A method for determining the optimal placement of litchi clusters using improved YOLACT and a distributed target fault-tolerance mechanism

Yuanhong Li[1,2,3], Jing Wang[1], Jiapeng Liao[1], Yubin Lan[1,2,3*]

(1. *South China Agricultural University, College of Electronic Engineering, Guangzhou 510642, China*;
2. *Guangdong Laboratory for Lingnan Modern Agriculture, Guangzhou 510642, China*;
3. *National Center for International Collaboration Research on Precision Agricultural Aviation Pesticides Spraying Technology (NPAAC), South China Agricultural University, Guangzhou 510642, China*)

Abstract: At present, there is no efficient and accurate method for locating the litchi picking point. Different from the grapes and tomatoes picking, litchi have lush leaves, thick and hard stems, and the biological characteristics (picking points) are random. This paper proposes a fault-tolerant mechanism for distributed target picking. This mechanism combines the morphological distribution characteristics of single litchi and the occluded targets completion method, and transforms the image processing problem into supervised learning and nonlinear regression question. We researched the characteristics of litchi stems and growth laws, and divided the picking situation into two categories. For the first time, we design the target fault-tolerant shearing path by utilizing the projection distribution of the normal vector of a single litchi onto the image coordinate system. This approach addresses the challenge of litchi picking with irregular deviation angles caused by the influence of gravity. To sum up, the distributed target fault-tolerance mechanism proposed in this paper combines the morphological characteristics of litchis and artificial intelligence technology, which fundamentally improves the positioning accuracy of litchi picking points and creates a common and intelligent picking positioning technology method for fruit agricultural robots.

Keywords: distributed, litchi, smart picking, mask, target fault-tolerance
DOI: 10.33440/j.ijpaa.20230601.217

Citation: Lia Y H, Wang J, Liao J P, and Lan Y B. A method for determining the optimal placement of litchi clusters using improved YOLACT and a distributed target fault-tolerance mechanism. Int J Precis Agric Aviat, 2023; 6(1): 23–32.

1 Introduction

Zou et al., conducted in-depth research on litchi images; they carried out the accurate instance segmentation for litchi bunches and used the binocular vision positioning to obtain fruit position information. Eventually, they did a simulation analysis in dynamic and occluded scenarios and achieved good results[1,2]. Xiong Juntao et al. analyzed the HSV color gamut model of litchis in natural environment, and used fuzzy C-means clustering (FCM) to segment fruits and stems; at last, the depth error of picking points after image matching and limit constraints rate is less than 5.64%[3]. The current visual solution of litchi bunching has the following shortcomings: (1) the fitting accuracy of the fruit contour is poor[4,5]. (2) It possesses insufficient computational capability for intricate environments. This paper proposes for the first time divided the litchi picking point location scene into A and B categories from the visual system. The type A picking is the same as most bunch fruit picking, such as grapes, cherry tomatoes, etc. The bunch fruit picking points are roughly distributed on the mid-line of the geometric center of the bunch fruit included the occlusion and non-occlusion scenarios[6,7]. It is worth summarizing that a single litchi fruit weighs about 21.4-31.8 grams, and the number of litchi bunches generally ranges from 3 to 15. Such a weight makes the litchis easy to fall under the gravity[8-10]. For the vision system, the real picking point P' will form an inclination angle between the predicted picking point P and the main stem on the mid-line of the bunch contour shape. This is the main reason for low precision of litchi piking[11,12]. Obviously, it is the characteristics of this kind of arbor that makes it very difficult for the visual system to locate the picking point, resulting in the slow progress of mechanized and intelligent picking of litchis[13-16].

2 Materials and Methods

2.1 Mask data encoding

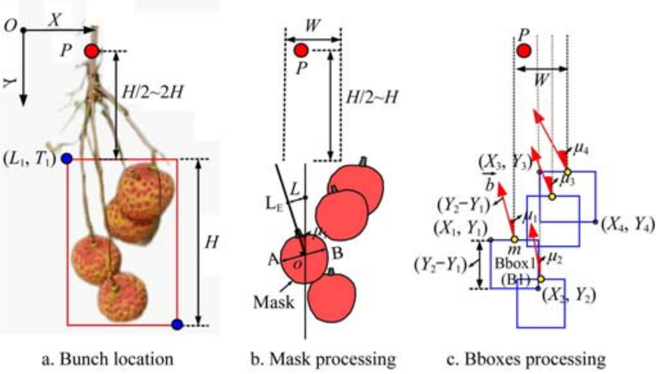
There are complex growth rules among litchi bunch fruit, branch and main stem. In this paper, firstly, we marked litchi bunch to find their position in the coordinate system (XOY) (as shown in Figure 1). According to the upper left corner coordinate value which is (L_1, T_1), and the lower right corner coordinate value which is (R_1, B_1), it can obtained the height H (B_1-T_1) of the litchi bunches. It mark the picking (target) point as P. According to the empirical value, it can get the distance between P and the upper surface of bounding boxes (Bbox) of litchi bunch as $\frac{H}{2} \sim H$. Secondly, it is assumed that after the mask of each fruit had extracted, calculating normal vector of the single litchi contour with the contour approximation method, and then it can get the angle μ which between the normal vector (NV) and Y axis[17,18].

Received date: 2023-11-25 **Accepted date:** 2023-12-26

Biographies: Yuanhong Li, PhD, research interests: agricultural harvesting robot, machine vision, Email: liyuanhong@stu.scau.edu.cn; **Jing Wang**, Master degree, research interests: machine vision and deep learning algorithms, Email:jingzai@stu.scau.edu.cn; **Jiapeng Liao**, Master degree, research interests: image processing and robotic harvesting path analysis, Email: jiapengliao@stu.scau.edu.cn.

***Corresponding author: Yubin Lan**, PhD, Professor, research interests: precision agriculture aerial technology, applied technology of agricultural aviation. College of Electronic Engineering, South China Agricultural University, Guangzhou, 510642, china. Tel: +86-13922707507, Email: ylan@scau.edu.cn.

As showing in Figure 1c, by the object tracking algorithm, it can get all Bbox and its midpoint of each litchi fruit under the dynamic conditions. Taking the first Bbox in Figure 1c as an example, we can mark the corresponding mask normal vector as $\vec{b}$, the value (length) of the normal vector as Y_2-Y_1. Finally, the normal vector is extracted for each Bbox. So far, it have transformed the engineering problem of locating the litchi cluster into mathematical probability distribution problem. With the feature vector of all litchi, it can use the regression algorithm to predict the pixel position $P(P_x, P_y)$. We mark the mask and the coordinate position of the branch intersection point P on the litchi bunch and encode it into a neural network format output.#



a. Bunch location b. Mask processing c. Bboxes processing

Figure 1 Litchi mask acquisition and contour NV extraction (W represents the fault-tolerance distance for picking executed by the end effector at the picking point)

The processing of the fruit mask is as shown in the Figure 1b. First, it is obtain the pixel coordinate points (M_{x1}, M_{y1}), $(M_{x2}, M_{y2})\ldots(M_{xi}, M_{yi})$ on the both sides of mask outline based on the midline L. The distance formula between the two points can be defined by the following formula:

$$|AB|=\sqrt{(M_{xi}-M_{xi-1})^2+(M_{yi}-M_{yi-1})^2} \qquad (1)$$

where, M represents the geometric coordinate point of a single litchi along its Mask outline, and i represents the i-th coordinate value of the point set. Assuming that L_E is a straight line perpendicular to the line AB, according to a point on the slope μ_1 of line, the equation for L_E can be defined by:

$$Y=Y_{i-1}+\tan\mu_i(X-X_i-\frac{X_i-X_{i-1}}{2}) \qquad (2)$$

When processing multiple Bbox normal vectors, it is assumed that the starting point of each litchi outline NV is marked as B_1, B_2……B_n; The position of the starting point of each NV from point P is Y_1, Y_2……Y_n; In this paper, the direction of the normal vector is marked in the order from left to right, such as μ_1, μ_2, μ_3.....μ_n, their size of the normal vector is Y_2-Y_1, Y_4-Y_3, Y_6-Y_5, ……Y_n-Y_{n-1}. Correspondingly, the starting point of each Bbox NV are $\left(X_1+\frac{X_2-X_1}{2},Y_1\right)$, $\left(X_2+\frac{X_3-X_2}{2},Y_2\right)$, $\left(X_3+\frac{X_4-X_3}{2},Y_3\right)$…… $\left(X_{n-1}+\frac{X_n-X_{n-1}}{2},Y_n\right)$. It should be definite that, according to the growth rules of litchi branches, the range of normal vector angle of each Bbox is between 45 and 135 degrees parallel to the positive direction of the X axis, and the normal vectors exceeding this angle range will be regarded as invalid vectors. At last, we added loss function $L_{litchi_{mask}}$ to Mask RCNN[19,20], training with average binary cross-entropy loss. The multi-task loss function is defined by:

$$L=L_{class}+L_{boxes}+L_{litchi_{mask}} \qquad (3)$$

where, L_{class} is the classification loss and L_{boxes} is the bounding regression loss; This multi-task loss function can select the anchor that contains the maximum target and adjust the position and size of the bounding box. Finally, the bounding box and mask will be generated. The backbone uses ResNet-101 C4 to first obtain 7×7×1024 features through ROI Align, and then obtain 7×7×2048 features by Res5. Here 2048 channels are divided into two branches which are classification and regression, and another branch is responsible for generating 14×14×80 litchi Mask. Obviously, this is faster than learning from scratch the mapping between input and output converges fast. For occlusion targets, we use the width and height ratio (WHR) of the Mask output to redefine the bounding boxes, and complete the mask of the occlusion part to obtain an accurate mask normal vector [21,22]. The mask redefinition procedure is presented as Algorithm 1.

```
Algorithm 1. Redefine bounding boxes procedure
Input: All litchi instance masks
Output: redefine bounding boxes and its normal vector (NV)
1. Obtain contour coordinate points$(M_{x1}, M_{y1})$, $(M_{x2}, M_{y2})\ldots(M_{xi}, M_{yi})$ and calculate $|AB_1|$, $|AB_2|$……$|AB_n|$ by distance formula.
2. For $i$ = 1,2……$n$
    (1) Defined the max = $|AB_i|$, and find the maximum value of $|AB_i|$, if max < $|AB_i|$, then the max = $|AB_i|$.
    (2) Assuming that $L_E$ is a line perpendicular to segment AB, according to the slope $\mu_i$, the equation of $L_E$ can be defined by: $Y=Y_{i-1}+\tan\mu_i(X-X_i-\frac{X_i-X_{i-1}}{2})$ .
    (3) For a specific slope $\mu_i$, if $\mu_i=\begin{cases}45^\circ<\mu_i<135^\circ & valid\ NV\\ & invalid\ NV\end{cases}$
3. Extracted Bbox coordinates based on mask, then calculate the aspect ratio WHR = $=\frac{Y_2-Y_1}{X_2-X_1}$, If WHR< threshold, mark it as occlusion litchi, then record this mask and redefine the bounding boxes according to its mean value.
4. Calculated the starting coordinate $\left(X_1+\frac{X_2-X_1}{2},Y_1\right)$…… $\left(X_{n-1}+\frac{X_n-X_{n-1}}{2},Y_n\right)$ of valid NV and output the result.
```

2.2 Target Localization Fault Tolerance Mechanism

The fault-tolerance mechanism of positioning means that within the picking scope of the litchi gripper, when there is an error in the positioning picking point, the end effector can take the corresponding error compensation, sacrificing a certain part of the performance to ensure the system works within an acceptable range. Aiming at the bunch picking scene, this paper proposes a “distributed target localization method”, which is based on the fruit normal vector and target tracking algorithm proposed above. The fault tolerance distance for litchi picking exhibits specific rules and a probability distribution. Statistical errors occur in three dimensions. Initially, there is a visual depth error in the Z direction. Subsequently, the width error in the X direction significantly influences the success of the picking process. The Y direction pertains to the reserved length of fruit bunches along the main stem, with a relatively high fault-tolerant positioning range in this direction. Assuming the estimated value in the X direction of the picking point P follows a normal distribution, it employ the Shapiro-Wilke W test method for verification. the specific steps are as follows[23]: 1) With statistical assumption factor H_0, the X distance values of the picking points are all from normal distribution; 2) According to the estimated normal vector value X_i of each bunch of litchi rearrange X_1, X_2, X_3……X_i from large to small; 3) According to the Shapiro-Wilk coefficient table, find out the Shapiro-Wilk coefficient α_{in} corresponding to the sample size i. 4) Calculate the value of the statistic W:

$$W=\frac{(\sum_i \alpha_{in}(X_{n+i-1}-X_n))^2}{\sum_{i=1}^{n}(X_{(i)}-\bar{X})^2} \tag{3}$$

The numerator $\sum_i$ is $\sum_{i=1}^{\frac{n}{2}}$ when n is even and $\sum_{i=1}^{\frac{n+1}{2}}$ when n is odd. X is the average estimated values of the target normal vector in the X direction. 5) Select the test level β factor (β=0.10, 0.05 or 0.01), and obtain the corresponding $W(n, \beta)$ value according to the number of samples n and the test level factor β difference W distribution table. 6) When $W \leq W(n, \beta)$, the overall sample is not normally distributed. If $W > W(n, \beta)$, the assumed H_0 follow a normal distribution. As shown in Figure 2, between connecting the litchi fruit and the main stem, it mark the normal vector of the litchi according to the midpoint position of the Bbox, and then cluster the picking points from main stem according to the marked normal vector. The picking target point is the midpoint of the main stem, and the fault tolerance radius can be defined as R. If it is less than R_2, it is the maximum limit fault tolerance radius, and if it is less than R_1, it is the ideal fault tolerance radius. The normal vector of a single fruit of litchi follows the "unsigned" gradient principle. First, the direction, size, and distance between the starting point and picking point of the normal vector between 45 and 135 degrees are formed into three digital matrices. It is assumed that the position of the starting point of each normal vector from the point P is Y_1, Y_2……. Y_n then the vector size and Y_n are normalized relative to the ground and then accumulated into 9 array intervals to form a gradient histogram. These nine array intervals 45, 55, 65, 75, 85, 95, 105, 115, 125 and 135, respectively. Finally, we judge whether the picking target belongs to the right, left or normality distribution according to the number distribution map.

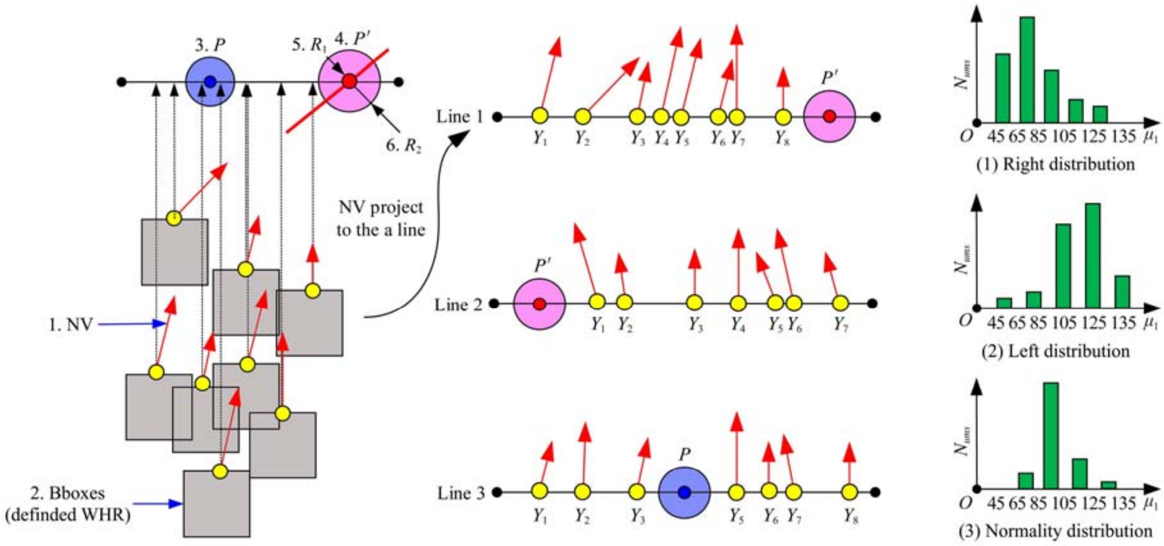

3. P
5. R_1
4. P′
6. R_2
1. NV
2. Bboxes
(definded WHR)
NV project
to the a line
Line 1
Y_1 Y_2 Y_3 Y_4 Y_5 Y_6 Y_7 Y_8
P′
Line 2
P′
Y_1 Y_2 Y_3 Y_4 Y_5 Y_6 Y_7
Line 3
P
Y_1 Y_2 Y_3 Y_5 Y_6 Y_7 Y_8
N_{ums}
O 45 65 85 105 125 135 μ_1
(1) Right distribution
N_{ums}
O 45 65 85 105 125 135 μ_1
(2) Left distribution
N_{ums}
O 45 65 85 105 125 135 μ_1
(3) Normality distribution

Figure 2 Distributed target location method (The blue area is the A-type picking targets, and red area is the B-type target picking)

Finally, it construct the Shapiro-Wilk distribution learning model by projecting each normal vector on a line parallel to the X axis with taking the projected size of the normal vector $ProL_i(Y_i)$ and the known picking points. We predict the coordinate position of Px through supervised learning, and then judge whether the picking stem diameter 1 falls within the target circle with R_2 as the radius, so there are two expected output values of supervised learning: 1) Assuming that the target circle with radius R_1 is the preferred picking (PP), 2) A ring of radius greater than R_1 and less than R_2 is Alternative picking (AP). The algorithm workflow of distributed target fault-tolerant mechanism is presented as Algorithm 2.

Algorithm 2. Target fault-tolerance procedure

Input: All litchi normal vector

Output: The predicted coordinate value of the Px

1. From the algorithm 1, it can get the starting coordinates of the valid $NV\left(X_1+\frac{X_2-X_1}{2}, Y_1\right)$……$\left(X_{n-1}+\frac{X_n-X_{n-1}}{2}, Y_n\right)$, and its sizes or angles μ_1……μ_n, and then moved all NV starting points of their bounding boxes to the line segment paralleled to the X-axis. For multiple bounding boxes, the Line segment is labeled as Line 1……Line i.

2. Through the angle between NV and Line i (μ_1+90)……(μ_n+90), For i in n, the number of vectors μ_n+90 in the nine array intervals 45, 55, 65, 75, 85, 95, 105, 115, 125, 135 was calculated to determine which kind of distribution the picking points belonged to.

3. For i in n, it calculate the size of $||NV_i||\cos(\mu_n+90)$, assuming that the projected size of the vector is denoted as $ProL_i(Y_i)$.

4. According to the Shapiro-Wilk coefficient table, find the Shapiro-Wilk coefficient α_{in} of the corresponding vector projection size $ProL_i(Y_i)_i$. Calculate the value of the statistic $ProW_i$, $ProW_i=\frac{(\sum_i \alpha_{in}(X_{n+i-1}-X_n))^2}{\sum_{i=1}^{n}(X_{(i)}-X)^2}$, the numerator here, $\sum_i$, is $\sum_{i=1}^{\frac{n}{2}}$ when n is even and $\sum_{i=1}^{\frac{n+1}{2}}$ when n is odd.

5. Output the subtraction value of the $ProW_i - P'x$, import to supervised learning model. Finally, it output Px position again.

2.3 Handling Occluded Class B Targets

For the occluded target, this paper firstly processes the depth information of the constructed single litchi, and then determines the spatial position by combining the target location fault-tolerance mechanism. Assuming that the depth distance of litchi in the RGB-D depth camera is L, then the distance between the depth camera and the geometric center of the single fruit is $(L+\frac{3.5}{2})$ cm.

Because the diameter of litchi is generally 3~4 cm, we take 3.5 cm here. As show in Figure 3, this paper draws a three-dimensional schematic diagram of the litchi picking area. Firstly, the top layer represents the longest distance that the robot end effector can reach along the litchi stem diameter. In single fruit detection, this distance is twice the longitudinal length of the fruit. In the brunch fruit detection, the distance is the sum of the preset fault-tolerance distances T_1 and T_2. Secondly, here is the central layer, which is the optimal picking space. The Point P in the figure is the optimal picking point. Assuming that the height of the litchi target detection frame at this point is H, then the height of point P should between $H/2$ and $2H$. Whenever the position of point P is determined, the bottom layer is along the positive Y-axis direction T_2 and parallel to the XOZ plane. The steps for determining the depth distance of point P for occluded litchis are as follows: 1) First, the value of H is determined by the object detection, and then the range of the value of P_y is calculated as $\frac{(T_1+T_2)}{2}$ by using the coordinates of the lower right corner of the object detection.

2) Through the mathematical statistics method, it is use the fault-tolerance mechanism in Section 2.2 to estimate the P_X coordinate value. 3) Obtain the depth information of each fruit through the depth camera, assuming that the depth information of n litchis is L_1, L_2……L_n,

Through the conversion between the camera and the world coordinate system, we can get the depth information D_1, D_2……D_n of each litchi in the XOY coordinate system. After the calculation of the normal vector starting point, all position of the point P is Y_1, Y_2…… Y, then we can get the coordinates of P_Z and the values of R_1 and R_2.

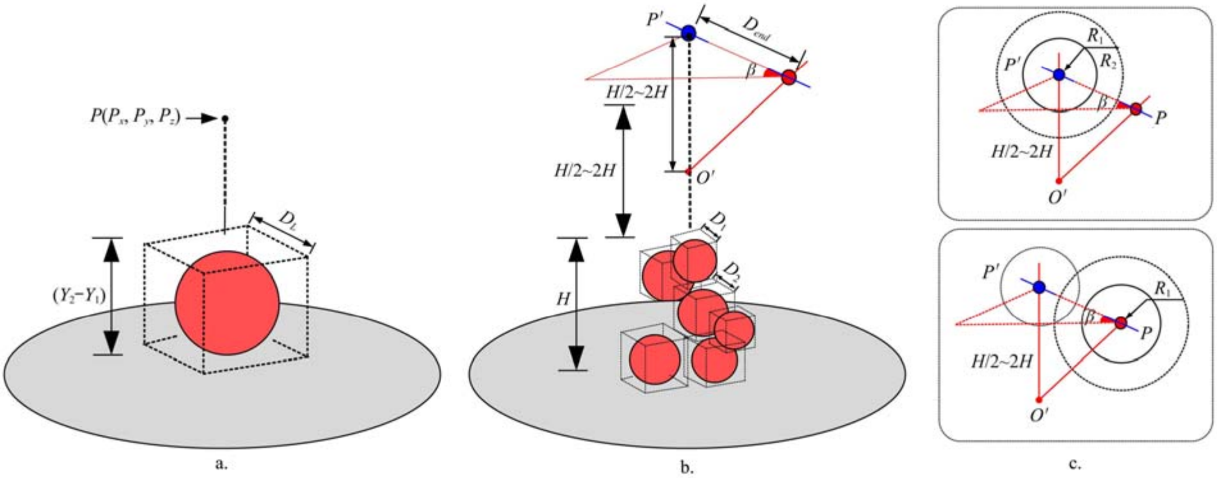


Figure 3 3D model of single and bunch fruit picking

As mentioned in the section 1, the mathematical models are explained according to the end-effector shear position. As for the fruit picking in Figure 3a, we directly used the visual system to locate the target, and conducted the test on whether there existed occlusion. For the scenario shown in Figure 3b, this paper made a compatible path design based on the path planning of end-effector. Since litchi fruit distribution is random, it is difficult for the visual system to judge type A and B based on the fruit distribution. The solution of this paper is to design the cutting route of the end-effector. As shown in Figure 4 below, It is assumed that the angle between the horizontal line (X-axis) and the line which between the predicted picking point P and real picking point P' is β, and the shear angle of the end-effector is also set to β. After determining the picking point of type A, the vision system will feedback whether it can cut the cross-fruit rod diameter at position P through force sensing. If it fails to cut at point P, the end-effector will take β as the cutting angle and D_{end} as the linear moving distance for secondary cutting. In this way, the picking success rate of litchi fruits can be improved among target range. Similarly, fault-tolerant design was carried out at P and P' for the picking range of the target, as shown in Figure 4. The target P was cut within the effective fault-tolerant radius of R_2. For litchi cross fruit picking of type B, the target P will be repositioned to P' for picking by calculating D_{end}.

2.4 Dynamic Target Tracking and Matching Strategy

There are four main requirements for target tracking in litchi picking: 1) In the case of dynamic situation, litchi are not detected due to occlusion, and the tracker can predict the target object[24, 25]. 2) Target tracking can assign a unique ID to each string of litchi, and bind each three-dimensional coordinate to the corresponding box[24]. 3) Trackers are usually faster than detectors and can increase real-time performance[26]. 4) The visual tracking and precise positioning of picking points in a dynamic environment can fundamentally improve the work efficiency of fruit picking robots. Multiple object tracking (MOT), is a versatile experimental paradigm developed by Zenon Pylyshyn for studying sustained visual attention in a dynamic environment in 1988[27]. In case of low frame rate and large camera motion, we must improve the MOT methods to learn object motions and achieve more robust results. Bytetrack uses the similarity between the detection frame and the tracking trajectory to remove the background from the low-scoring detection results while retaining the high-scoring detection results, mining occlusion, blur and other samples, thereby reducing missed detection and improving the coherence of target tracking trajectories[28].

In terms of detectors, here we use the YOLOX as the detector of the litchi fruit[29,30]. YOLOX boasts a more robust advanced label assigning strategy (SimOTA) and a decoupled detection head compared to the YOLO series. The YOLOX Head utilizes three feature maps generated by the Feature Pyramid Network (FPN) to ascertain the presence of objects at the corresponding feature points. In contrast to the previous YOLO Head, which combined classification and regression in a single convolution, the YOLOX Head independently handles classification and regression before integrating them.In terms of tracker, this paper uses the Kalman filter algorithm to define the KalmanFilter class[31,32]. The code implementation process is divided into 6 parts: (1) Class initialization __init__, (2) Initialize the function of state (mean) and state covariance (covariance), (3) Prediction stage function predict, (4) Distribution conversion function project, (5) Update phase function update, (6) Calculate the distance function gatingdistance between the state distribution and the measurement (detection frame). In terms of matching strategy, this paper uses the IoU score to judge. With the movement of the lens or the relative movement of the object and the camera, the aspect ratio of the object will also change; the implementation steps of this part of the code are: 1) There are two categories: high-scoring box + low-scoring box. 2) For the first time, use the high score box to match the previous trajectory. 3) For the second time, use the low-scoring box to match the tracked trajectory of the high-scoring box that did not match the first time. 4) For the high-scoring boxes that do not match the upper track, create a new tracking track. For the tracked trajectories that do not match the detection frame, 30 frames are reserved so that they can be matched again later. Suppose a video sequence **V** is input through the camera, the detection result returned by the target detector is **Det**, and the **Det** is a data container that contains bounding boxes, scores and class ID information output by the detector. For each frame from video sequence **V**, we set a detection score threshold α. Based on this threshold, we separate all the detection boxes into high part M_{high} and low part M_{low}. The method used in this paper is to first combine the tracks **T** and high-scoring M_{high} of the tracker to calculate the IoU and Re-ID feature distances of the target to be tracked. The Hungarian Algorithm is then used for similarity matching[33]. We mark the remaining unmatched detection boxes and tracks as M_{remain} and T_{remain}, respectively. Second, we combine M_{low} and tracks T_{remain} that have not been matched for similarity evaluation. The purpose of this is to track some litchi fruits that are occluded, blurred, or whose shape features are not very obvious. IoU (Intersection over Union) is a standard performance measure for image class segmentation problems. For a given set of images, this project improves MST-IoU by giving the ratio of the intersection and union of predicted and ground-truth candidate boxes. Suppose t is the probability output of the pixel set N after the activation function in the fruit or cluster feature layer, and Y is the real candidate box dataset. Here $Y \in \{0,1\}^M$ means 0 is a non-target pixel and 1 is a detection target pixel. So:

$$\text{MST-IoU} = \frac{I(t)}{U(t)} = \frac{\sum_{n \in N} t_n * Y_n}{\sum_{n \in N}(t_n + Y_n - t_n * Y_n)} \tag{4}$$

In the multi-target detection training process, NMS calculates and sorts the candidate box list and confidence C of the candidate region, and selects the test box with the largest score. Then calculate the IoU coefficients of other scoring boxes and the current box, and delete if the IoU is greater than the set threshold. This is an iterative process. In this process, NMS is used to select a certain maximum score box. Then the second iteration will select the highest score in the remaining boxes and delete those that exceed the set IoU threshold until all possible targets in the picking target are obtained. After the residual network and the 1×1 convolutional layer, a large number of candidate boxes will be generated on the candidate region output by the output feature map.

In terms of matching strategy, the visual strategy from “coarse” to “fine” enables the robot to quickly complete the target search task. Aiming at the long-range image of fruit clusters with a large number of litchi and inconspicuous branches, K-means density clustering is proposed to study the field of view division. This enables the litchi fruit field data to be divided into K predefined categories, and each data point is clustered into a picking object. The specific principle function is as follows: Assuming that there are m single litchi samples, first of all, the multi-sample function on the K value is introduced here,

$$F = \sum_{i=1}^{m} \sum_{k=1}^{k} \sigma_{ik} \| t(x,y) - \mu_k \|^2 \tag{5}$$

If the single litchi center point $t(x,y)$ belongs to cluster K, then σ_{ik}=1, or σ_{ik} =0; This time μ_k is the centroid $t(x,y)$. If the derivative of the F function minimizes the solution of the equation, the problem is transformed into:

$$\frac{\partial F}{\partial \sigma_{ik}} = \sum_{i=1}^{m} \sum_{k=1}^{k} \| t(x,y) - \mu_k \|^2 \tag{6}$$

We need to differentiate F, and recalculate the centroids after the previous clustering iteration, obviously data points $t(x,y)$ will be assigned to clusters that are close. The centroids of each cluster are naturally recalculated according to equation (3) below to reflect the new data point assignments.

$$\frac{\partial F}{\partial \mu_k} = 2\sum_{i=1}^{m} \sigma_{ik}(t(x,y) - \mu_k) = 0 \tag{7}$$

$$\mu_k = \frac{\sum_{i=1}^{m} \sigma_{ik} t(x,y)}{\sum_{i=1}^{m} \sigma_{ik}} \tag{8}$$

After the screening of K-means, in the previous experimental test, the multi-objective fruit-picking field-of-view partitioning algorithm in this paper is direct and efficient. With the self-developed vision system, the computing speed performance is greatly improved, and the data dimension is effectively reduced.

3 Results and Discussion

3.1 Experiment setting

The litchi picking in this paper can be divided into A and B scenarios (as show in Figure 2). The litchi pictures were collected in Guangdong Province, China. If the occlusion area of leaves or branches is greater than 30%, the sample will be considered as an occlusion target. The dataset presented in this paper comprises 2000 litchis captured in their natural environment, each with dimensions of 1061 by 640. Additionally, there are 200 validation sets equipped with RGB-D depth information. The experimental setup utilizes the TensorFlow framework, with hardware specifications including an Intel Core i7 CPU, 16GB of memory, and a GeForce GTX 3060 Ti GPU featuring 8GB of video memory. The GPU runs on CUDA version 11.0 and CUDNN 7.4. The operating system employed is Linux, specifically Ubuntu 18.04 LTS. In addition, in this paper, the ProtoBuf serialized structure protocol is used to store the weight information which called Open Neural Network Exchange (ONNX) Neural Network[34]. This is a platform independent and programming language independent efficient protocol. In order to obtain the optimal training results, transfer learning is used to train the pre-trained model on COCO dataset. After testing, the pre-training based on COCO data has fast convergence speed and high initial performance of the model. In the training process, the transfer learning model can get the improvement fast.

3.2 Object Detection Evaluation

This paper employ YOLOX, YOLOv5, CenterNet, and EfficientDet as multi-target trackers to assess target detection using both individual litchi instances and litchi clusters[35,36]. As can be seen from Table 1, YOLOX-DarkNet53 achieved the best mAP, followed by YOLOV5 with 60.68% mAP. The difference between YOLOX and YOLOV5 is in the head part of the detection network. YOLOX is anchor free, and its head directly predicts four target Bbox parameters at each position. The Figure 5 below shows that YOLOV5 uses the K-means algorithm to screen anchor boxes.

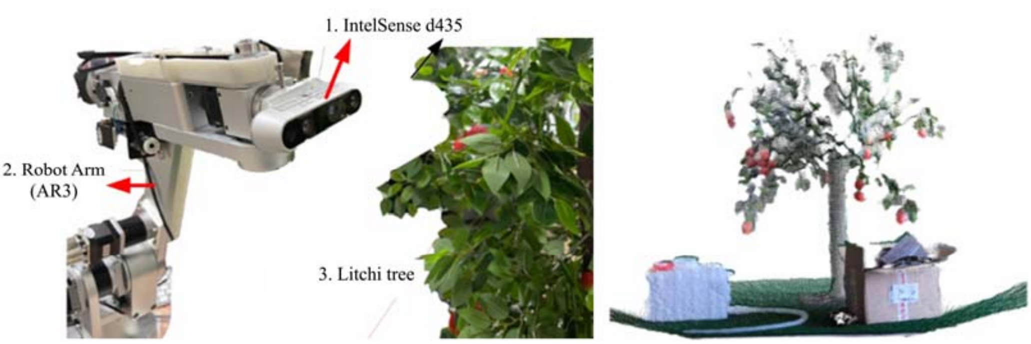


Figure 4 In-depth information collection platform

Table 1 Detector Performance Comparison

Model	AP/%	FPS	Parameters	AP50	AP65	AP80
CenterNet	48.34	12	74.99M	62.8	59.3	55.2
YOLOX-DarkNet53	**65.30**	30	87.80M	86.5	**82.7**	**78.6**
EfficientDet	58.50	15	52.40M	75.4	70.8	66.5
YOLOV5-DarkNet53	60.68	22	90.43M	73.2	72.4	60.2
YOLOV3- DarkNet53	58.92	28	63.72M	**87.9**	76.1	70.9

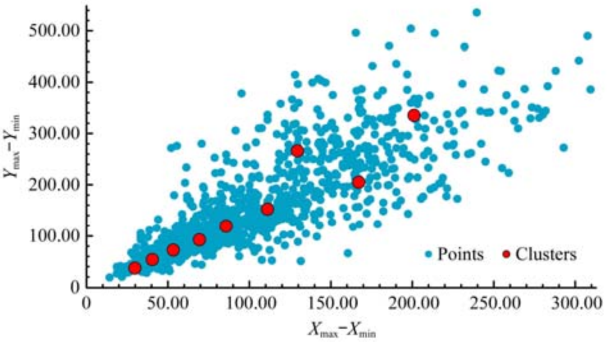


Figure 5 K-means cluster of YOLO V5 anchor boxes

The detection head of YOLOV5 can simultaneously predict the category score, bounding box regression parameters and objectless; YOLOX replaces the coupled head with the decoupled detection head, greatly improving the convergence speed of the network. In terms of FPS comparison effect, YOLOX still reached the highest 30fps. Although the mAP of YOLOV3 is the highest when the IoU is 50, when the IoU value is 65 or 80, YOLOX has a significant effect on improving the accuracy. In summary, the following will focus on comparing the detection effects of different versions of YOLOX, so that we can select the fastest and best detector for target tracking of a cluster of litchi.

YOLOX and YOLOV5 have different versions. For single litchi detection, this paper only compares their S versions. We use 640×640 as the image input. As shown in Figure 6 below, the model training is divided into two steps. The first step is to freeze the training, that is, only train the backbone part. And the learning rate is set to 0.01-0.001, the number of iterations is 50, and the number of samples for each iteration is 4. The second step of training is the entire detection network, the initial learning rate is set to 0.001, the number of iterations is 150-200, and other setting parameters are shown in Table 2.

In the 200 training epochs of YOLO5 and YOLOV5, the convergence speed of the loss function is relatively fast in the first 50 iterations. This shows that freezing the network is a relatively optimal solution. The training loss function of YOLOV5 starts to slow down at the 80th iteration, and starts to converge after about 160 iterations. The validation loss function curve of YOLOV5 began to oscillate relatively large at the 60th iteration, and did not begin to converge until the 180th iteration. Both the training and validation loss function curves of YOLOX gradually stabilized after the 150th iteration, and did not change significantly after the 200th iteration. Therefore, this article finally selects the weight file saved in the 200th (non-frozen) epoch of YOLOX.

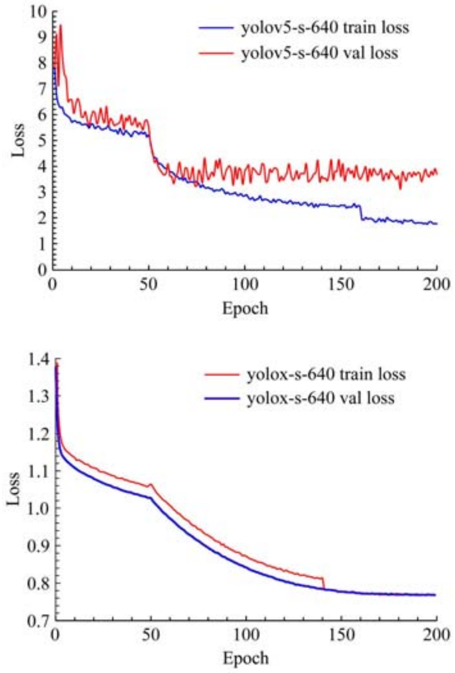


Figure 6 YOLOX-S-640 VS YOLOV5-S-640

Table 2 Training parameter setting

Parameters	YOLOX-m	YOLOX-MaskLabel	YOLOV5-m	YOLOV5-MaskLabel
Mosaic	True	True	False	False
Mixup	True	True	False	False
Freeze Epoch	50	0	100	50
Unfreeze Epoch	150	200	150	200
Initial lr	0.001	0.001	0.01	0.01
Optimizer	SDG	Adam	SDG	Adam

There are multiple versions of YOLOX. Considering the demand to deploy YOLOX on mobile robotic arms or embedded systems, this article tests and compares the nano, tiny, s, and m versions of YOLOX. As can be seen from Figure 7 below, when training 200 epochs and the score threshold is both 0.5, the YOLOX-nano-416 version with 416×416 as input has the highest accuracy of 84.96%. It is worth noting that in the model with the same 640×640 input, the accuracy of YOLOX-m is 74.29%, while the accuracy of YOLOX-s is only 69.23%.

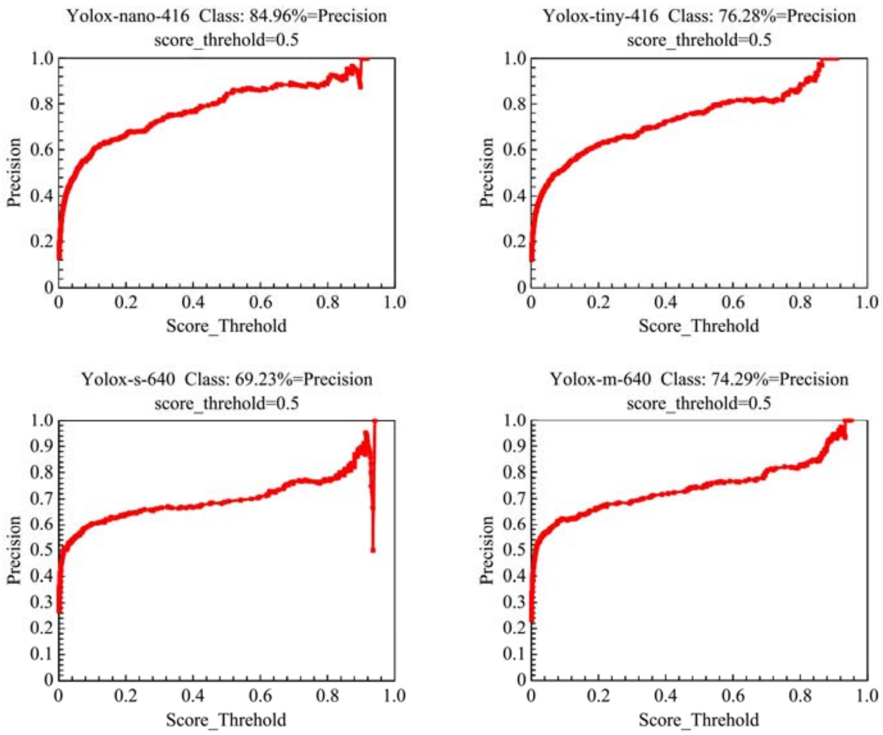


Figure 7　YOLOX precision with different version

From the F1 curve of YOLOX nano (Figure 8), the confidence value of optimization precision and recall rate is 0.5. In most cases, a higher confidence level is desirable. But for this model, the best choice may be when F1 is 0.70, and the confidence level is 0.43. Because when F1 is 0.72, the confidence of the model will reach the maximum, and 0.70 and 0.72 are not far away. Observing precision and recall values with a confidence value of 0.5 also confirms that this may be a suitable design point. From around 0.5, the recall rate starts to decrease, and the accuracy rate is still roughly at the maximum value. In addition, it can be seen from the F1 curve of YOLOX-m that the confidence of the optimal precision and recall is 0.73, which starts to decrease from 0.83. Overall, the F1 curve of YOLOX-m is better than that of YOLOX-nano, but this cannot accurately evaluate all models.

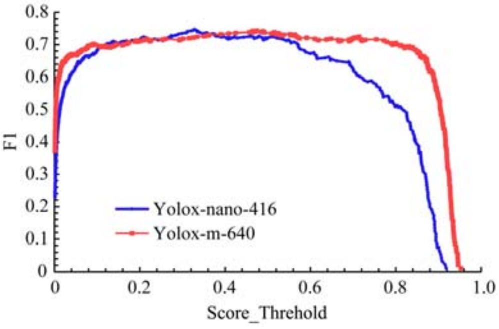


Figure 8　YOLOX-NANO VS YOLOX-M

For the single fruit detection model of litchi that are ripe and ready to be picked, the mAP of YOLOX-nano reached a maximum of 78%, followed by the mAP of YOLOX-m, which also achieved an ideal 76%. YOLOX-tiny and YOLO-S are 0.73 and 0.69 respectively. The analysis of experimental data shows that the target detection using YOLOX can meet the requirements of target tracking detector to a certain extent. In addition, different versions of YOLOX algorithm can run on embedded and mobile systems. For single fruit detection, YOLOX-m is used in this paper. As can be seen from Figure 13 below, target detection of single fruit can achieve good results at different distances, and can also achieve high precision positioning. In addition, the algorithm can also recognize single fruits with occlusion. The following will be based on target detection, plus target tracking to analyze the positioning effect of litchi dynamic picking. Figure 14 shows the effect of fruit bunching detection. The fruit bunching detection mainly determines the approximate size of the fruit bunching outline through "coarse" positioning, and then determines the values of H, L_1, T_1, R_1 and B_1, according to the mask data encoding in Section 2.1 of this article.

3.3　Target Tracking Evaluation

For the dynamic tracking of litchi bunches, this paper uses the multi-target tracking algorithm for comparison. After the litchi is occluded, the target detection often loses the tracking ID. The tracking ID can affect the relative position of the litchi fruit and its direction of the normal vector. After improving the accuracy of target detection, In addition, the impact of the target fault-tolerance mechanism on positioning in a dynamic environment must be taken into account. We utilize multiple object tracking accuracy (MOTA) and multiple object tracking precision (MOTP) as metrics to assess the tracking performance. The MOTA can be defined by formula:

$$\text{MOTA} = 1 - \frac{\sum_t (m_t + fP_t + mme_t)}{\sum_t g_t} \tag{9}$$

where, m_t is FP, the number of missing (missed detection), that is, the target has no hypothetical position match in frame P_t is the number of false positives, that is, there is no tracking target match at the hypothetical position given in the t frame. The target tracking experimental samples in this paper include ripe and immature lychee fruit bunches. Initially, we capture an indoor mp4 video, which is subsequently imported into the program for tracking. The tracking parameters are configured with a track threshold of 0.5 and a track buffer of 30. Additionally, the match threshold is set at 0.5, and the minimum bounding box area is specified as 10. This paper compares multi-target tracking algorithms such as sort, deepsort, and bytetrack. As showing in Table 3, for different detection score thresholds, the IDF1 and MOTA of bytetrack are relatively high. When the score is equal to 0.5, the IDF1 of bytetrack is 89, and the IDF1 of deepsort is 85. In the comparison of MOTA, we found that bytetrack can reach up to 88, but deepsort can only reach up to 79. To sum up, this article adopts YOLOX for the detector's bytetrack, locate and track the single or bunch fruit target.

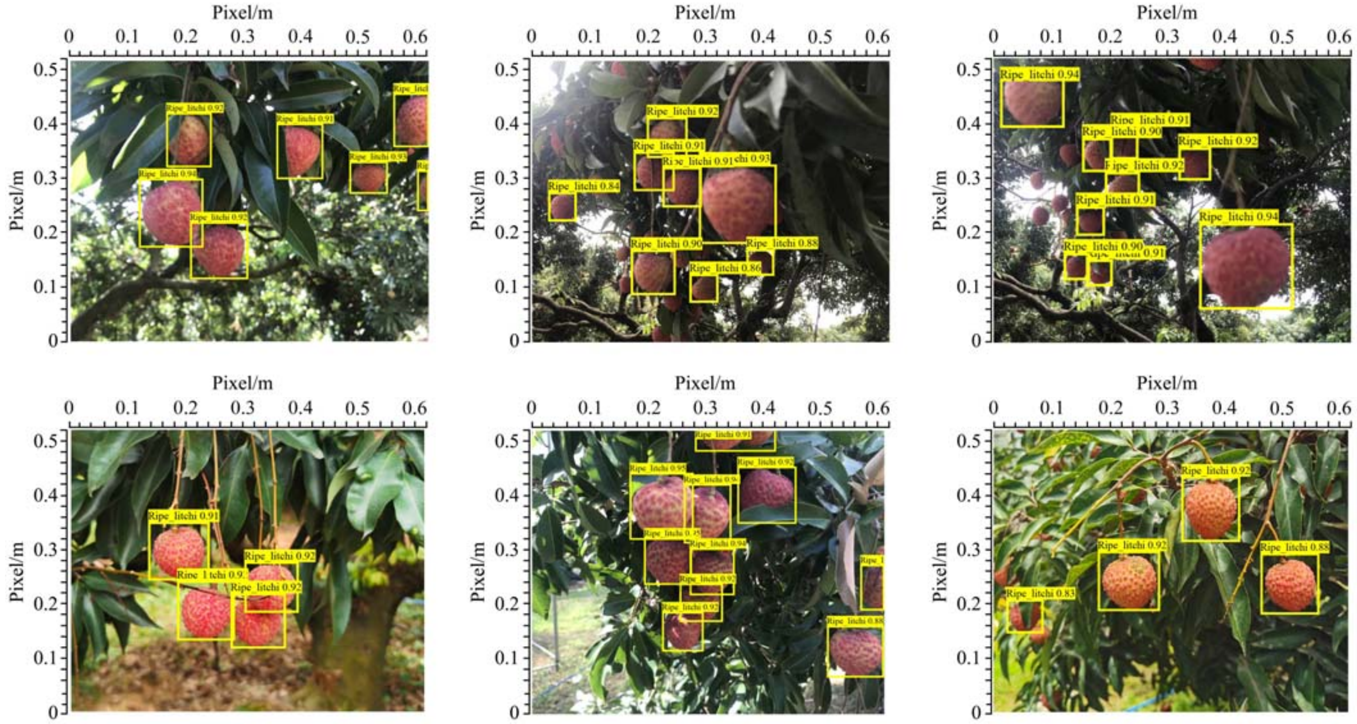


Figure 9 The detection effectiveness of litchi clusters based on instance segmentation algorithms

Table 3 IDF1 and MOTA comparison

SCORE	IDF1-BYTE	IDF1-SORT	IDF1-DEEPSORT	MOTA-BYTE	MOTA-SORT	MOTA-DEEPSORT
0.2	75	52	72	62	66	72
0.3	81	62	78	79	68	78
0.4	80	70	80	82	70	74
0.5	89	66	85	88	77	79
0.6	88	60	80	87	75	51
0.7	84	55	76	82	55	52
0.8	64	51	72	65	44	50

In addition, this paper uses bytetrack to conduct object tracking tests on mature and immature litchis. The evaluation metrics are MOTA, MOTP, MT (Mostly Tracked) and ML (Mostly Lost). MT refers to the proportion of tracks that meet Ground Truth at least 80\% of the time in all tracking targets. MT mainly measures the performance of the detector. ML refers to the proportion of tracks that satisfy Ground Truth that only matches successfully in less than 20\% of the time among all tracking targets. The data in Table 4 illustrates that mature litchis are more readily tracked, evidenced by a MOTA value of 51.30 and MOTP value of 82.82. In comparison, immature litchis exhibit less distinct spectral information; however, they still align with the mature MT and ML indicators, with litchi values showing relatively minor differences. This indicates that the bytetrack algorithm is proficient in effectively tracking litchi bunches of varying maturities.

Table 4 Bytetrack for tracking lychee at different ripeness levels

	MOTA	MOTP	MT	ML
Ripe_litchi	51.3	82.82	26.11%	38.70%
Unripe_litchi	42.8	75.23	21.66%	35.70%

In the detection of 3 consecutive frames, it can be seen that bytetrack can track single litchi when the leaves are occluded. Such tracking information can not lose the its normal vector in the dynamic environment when the target is located, thereby improving the target location accuracy in Section 2.2. The detection boxes of different colors in the figure represent different tracking IDs. In the case of no occlusion, our locked IDs can perform continuous tracking between 3$\sim$5 frames.

4 Conclusions

This paper breaks through the traditional fruit cluster location method, and proposes a litchi picking point location method based on visual system and engineering technology. This method combines the morphological characteristics of litchi with the data association method of occluding targets for visual collaboration, transforms the image processing problem into a non-linear regression problem, and overcomes the common fault-tolerance technology of fruit cluster picking with irregular offset. In general, the conclusions of this paper mainly include the following parts: (1) It is proposed that the location scene of litchi fruit cluster picking points can be divided into two categories: one scene is the same as grapes and tomato, and the picking points are distributed near the centerline of the geometric shape of the fruit cluster; In view of the irregular offset angle caused by the drooping of litchi clusters under the action of gravity, this paper defines it as other scene, which is solved by the distributed single fruit positioning and the design of picking targets for the first time. (2) Breaking through the traditional research on litchi target detection, combined with the characteristics of crop production technology and growth law of fruit clusters, a target location fault tolerance mechanism based on the geometric shape distribution of litchi single fruit (mask method vector) is proposed. This mechanism can not only redefine the occlusion target according to the width to height ratio (WHR), but also accurately and efficiently locate the location of litchi fruit cluster picking target. (3) This paper investigates the target picking mechanism under both static and dynamic conditions. The experimental findings demonstrate that the litchi fruit cluster, utilizing the target location mechanism, yields favorable outcomes in both static and dynamic environments. Notably, when dealing with occluded cross-fruit targets, dynamic target tracking is employed to compute the mathematical model of the picking point. This approach effectively addresses the challenging task of locating the picking point under occluded conditions, offering a versatile technology with high efficiency and accuracy for cross-fruit picking.

The paper faces challenges in litchi picking accuracy relying on a precise camera calibration system, hindering accurate quantification of target point positioning. Efforts focus on deploying a robotic arm for experiments to enhance positioning accuracy and address issues like low frame rates and detection timeliness in litchi detection through the camera.

Acknowledgments

This work was financially supported by the Laboratory of Lingnan Modern Agriculture Project (Grant No. NT2021009); Guangdong Basic and Applied Basic Research Foundation (Grant No. 2021A1515110554);Guangdong University Key Field (Artificial Intelligence) Special Project (No. 2019KZDZX1012) and the 111 Project (D18019); China Postdoctoral Science Foundation (Grant No. 2022M721201); the National Natural Science Foundation of China (Grant No. 31901411); The Open Competition Program of the Top Ten Critical Priorities of Agricultural Science and Technology Innovation for the 14th Five-Year Plan of Guangdong Province (No. 2022SDZG03).

Declaration of Interest Statement

None is declared.

[References]

[1] Xie, J., Jing, T., Chen, B., Peng, J., Zhang, X., He, P., Yin, H., Sun, D., Wang, W., Xiao, A. and Lyu, S. Method for Segmentation of Litchi Branches Based on the Improved DeepLabv3+. Agronomy, 2022, 12(11): 2812. https://doi.org/10.3390/agronomy12112812

[2] Peng, H., Huang, B., Shao, Y., Li, Z., Zhang, C., Chen, Y. and Xiong, J. General improved SSD model for picking object recognition of multiple fruits in natural environment. Transactions of the Chinese Society of Agricultural Engineering, 2018, 34(16): 155–162.

[3] Q. Zhu, R. Lu, J. Lu, F. Li. Research status and development trend of litchi picking machinery. Forestry Machinery and Woodworking Equipment, 2021, 49(08): 11–19. (in Chinese)

[4] J. Li, Y. Tang, X. Zou, G. Lin, H. Wang. Detection of fruit-bearing branches and localization of litchi clusters for vision-based harvesting robots. IEEE Access, 8 (2020) 117746–117758.30. DOI: 10.1109/ACCESS.2020.3005386

[5] J. Xiong, R. Lin, Z. Liu, Z. He, L. Tang, Z. Yang, X. Zou. The recognition of litchi clusters and the calculation of picking point in a nocturnal natural environment. Biosystems Engineering, 2018, 166: 44–57.440. DOI：10.1016/j.biosystemseng.2017.11.005

[6] J. Xiong, X. Zou, L. Chen, H. Peng, D. Wu, et al. Fruit recognition and positioning technology of litchi picking manipulator. Journal of Jiangsu University-Natural Science Edition, 2012, 33(1): 1–5. DOI: 10.3969/j.issn.1671-7775.2012.01.001

[7] Pérez-Zavala, R., Torres-Torriti, M., Cheein, F.A., Troni, G. A pattern recognition strategy for visual grape bunch detection in vineyards. Comput. Electron. Agric., 2018, 151, 136–149.

[8] Wu, G., Zhu, Q., Huang, M., Guo, Y. and Qin, J. Automatic recognition of juicy peaches on trees based on 3D contour features and colour data. Biosystems Engineering, 2019, 188, pp.1–13.

[9] Li, Y., Feng, Q., Liu, C., Xiong, Z., Sun, Y., Xie, F., Li, T. and Zhao, C. MTA-YOLACT: Multitask-aware network on fruit bunch identification for cherry tomato robotic harvesting. European Journal of Agronomy, 2023, 146, p.126812.

[10] Tian, D., Han, Y., Wang, B., Guan, T., Gu, H. and Wei, W. Review of object instance segmentation based on deep learning. Journal of Electronic Imaging, 2022, 31(4): 041205-041205.

[11] He, K., Gkioxari, G., Dollár, P. and Girshick, R. Mask r-cnn. In Proceedings of the IEEE international conference on computer vision, 2017, pp. 2961-2969.

[12] Gong, T., Chen, K., Wang, X., Chu, Q., Zhu, F., Lin, D., Yu, N. and Feng, H. May. Temporal ROI align for video object recognition. In Proceedings of the AAAI Conference on Artificial Intelligence, 2021, 35(2): 1442–1450.

[13] Li, Y., Qi, H., Dai, J., Ji, X. and Wei, Y. Fully convolutional instance-aware semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 2359–2367.

[14] Bolya, D., Zhou, C., Xiao, F. and Lee, Y. J. Yolact: Real-time instance segmentation. In Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 9157–9166.

[15] Leung, T. and Malik, J. Representing and recognizing the visual appearance of materials using three-dimensional textons. International journal of computer vision, 2001, 43, pp.29–44.

[16] Sivic and Zisserman. October. Video Google: A text retrieval approach to object matching in videos. In Proceedings ninth IEEE international conference on computer vision, 2003, pp. 1470-1477. IEEE.

[17] Kim, E., Kim, S., Seo, M. and Yoon, S. XProtoNet: diagnosis in chest radiography with global and local explanations. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021, pp. 15719-15728.

[18] Chen, C., Li, O., Tao, D., Barnett, A., Rudin, C. and Su, J. K. 2019. This looks like that: deep learning for interpretable image recognition. Advances in neural information processing systems, 32.

[19] Zhang, Q. and Gao, G. Grasping point detection of randomly placed fruit cluster using adaptive morphology segmentation and principal component classification of multiple features. Ieee Access, 2019, 7, pp.158035-158050.

[20] R. Bogue. Fruit picking robots: has their time come?, Industrial Robot:the international journal of robotics research and application.

[21] Z. Li, F. Miao, Z. Yang, P. Chai, S. Yang. Factors affecting human hand grasp type in tomato fruit-picking: A statistical investigation for ergonomic development of harvesting robot. Computers and Electronics in Agriculture, 2019, 157, 90–97. DOI: 10.1016/j.compag.2018.12.047

[22] H. Si, J. Lv, K. Lin, J. Wu, J. Chen. A review of application of computer vision in fruit picking robot, in: International Conference on Intelligent Computing. Communication & Devices, Springer, 2019, pp. 346–355. DOI: 10.1007/978-981-15-5887-0_50

[23] J. Zhang. Target extraction of fruit picking robot vision system, in: Journal of Physics: Conference Series, Vol. 1423, IOP Publishing, 2019, p. 012061.

[24] G. Wang, Y. Lan, H. Qi, P. Chen, A. Hewitt, Y. Han. Field evaluation of an unmanned aerial vehicle (uav) sprayer: effect of spray volume on deposition and the control of pests and disease in wheat. Pest Management Science, 2019, 75(6): 1546–1555. DOI: 10.1088/1742-6596/1423/1/012061

[25] Y. Zhan, P. Chen, W. Xu, S. Chen, Y. Han, Y. Lan, G. Wang. Influence of the downwash airflow distribution characteristics of a plant protection uav on spray deposit distribution. Biosystems Engineering, 2022, 216, 32–45. DOI: 10.1016/j.biosystemseng.2022.01.016

[26] N. Saranya, K. Srinivasan, S. Pravin Kumar, V. Rukkumani, R. Ramya,450 Fruit classification using traditional machine learning and deep learning approach, in: International Conference On Computational Vision and BioInspired Computing, Springer, 2019, pp. 79–89. DOI:10.1007/978-3-030-37218-7_10

[27] J. Zhuang, C. Hou, Y. Tang, Y. He, Q. Guo, Z. Zhong, S. Luo. Computer vision-based localisation of picking points for automatic litchi harvesting applications towards natural scenarios. Biosystems Engineering, 2019, 187, 1–20. DOI: 10.1016/j.biosystemseng.2019. 08.016

[28] K. Nagraj, G. Diwan, N. L, et al. Effect of fruit load on yield and quality of litchi (litchi chinensis sonn.). Journal of Pharmacognosy and Phytochemistry, 2019, 8(6): 1929–1931.

[29] K. Kumar, K. Madhumala, S. Sahay. Response of different sources of

potas-sium on fruit quality and fruit colour enhancement in litchi. Journal of Pharmacognosy and Phytochemistry, 2019, 8(6): 1990–1993.

[30] L. Wang, Y. Zhao, S. Liu, Y. Li, S. Chen, Y. Lan. Precision detection of dense plums in orchards using the improved yolov4 model. Frontiers in Plant Science, 2022, 13, 839269–839269. DOI: 10.3389/fpls.2022.839269

[31] Zhang, Q. and Gao, G. Grasping point detection of randomly placed fruit cluster using adaptive morphology segmentation and principal component classification of multiple features. IEEE Access, 2019, 7, pp.158035–158050.

[32] Barisoni, L., Lafata, K. J., Hewitt, S. M., Madabhushi, A. and Balis, U. G. Digital pathology and computational image analysis in nephropathology. Nature Reviews Nephrology, 2020, 16(11): 669–685.

[33] Bolya, Daniel, Chong Zhou, Fanyi Xiao, and Yong Jae Lee. Yolact: Real-time instance segmentation. In Proceedings of the IEEE/CVF international conference on computer vision, pp. 9157–9166. 2019.

[34] Bharati, P. and Pramanik, A., 2020. Deep learning techniques—R-CNN to mask R-CNN: a survey. Computational Intelligence in Pattern Recognition: Proceedings of CIPR 2019, pp.657-668.

[35] Gong, Yuqi, Xuehui Yu, Yao Ding, Xiaoke Peng, Jian Zhao, and Zhenjun Han. Effective fusion factor in FPN for tiny object detection. In Proceedings of the IEEE/CVF winter conference on applications of computer vision, pp. 1160-1168. 2021.

[36] Chen, Chaofan, Oscar Li, Daniel Tao, Alina Barnett, Cynthia Rudin, and Jonathan K. Su. This looks like that: deep learning for interpretable image recognition. Advances in Neural Information Processing Systems, 2019, 32.

[37] J. A. Villasenor Alva, E. G. Estrada. A generalization of shapiro–wilk's test for multivariate normality. Communications in Statistics—Theory and Methods, 2009, 38(11): 1870–1883. DOI: 10.1080/03610920802474465

[38] Y. Ge, Y. Xiong, G. L. Tenorio, P. J. From, Fruit localization and environment perception for strawberry harvesting robots. IEEE Access 7 (2019) 147642–147652. DOI: 10.1109/ACCESS.2019.2946369

[39] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask r-cnn, in: Proceedings of the IEEE international conference on computer vision, 2017, pp. 2961–2969.

[40] Y. Yu, K. Zhang, L. Yang, D. Zhang. Fruit detection for strawberry harvest-ing robot in non-structural environment based on mask-rcnn. Computers and Electronics in Agriculture, 2019, 163: 104846. DOI: 10.1016/j.compag.2019.06.001

[41] X. Liu, D. Zhao, W. Jia, W. Ji, C. Ruan, Y. Sun. Cucumber fruits detection in greenhouses based on instance segmentation. IEEE Access, 2019, 7: 139635–139642. DOI: 10.1109/ACCESS.2019.2942144

[42] R. Yang, Y. Hu, Y. Yao, M. Gao, R. Liu. Fruit target detection based on bco-yolov5 model. Mobile Information Systems, 2022. DOI: 10.1155/2022/8457173

[43] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, Q. Tian. Centernet: Keypointtriplets for object detection, in: Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 6569–6578