Standard Issue – Article

Journal of Agricultural Engineering

https://www.agroengineering.org/

Zanthoxylum infructescence detection based on adaptive density clustering

Diwei Wu, Shaohua Zeng, Shuai Wang, Yanan Chen, Yidan Xu

Publisher's Disclaimer

E-publishing ahead of print is increasingly important for the rapid dissemination of science. The *Early Access* service lets users access peer-reviewed articles well before print/regular issue publication, significantly reducing the time it takes for critical findings to reach the research community. These articles are searchable and citable by their DOI (Digital Object Identifier).

Our Journal is, therefore, e-publishing PDF files of an early version of manuscripts that undergone a regular peer review and have been accepted for publication, but have not been through the typesetting, pagination and proofreading processes, which may lead to differences between this version and the final one.
The final version of the manuscript will then appear on a regular issue of the journal.

Please cite this article as doi: 10.4081/jae.2024.1568

Submitted: 08/11/2022
Accepted: 09/05/2023

Note: The publisher is not responsible for the content or functionality of any supporting information supplied by the authors. Any queries should be directed to the corresponding author for the article.

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article or claim that may be made by its manufacturer is not guaranteed or endorsed by the publisher.

Zanthoxylum infructescence detection based on adaptive density clustering

Diwei Wu,[1,2] Shaohua Zeng,[1,2] Shuai Wang,[3] Yanan Chen,[4] Yidan Xu[5]

[1]College of Computer & Information Science, Chongqing Normal University, Chongqing; [2]Chongqing Center of Engineering Technology Research on Digital Agricultural & Service, Chongqing; [3]Chongqing Master Station of Agricultural Technology Promotion, Chongqing; [4]Chongqing Wanzhou District Station of Soil Fertilizer and Agricultural Ecological Protection, Chongqing; [5]Chongqing Beibei District Agriculture and Rural Committee, Chongqing, China

Correspondence: Shaohua Zeng, College of Computer & Information Science, Chongqing Normal University, Chongqing 401331, China.
E-mail: zsh_cqu@126.com

Key words: adaptive density clustering; infructescence detection; yield estimation; Zanthoxylum.

Acknowledgments: this research was supported by Chongqing University Innovation Research Group Funding [Grant No. CXQT20015] and Research Project of Chongqing Normal University [Grant No. YKC21060].

Conflict of interest: the authors declare no potential conflict of interest.

Availability of data and material: the raw data required to reproduce these findings cannot be shared at this time as the data also form part of an ongoing study.

Abstract

Infructescence detection during the early fruiting stage is a necessary preliminary work to estimate the yield of Zanthoxylum. The purpose of this research is to detect and quantify the infructescences on the images of early fruit-bearing branches of Zanthoxylum which are collected in the natural environment. Thus, a machine vision-based algorithm for detecting Zanthoxylum infructescences is proposed, which contains of two phases. The first is to segment fruit-like and non-fruit region from Zanthoxylum branch image by extracting the Histogram of Oriented Gradient (HOG) feature map and Excess Green minus Excess Red (ExGR) index to obtain the fruits of Zanthoxylum infructescences. The second is to identify fruits adaptively and roughly in accordance with the density of their position distribution, then an optimization model is established to merge rough clusters and the optimal clustering result is obtained. Experiments with normal samples demonstrate that the proposed approach receives a Precision of 96.67%, a Recall of 91.07%, and an F1-score of 0.93. Compared with ADPC-kNN, DBSCAN and OPTICS, the results reveal that the proposed algorithm has an improved robustness and achieves higher Recall and F1-score. Meanwhile, its competitiveness is displayed in the experiments with deep learning-based methods. It can adaptively detect the infructescences of Zanthoxylum branch images, and the experiments prove that it is effective.

Introduction

Estimating the yield of Zanthoxylum during the early fruiting stage is indispensable for growers. It can provide growers with the necessary information to support logistics, crop storage and marketing in advance (Zhang *et al.*, 2022). In Chongqing, Zanthoxylum Schinifolium, one of the Zanthoxylum species, is widely grown. Its planting area is expanding and output is increasing year by year. In 2019, the planting area and yield of Zanthoxylum in Chongqing increased over the previous year by 4.3% and 17.8%, respectively, reached 73,000 hectares and 435,000 tons, respectively (Kuang *et al.*, 2020). The infructescence number of immature Zanthoxylum can to a certain extent predict the future yield. Manual sampling is labor-intensive and inefficient. Therefore, it is crucial to provide a machine vision-based method for identifying Zanthoxylum infructescences.

When extracting the infructescence regions of Zanthoxylum, each infructescence is regarded as a cluster composed of a series of fruits. In recent years, various techniques have been developed for automatic fruit detection, which can help improves the efficiency, functionality, intelligence and remote interactivity of harvesting robots in complex agricultural environments (Tang *et al.*, 2020; Li *et al.*, 2022). And the follows are commonly adopted to achieve the goal, such as deep learning, Hough Circle Transform (CHT), Local Binary Pattern (LBP), and stereo vision.

There is no doubt that deep learning has gained wide application in intelligent agriculture recently. A video processing method was developed by Gao *et al.* (2022) to improve the detection accuracy of apple fruits in orchard environment with modern vertical fruiting-wall architecture by introducing YOLOv4-tiny network. Lv *et al.* (2022) drew a visual recognition way of apple growth morphology. They designed a model named YOLOv5-B, which embedded BiFPN-5 and ACON-C in YOLOv5 to improve the performance. It achieved a high accuracy in real time. Zhou *et al*. (2022) and Tang *et al*. (2023) presented methods to recognize and locate Camellia oleifera fruits based on YOLOv7 and YOLOv4 respectively. Both these two algorithms display their excellent performance in locating oleifera fruits. In addition to its application in fruit recognition, deep learning-based methods have also been applied to some other fields. Researchers (Ji *et al*., 2023) designed a target detection method based on multi-scale pyramid fusion image enhancement and the MobileCenterNet model to achieve rapid and accurate detection of pond cultured river crab. Xu *et al*. (2023) improved YOLOv5 to address the problems of low grading accuracy and slow grading speed in the apple grading process.

However, there are some limitations to deep learning-based approaches, training is expensive in certain circumstances due to their reliance on high-performance hardware, high time cost, a large number of labeled samples, and a multitude of parameters. This gives traditional machine learning algorithms an edge over deep learning under constrained conditions. Lin *et al.* (2020) proposed a novel technique for fruit detection in natural environments which is applicable. A novel probabilistic CHT is developed to obtain fruit candidates. It is competitive for detecting most type of fruits in natural environments. For the purpose of extracting possible fruit regions, an algorithm (Lu *et al.*, 2018) combining LBP and edge hierarchy was designed. It can obtain 82.3% accuracy only relying on texture and intensity distribution features. Researchers in China Agricultural University (Zhang *et al.*, 2021) adopted 3D point cloud images obtained by an RGB-D camera to recognize pomegranate fruits. The algorithm finally obtained a Recall of 87.74%. In recent years, Histogram Oriented Gradients (HOG) features (Dalal and Triggs, 2005), which are often used in human detection, have recently been introduced for fruit detecting. Scholars (Tan *et al.*, 2018) developed an approach to recognize blueberry fruit of different maturity in outdoor scenes. HOG feature vectors are calculated from 1374 patches which were cropped from the original color images, and a linear Support Vector Machine (SVM) classifier is trained to detect fruit-like regions rapidly.

Most of the above methods are presented for detecting individual fruits, while the infructescence of Zanthoxylum is a string of fruits. Thus, it is necessary to implement clustering based on distribution information for detecting infructescences after obtaining the fruit-like regions. To fit crop rows by location clustering, Zhang *et al.* (2018) developed a method based on the extracted feature regions. It obtains the feature points of final cluster by combining location clustering and the shortest path. The crop rows are fitted with a linear regression algorithm. Ma *et al.* (2021) raised a robust crop root row detection algorithm based on line clustering and supervised learning, which obtains the crop rows through the linear clustering algorithm and performs anomaly detection. An approach (Biglia *et al.*, 2022) was established to detect vine rows automatically within 3D point clouds of vineyards based on the detection of key points and a density-based approach. The results showed that the detection was found to be 100% in accordance with the manual one.

Zanthoxylum infructescence has cluster-like construction in natural environment. Mature Zanthoxylum are generally red and different from the background. There are few studies focus on Zanthoxylum detection. Xu *et al.* (2022) presented a Zanthoxylum-picking-robot target detection method based on improved YOLOv5s. Firstly, CBF module based on the CBH module is improved in the backbone to promote the detection accuracy. Then, a Specter module based on CBF is presented to replace the bottleneck CSP module, which improves the speed of detection with a lightweight structure. Finally, the algorithm is checked by the improved YOLOv5 framework, and the differences in detection between YOLOv3, YOLOv4 and YOLOv5 are analyzed and evaluated.

Nevertheless, the fruits of Zanthoxylum Schinifolium are green and the images of the Zanthoxylum Schinifolium's infructescence collected in natural environment are complex because there are weeds in the background while the fruits are small. Thus, they are difficult to identify. Meanwhile, the available sample set is small. These cause that the existing methods cannot be applied directly to detect the infructescences of Zanthoxylum. For detecting Zanthoxylum infructescence, a framework based on adaptive density clustering is proposed to support the further studies. More precisely, the main work in this paper is summarized below:

(1) A feasible framework is developed to detect the infructescences of Zanthoxylum automatically in complex environment, which transform the infructescence detection into density

clustering for fruit regions.

(2) A method for extracting fruit-like region is designed in the framework, which integrates color and morphological features.

(3) A novel density-based method, using a new density metric and an improved clustering validity index, is designed to solve the problem of the existing density clustering relying on hyperparameters.

To the author's best knowledge, such efforts have never been seen in any prior work.

The details of collecting samples and proposed method are described in Section 2. Experiments on the proposed algorithm are carried out, and the effectiveness of the algorithm are demonstrated in Section 3. In the last section, the conclusion is given.

Materials and Methods

Data acquisition

Image acquisition

According to the growth stages of Zanthoxylum, the samples were collected between late March and early April 2021, after the blooming stage. These images were taken in Bishan District, Chongqing, China (near the 29°36′1.95″N, 106°11′14.48″E), and 307 images in total were captured manually using a portable device with fixed camera parameters over several days. Samples were saved as 24-bit color JPG images with a resolution of 6016px × 4512px and a focal length of 7mm. The distance between the infructescences and lens is 150 - 250 mm.

Dataset partitioning

Seventy-five typical samples were randomly selected to verify the performance of the proposed algorithm. They were resized to 1200px × 900px by bilinear interpolation. And two sample sets with different conditions were created as follows. The detailed distribution of the samples is shown in Table 1. And some typical images with different conditions are displayed in Figure 1.

a. Normal sample set: Images were collected under normal natural conditions without complex situations listed below;

b. Robust sample set: Images were collected with one or more of following conditions.

Four complex situations:

1) The leaves cover part of the infructescence area;

2) The infructescence overlap each other;

3) Complex elements in background with wildflowers or large areas of weeds;

4) The environment was either too bright or too dark.

Description of infructescence detection framework

This research proposes a feasible method for detecting infructescences, the framework is shown as Figure 2. It can be divided into two parts. The first part is fruits region extraction and the second part is density-based clustering on the distribution of fruit regions.

As the first part shown in Figure 2, HOG operator and vegetation index are introduced for extracting morphologic feature and segmenting the fruit areas from the image. First, the feature map of the HOG is obtained, thresholding is performed on the feature map later, and the Joint-Direction-Intensity feature is constructed. Then the excess green minus excess red (ExGR) index (Meyer *et al.*, 2004) of the original image is calculated, and the plant area and non-plant area is segmented on the basis. The results of the above steps are then combined to obtain fruit regions.

The second part in the framework, density-based clustering, consists of three main steps: Firstly, the fruit regions are roughly clustered based on the density information of the fruit regions. Then, the clustering validity index (CVI) is adopted to further merge the results of the rough clustering, and the optimal clustering result is obtained. Finally, based on the clustering results, the minimum bounding rectangle of each cluster are found to realize the detection and counting of the infructescences.

Fruits extraction

Joint-direction-intensity feature

An infructescence of Zanthoxylum consists multiple spherical fruits, as shown in Figure 2(a), and descriptions in this section are all based on this figure. To strengthen the distinction between infructescences and background, HOG is imported to extract morphological features of fruits, which is shown as Figure 2(b). Since the diameter of each fruit is about 10-18px, the size of each cell is set to 14, named *cell_size*.

HOG features indicate that if a cell contains infructescence, then its gradients on each direction are similar and large. The HOG feature map is binarized with a threshold of 127, simple but effective, as shown in Figure 2(c). Using the thresholding approach, the Joint-Direction-Intensity feature map is obtained for further fruit extraction. The binarized HOG feature map undergoes further processing to determine the number of directions for each cell, which is typically proportionate to the number of white pixels within it. The pixels in a cell with more than 5 directions are marked as 1 and the rest is 0. Joint-Direction-Intensity feature map is named as *I*.

Segmentation of plant and non-plant

Various background clutters in images collected in wild environment. To reduce the influence of background and improve the accuracy of the algorithm, a grayscale image is created to mark a pixel is in green plant (infructescence regions, leaves, and weeds) or non-plant (soil, branches, and shadows, etc.) by calculating ExGR, which is defined as Eq. 1:

$$\begin{aligned} ExGR &= ExG - ExR \\ &= (2g - r - b) - (1.4r - g) \qquad (1) \\ &= 3g - 2.4r - b \end{aligned}$$

where *r*, *g* and *b* are the normalized chrominance channel values. An example is shown as Figure 2(d).

The histograms of ExGR images exhibit bimodal characteristics, the valley between its two peaks can be regarded as the threshold to segmentate of plant and non-plant area. The outcome of threshold segmentation is a binary image, denoted as *PM*, in which plant area is 1 and 0 means non-plant area, as shown in Figure 2(e).

Fruit region extraction

The Joint-Direction-Intensity feature map *I* is built to identify circulars in an image. ExGR is introduced to segment plant and non-plant area, then produce a binary map named *PM*. Since fruits of Zanthoxylum are usually small green circulars, it's difficult to locate fruits using only Joint-Direction-Intensity feature or ExGR segmentation. Therefore, these two features are considered together, and one region is a fruit-like region only when both features are positive. In another words, a region is identified as a fruit when it is both circular-like and green. The Hadamard product of *I* and *PM* can help find the fruit-like regions, which is shown as Eq. 2.

$$F = I \odot PM \qquad (2)$$

where *I* and *PM* are defined in 2.3.1 and 2.3.2, $\odot$ represents the Hadamard product.

Then, *F* is divided into *m*n* cells with *cell_size*, where *m=width/cell_size*, *n=height/cell_size*, *width* and *height* are the size of origin image. Finally, cells with more than 1/3 non-zero elements are defined as fruit regions, each of which is equated to a pixel of *F′*, and then marked as 1; the others are non-fruit regions and marked as 0. *F′* is the result of fruit region extraction and its size is $m*n$, which is shown as Figure 2(f).

Density-based rough clustering

Zanthoxylum infructescences can be regarded as a series of non-spherical clusters. To identify infructescences, a density-based clustering approach is implemented. However, existing density clustering algorithms, such as DBSCAN (Ester *et al.*, 1996), OPTICS (Ankerst *et al.*, 1999) and DPC (Rodriguez and Laio, 2014), are difficult to apply to this study due to the inconsistent amount of infructescences, the large density differences among clusters, and the vast number of required hyperparameters. Hence, an adaptive rough clustering method based on density is proposed in this study.

Local density map

Since the cluster centers are generally the points of maximum density value in their neighbors, the maximum points of local density are chosen to be the candidate cluster centers. The clusters are then expanded from them. An excellent density metric can help find the true center of clusters more accurately. Therefore, a novel density metric is designed for measuring the density of each point.

Due to the coordinates of the candidate regions extracted in the previous step are all integers, *F′* can be seen as a grid graph, each grid is a point in *F′*. Focusing on the 8-direction, $\Theta = \{0, \pi/4, \pi/2, 3\pi/4, \pi, 5\pi/4, 3\pi/2, 7\pi/4\}$, the density ρ at $F'(x, y)$ can be defined as Eq. 3:

$$\rho(x,y)=\sum_{\theta\in\Theta} den(x,y,\theta) \qquad (3)$$

where $den\ (x, y, \theta)$ is the density of the $F'(x, y)$ on the direction θ. The local density map obtained by Eq. 3 is named as *DM*.

The calculation of the density on the direction θ is an iterative process which executes when the grid's value is 1. During the iteration, the grids on the direction θ are traversed, and the feature values on the path are weighted and accumulated. The iteration stops until reaching the boundary or feature value is 0. An example is showed as Figure 3.

The density of q_{th} iteration is denoted as $den\ (x, y, \theta)^{(q)}$, and the calculation formula is shown as

Eq. 4:

$$den\left(x,y,\theta\right)^{(q)}=\begin{cases}0,q=0\\ den\left(x,y,\theta\right)^{(q-1)}+f_{travel}\cdot g\left(q\right),q>0\,\&\,f_{travel}\neq 0\end{cases}\qquad(4)$$

in which, $g(\cdot)$ is a Gaussian function. It implies that the contribution of its feature values decreases as distance increases. f_{travel} represents the feature value of the point being iterated, which is defined as Eq. 5.

$$f_{travel}=\begin{cases}F'\left(x+sign\left(\tan\theta\right)\Delta x,y+sign\left(\sin\theta\right)\Delta y\right),\left|\tan\theta\right|\leq 1\\ F'\left(x,y+sign\left(\sin\theta\right)q\right),\left|\tan\theta\right|=+\infty\end{cases}\qquad(5)$$

where Δx and Δy are the offsets to $F'(x, y)$, $sign(\cdot)$ is a sign function, they are defined as Eq. 6.

$$\Delta x=\begin{cases}q,\left|\tan\theta\right|\leq 1\\ q\left|\cot\theta\right|,1<\left|\tan\theta\right|<+\infty\end{cases},\Delta y=\begin{cases}q,1<\left|\tan\theta\right|\leq+\infty\\ q\left|\tan\theta\right|,\left|\tan\theta\right|\leq 1\end{cases},sign(x)=\begin{cases}1,x>0\\ -1,x<0\end{cases}\qquad(6)$$

Feature weight map and feature density map

In Joint-Direction-Intensity feature map I, the number of directions is proportional to the ratio of white pixels to the entire cell, it also indicates the possibility that a cell is a fruit-like region. Hence, the proportion of white pixels in a cell can be used as a weight to measure the likelihood of a cell is a fruit-like region. And each of the cell will be mapped to a pixel of feature weight map FW with a size as F'. The feature value of $cell_i$ can calculate as Eq. 7:

$$FW\left(cell_i\right)=\sum_{x\in cell_i}\chi\left(x\right)\Big/cell\_size^2\qquad(7)$$

where $\chi\left(x\right)$=1 if $x = 255$ and $\chi\left(x\right)$= 0 otherwise. A point is more likely to be the center of a cluster if it is more possible to be a fruit-like region and has a larger local density. Hence, Hadamard product of DM and FW is calculated, which is named as feature density map FDM.

Maximum density points

Maximum density points are found to start clustering after getting the FDW. Let FDW be a bivariate function, which has first and second derivatives. The second derivatives f_{xx}, f_{yy} and f_{xy} of the point (x, y) in FDW compose its Hessian matrix, denoted as $H(x, y)$. It is a maximum density point if the corresponding Hessian matrix $H(x, y)$ is positive definite. An example is shown as Figure 2(g), in which the red plus signs are the maximum density points.

Density-based rough clustering

The following demonstrates the fundamental idea of the proposed density-based rough clustering: Initially, the maximum density points of *FDM* are seen as initial centers, and each is labeled uniquely. The neighbors of centers are then connected to the corresponding center. Framework of the density-based rough clustering algorithm is shown as Algorithm 1.

Algorithm 1: *Framework of the density-based rough clustering*

```
Input:    Fruit regions.
Output:   Result of density-based rough clustering.
1.        Calculate the density of each point according to Eq. 3, and get the density map DM.
          Obtain feature weight map FW according to Eq. 7.
2.        Calculate the Hadamard product of DM and FW to get the feature density map FDM.
3.        The non-zero elements in FDM are extracted in the order of row-first to form $X = \{x_1,$
          $x_2, \ldots, x_N\}$.
          Find the maximum density points of FDM as the initial cluster centers.
4.        Assign labels to centers and form a series of cluster sets called cluster_list which only
5.        contains corresponding centers.
          Allocate unmarked points to the possible cluster sets according to Algorithm 2.
6.        Merge clusters with duplicate elements into one cluster to update cluster_list by
          Algorithm 3.
7.        Return cluster_list.

8.
```

Step 6 of Algorithm 1 is described in detail below: Firstly, a flag *is_changed* is initialized as 0, which indicates whether the *cluster_list* has been modified. Then, the algorithm traverses points and appends the unallocated points to the cluster sets that contain any of their 4-neighbors. If there is any append operation during current loop, *is_changed* will be changed to 1. The loop will terminate when *is_changed* is 0 after a certain loop. In particular, *is_changed* will be set to 0 in every start of a loop. During the point allocation, a point may belong to multiple cluster sets in Step 6 of Algorithm 1. For further data process, the repeatedly assigned points are temporarily stored into the cluster sets. The pseudocode of points allocation is shown as Algorithm 2.

Algorithm 2: *Points allocation* (Step 6 of Algorithm 1)

```
Input:  Points $X = \{x_1, x_2, \dots, x_N\}$, cluster sets cluster_list.
Output: The updated cluster set cluster_list.
Initializing: is_changed=0
while True do
    is_changed=0
    for $i$ from 1 to len($X$) do
        if $x_i$ not in any cluster_list then
            for each $x_j$ in 4-neighbors of $x_i$ do
                if $x_j$ in cluster_list then
                    cluster_list[index of cluster set $x_j$ belongs].append($x_i$)
                    is_changed = 1
                end if
            end for
        end if
    end for
    if is_changed == 0 then
        break
    end if
end while
```

The remaining unallocated points are considered as outliers and will be ignored in the further process. Algorithm 2 captures a series of clusters are separated from each other. However, the same points may be assigned to more than one set of clusters due to flaws in Algorithm 2. That is, there are the same points exist in different sets of clusters. Hence, it is necessary to merge these clusters.

The main idea: An intersection between two cluster sets is found within a loop. If the intersection is not empty, the elements in the cluster set with larger index will be copied to another set and elements in the cluster with larger index will be deleted, that is, it will be marked as empty. This process is repeated until the intersection of any two sets is empty in *cluster_list*. The process is shown as Algorithm 3.

Algorithm 3: *Cluster sets merging* (Step 7 of Algorithm 1)

Input: Cluster set *cluster_list*.

Output: Density-based rough clustering results.

```
for i from 1 to len(cluster_list)-1 do
    for j from (i+1) to len(cluster_list) do
        if cluster_list [i]∩cluster_list [j] is not empty then
            cluster_list [i] = cluster_list [i]∪cluster_list [j]
            cluster_list [j] = empty
        end if
    end for
end for
```

The new cluster sets, reconstructed by the nonempty cluster sets in *cluster_list*, are obtained after merging the clusters with duplicate elements. Labels are re-assigned to points in the new cluster sets and fruit regions are divided. The result of density-based rough clustering is shown as Figure 2(h).

Merge clusters based on Local Calinski-Harabasz index

Results of the density-based rough clustering algorithm, shown as Figure 2(h), demonstrated that multiple cluster groups could belong to a single cluster, such as cluster 2 & 4 and cluster 15 & 16. The number of clusters can generally be determined by clustering validity index (CVI) because it is proved that the optimal CVI corresponds to the optimal number of clusters and the optimal clusters by Zhu and Ma (2018). Therefore, CVIs are introduced to obtain the optimal cluster by iteratively merging the rough clustering results.

Original Calinski-Harabasz index

The original CH index (Caliński and Harabasz, 1974) is the ratio of the inter-cluster dispersion degree and the intra-cluster dispersion degree. That is, the CH index increases when intra-cluster dispersion degree decreases and inter-cluster dispersion degree increases, which is defined as Eq. 8:

$$CH\left(k\right)=\frac{Tr\left(B_k\right)}{Tr\left(W_k\right)}\times\frac{N-k}{N-1} \tag{8}$$

where N is the number of points, B_k and W_k are the inter-cluster scatter matrix and intra-cluster scatter matrix, respectively. $Tr(\cdot)$ represents the trace of a matrix, which indicates the dispersion degree. The definitions of B_k and W_k are shown in Eq. 9 and Eq. 10.

$$B_k = \sum_i |C_i| (v_i - v)(v_i - v)^T \tag{9}$$

$$W_k = \sum_{i=1}^{k} \sum_{x \in C_i} (x - v_i)(x - v_i)^T \tag{10}$$

where v_i is the center of the cluster C_i, v is the global center, k is the number of clusters and $|C_i|$ represents the number of points contained in cluster C_i, v_i and v are defined as follows.

$$v_i = \frac{1}{|C_i|} \sum_{x \in C_i} x \tag{11}$$

$$v = \frac{1}{N} \sum_{i=1}^{k} \sum_{x \in C_i} x \tag{12}$$

Local Calinski-Harabasz index

The original CH index is a global evaluation index to evaluate the quality of clustering. Its inter-cluster dispersion is transformed from a global metric to a local one, which allows it to be utilized to evaluate the scatter between any two adjacent clusters. The new indicator is denoted as Local Calinski-Harabasz index, referred to as *LCH*, which is defined as Eq. 13.

$$LCH(p, q, k) = \frac{Tr(B_k)}{Tr(W_k')} \times \frac{N - k}{k - 1} \tag{13}$$

where p and q are the labels of the clusters that are being merged. C' is the cluster that is formed by merging cluster p and cluster q, that is $C' = C_p \cup C_q$. W_k' is the local scatter matrix, representing the intra-cluster dispersion degree of the cluster C'. W_k' is defined as Eq. 14:

$$W_k' = \sum_{x \in C'} (x - v')(x - v')^T \tag{14}$$

where v' is the center of cluster C'.

Local Calinski-Harabasz-based cluster mergence

The *LCH* is introduced to evaluate the suitability of merging two adjacent clusters. The higher the *LCH* value, the more appropriate it is to combine the two clusters. Multiple merge operation based on *LCH* are performed to get the optimal clustering result. During each mergence, the maximum value of *LCH* between all clusters and their nearest neighbors are found out and corresponding *p* and *q* are the cluster labels which are taken as the optimal choice for this mergence. They are found by Eq. 15.

$$\underset{p,q}{\operatorname{argmax}} LCH\left(p,q,k\right) \qquad (15)$$

The next mergence will be performed based on the previous mergence. *SC*, as known as silhouette coefficient (Maulik and Bandyopadhyay, 2002), serves as the global assessment indicator to determine a termination condition of the merging progress. The silhouette coefficient of t_{th} mergence is regarded as $SC^{(t)}$ whose range is [-1, 1]. Increases in the coefficient indicate more effective clustering. Thus, the termination condition is as follows:

$$\underset{t}{\operatorname{argmax}} SC^{(t)} \qquad (16)$$

The pseudocode of the *LCH*-based cluster merging algorithm is shown in Algorithm 4.

Algorithm 4: *LCH-based cluster mergence*

```
Input:   Result of density-based rough clustering.
Output:  The optimal clusters.
while True do
|   for $v_j$ in centers do
|   |   $v_i = \underset{v_x}{\operatorname{argmin}} d\left(v_x, v_j\right), \text{where } x \neq j$
|   |   Calculate the LCH between $v_i$ and $v_j$ by Eq. 13
|   end for
|   Find the optimal merging with Eq. 15 and record the p and q
|   Calculate the silhouette coefficient of the current merging, and remember it
|   if number of clusters==2 then
|   |   break
|   end if
end while
Find the optimal merge result by Eq. 16 and return the corresponding mergence.
```

Adaptive density-based clustering algorithm

After the clusters are merged, which is displayed in Figure 2(i), the number of clusters obtained is the number of Zanthoxylum infructescence. The infructescence detection result is shown in Figure

2(j). The minimum bounding rectangle of each cluster is found in units of clusters and mapped to the corresponding coordinates in the original image. The detailed framework of the proposed algorithm is shown as Algorithm 5.

Algorithm 5: *Framework of adaptive-density-clustering-based Zanthoxylum infructescence detection*

```
Input:   The Zanthoxylum infructescence image img.
Output:  The number of infructescence and the location of infructescence.
1.       Extract the HOG feature map $I_{HOG}$ from original image img.
2.       Obtain the thresholded HOG feature map $I_{HOG}'$.
3.       Extract the Joint-Direction-Intensity feature map $I$.
4.       Calculate the ExGR index, and segment the plant and non-plant area to
         get $PM$.
5.       Fuse $I$ and $PM$ to obtain the fruit regions $F'$.
6.       Perform density-based rough clustering (Algorithm 1).
7.       Iteratively merge the rough clustering results of Step6, obtain the optimal
         clustering by Algorithm 4.
8.       Mark the infructescence positions in the original image img and calculate
         the number of infructescence of infructescence based on the clustering
         results.
```

Results

The experiments are designed as two parts to verify the effectiveness of the proposed algorithm. The first is the comparative experiment with traditional methods, it compares the proposed algorithm with ADPC-kNN (Yaohui *et al.*, 2017), which is an extension of DPC (Rodriguez and Laio, 2014), DBSCAN (Ester *et al.*, 1996) and OPTICS (Ankerst *et al.*, 1999) to test the performance of clustering. In the second part of experiment, several classical deep learning models are introduced for comparison in this paper, including the multi-stage Faster RCNN (Ren *et al.*, 2015) and the single-stage YOLOv5 (Jocher *et al.*, 2022) and YOLOv7 (Wang *et al.*, 2022), to demonstrate the competitiveness of the proposed algorithms under small data sets. Each part of the experiments will be conducted on normal sample set and robust sample set mentioned in sub-section 2.1.2 to prove the effectiveness of the proposed algorithm.

The process of Zanthoxylum infructescence detection described in this work was all implemented on a device with a CPU of AMD Ryzen 7 5800H @ 3.20GHz, a RAM of 24.0GB, and a graphics card of NVIDIA GeForce RTX3060 Laptop with video memory of 6GB. The process of image is implemented in Python 3.6.8 and opencv-python 4.4.4 environments. The deep learning models were also trained on this device.

Evaluation metrics

To assess and quantify the performance of the proposed adaptive density-based Zanthoxylum infructescence detection method, three commonly used indices are introduced: Precision, Recall and F1-score. Their formulas are shown as Eq. 17-19:

$$Precision = \frac{TP}{TP + FP} \tag{17}$$

$$Recall = \frac{TP}{TP + FN} \tag{18}$$

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{19}$$

where *TP* is the number of infructescences that are correctly identified by the proposed algorithm; *FN* represents the number of non-Zanthoxylum infructescence regions that are misclassified to infructescence regions; *FP* refers to the number of Zanthoxylum infructescences that are misidentified by the algorithm. Since the algorithm proposed in this paper is only adopted for detecting the infructescences, it is defined that Zanthoxylum infructescence is correctly identified when the center of the detection box falls in the ground truth and will be marked as *TP*. At the same time, a box of ground truth can only be correctly recognized by one detection box. If multiple detection boxes identify a same infructescence, the remaining detection boxes will be seen as failed detections, and they will be marked as *FN*.

Performance comparison with traditional methods

Although there are some reports of fruits counting and detection with good results (Lu *et al.*, 2018; Tan *et al.*, 2018; C. Zhang *et al.*, 2021), it is difficult to compare our research with them because the data processed by each research are all captured from different species, and even the features vary greatly. Therefore, many classical clustering algorithms (Ankerst *et al.*, 1999; Ester *et al.*, 1996; Yaohui *et al.*, 2017) are chosen to prove the effectiveness of the proposed clustering algorithm. Part of the experimental results are shown as Figure 4.

Figure 4(a-d) show the results of DBSCAN, OPTICS, ADPC-kNN and ours. The results of infructescence detection are displayed with the red boxes, and ground truth is shown with blue borders. The experimental results on different sample sets in terms of Precision, Recall and F1-score are compared, and are shown in Table 2. In all following tables, the best results are bolded, and the second best are underlined.

The comparison algorithm and the clustering proposed algorithm was analyzed and their required parameters and corresponding numbers are shown in Table 3.

It is found that our algorithm obtains better accuracy than ADPC-kNN and OPTICS while using fewer parameters, and the overall accuracy is slightly better than DBSCAN. Since OPTICS only considers density information, the necessary cluster merging operation is missing. And as the result, many small clusters are detected, which leads that OPTICS has high Precision, but low Recall and F1-score, as shown in Figure 5(b). The ADPC-kNN needs to manually specify the number of clusters, which is not suitable for this work. At the same time, when a larger number of clusters is specified, some outliers may be mistakenly selected as the cluster center due to their large distance between each other, as shown in Figure 5(c). Although the accuracy of the proposed algorithm is only slightly better than that of DBSCAN, it requires fewer parameters and is more adaptive. Therefore, the proposed algorithm has higher accuracy and better adaptability than the comparison algorithms. Though the scores are decreased slightly on robust sample set, our results, and the standard deviation of ours on average Recall and average F1-score is better than that of the comparison algorithms. That is, the proposed algorithm also shows better performance and better stability than other algorithms. It proves the application value of the proposed algorithm in complex environment.

In addition to Precision, Recall, and F1-score of infructescence detection, several other external evaluation indicators of clustering algorithms such as Rand index (RI), adjusted Rand index (ARI), normalized mutual information (NMI) and Fowlkes-Mallows scores (FMI) are introduced to measure how well the clusters obtained by each algorithm match the ground truth to evaluate the performance of different clustering algorithms. The detailed data of which is shown in Table 4. It can be found that the proposed method obtains the best clustering results for each index.

Performance comparison with deep learning-based methods

There is no doubt that deep learning-based methods are competitive in the field of fruit detection. And there are several neural network models like Faster RCNN, YOLOv5, and YOLOv7, which are representative models for multi-stage and single-stage target detection. The key parameters of training these models are shown in Table 5. And the learning rate will decay with the increases of training epochs.

In this section, the optimal result of each model is chosen to compare with our algorithm. The optimal detection results of each algorithm are shown in Table 6, and several detection results of these models and ours are randomly displayed in Figure 6.

During the experiment, it can be found that most of the Zanthoxylum infructescences can be detected at low confidence, but the misclassification rate and repeat detection rate are higher at this time, and these lead to a lower Recall and F1-score in a comprehensive view. With the appropriate confidence, its misclassification rate and misdetection rate are reduced, but it also sacrifices its accuracy in complex situations, and its Recall rate and F1-score of normal samples are improved somewhat. In the high-confidence case, more infructescences are ignored and the misdetection rate is further increased, resulting in a decrease in all indicators at this time.

Overall, the results of the deep learning model for detecting Zanthoxylum infructescences vary considerably with the confidence settings. However, even when the detection results are optimal, the detection accuracy of YOLOv5 and Faster RCNN is still worse than the ones of the proposed algorithm. Meanwhile, the results obtained by the algorithm proposed in this paper are also competitive compared to the latest YOLOv7 though the pre-trained weight is introduced. It proves that the algorithm in this paper is valuable in current application scenarios and small sample cases.

Discussion

The proposed method exhibits excellent infructescences detection performance under natural conditions. The average precision is 94.17% in total (in normal sample set, its precision is 96.67%, and the precision is 94.09% for robust samples), although there are still some misclassifications caused by defects in feature extraction and under-merging. Due to the complexity of the situation, some fruit regions may be misclassified during the feature extraction, which is shown as Figure 7(a).

In some cases, the extracted fruit areas are separated when the infructescence is large and the internal fruits are sparser, which further leads to under-merging, as shown in Figure 7(b). While in other conditions, several infructescences may overlap together and the clustering algorithm may fail to work, which is shown as Figure 7(c). However, the task does not require to detect the precise position of infructescence, and most of the infructescences can be successfully identified, so the algorithm is able to preliminarily complete the task of infructescence counting and estimating yield. Since HOG and ExGR are both normalized in the calculation and they are insensitive to illumination conditions, the algorithm can effectively identify the infructescences under different illumination conditions.

Conclusions

In this paper, a study on the density-based infructescence detection method for Zanthoxylum Schinifolium is conducted, which is focused on the images of Zanthoxylum branches collected under natural environment. A feasible automatic recognition approach for Zanthoxylum infructescences is provided in view of its complex background, difficulty in recognition of Zanthoxylum and small sample size, which could provide data support for estimating its yield in the future. The major findings of this study can be summarized as following:

(1) Combine Joint-Direction-Intensity feature, which is an extension of HOG, and ExGR index to extract fruit-like regions as accurately as possible from images with complex background.

(2) Build an optimization model to optimize clustering results, and proposed a novel adaptive density-based clustering algorithm for detecting the infructescences according to distribution of fruit-like regions.

(3) Experiments show that the proposed algorithm exhibits excellent infructescences detection performance with an average Precision of 94.17%.

(4) Compared with OPTICS, ADPC-kNN and DBSCAN, the proposed adaptive density clustering has a higher accuracy in locating infructescences under normal condition while it requires fewer parameters. Meanwhile, it also has higher applicability for robust samples.

(5) Compared with deep learning-based object detection algorithms, the algorithm proposed in this paper is proved to be valuable in the case of small samples. All indicators are significantly higher than Faster RCNN and YOLOv5, and is also competitive to the latest YOLOv7.

To realize the automatic yield estimation of Zanthoxylum, further research on the growth density and area size of Zanthoxylum is also needed to build the relationship between them and the yield. In the future, experiments can be conducted on more scenes in the natural environment, such as more complex lighting conditions, and Zanthoxylum at different growth stages. Meanwhile, more samples should be collected so that the advanced technologies, such as deep learning, can be introduced to conduct further research. Moreover, the precented algorithm can be transferred to other related studies of varieties with similar characteristics, such as the identification of immature grape.

References

Ankerst, M., Breunig, M. M., Kriegel, H.-P., Sander, J. 1999. OPTICS: Ordering points to identify the clustering structure. Sigmod Rec. 28:49–60.

Biglia, A., Zaman, S., Gay, P. 2022. 3D point cloud density-based segmentation for vine rows detection and localisation. Comput. Electron. Agr. 199:107166.

Caliński, T., Harabasz, J. 1974. A dendrite method for cluster analysis. Commun. Stat-Theor. M. 3:1–27.

Dalal, N., Triggs, B. 2005. Histograms of oriented gradients for human detection. 2005 IEEE. Comput. Soc. Conf., San Diego, CA, USA, 1:886–893.

Ester, M., Kriegel, H.-P., Sander, J., Xu, X. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. ACM SIGKDD Conference on Knowledge Discovery and Data Mining., Portland, OR, USA, 96:226–231.

Gao, F., Fang, W., Sun, X., Wu, Z., Zhao, G., Li, G., Li, R., Fu, L., Zhang, Q. 2022. A novel apple fruit detection and counting methodology based on deep learning and trunk tracking in modern orchard. Comput. Electron. Agr. 197:107000.

Ji, W., Peng, J., Xu, B., Zhang, T., 2023. Real-time detection of underwater river crab based on multi-scale pyramid fusion image enhancement and MobileCenterNet model. Comput. Electron. Agr. 204:107522.

Jocher, G., Ayush Chaurasia, Stoken, A., Borovec, J., NanoCode012, Yonghye Kwon, Kalen Michael, TaoXie, Jiacong Fang, Imyhxy, Lorna, Zeng Yifu, Wong, C., Abhiram V, Montes, D., Zhiqiang Wang, Fati, C., Jebastin Nadar, Laughing, UnglvKitDe, Sonck, V., Tkianai, YxNONG, Skalski, P., Hogan, A., Dhruv Nair, Strobel, M., Jain, M., 2022. YOLOv5 by Ultralytics, GitHub. Available from: https://github.com/ultralytics/yolov5.

Kuang, M., Zhang, L., Li, S., Yang, S., Qu, S., Dong, P. 2020. Problems and countermeasures of pepper industry development in Chongqing. South China Agriculture. 11–13.

Li, C., Tang, Y., Zou, X., Zhang, P., Lin, J., Lian, G., Pan, Y., 2022. A Novel Agricultural Machinery Intelligent Design System Based on Integrating Image Processing and Knowledge Reasoning. Appl. Sci-Basel. 12:7900.

Lin, G., Tang, Y., Zou, X., Cheng, J., Xiong, J., Fruit detection in natural environment using partial shape matching and probabilistic Hough transform[J]. Precis. Agric. 2020, 21:160-177.

Lu, J., Lee, W. S., Gan, H., Hu, X. 2018. Immature citrus fruit detection based on local binary pattern feature and hierarchical contour analysis. Biosyst. Eng. 171:78–90.

Lv, J., Xu, H., Han, Y., Lu, W., Xu, L., Rong, H., Yang, B., Zou, L., Ma, Z. 2022. A visual identification method for the apple growth forms in the orchard. Comput. Electron. Agr. 197:106954.

Ma, Z., Tao, Z., Du, X., Yu, Y., Wu, C. 2021. Automatic detection of crop root rows in paddy fields based on straight-line clustering algorithm and supervised learning method. Biosyst. Eng. 211:63–76.

Maulik, U., Bandyopadhyay, S. 2002. Performance evaluation of some clustering algorithms and validity indices. IEEE T. Pattern Anal. 24:1650–1654.

Meyer, G. E., Neto, J. C., Jones, D. D., Hindman, T. W. 2004. Intensified fuzzy clusters for classifying plant, soil, and residue regions of interest from color images. Comput. Electron. Agr. 42:161–180.

Ren, S., He, K., Girshick, R., Sun, J., 2015. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. Adv. Neur. In. 28.

Rodriguez, A., Laio, A. 2014. Clustering by fast search and find of density peaks. Science. 344:1492–1496.

Tan, K., Lee, W. S., Gan, H., Wang, S. 2018. Recognising blueberry fruit of different maturity using histogram oriented gradients and colour features in outdoor scenes. Biosyst. Eng. 176:59–72.

Tang, Y., Chen, M., Wang, C., Luo, L., Li, J., Lian, G., Zou, X., 2020. Recognition and Localization Methods for Vision-Based Fruit Picking Robots: A Review. Front. Plant. Sci. 11:510.

Tang, Y., Zhou, H., Wang, H., Zhang, Y., 2023. Fruit detection and positioning technology for a Camellia oleifera C. Abel orchard based on improved YOLOv4-tiny model and binocular stereo vision. Expert. Syst. Appl. 211:118573.

Wang, C.-Y., Bochkovskiy, A., Liao, H.-Y.M., 2022. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. arXiv Preprint. arXiv: 2207.02696v1.

Xu, B., Cui, X., Ji, W., Yuan, H., Wang, J., 2023. Apple Grading Method Design and Implementation for Automatic Grader Based on Improved YOLOv5. Agriculture. 13:124.

Xu, Z., Huang, X., Huang, Y., Sun, H., Wan, F. 2022. A Real-Time Zanthoxylum Target Detection Method for an Intelligent Picking Robot under a Complex Background, Based on an Improved YOLOv5s Architecture. Sensors. 22:Article 2.

Yaohui, L., Zhengming, M., Fang, Y. 2017. Adaptive density peak clustering based on K-nearest neighbors with aggregating strategy. Knowl-Based. Syst. 133:208–220.

Zhang, C., Zhang, K., Ge, L., Zou, K., Wang, S., Zhang, J., Li, W. 2021. A method for organs classification and fruit counting on pomegranate trees based on multi-features fusion and support vector machine by 3D point cloud. Sci. Hortic-Amsterdam. 278:109791.

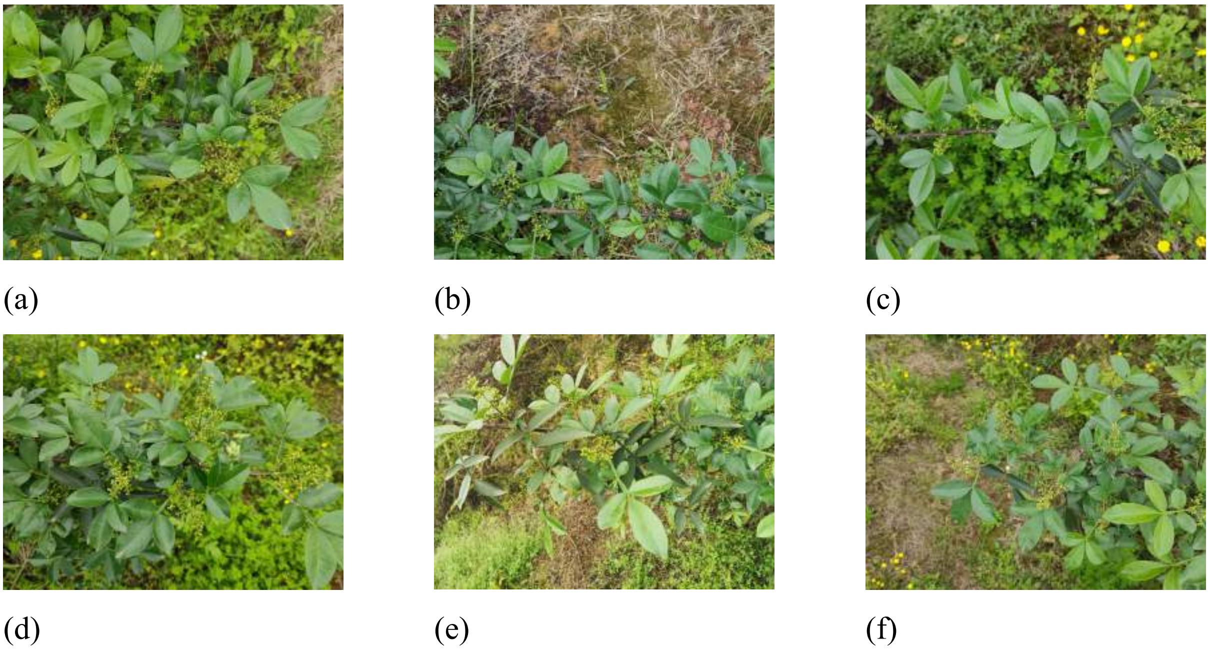
Zhang, X., Li, X., Zhang, B., Zhou, J., Tian, G., Xiong, Y., Gu, B. 2018. Automated robust crop-row detection in maize fields based on position clustering algorithm and shortest path method. Comput. Electron. Agr. 154:165–175.

Zhang, X., Toudeshki, A., Ehsani, R., Li, H., Zhang, W., Ma, R. 2022. Yield estimation of citrus fruit using rapid image processing in natural background. Smart Agricultural Technology. 2:100027.

Zhou, Y., Tang, Y., Zou, X., Wu, M., Tang, W., Meng, F., Zhang, Y., Kang, H., 2022. Adaptive Active Positioning of Camellia oleifera Fruit Picking Points: Classical Image Processing and YOLOv7 Fusion Algorithm. Appl. Sci-Basel. 12:12959.

Zhu, E., Ma, R. 2018. An effective partitional clustering algorithm based on new clustering validity index. Appl. Soft Comput. 71:608–621.

Figures



(a) (b) (c)

(d) (e) (f)

Figure 1. Images of Zanthoxylum under different conditions: (a) Normal sample; (b) Infructescences covered by leaves; (c) Overlapping infructescences; (d) Samples with complex background; (e) Samples with bright light and (f) Samples with several conditions.

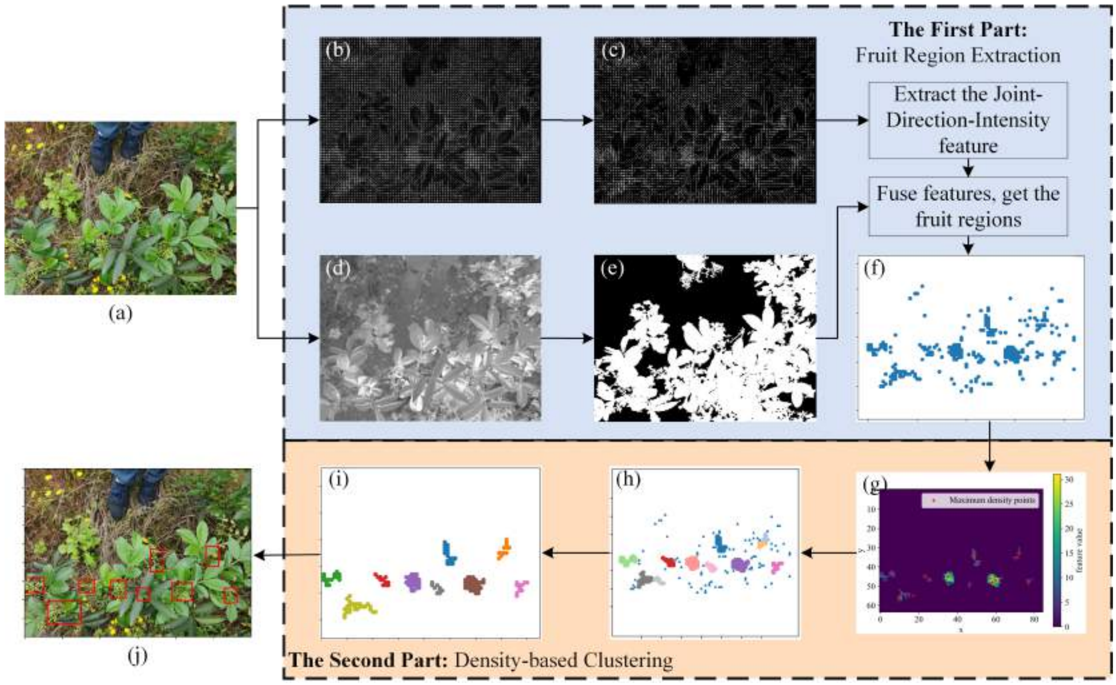
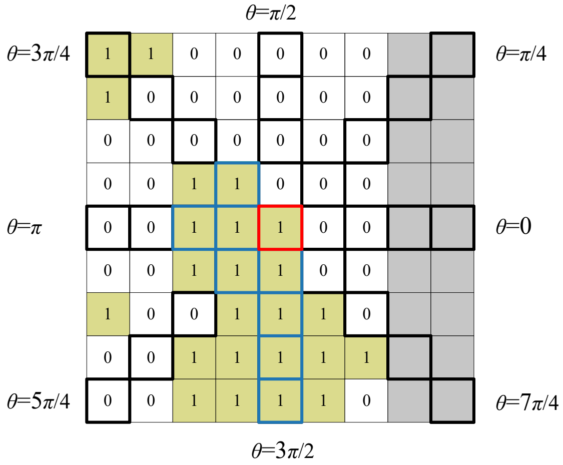

The First Part:
Fruit Region Extraction
Extract the Joint-Direction-Intensity feature
Fuse features, get the fruit regions
(a)
(b)
(c)
(d)
(e)
(f)
(g)
Maximum density points
feature value
(h)
(i)
(j)
The Second Part: Density-based Clustering

Figure 2. Overview of the proposed framework: (a) The original image; (b) The HOG feature map; (c) The binarized HOG feature map; (d) The ExGR image; (e) The segmenting result of plant & non-plant area which is named *PM*; (f) The coordinates of extracted fruits in *F'*; (g) *FDW* and density maximum points; (h) Result of density-based rough clustering; (i)The result of adaptive density clustering algorithm for fruit area distribution; (j) Result of the proposed method, and the infructescence positions are marks with boxes in red borders.



1	1	0	0	0	0	0		
1	0	0	0	0	0	0		
0	0	0	0	0	0	0		
0	0	1	1	0	0	0		
0	0	1	1	1	0	0		
0	0	1	1	1	0	0		
1	0	0	1	1	1	0		
0	0	1	1	1	1	1		
0	0	1	1	1	1	0		

Figure 3. The 8-directions of the grid with red borders: Local density of the red border grid is calculated with the grids with blue borders, in which, grids in grey are out of the map.

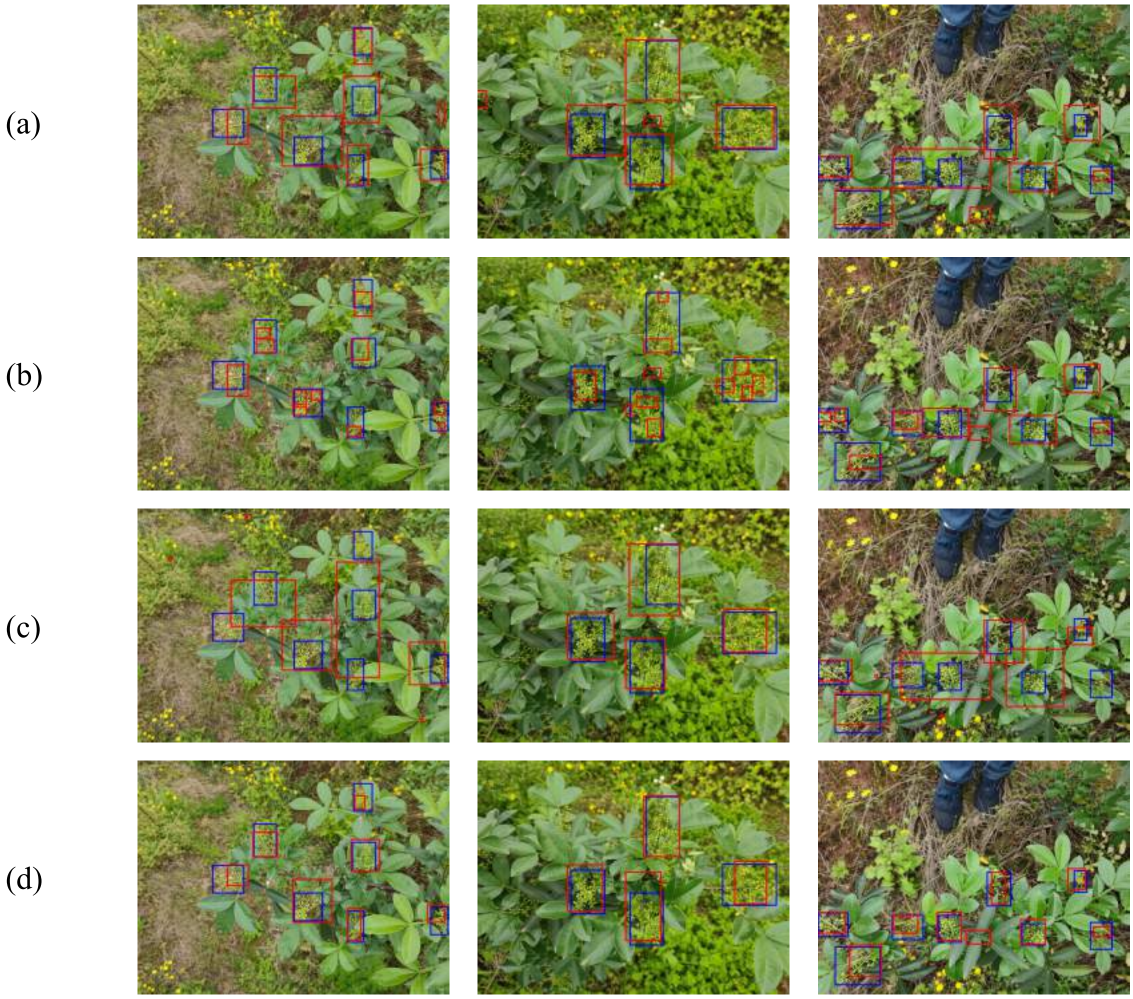


Figure 4. The results of comparison experiments with traditional methods: (a) Results of DBSCAN; (b) Results of OPTICS; (c) Results of ADPC-kNN and (d) Results of the proposed method. In the comparison experiments, the *eps* and *minPts* setting of DBSCAN and OPTICS are 3.5 and 5, respectively. ADPC-kNN introduces kNN to improve the calculation of density, where *k*=25. And the boxes with blue borders are the target boxes, and the red boxes are the results gotten by the corresponding method.

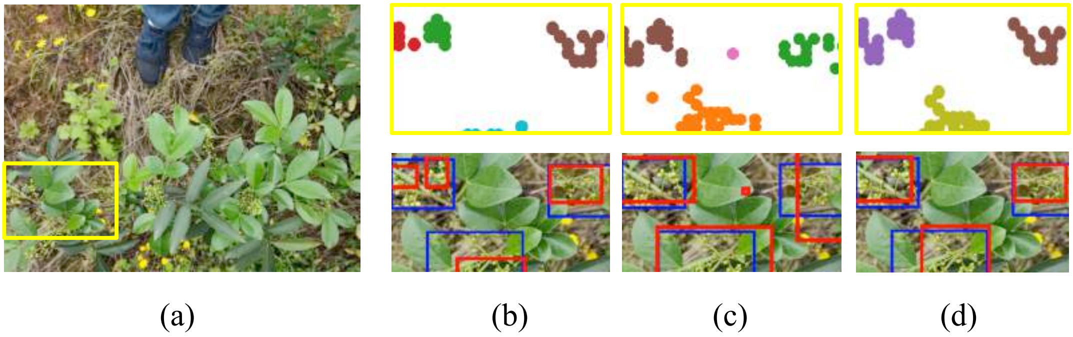


Figure 5. An example of the comparison algorithms: (a) Original image and the interest area; (b) Result of OPTICS; (c) Result of ADPC-kNN and (d) Result of Ours. The box with yellow borders shown in (a) is the area where mistakes happened. The cluster results without outliers and detection results of three algorithms are shown in (b-d). The boxes with blue borders are the ground truth, and the red boxes are the detection results of corresponding algorithm.

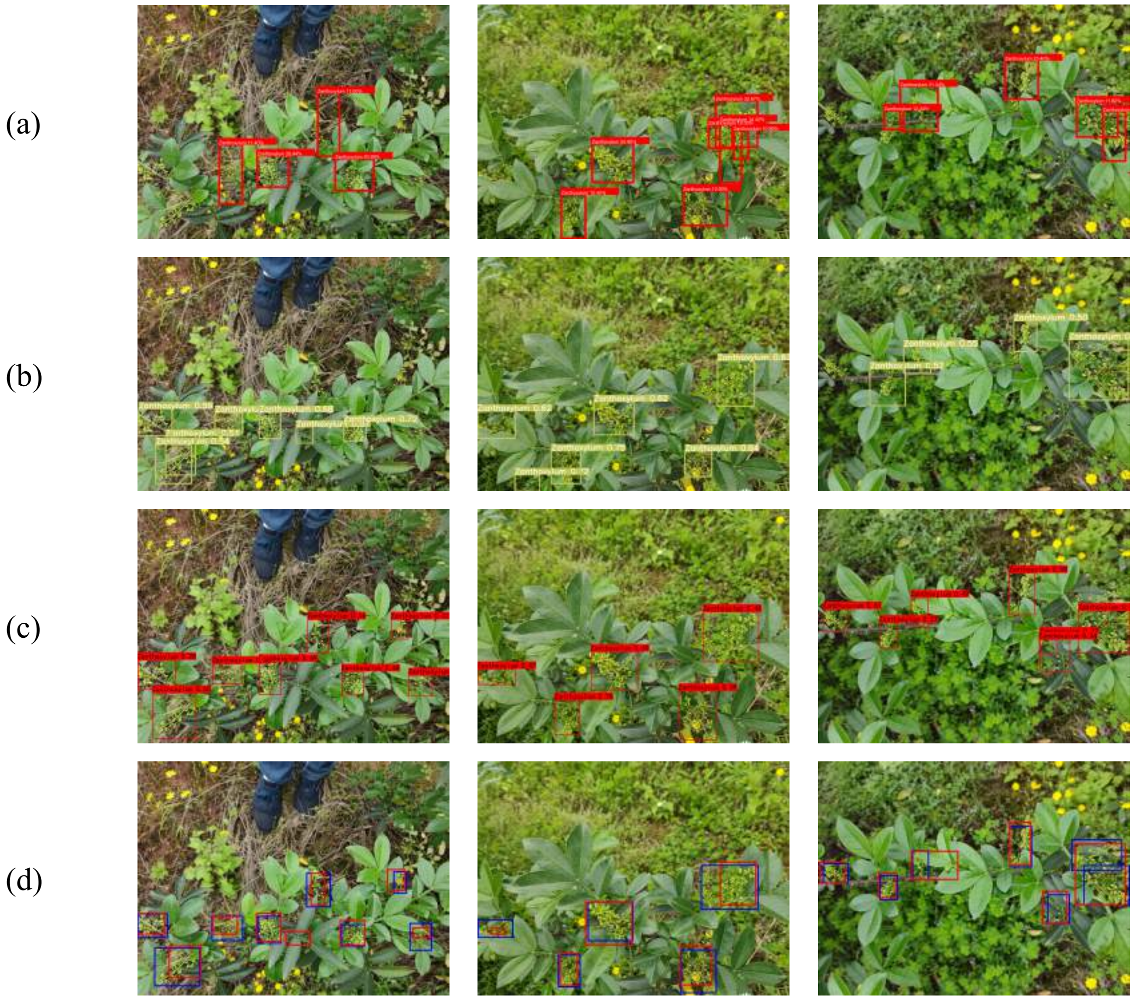


Figure 6. The results of comparison experiments with deep learning-based methods: (a) Results of Faster RCNN; (b) Results of YOLOv5; (c) Results of YOLOv7 and (d) Results of the proposed method.

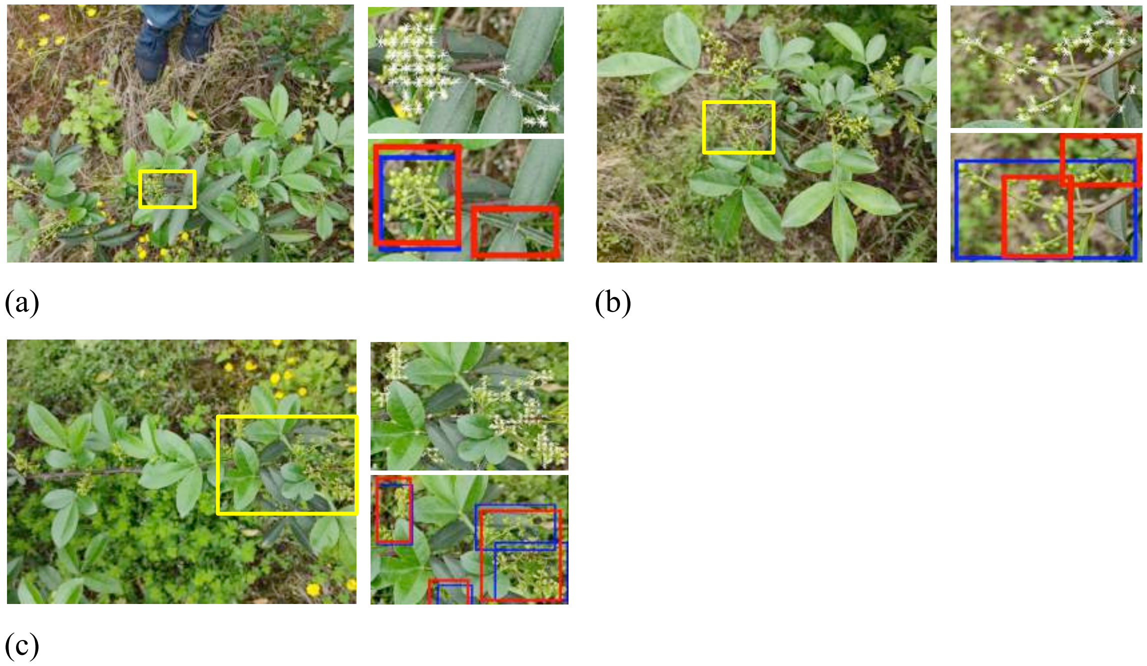


Figure 7. Some examples of failed detection: (a) An example of failed feature extraction; (b) An example of under-merging and (c) An example of overlapping clusters. In Figure 7, the extracted feature maps and detection results are enlarged in the right panel of each example which are corresponding to the location of yellow boxes marked in the left. The boxes with blue borders are the ground truth, and the red boxes are the results gotten by the proposed method.

Tables

Table 1. Distribution of the created sample set.

Conditions	**Normal**	**Robust**			
		Leaves cover the targets	Overlapping infructescences	Complex background	Bright light / Dark light
Number of samples	25	37	24	8	12
Number of infructescences	134	215	144	47	72

*Note: Since a sample in robust sample set may have multiple complex conditions at the same time, the number of robust samples is greater than 50.

Table 2. Detailed data of precision comparison experiments.

Algorithm	**Sample set**	**Average Precision [%]**	**Average Recall [%]**	**Average F1-score**
OPTICS	Normal	**98.33±4.08**	57.43±18.35	0.71±0.14
	Robust	**100.00±0.00**	62.14±18.19	0.75±0.13
ADPC-kNN	Normal	79.76±17.29	79.76±17.29	0.80±0.17
	Robust	82.34±14.34	82.34±14.34	0.82±0.14
DBSCAN	Normal	90.95±16.20	86.81±16.33	0.87±0.12
	Robust	93.93±8.47	81.50±15.29	0.87±0.11
Ours	Normal	96.67±8.16	**91.07±13.88**	**0.93±0.08**
	Robust	94.09±13.56	**84.64±14.27**	**0.88±0.10**

Table 3. Parameters required for these clustering algorithms.

Algorithm	**Parameters**	**Meaning**	**Number of Parameters**
DBSCAN	*eps*	Radius of neighborhood	2
	minPts	Threshold of density to distinguish type of points	
OPTICS	*eps*	Radius of neighborhood	2
	minPts	Threshold of density to distinguish type of points	
ADPC-kNN	*k*	The number of nearest neighbors which is introduced to calculate the density	2
	c	Number of clusters	
Ours	**-**	**-**	**0**

Table 4. Comparison of four indicators for 4 clustering algorithms.

Indicators	OPTICS	ADPC-kNN	DBSCAN	Ours
RI	0.75	0.87	0.91	**0.95**
ARI	0.32	0.60	0.70	**0.75**
NMI	0.59	0.71	0.80	**0.88**
FMI	0.49	0.69	0.76	**0.89**

Table 5. Key parameters of training neural network models.

Parameters	Faster RCNN	YOLOv5	YOLOv7
batch size	2	16	16
epoch	100	350	350
initial learning rate	0.005	0.01	0.01
optimizer	SGD	Adam	SGD
pre-trained weights	Yes	Yes	Yes

Table 6. Comparison of ours with deep learning-based methods.

Sample Set	Indicators	Faster RCNN	YOLOv5	YOLOv7	Ours
Normal	Precision [%]	72.62±21.00	**100.00**±0.00	76.25±18.81	96.67±8.16
	Recall [%]	82.38±27.73	85.71±24.74	**98.33**±4.08	91.07±13.88
	F1-score	0.76±0.21	0.91±0.16	0.85±0.10	**0.93**±0.08
Robust	Precision [%]	73.21±15.96	79.21±18.27	79.90±18.60	**94.09**±13.56
	Recall [%]	80.14±23.37	83.89±18.37	**91.57**±18.17	84.64±14.27
	F1-score	0.72±0.15	0.79±0.11	0.84±0.15	**0.88**±0.10
Average	Precision [%]	72.63±17.24	83.11±18.36	80.36±17.90	**94.17**±14.26
	Recall [%]	80.92±24.14	84.23±18.77	**92.54**±14.96	87.64±12.39
	F1-score	0.73±0.17	0.81±0.12	0.85±0.13	**0.90**±0.10