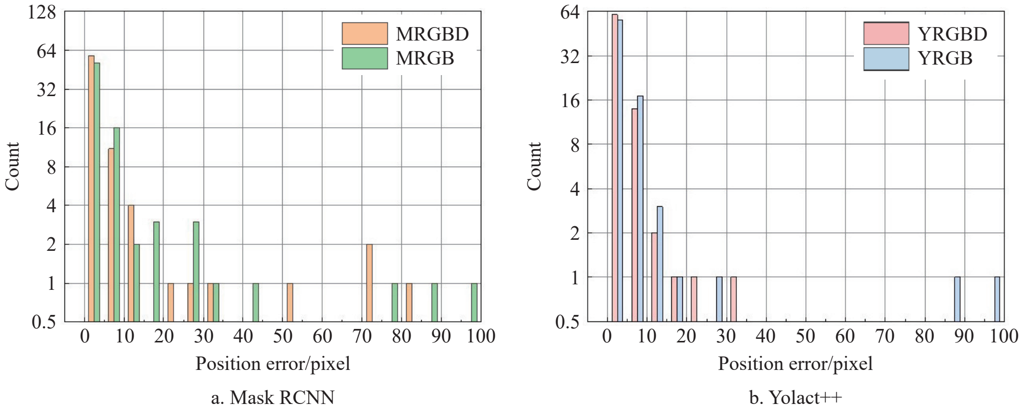
3.3.2 Results of grab line prediction

The mean value and standard deviation of position and angle errors of the four combination prediction values are listed in Table 3.

The distribution of position errors of the four combinations is shown in Figure 11. Specifically, positioning errors are divided into 20 groups within the range of 0-100 pixels, taking 5 pixels as an interval. Compared with the other combinations, the distribution of YRGBD was the most concentrated. The samples predicted by YRGBD with a position error of less than 10 pixels accounted for 93.75% of the total. The proportions of the other three combinations were 86.25%, 83.75%, and 91.25%, respectively. As shown in the first two columns of Table 3, the mean, and standard deviation of YRGBD positioning error are the smallest.

Table 3 Accuracy evaluation of grab line

Combination	Mean position error/pixels	Standard deviation of position error/ pixels	Mean angle error/(°)	Standard deviation of angle error/(°)
YRGBD	**3.7941**	**5.1552**	**0.7357**	**0.8860**
YRGB	6.3814	14.7962	1.9502	7.1231
MRGBD	7.8367	15.3753	1.7870	5.1424
MRGB	9.2062	17.6933	2.1865	6.5587

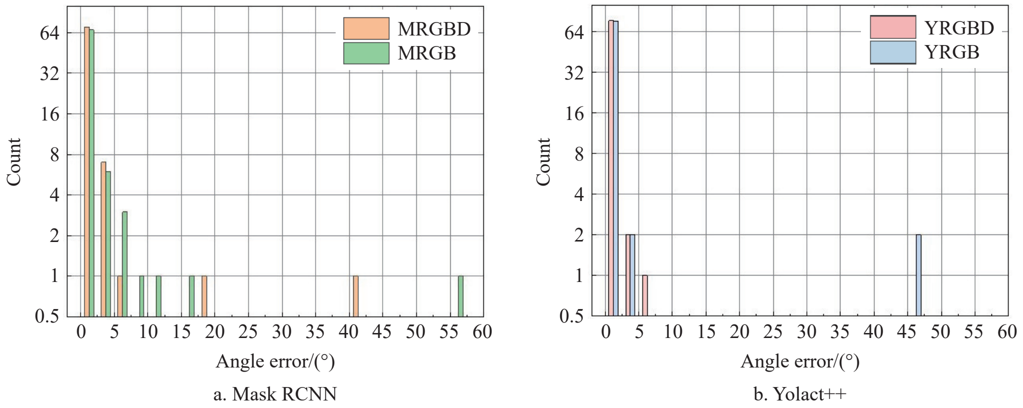


a. Mask RCNN

b. Yolact++

Figure 11 Distribution diagrams of the position errors of predicted grab lines

The distribution of angle errors of the four combinations is shown in Figure 12. Specifically, the angle errors were divided into 24 groups at the range of 0°-60°, taking 2.5° as an interval. The angle error distribution trend of the four combinations was also roughly the same as the position errors. The angle errors of YRGBD were less than 7.5°.



a. Mask RCNN

b. Yolact++

Figure 12 Distribution diagrams of the angle errors of predicted grab lines

Combined with the data of the last two columns in Table 3, the angle error result of YRGBD is the minimum. According to the results, it is worth noting that compared with the RGB dataset, the prediction effect in the two models has been improved by the F-RGBD dataset. Moreover, each index has been significantly improved when using Yolact++.

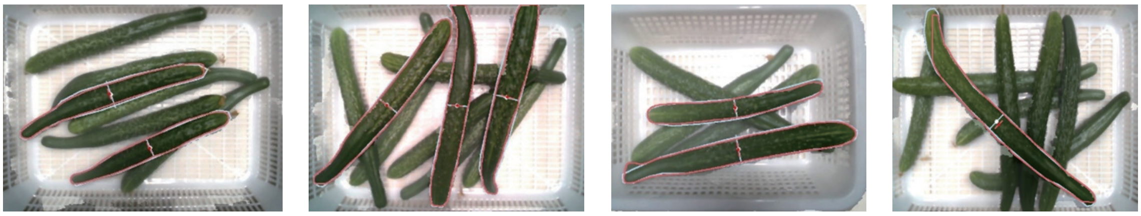
3.3.3 Visualization of grab line

Figure 13 shows the prediction results of YRGBD in different scenarios. In most scenarios, the top cucumber to be sorted is separated by a certain distance, and the prediction contour and grab line of YRGBD is very close to the ground truth, as shown in Figures 13a and 13b. However, when there is other interference information or the depth values of the same instance are greatly different, the predicted contour will be different from the ground truth at the edge, and the corresponding grasping pose will also deviate, as shown in Figures 13c and 13d.

In general, the advantages of position error and angle error of YRGBD are significant.

3.4 Detection and Grasping tests

The tests based on Yolact++ and F-RGBD data sets are conducted by an eye-in-hand system. In the indoor environment, the experiment was carried out in three different time periods: 10:00 to 11:00, 15:00 to 16:00, and 20:00 to 21:00. First, a certain number of 50 cucumbers were randomly selected and scattered in the box. Then the test is conducted according to the workflow. In each test, a total of 120 grasping actions were executed. Table 4 lists the



a. Correct example containing two target

b. Correct example containing three target

c. False prediction of contour caused by interfering information

d. Deviation of grab line caused by extreme deviation of depths

Note: The white contour, white line, and the white filled dot are the ground truth of the contour and grab line, and the grab position. The red contour, white line, and the white filled dot are the prediction of the contour and grab line, and the grab position.

Figure 13 Prediction results of YRGBD of cucumbers in different scenarios

success rate of three experiments, the average results of the three experiments, and the average cycle time of a single cucumber. It must be mentioned that bad light condition leads to a relatively lower success rate at night. Auxiliary lighting could be helpful for cucumber detection at night.

Table 4 Results of robotic grasping system tests

Index	Test1 10:00-11:00	Test2 15:00-16:00	Test3 20:00-21:00	Mean
Success rate of picked	94.17%	95.00%	92.50%	93.67%
Average sorted time/s	9.70	10.10	9.80	9.87

In the grabbing tests, grabbing failures are mainly the following cases.

1) In the case of a partial close arrangement, the gripper was disturbed by cucumbers on both sides of the target cucumber, resulting in failure of grasping.

2) When some cucumbers are located at the boundary of the box, the gripper interferes with the frame of the box, which leads to the failure of grasping.

Despite all this, the cucumber that fails to be picked will continue to be identified and picked in the next action until there are no remaining cucumbers in the box. Because the physical limitations of all the joints, and the inherent singularities of a 6-DOF manipulator can restrict the overall working space[23]. The 6-DOF manipulator used in this study cannot get a reasonable Inverse solution for some 6D poses (including translation along the x, y, and z axes and rotation around the x, y, and z axes) in the workspace but can get a reasonable Inverse solution for any 4D grasp line (including translation along the x, y, and z axes and the angle of the grasp line). Compared with the 6D pose detection, which is based on 3D point cloud processing and with high computational cost[24], the 4D grab line detection algorithm used in this study was more effective. Both the success rate of grabbing and the processing speed are taken into account.

4 Conclusions

Aiming at cucumber grasping in the unordered scenes, a method based on the Yolact++ network for a 4D grab line of cucumber was proposed. In addition, a new early fusion method was proposed, named F-RGBD, fusing RGB image and depth image into a new three-channel image. A better convergence and better discriminative ability were shown on F-RGBD dataset. F-RGBD fused RGB image and depth image into a new three-channel image, and compatible with the weights pre-trained by common public datasets naturally, such as the COCO dataset. Different instance segmentation models Mask RCNN and Yolact++ were used to compare the effect of the F-RGBD dataset and RGB dataset. From comparative experiments of four combinations, the combination of Yolact++ and F-RGBD data got higher quality masks faster than others. Its Mask IoU was concentrated around 0.8895 with a standard deviation of 0.0894, and the rate of continuity of the predicted mask was 93.62%. The positioning error of 93.75% predicted grab lines were less than 10 pixels. The angle errors of all grab lines are less than 7.5°. The experimental results show that YRGBD has obvious advantages in high-quality mask prediction, and has less position error and angle error. In a real-world experiment, the average success rate was 93.67%, and the average speed was 9.87 s per attempt. The effectiveness of the cucumber segmentation and grab line acquisition method in cluttered scenes was verified by experimental results.

In the future, the robustness of the recognition algorithm should be further improved, and the structure of end effector should be optimized for the case of partial dense arrangement, to adapt to a variety of scenes in the cluttered environment.

Acknowledgements

This work was financially supported by the Beijing Innovation Consortium of Agriculture Research System (BAIC12).

[References]

[1] Cheng W-C, Hsiao H-C, Lin Y-P, Liu Y-H. Robotic arm pick-and-place system for L-shaped water pipe joint with arbitrary angle. In: 2020 International Automatic Control Conference (CACS), Hsinchu: IEEE, 2020; pp.1–5. doi: 10.1109/CACS50047.2020.9289721.

[2] Triantafyllou P, Mnyusiwalla H, Sotiropoulos P, Roa M A, Russell D, Deacon G. A Benchmarking Framework for systematic evaluation of robotic pick-and-place systems in an industrial grocery setting. In: 2019 International Conference on Robotics and Automation (ICRA), Montreal: IEEE, 2019; pp.6692–6698. doi: 10.1109/ICRA.2019.8793993.

[3] Ye H J, Liu C Q, Niu P Y. Cucumber appearance quality detection under complex background based on image processing. Int J Agric & Biol Eng, 2018; 11(4): 193–199.

[4] Feng Q C, Zou W, Fan P F, Zhang C F, Wang X, Design and test of robotic harvesting system for cherry tomato. Int J Agric & Biol Eng, 2018; 11(1): 96–100.

[5] Cornelius H, Kragic D, Eklundh J O. Object and pose recognition using contour and shape information. In: ICAR '05 Proceedings 12th International Conference on Advanced Robotics, 2005; pp.613–620. doi: 10.1109/ICAR.2005.1507472.

[6] Rahardja K, Kosaka A. Vision-based bin-picking: recognition and localization of multiple complex objects using simple visual cues. In: Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS '96, Osaka: IEEE, 1996; pp.1448–1457. doi: 10.1109/IROS.1996.569005.

[7] Hinterstoisser S, Cagniart C, Ilic S, Sturm P, Navab N, Fua P, et al. Gradient response maps for real-time detection of textureless objects. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2012; 34(5): 876–888.

[8] Liu W H, Pan Z Y, Liu W J, Shao Q Q, Hu J, Wang W M, et al. Deep learning for picking point detection in dense cluster. In: 2017 11th Asian Control Conference (ASCC), Gold Coast: IEEE, 2017; pp.1644-1649. doi:

10.1109/ASCC.2017.8287420

[9] Redmon J, Angelova A. Real-time grasp detection using convolutional neural networks. In: 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle: IEEE, pp.1316–1322. doi: 10.1109/ICRA.2015.7139361.

[10] Kehl W, Manhardt F, Tombari F, Ilic S, Navab N. SSD-6D: Making RGB-based 3D detection and 6D pose estimation great again. In: 2017 IEEE International Conference on Computer Vision (ICCV), Venice: IEEE, 2017; pp.1530–1538. doi: 10.1109/ICCV.2017.169.

[11] Mao S H, Li Y H, Ma Y, Zhang B H, Zhou J, Wang K. Automatic cucumber recognition algorithm for harvesting robots in the natural environment using deep learning and multi-feature fusion. Computers and Electronics in Agriculture, 2020; 170: 105254.

[12] Liu X Y, Zhao D A, Jia W K, Ji W, Ruan C Z, Sun Y P. Cucumber fruits detection in greenhouses based on instance segmentation. IEEE Access, 2019; 7: 139635–139642.

[13] Zhao W R, Wang J C, Chen W D, Huang Y. Planning for grasping cluttered objects based on obstruction degree. International Journal of Advanced Robotic Systems, 2021; 18(6): 17298814211040632. doi: 10.1177/17298814211040632

[14] Tu S Q, Pang J, Liu H F, Zhuang N, Chen Y, Zheng C, et al. Passion fruit detection and counting based on multiple scale faster R-CNN using RGB-D images. Precision Agriculture, 2020; 21: 1072–1091.

[15] Sa I, Ge Z Y, Dayoub F, Upcroft B, Perez T, McCool C. DeepFruits: A fruit detection system using deep neural networks. Sensors, 2016; 16(8): 1222.

[16] Russell B C, Torralba A, Murphy K P, Freeman W T. LabelMe: a database and web-based tool for image annotation. International Journal of Computer Vision, 2008; 77: 157–173.

[17] Lin T-Y, Maire M, Belongie S, Hays J, Perona P, Ramanan D, et al. Microsoft COCO: Common objects in context. In: Computer Vision – ECCV 2014; 8693: 740–755. doi: 10.1007/978-3-319-10602-1_48.

[18] Bolya D, Zhou C, Xiao F Y, Lee Y J. YOLACT: Real-time instance segmentation. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul: IEEE, 2019; pp.9156–9165. doi: 10.1109/ICCV.2019.00925.

[19] Bolya D, Zhou C, Xiao F Y, Lee Y J. YOLACT++: Better real-time instance segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2022; 44(2): 1108–1121.

[20] Lin T-Y, Dollar P, Girshick R, He K M, Hariharan B, Belongie S. Feature Pyramid Networks for object detection. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu: IEEE, 2017; pp.936-944. doi: 10.1109/CVPR.2017.106.

[21] He K M, Gkioxari G, Dollar P, Girshick R. Mask R-CNN. In: 2017 IEEE International Conference on Computer Vision (ICCV), Venice: IEEE, 2017; pp.2980-2988. doi: 10.1109/ICCV.2017.322.

[22] Wu M H, Yue H H, Wang J, Huang Y X, Liu M, Jiang Y H, et al. Object detection based on RGC mask R-CNN. IET Image Processing, 2020; 14(8): 1502–1508.

[23] Cheng F-T, Hour T-L, Sun Y-Y, Chen T-H. Study and resolution of singularities for a 6-DOF PUMA manipulator. IEEE Transactions on System, Man, Cybernetics, 1997; 27(2): 332–343.

[24] Yu S S, Wang C, Wen C L, Cheng M, Liu M H, Zhang Z H, et al. LiDAR-based localization using universal encoding and memory-aware regression. Pattern Recognition, 2022; 128: 108685.