IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received September 17, 2019, accepted October 14, 2019, date of publication October 24, 2019,
date of current version November 4, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2949343

Automatic Detection of Single Ripe Tomato on Plant Combining Faster R-CNN and Intuitionistic Fuzzy Set

CHUNHUA HU[1], XUAN LIU[1], ZHOU PAN[1], AND PINGPING LI[2]
[1]School of Information Science and Technology, Nanjing Forestry University, Nanjing 210037, China
[2]School of Biology and Environment, Nanjing Forestry University, Nanjing 210037, China

Corresponding author: Pingping Li (lipingping@ujs.edu.cn)

This work was supported in part by the Jiangsu Science and Technology Support Plan under Grant BE2018321-02, and in part by the Priority Academic Program Development of Jiangsu Higher Education Institutions.

ABSTRACT Fast and accurate detection of ripe tomatoes on plant, which replaces manual labor with a robotic vision-based harvesting system, is a challenging task. Tomatoes in adjacent positions are easily mistaken as a single tomato by image recognition methods. In this study, a ripe tomato detection method that combines deep learning with edge contour detection is proposed. Our approach efficiently separates target tomatoes from overlapping tomatoes to detect individual fruits. This approach yields several improvements. First, deep learning requires less time and extracts deeper features than traditional methods for assessing candidate ripe tomato regions. Second, we use Gaussian density function of H and S in the HSV color space to help segment tomato regions from the background, followed by erosion and dilation on the tomato body to separate adjacent tomatoes and remove peripheral subpixels from all detected ripe tomatoes. Third, an adaptive threshold intuitionistic fuzzy set (IFS) method was developed to identify the tomato's edge, and it performs well in detecting blurred edges in overlapping regions. To improve the efficiency and stability of edge detection under natural conditions, we adopted an illumination adjustment algorithm for the tomato image before edge detection. As samples, we collected images showing tomatoes that were separated, adjacent, overlapped, and even shaded by leaves. The widths and heights of these tomato samples were calculated and analyzed to evaluate the detection performance of the proposed method. The root mean square error (RMSE) results for tomato width and height using the proposed method are 2.996 pixels and 3.306 pixels, respectively. The mean relative error percent (MRE%) values for horizontal and vertical center position shift are 0.261% and 1.179%, respectively. These results demonstrate that the proposed method improves tomato detection accuracy and that it can be further applied in the harvesting process of agricultural robots.

INDEX TERMS Tomato detection, deep learning, background subtraction, intuitionistic fuzzy set theory (IFS), contour segmentation.

I. INTRODUCTION

A. BACKGROUND

Tomatoes are one of the most important and popular fruit crops. Tomatoes offer humans many essential and beneficial nutrients such as antioxidants and vitamins C and A. As tomato demand increases, tomatoes are increasingly grown in greenhouses. However, manual harvesting is time consuming and costly, and as China's labor costs rise, the adoption of agricultural automation processes is inevitable. Such processes are of great significance for reducing agriculture labor costs and improving a country's industrial structure. Therefore, it is necessary to develop automatic tomato pickers. Although most agricultural robots—fruit harvesting systems in particular—use computer vision to detect fruit targets, accurate fruit detection is a challenging research topic. It is difficult to develop a vision system that functions as intelligently as a human and can easily identify fruit, especially in the presence of overlapping fruits or large leaf occlusions. The performance of the robot's

The associate editor coordinating the review of this manuscript and approving it for publication was Mohan Venkateshkumar.

visual system directly affects tomato picking and operational safety. Improving the recognition rate of the visual system can increase the locating accuracy of the robot arm. In this study, we mainly aimed to identify ripe tomatoes based on a vision system. Systems designed to count or harvest fruit require accurate detection schemes that can overcome challenges such as naturally occurring changes in illumination, shape, pose, color and viewpoint.

Many fruit detection and recognition methods based on vision systems have been proposed. These methods include color-based, shape-based, feature-fusion-based and deep-learning-based detection methods. Color is one of the most prominent features used to distinguish mature fruit from complex backgrounds. In studies that focus on color-based detection, the image pixels are clustered into two classes based on a color threshold to determine whether a pixel belongs to a fruit or to the background. But the fruit detection method based on color only will keep the background with similar color features to the fruit in the image, and it is difficult to only obtain the fruit. Commonly used methods are color difference processing, HSV, YCbCr color space transformation. Wei *et al.* [1] proposed a method using the OHTA color space that achieved a detection accuracy above 95%. Zhou *et al.* [2] used visible spectrum images and a color threshold method to detect both green and red apples. Arefi *et al.* [3] developed a ripe tomato recognition method by combining the RGB, HSI and YIQ color spaces and considering fruit morphological features; this method achieved an accuracy of up to 96.36%. Teixidó *et al.* [4] defined different linear color models in the RGB vector color space to detect red peaches in orchard images. Ostovar *et al.* [5] proposed a method based on reinforcement learning to adaptively find hue (H) and saturation (S) thresholds in images to detect yellow peppers in greenhouses. Luo *et al.* [6] developed an approach that combined the AdaBoost framework and multiple color components to identify grape clusters in a vineyard, and the approach was able to effectively extract color components from multiple color spaces.

Shape-based detection methods mainly extract the geometric features of targets, including edge contour features and features of the whole region. Nevertheless, it has the disadvantage of high time complexity. Hough transform is one of the commonly used method. Some recent papers on the fruit detection using shape-based methods are listed as follows. Xie *et al.* [7] proposed an improved randomized circular Hough transform method to rapidly and accurately calculate the center coordinates and radii of quasi-circular fruits. Liu *et al.* [8] proposed a method for constructing a multi-elliptical boundary model in Cr-Cb coordinates to detect citrus fruit and tree trunks under natural lighting conditions. Nyarko *et al.* [9] proposed a nearest neighbor approach for fruit recognition in RGB-D images based on detecting convex surfaces. The paper also proposed a novel descriptor of approximately convex surfaces, called the convex template instance (CTI), which approximated surfaces by convex polyhedrons with quantized face orientations; every polyhedron face corresponded to one descriptor component.

It is hard to recognize specific fruits or locate them accurately based solely on color or shape features. Hence, multi-feature information fusion was adopted by researchers, which used both the color feature and the shape feature to improve the recognition rate of fruits. Wu *et al.* [10] developed an improved method that combined multiple features, feature analysis and selection, a weighted relevance vector machine (RVM) classifier, and a bilayer classification strategy to recognize ripening tomatoes. Yamamoto *et al.* [11] proposed an image processing method for accurately detecting individual intact tomatoes on plants, including mature, immature and young fruits, using a conventional RGB digital camera in conjunction with machine learning approaches. The detection method was based on classification models generated in accordance with the colors, shapes, textures and sizes of the images. Fernández *et al.* [12] proposed a unique, modular and easily adaptable multisensory system and a set of associated preprocessing algorithms for detecting and locating fruits from different crop types in natural scenarios. Gan *et al.* [13] combined color and thermal images to detect immature green fruits. Monta and Namba [14] used a cascading technique to detect tomatoes; depth data were employed to distinguish individual fruits that are part of a single color segment. Fruit candidate regions were generated by setting thresholds for the color channels, and single fruits were separated by examining and setting a threshold for the spatial distance between adjacent pixels in the candidate regions. Patel *et al.* [15] developed an algorithm for fruit detection based on multiple features; different weights were assigned to different image features such as intensity, color, orientation and edge. Li *et al.* [16] developed a method for detecting and counting immature green citrus fruits using outdoor color images as part of the development of an early yield mapping system; multiple features, including color, shape and texture, were combined to remove false positives. Seng and Mirisaee [17] proposed a recognition approach that combined color-based, shape-based and size-based methods and used a nearest neighbor model to classify fruit pixels. The class 'tomato' may include several intensity subclasses, such as ripe and unripe tomatoes. Senthilnath *et al.* [18] proposed a method for detecting tomatoes using spectral-spatial methods in remotely sensed RGB images captured by a UAV that used the Bayesian information criterion (BIC) to determine the optimal number of clusters for the image. Spectral clustering was conducted using K-means, expectation maximization (EM) and self-organizing map (SOM) algorithms to categorize the pixels into two groups i.e., tomatoes and non-tomatoes. Barnea *et al.* [19] presented a color-agnostic, shape-based, 3D fruit-detection method for crop harvesting robots; they proposed exploiting both RGB and range data to analyze the shape-related features of objects in both the image plane and 3D space.

Feature extraction and selection are major challenges to improve the recognition accuracy of any computer-based

application. In recent years, deep learning approaches have been widely used in image detection and classification. Sa *et al.* [20] presented an approach using a deep convolutional neural network (DCNN) that combined multi-modal (RGB and near-infrared, NIR) features to improve single-feature DCNNs. Lin *et al.* [21] introduced a guava detection and pose estimation method based on low-cost RGB-D sensors in the field by utilizing a state-of-the-art fully convolutional network to segment the RGB image and output a fruit and branch binary map. Stein *et al.* [22] proposed a novel multisensor framework to identify, track, localize and map every piece of fruit by combining a faster region-based convolutional neural network (Faster R-CNN) with LiDAR data. Region detection with CNN features [23] is a widely used method for detecting objects from images. The R-CNN wraps all the proposed object pixels in a tight bounding box, which can change the proposal information. Therefore, the Fast R-CNN model [24] was proposed to detect objects by adding a spatial pyramid pooling module [25]. Later, to improve the region proposal selection time, Ren *et al.* [26] proposed the Faster R-CNN model that detected objects by adding a region proposal network. There are other ways to detect fruit; for example, Korostynska *et al.* [27] used microwave spectroscopy based on a planar electromagnetic wave sensor to assess strawberry ripeness in real time.

Although all the above studies, which utilize color or shape features and machine learning, have made some progress toward automatic fruit detection and localization, several issues remain in the ripe tomato detection problem: (1) some color-feature-based methods cannot recognize single fruits; (2) some shape-feature-based methods cannot quickly locate a single fruit; (3) some methods cannot fully handle fruit overlap and leaf occlusion; and (4) classical edge operators do not work well in detecting the edges of overlapping tomatoes.

B. OBJECTIVES

The primary goal of this paper is to assess the feasibility of combining deep learning with edge segmentation to detect individual tomatoes in complex environments. To achieve this goal, several sub-goals must be met, including (1) extraction of ripe tomato features using Faster R-CNN to recognize and locate candidate ripe tomato regions in complex greenhouse environments; (2) utilization of the Gaussian density function of hue (H) and saturation (S) in the HSV color space to remove background content from the candidate regions; (3) separation of target tomatoes from adjacent tomatoes and removal of sparsely connected pixels based on morphological processing; (4) performance of illumination compensation on the tomato image using an illumination adjustment algorithm; (5) detection of a single tomato in an image block using IFS; and (6) calculation of tomato parameters such as width, height and center.

II. MATERIALS AND METHODS

A. MATERIALS

Tomato images were captured from plants in a greenhouse situated at the Jiangsu Academy of Agricultural and Sciences, Nanjing, China, between 9:00–11:00 a.m. and 4:00–6:00 p.m., using a consumer-level regular digital camera (USB) and a Kinect v1.0 device. During the experiment, 800 photos of tomato plants were taken under complex backgrounds; these photos included green and red fruits in the same bunch and under uneven lighting, as shown in Fig. 1. The acquired images were saved at a resolution of 640×480 pixels in JPG format. The study was executed on a computer equipped with a 12 core CPU running @ 3.70 Hz with 32 GB of RAM, a Nvidia GTX1080Ti GPU processor and a Windows 10 operating system. The programming environment was Visual Studio 2015. In this paper, GTX1080Ti was used to perform GPU-accelerated calculations.

FIGURE 1. Collection scene.

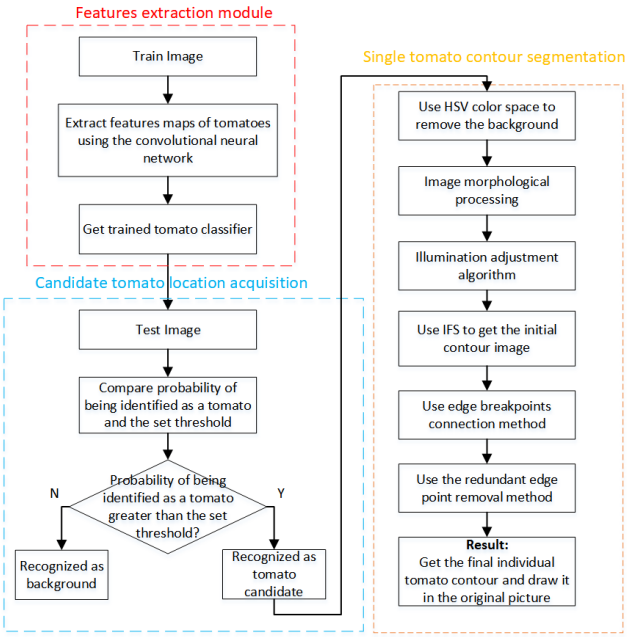


FIGURE 2. Flowchart of the tomato recognition and boundary segmentation algorithm.

B. METHOD FRAMEWORK

In this section, we discuss in detail the method used in this study to segment the contour of a single ripe tomato. The flowchart of the recognition method for clustered tomatoes is shown in Fig. 2.

The Kinect v1.0 sensor was used to acquire tomato images in a greenhouse. Subsequently, a large number of ripe tomato

images, including separated, adjacent, overlapping and leaf-shaded images, were manually labeled and used in training the Faster R-CNN detector. In a previous fruit segmentation method, Wei *et al.* [1] used a combined OHTA color space and Otsu threshold algorithm to segment mature fruit with high accuracy; however, that approach cannot handle occluded fruit. Wu *et al.* [10] developed a multi-feature fusion method that included the iterative RELIEF (I-RELIEF) algorithm, a weighted relevance vector machine (RVM) classifier, and a two-layer classification strategy to recognize ripening tomatoes. Sa *et al.* [20] adopted a state-of-the-art object detector termed the Faster Region-based CNN (Faster R-CNN) model and combined color (RGB) with near-infrared (NIR) information, which led to a novel multi-model Faster R-CNN and achieved good results in fruit detection. Compared with these methods, deep learning can overcome feature extraction difficulties and identify overlapping tomatoes. However, in some instances, deep learning misidentifies overlapping fruits as a single whole fruit. Therefore, morphological processing, contour segmentation, and redundant edge removal were conducted subsequently to separate overlapping ripe tomatoes.

C. RIPE TOMATO RECOGNITION BASED ON FASTER R-CNN

A CNN recognizes objects by performing convolutions, pooling, rectified linear unit (ReLU) application and other operations on an entire image. In 2012, Krizhevsky *et al.* [28] captured widespread attention for deep learning when they obtained superior recognition results on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) by increasing the depth of the CNN model and adopting ReLU and dropout [29] technology. The CNN performs an exhaustive selection process on the image to find all possible bounding boxes for all objects. The features of these regions are extracted first; then, an image recognition method is used to classify them. During the recognition process, non-maximum suppression (NMS) is used to obtain the candidate boxes with the highest probability of classification. The softmax function is used to form the final output before the last fully connected layer. The softmax activation function is expressed as follows:

$$y_c = \frac{e^{x_c}}{\sum_{c=1}^{C} e^{x_c}} \tag{1}$$

where x_c denotes the input imported from class c in the last output layer of the fully connected layer, and y_c denotes the output of the softmax activation function for class c. The total number of classes is represented by C.

Compared with a CNN, an R-CNN [23] scales the regions to a uniform size and then uses a CNN to extract features from every detected region. Selective search [30] is used to detect the candidate boxes with the highest classification probability, which saves time. Finally, the R-CNN features are sent to an SVM classifier for each class to predict whether they belong to that class.

Because an R-CNN conducts a forward transmission of the convolution calculations for each candidate box, it reduces the calculation time. He *et al.* [25] proposed the spatial pyramid pooling net (SPPnet) model, which involved a shared computing approach that calculated a convolutional feature graph of the entire image and then used the extracted feature vectors to classify each region's proposal box. Each training region of interest (RoI) is labeled with a ground-truth class u and a ground-truth bounding-box regression target v. Fast R-CNN introduces multi-task loss, L, to jointly train for classification and bounding-box regression as follows [24]:

$$L(p, u, t^u, v) = L_{cls}(p, u) + \lambda[u \geq 1]L_{loc}(t^u, v) \tag{2}$$

where $L_{cls}(p, u) = -\log p_u$ is the log loss for the true class L_{loc}, denotes the bounding-box regression loss, and λ represents a loss-balancing parameter. Here, (v_x, v_v, v_w, v_h) and $t^u = (t^u_x, t^u_v, t^u_w, t^u_h)$. $[u \geq 1]$ equals 1 when $u \geq 1$ and 0 otherwise.

Although Fast R-CNN achieves better results than R-CNN does, it expends a considerable amount of time on its selective search of the identified bounding boxes. In Faster R-CNN [26], the region proposal network (RPN) replaces the selective search for extracting the proposed regions. This replacement greatly improves the model's speed and the accuracy of the results, largely because selective search uses the CPU for calculations while RPN uses a GPU.

The RoI pooling layer is also known as a downsampling layer, and it is located after a convolutional layer. After processing by the RoI pooling layer, the size of the feature map extracted from the convolutional layer is reduced; it retains the effective information while reducing the amount of data to be processed and preventing overfitting. For example, after an image with 32×32 pixels is sent to an RoI pooling layer with 2×2 pixels and a stride of 2, the size of the output feature map will be 16×16 pixels. RoI pooling layers are generally divided into two types: average pooling and max pooling. Average pooling sums the elements in the pooling layer area and then divides the result by the area of the pooling layer. Max pooling, which is more common, replaces the current area with the maximum element value in the pooling layer area. The convolutional network structure for ripe tomato image detection used in this paper is depicted in Fig. 3.

The network structure used in this experiment has two convolutional layers, one max pooling layer and two fully connected layers. We adopted the ReLU activation function. The training architecture for ripe tomato detection based on Faster R-CNN is illustrated in Fig. 4.

First, images in different states of the environment were chosen and all the tomatoes including those far from the camera were labeled in the images to ensure the robustness of the dataset. Of the labeled tomatoes, some are separated (separated tomato, Fig. 5a), some are adjacent (adjacent tomato, Fig. 5b), some are overlapping (overlapping tomato, Fig. 5c), and some are shaded by leaves or branches (shaded tomato, Fig. 5d). These conditions increase the robustness of the

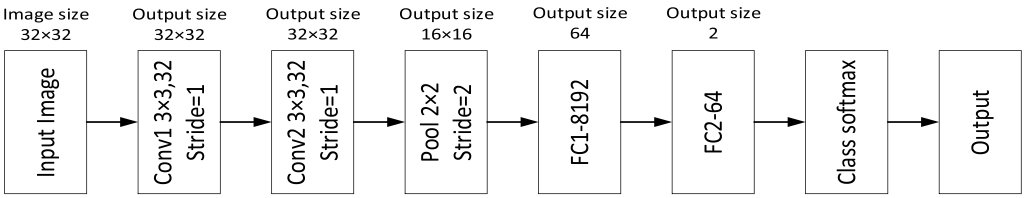


FIGURE 3. The convolutional network structure for ripe tomato image detection (Conv x represents the xth convolutional layer; Pool represents the maxpooling layer; FC x represents the xth fully connected layer).

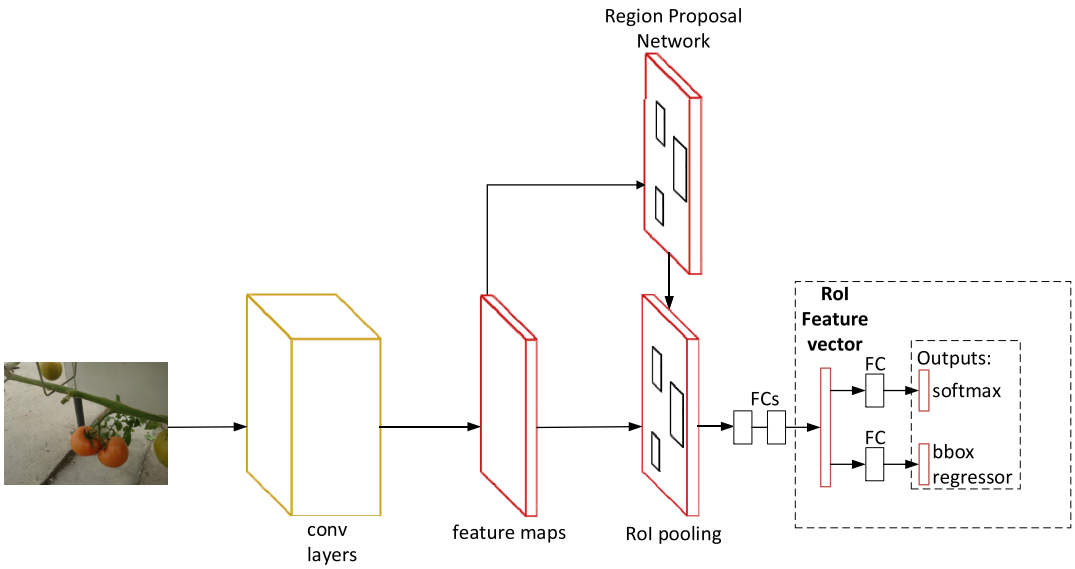


FIGURE 4. The training architecture for ripe tomato detection based on faster R-CNN.

FIGURE 5. Four state images from the greenhouse dataset. (a) Separated. (b) Adjacent. (c) Overlapping. (d) Shaded.

training set to meet the needs of practical applications in tomato detection in complex greenhouse environments.

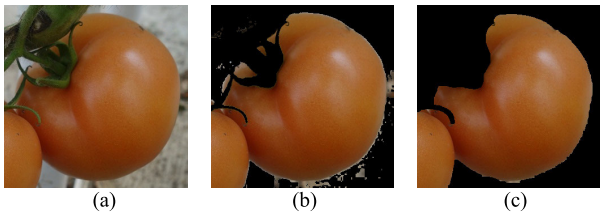
Second, we used Faster R-CNN to train the labeled images using the network structure shown in Fig. 3. We fine-tuned our network and then tested the effect of the trained detector on the validation dataset. For example, an original image is shown in Fig. 6a. Faster R-CNN was used to recognize and locate tomatoes, as illustrated in Fig. 6b. Because background content such as leaves, stems and other tomatoes occurs near the target tomatoes, the tomatoes cannot be detected accurately. To accurately acquire a single tomato, further processing for the tomatoes in the bounding boxes created based on Faster R-CNN recognition and location is necessary.

D. DETECTING A SINGLE RIPE TOMATO BASED ON CONTOUR

To accurately and quickly find ripe tomatoes in the bounding boxes obtained from the Faster R-CNN, we collected a large number of ripe tomato images from different directions and at different times. After a large number of experimental analyses, the H and S color space values can be used to quickly distinguish a sample from the background. Therefore, in this study, we converted the RGB images to HSV color space and established the Gaussian density function of H and S to further segment tomatoes from backgrounds at a threshold of 0.85. The Gaussian density function of H and S is defined

(a) (b)

FIGURE 6. The detection results using Faster R-CNN. (a) Original image. (b) Candidate tomato regions.



(a) (b) (c)

FIGURE 7. The processing result using Gaussian density function and morphology. (a) The candidate tomato region image using faster R-CNN. (b) Image processed using the Gaussian density function of H and S. (c) Morphological processing result.

as follows:

$$F(X) = (2\pi \sum)^{-\frac{1}{2}} \exp\left(-\frac{1}{2}(X-\mu)^T \sum^{-1}(X-\mu)\right) \quad (3)$$

where $X = (H, S)$ denotes the hue and saturation of tomatoes, $\sum$ is the variance of X, and $\mu = (\mu_H, \mu_S)$ is the average of H and S.

If a pixel point function value in the image is greater than the threshold, it belongs to a ripe tomato; otherwise, it belongs to the background. Taking the right-hand tomato block shown in Fig. 6b as an example for further segmentation, the ripe tomato was extracted from the background using the Gaussian density function of H and S in the HSV color space. Taking the tomato in the bounding box on the right in Fig. 6b as an example, the process of obtaining the edge of the tomato is shown in Fig. 7. The original image is shown in Fig. 7a. Fig. 7b illustrates the result obtained using the Gaussian density function of H and S, which leaves some subpixels at the edges of the target tomato. Hence, the further segmentation for ripe tomatoes from backgrounds is necessary.

A few connected pixels also exist that are outside the body of the tomato in Fig. 7b. To remove as many of these as possible while retaining the target image. We can use the open function to separate overlapping tomatoes by interrupting the connection area in the overlapping area. This method is commonly used to separate objects with subtle connections in an image and eliminate noise. The open function is defined as follows:

$$A \circ B = (A \ominus B) \oplus B \quad (4)$$

where $\ominus$ represents erosion and $\oplus$ is dilation.

After morphological processing on the tomato image, most of the background pixels are removed, as shown in Fig. 7c, making contour extraction of the target tomato more precise and simpler.

However, in some instances, the area of overlap between tomatoes (as shown in Fig. 7c) is too large, and using the open function to remove the intersecting parts will lead to a considerable loss of tomato contour information. In these instances, to detect a single tomato, it is important to fully utilize the edge features of the overlapping area. Traditional edge detection methods such as Canny, Prewitt and Sobel are based on derivative filtering and are affected when the edges are blurred, noisy and inflexible [31]. Previous studies [32], [33] developed edge feature descriptors for detecting edges, but these methods have difficulty identifying edges in fuzzy states, and the edge detected is not continuous.

Since Atanassov [34] introduced the concept of an intuitionistic fuzzy set (IFS), IFS has attracted the attention of scholars all over the world. IFS add an attribute called the non-membership degree, which describes a neutral state and considers uncertainties. Recently, IFS theory was used to improve detection accuracy. Chaira [35] detected edges in medical images using the IFS technique. Melo-Pinto *et al.* [36] used Atanassov's intuitionistic index values to represent hesitance in image segmentation. Our previous work [31] used IFS theory with adaptive thresholding to segment hardwood seedling leaves. To obtain the complete contour of the whole tomato, in this paper, we adopt the IFS method for the extraction of the fuzzy contour in the overlapped area.

Because of non-uniform illumination, the edge segmentation algorithm can easily identify locations with large differences in illumination on the tomato surface as edges. Hence, before conducting IFS on an image, we conduct illumination compensation on the tomato image. There are many solutions to the problem of non-uniform illumination (e.g., Retinex algorithm, histogram equalization algorithm, gamma correction). In this paper, a novel Retinex-based light-processing algorithm is used for image illumination adjustment [37] because this method works best with our tomato dataset. Following this step, we transformed the image without background to grayscale and used IFS to perform edge segmentation on the grayscale image.

In [31], we used IFS theory to handle fuzzy boundaries and compared the results to those achieved with the classical method, which proved that the IFS method works better when the edges of objects are more complex. In the present paper, a set of 4 fuzzy templates of size 3×3 representing different types of edge profiles were used, as shown in Fig. 8.

In [31], the Otsu algorithm [38] was employed to adaptively select threshold T, which was used to compute an edge image of the IFS clustering image. The Otsu algorithm is a discriminant criterion used to select an optimal threshold, and it can be formulated as follows:

$$f(t) = \omega_0\omega_1(\mu_0 - \mu_1)^2 \quad (5)$$

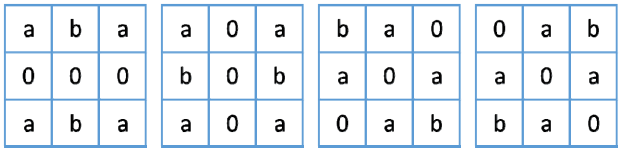


FIGURE 8. Set of 4 3×3 templates. (a) Template of x direction. (b) Template of y direction. (c) Template of 45° direction. (d) Template of 135° direction.

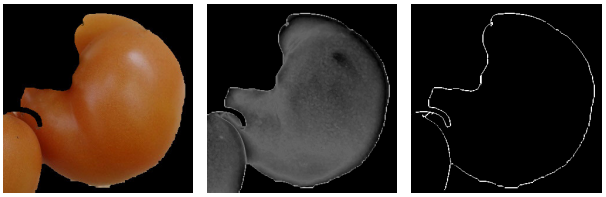


FIGURE 9. Edge detection using IFS. (a) Illumination compensation result. (b) IFS clustering image. (c) Edge image of IFS clustering image.

The class probability is computed from the histogram as follows:

$$p_i = \frac{n_i}{N}, \quad p_i \geq 0, \quad \sum_{i=1}^{L} p_i = 1 \tag{6}$$

where L is the maximum luminance value in the image and n_i is the number of pixels at level i.

The foreground and background probabilities of occurrence are given by

$$\omega_0 = \sum_{i=1}^{k} p_i, \quad \omega_1 = \sum_{i=k+1}^{L} p_i \tag{7}$$

The class mean levels are denoted as follows:

$$\mu_0 = \sum_{i=0}^{k-1} i * p_i/\omega_0, \quad \mu_1 = \sum_{i=k}^{L} i * p_i/\omega_0 \tag{8}$$

The best threshold is computed by

$$T = \arg\max_t f(t) \tag{9}$$

However, our previous work [31] selected the thresholds of the fuzzy template manually. To find better edge detector mask thresholds, according to the automatic threshold selection method in [5], we used reinforcement learning to select fuzzy template parameters a and b in this study. Fig. 9a illustrates the result of morphological processing. Fig. 9b is the IFS clustering image. Fig. 9c is the edge image of the IFS clustering image.

In a natural environment, tomatoes are distributed in various positions. For adjacent tomatoes, as shown in Fig. 9c, it is highly probable that the contour calculated by the edge detector from an image will be connected with the contour of an adjacent tomato. Therefore, a tomato edge contour detection method that connects edge breakpoints and removes redundant edge points was developed to extract the contour

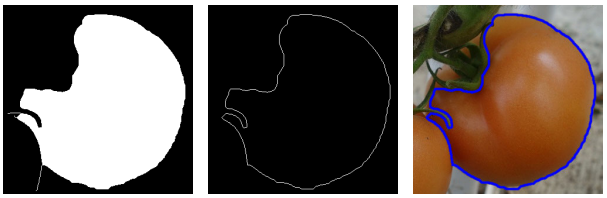


FIGURE 10. Target tomato detection using the edge contour detection method. (a) Contour image after filling in the main body of the tomato. (b) Contour image after removing pixels belonging to the contour of the adjacent tomato. (c) Target tomato obtained after marking the original image with a blue line.

of a single ripe tomato. Broken contours are completed using the edge breakpoint connection method. The edge breakpoint connection method for the tomato is a two-step process based on the breakpoint positions. In the first step, the edge of the tomato is truncated by the Faster R-CNN bounding box, and the truncated edge breakpoint connection method is used. If there are two breakpoints on the same block edge, that edge is considered to be truncated by the Faster R-CNN bounding box. To complete the contour, we connected the two breakpoints on the truncated edge. In the second step, the edge contour obtained using the edge method has a missing part. To connect the breakpoints on the nonborder edges, we used the cubic polynomial function to fit the curve of the missing contour from several points near the two matching breakpoints. In a natural environment, tomatoes are distributed in various ways. For adjacent tomatoes, it is highly probable that the contour calculated by the edge detected from an image will include the contour of an adjacent tomato. To remove redundant contours, a redundant edge point removal method is required. First, the closed area of the main body of the tomato is filled. Then, a threshold method is employed to determine whether a point is a target point. For each point, we define the number of its 8 neighborhood pixels that have of a value of 1 as k. If k is less than a set threshold value T_1, the point belongs to the adjacent tomato. In this study, the threshold T_1 was set to 3.

As shown in Fig. 9c, the contours of the adjacent tomatoes in the target tomato contour are not removed. To remove these redundant contours, the redundant edge point removal method is required. First, the closed area of the main body of the tomato is filled, as shown in Fig. 10a. Then, a threshold method is employed to determine whether a point is a target point. The removal result is shown in Fig. 10b, and the boundary of the target tomato is presented with a blue line in Fig. 10c.

III. RESULTS

In this study, we conducted multiple steps to extract the final contour of a single tomato. The tomato images were obtained using a consumer-level regular digital camera (USB) and a Kinect v1.0 device at the Jiangsu Academy of Agricultural and Sciences, Nanjing, China, from 9:00–11:00 a.m. and from 4:00–6:00 p.m. We acquired 800 sample images under

different environmental states, including separated, adjacent, overlapping and shaded tomatoes. Of these images, we used 600 images for training and 200 images for testing. We conducted numerous experiments to evaluate the accuracy and practicability of the proposed method. In Section III.A, the accuracy and regression rate of Faster R-CNN for ripe tomato detection were evaluated. In Section III.B, the contour segmentation processes of tomatoes in four different states were described to show the applicability of our proposed method for tomato segmentation. In Section III.C, we report the results of comparison experiments conducted to analyze the error parameters for a single tomato.

A. RECALL OF FASTER R-CNN

The area under the P-R curve, known as the average precision (AP), can be used as a single metric to summarize the performance of an object detection model; P represents precision and R represents recall. The performance metrics selected for validation purposes in this study are as follows.

True positive (TP) is the number of ripe tomatoes that were correctly identified as ripe tomatoes.

False positive (FP) is the number of background areas that were misidentified as ripe tomatoes.

False negative (FN) is the number of ripe tomatoes that were misidentified as background areas.

True negative (TN) is the number of background areas that were correctly identified as backgrounds.

The precision of ripe tomato detection is a measure of accuracy defined as

$$\text{Precision} = \frac{TP}{TP + FP} \times 100\% \tag{10}$$

The recall of ripe tomato detection is defined as

$$\text{Recall} = \frac{TP}{TP + FN} \times 100\% \tag{11}$$

A model that achieves high precision at all levels of recall will have a high AP score, while a model that achieves high precision at only some recall levels will have a low AP score. Both recall and precision are important for characterizing the performance of a detector. Accuracy decreases as recall increases, but the precision of a detector with good performance remains high as recall increases, which means the detector will detect a high proportion of TP before it starts detecting FP. The trajectory of the precision-recall curve of the network is shown in Fig. 11.

Clearly, the precision is high at the beginning, but as the recall increases, the precision decreases rapidly. One reason for this phenomenon is that our tomato images are complicated. Another reason is that some of the training samples include small, fuzzy tomatoes that can easily be confused with the background and are difficult to detect. The study in [39] obtained an AP of 0.948. In that study, their images were obtained by lighting a dark environment; thus, their images contain only nearby, well-illuminated apples, and blurred objects in the background are not captured.

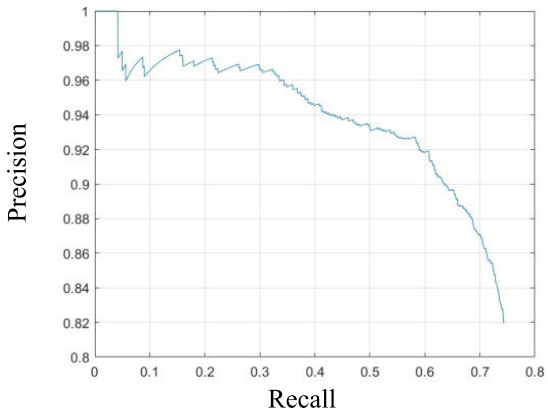


FIGURE 11. Precision-Recall curves for the faster R-CNN detector used to detect ripe tomatoes in the greenhouse.

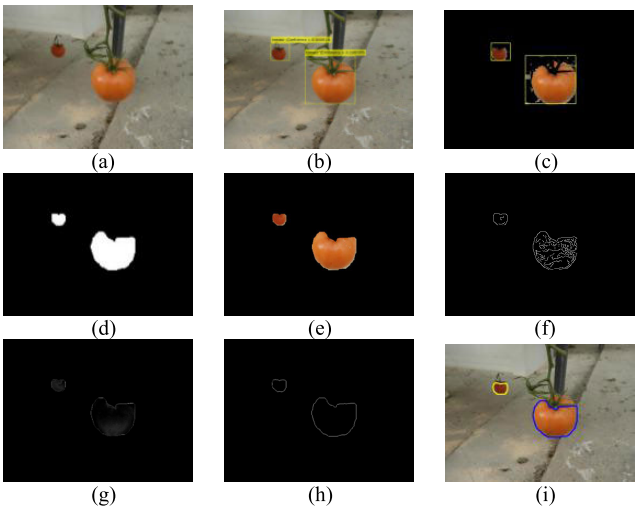


FIGURE 12. Segmentation process of separated tomatoes. (a) Original image of separated tomatoes. (b) Result after location using Faster R-CNN. (c) Result after using the Gaussian density function of H and S. (d) Result after morphological processing. (e) Resulting edge image after using the illumination adjustment algorithm. (f) Resulting edge image after using the Canny operator. (g) Result after using IFS for the image after illumination adjustment. (h) Adaptive threshold edge image of Fig. 12g. (i) Result obtained after marking the original image.

In contrast, our images were taken in a complex environment, which has a strong impact on recognition rates.

B. RESULTS OF SINGLE TOMATO DETECTION

In ripe tomato detection, the detection results of ripe tomatoes vary by tomato state. To conduct a comprehensive assessment of the performance of the proposed method, we implemented four experiments for different tomato states; the results are analyzed in this section.

Fig. 12 shows the details of the ripe tomato detection process under the separated condition. Fig. 12a is the original image, where the two tomatoes in the figure are separated from each other. Faster R-CNN was used to locate these tomatoes, and the results are shown in Fig. 12b. The two tomato blocks are separated, and the background around

them is not complicated. To accurately localize a single tomato, the tomato contour extraction method described in Section II.D was performed. First, the Gaussian density function of H and S was used to segment the candidate tomato region obtained by Faster R-CNN from the background, as shown in Fig. 12c. Because some subpixels still remain around the processed image, we conducted morphological processing to precisely extract the target region for each candidate region. The results are shown in Fig. 12d. However, the stem of the tomato on the right side separates the top part from the whole fruit, which is why, after morphological processing, the tomato lacks a small top portion. After morphological processing, we applied the illumination adjustment algorithm to the tomato image, as shown in Fig. 12e. Next, we used the Canny operator and IFS theory to detect the tomato edge. Fig. 12f shows the edges of the tomato detected using the Canny operator. The clustering image using IFS is shown in Fig. 12g. The edge image using the adaptive threshold is shown in Fig. 12h with a threshold of 0.1647. Parameters a and b for edge detection masks were 0.3 and 0.8, respectively. Tomatoes in a separated state are the easiest to segment in these states because there is no interference from neighboring tomatoes during the segmentation process. Compared with the edge detection result using Canny operator, which is shown in Fig 12f, the edge detection result using IFS in Fig. 12h can obtained better edges located at the boundary of the bounding box. That was why the tomato contour in Fig. 12h was further connected with the contour detection method involving edge breakpoint connection and redundant edge point removal. Finally, we marked the results of this process with different colors in the original image, as presented in Fig. 12i. The tomatoes are delineated by the yellow and blue lines.

Fig. 13 shows the experimental results for adjacent tomatoes. In the original tomato image in Fig. 13a, two tomatoes are adjacent. Faster R-CNN obtained the locations of the two tomatoes, as shown in Fig. 13b. In this image, the positions of the two tomatoes cause the two bounding boxes to overlap, which interferes with subsequent contour segmentation. Fig. 13e shows the result of the illumination adjustment algorithm on the tomato. Fig. 13f illustrates the results of segmentation using the Canny operator to make a comparison with the IFS detection method. In this figure, the Canny operator cannot detect continuous edge in the fuzzy area where two tomatoes overlap. The contours of the blurred area of the two tomatoes are disconnected from their main contours. Fig. 13g and Fig. 13h illustrate the clustering image using IFS and the edge image using adaptive threshold, respectively; $a = 0.3$, and $b = 0.8$. The threshold for Fig. 13h is 0.1137. As shown in Fig. 13h, IFS can detect the boundary of a continuous fuzzy region, which facilitates the extraction of individual tomatoes. So, we further processed the contour in Fig. 13h using the contour detection method involving edge breakpoint connection and redundant edge point removal and marked the results of this process with different colors in the original image, as presented in Fig. 13i. The tomatoes are

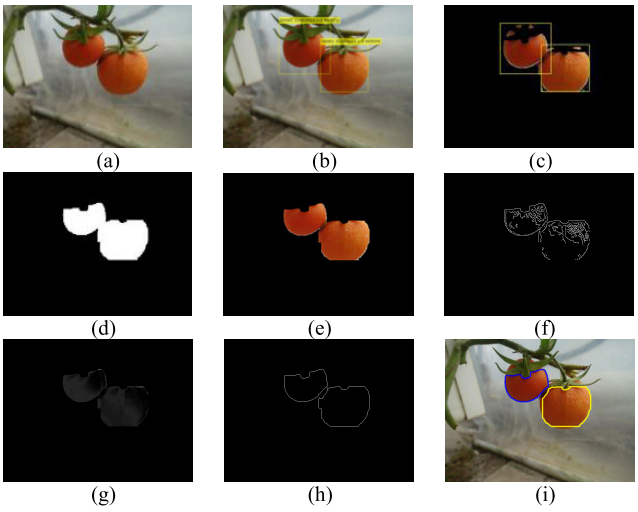


FIGURE 13. Segmentation process of adjacent tomatoes. (a) Original image of adjacent tomatoes. (b) Result after location using Faster R-CNN. (c) Result after using the Gaussian density function of H and S. (d) Result after morphological processing. (e) Resulting edge image after using the illumination adjustment algorithm. (f) Resulting edge image after using the Canny operator. (g) Result after using IFS for the image after illumination adjustment. (h) Adaptive threshold edge image of Fig. 13g. (i) Result obtained after marking the original image.

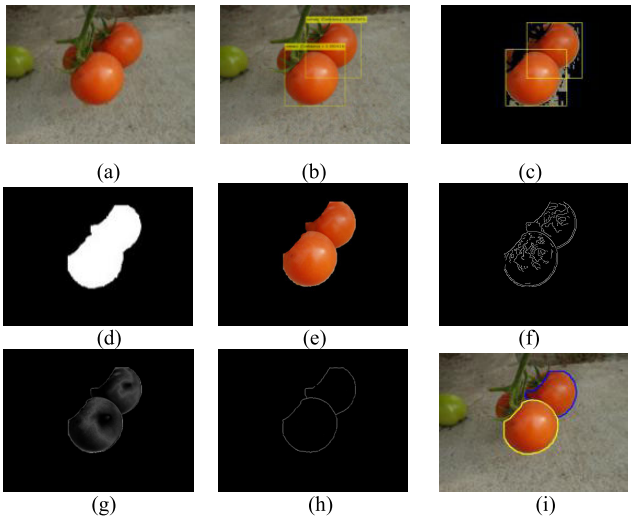


FIGURE 14. Segmentation process of overlapping tomatoes. (a) Original image of overlapping tomatoes. (b) Result after location using Faster R-CNN. (c) Result after using the Gaussian density function of H and S. (d) Result after morphological processing. (e) Resulting edge image after using the illumination adjustment algorithm. (f) Resulting edge image after using the Canny operator. (g) Result after using IFS for the image after illumination adjustment. (h) Adaptive threshold edge image of Fig. 14g. (i) Result obtained after marking the original image.

delineated by the yellow and blue lines. The results validate that the IFS detection method performs better than the Canny operator in detecting blurred edges and that it can accurately detect a single tomato in the adjacent state.

The process used to segment overlapping tomatoes is shown in Fig. 14. In Fig. 14a, three tomatoes overlap. Tomato images in this state are more difficult to segment than adjacent tomato images. Two of the candidate tomato regions detected using Faster R-CNN have a large overlap: the result

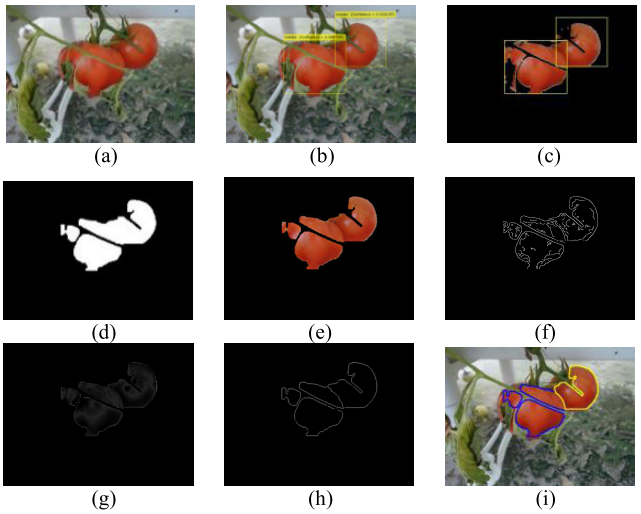


FIGURE 15. Segmentation process of shaded tomatoes. (a) Original image of shaded tomatoes. (b) Result after location using Faster R-CNN. (c) Result after using the Gaussian density function of H and S. (d) Result after morphological processing. (e) Resulting edge image after using the illumination adjustment algorithm. (f) Resulting edge image after using the Canny operator. (g) Result after using IFS for the image after illumination adjustment. (h) Adaptive threshold edge image of Fig. 15g. (i) Result obtained after marking the original image.

is shown in Fig. 14b. The large overlap makes it difficult for morphological processing to separate the tomatoes in each bounding box. In Fig. 14c, the background and stems connected to the tomatoes were removed. Then, morphological processing was conducted to extract the target region for each candidate region; these regions were not detected accurately by Faster R-CNN and were filtered out as background in the morphological process because the third tomato was too small, as shown in Fig. 14d. The result of the illumination adjustment algorithm is shown in Fig. 14e. Fig. 14f shows the result after using the Canny operator, in which the contours are interfered by the tomato texture and the contour of the blurred part is broken. The clustering image using IFS and its edge image using adaptive threshold for overlapping tomatoes are shown in Fig. 14g and Fig. 14h, respectively; a = 0.3, and b = 0.7. The threshold for Fig. 14h is 0.2529. Contrast to Fig. 14f, the IFS method filtered out the interference of surface texture and successfully detected the contour at the left end region of the intersecting part in Fig. 14h. After using the contour detection method for the edges in Fig. 14h, As shown in Fig. 14i, the results show the robustness of the proposed tomato detection method in detecting intersecting fuzzy region contours.

For shaded tomatoes, the segmentation process is shown in Fig. 15. In Fig. 15a, the front tomato is shaded by leaves and the back tomato is obscured by stems. First, the candidate regions of the two tomatoes were detected by Faster R-CNN, as shown in Fig. 15b. The bounding boxes of the two tomatoes include areas of the other tomato, which also greatly increases the difficulty of obtaining the complete tomato contour. By using the Gaussian density function of H and S, the leaves and stems of the tomatoes were removed, as shown in Fig. 15c. Next, morphological processing was used to extract the target region for each candidate region, and the good results shown in Fig. 15d were achieved. Fig. 15e shows the result of the illumination adjustment algorithm. In Fig. 15f, since the Canny operator has poor anti-interference performance, there are many interruptions on the contours where stem and nearby tomato cause errors in contour detection. Fig. 15h presents the edge image produced by using IFS with a threshold of 0.1255. Parameters a and b for edge detection masks were 0.3 and 0.75, respectively. As shown in Fig. 15h, the visible parts of contours with the stem and the adjacent tomato were detected with fewer redundant edges inside the contours, and the proposed method was able to detect more blurred contours near the stem than the Canny operator. The processed results for the edges in Fig. 15h are shown in Fig. 15i. Although the left tomato was shaded and split into several parts by tomato leaves and stems, the proposed method still accurately detected a single tomato in the shaded state.

After the training of the Faster R-CNN model, tomatoes in separated, adjacent, overlapped, shaded states were tested and analyzed statistically. The detection accuracy of separated, adjacent, overlapped and shaded tomatoes was 95.5%, 93.8%, 78.4% and 81.9%, respectively. The results indicate that the proposed method has a lower detection rate for overlapped tomatoes and shaded tomatoes, which is due to the fact that a considerable part of tomatoes under complex environment have a higher cover rate and some tomatoes in the image are too small to detect. What's more, to better evaluate the detection performance of overlapped tomatoes, we randomly selected 80 tomato images in the dataset and counted the number of all overlapped tomatoes in the images. We use cover rate to define the degree of overlap, which is defined as the ratio of the number of observed pixels to the actual pixels. The tomatoes in the picture are divided into three categories according to cover rate, and the detection accuracy of the proposed method for tomatoes with different overlapping degree is tested. The detection accuracy of tomatoes with cover rate less than 30% is 81.5%, the detection accuracy of tomatoes with cover rate between 30% and 50% is 59.3%, and the detection accuracy of tomatoes with cover rate greater than 50% is 13.3%. The results indicate that this method has a lower detection rate for tomatoes with larger overlap.

C. RESULTS OF PARAMETER ERROR ANALYSIS FOR SINGLE TOMATO DETECTION

To further validate the performance of the proposed tomato detection method, in this section, we measured the sizes of different forms of ripe tomatoes in the image dataset. The actual sizes, including the tomato width and height, were obtained by manually labeling the true tomato contours. We randomly chose 200 tomatoes from the testing sample group with different widths and heights (ranging from 15 pixels to 300 pixels). Two comparison experiments were carried out: one compared the tomato size measurements after tomato contour detection using the proposed method

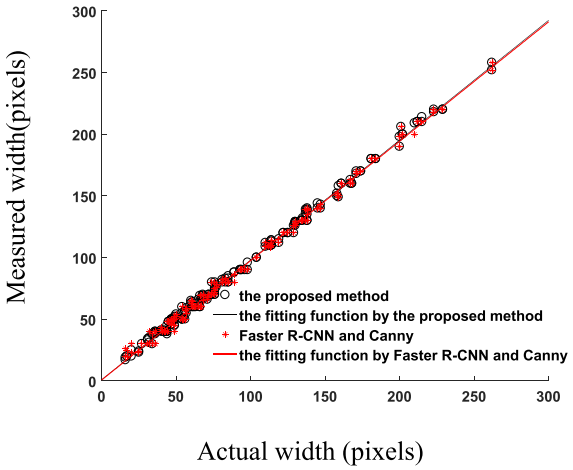


FIGURE 16. Comparison results for tomato width measurements using the proposed method and Faster R-CNN with the Canny operator vs. manual measurement.

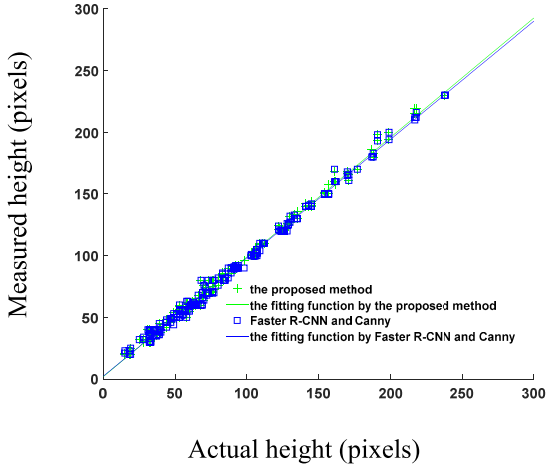


FIGURE 17. Comparison results for tomato height measurements using the proposed method and faster R-CNN with the Canny operator vs. manual measurement.

with the actual manual measurement, and the other compared the tomato size measurements using Faster R-CNN and the Canny operator with the actual manual measurement, as illustrated in Fig. 16 and Fig. 17.

Fig. 16 presents the results of the tomato width measurements, where the black circles (○) represent the measured width using this proposed method. The corresponding black line is a fitting function between the measured width and the actual width; the slope is 0.971, the offset is 0.988, the root mean square error (RMSE) is 2.996 pixels, the R-square value is 0.997, and the fitting function is $y = 0.971x + 0.988$. The red asterisks (*) and the corresponding red line represent the measured width and fitting function using Faster R-CNN with the Canny operator; the RMSE is 3.496 pixels, the R-square is 0.995 and the fitting function is $y = 0.968x + 0.814$. Fig. 17 shows the tomato height detection measurement results, where the green plus (+) and the green line represent the measured height after extracting tomatoes using the proposed method and its fitting function, respectively; the slope is 0.968, the offset is 2.046, the RMSE is 3.306 pixels, the R-square is 0.995, and the fitting function is $y = 0.968x + 2.046$. The blue squares (□) and the blue line represent the measured height after extracting the tomatoes using the Canny operator and its fitting function, respectively; the RMSE is 3.731 pixels, the R-square is 0.994, and the fitting function is $y = 0.959x + 2.356$. According to the experimental data analysis, IFS is better than Canny operator in edge detection. After the width and height were measured, the relative error of the central coordinates was calculated. The mean relative error percent (MRE%) values of the shifted horizontal and vertical distances of center positions using the proposed method are 0.261% and 1.179%, respectively. These results indicate that the proposed method can be used to effectively detect ripe tomatoes, even in the presence of overlap or leaf occlusion.

IV. DISCUSSION AND CONCLUSION

A. DISCUSSION

Numerous studies have been proposed to detect tomatoes based on different factors under natural conditions. Studies [39], [40] that used only deep learning to detect objects perform faster and extract deeper features but have greater difficulty detecting overlapping tomatoes than the method proposed in this article. Xiong *et al.* [41] proposed a method based on color analysis and vertical suspension angle analysis to detect green grapes. Wan *et al.* [42] used a color model and shape analyses to detect tomatoes. However, these approaches cannot extract deeper features and are prone to recognition errors in complex environments. Whittaker *et al.* [43] used a shape-based method (circular Hough transform) to detect tomatoes and demonstrated its ability to locate tomatoes based on shape only in images with substantial background noise. However, this method is computationally intensive (it expends considerable time on non-tomato processing); consequently, it is difficult to use in real-time robotic harvesting. The proposed method has a great advantage in terms of speed because it first uses Faster R-CNN to recognize and locate candidate tomatoes and then processes only tomatoes detected by Faster R-CNN. In our experiment, different tomato states were tested to evaluate the feasibility of the method. The experimental results prove that the proposed method can detect overlapping tomatoes under natural conditions. The proposed method using Faster R-CNN and IFS is robust to complex background conditions. Since the features of tomatoes can be extracted deeply based on Faster R-CNN, candidate tomatoes can be detected even though they overlap (Fig. 14a) or are divided into several parts by stems and leaves (Fig. 15a).

The focus of this work is to separate single tomatoes from overlapping tomatoes. In this paper, IFS is used to segment the target tomato contour to separate it from the overlapping tomatoes, and the illumination adjustment algorithm is adopted to compensate for the illumination of the candidate tomato region, reducing the probability of recognizing the

uneven illumination position as the edge. For the contour of overlapped areas, especially the two ends of an overlapped contour, there is considerable interference. Traditional edge detectors, such as the Canny operator, are too sensitive to noise, but IFS can blur the region with large interference on the edge of a tomato and effectively detect the edge of the region by setting an appropriate threshold.

We identified the primary causes of error, showing that the accuracy of the final contour segmentation result depends on the accuracy of the initial location and the recall of Faster R-CNN. We analyzed the detection results of Faster R-CNN classifier for four kinds of tomatoes. The detection rate of overlapped tomatoes and shaded tomatoes is relatively low, which is due to the large ratio of overlapping and shading for a part of tomatoes in the natural environment. For separated and adjacent tomatoes, the reason that cause an effect on the detection rate is that there are a few very small tomatoes in the images which are difficult to detect by Faster R-CNN. Those with an cover rate of more than 50% (Fig. 14a) have a 13.3% probability of detection, and for those with too much cover rate cannot even detect (seen in Fig. 14b). We found that the detection rate will decrease with the increase of cover rate of tomato.

The results show the applicability of the proposed method. First, the proposed method can be applied to the visual system for tomato location to increase the grasping accuracy of the tomato picking robot. In addition, it can be applied not only to tomatoes, but also to other partially shaded or overlapped fruit detection and location systems. However, there are two problems, one is that the computer visual system itself is greatly affected by the light, when the light is too weak or too strong, which result in the loss of image information. The other is that excessively large errors in the deep learning bounding box affect the tomato contour extraction, which leads to inaccuracies in calculating the tomato's height, width and center.

In future work, we plan to improve the performance of the deep learning classifier to detect distant tomatoes and tomatoes with high cover rates in the image and increase the recall rate of tomato detection. Further more, we will study multi-sensor fusion technology to solve the inherent problems of computer systems that RGB camera performs poorly in extreme light conditions. For example, the combination of infrared sensor and RGB camera can improve the performance of computer vision system in the case of too strong and too weak illumination.

V. CONCLUSION

In this paper, we introduced a method that can detect a single ripe tomato by combining IFS with the Faster R-CNN image detection method. The proposed method has several advantages over traditional methods. First, we labeled ripe tomatoes in different configurations (e.g., separated, adjacent, overlapping, and shaded) in a large number of images to train the Faster R-CNN detector. We identified candidate mature tomato regions in images using the trained Faster R-CNN classifier. The results showed that the trained Faster R-CNN classifier can accurately and quickly localize candidate ripe tomato regions. Then, we transformed the RGB color space for the candidate tomato region to the HSV color space. Different tomato samples were segmented manually and used to establish the Gaussian density function to remove the background from single tomatoes detected by Faster R-CNN to obtain the candidate tomato body. In some cases, subpixels or adjacent tomatoes that interfere with tomato contour extraction remain around the tomato body. Therefore, we conducted morphological processing on the tomato binary map to remove these extraneous subpixels and separate connected tomatoes to reduce the extra contour obtained by edge detection. Finally, we proposed a tomato contour extraction method to further detect tomatoes that uses the IFS edge detection method to obtain the edge and then applies a contour detection method to connect edge breakpoints and remove redundant edge points. Together, these operations connect the tomato contour and help to obtain accurate values for a tomato's width, height and center.

We conducted three experiments in this study. One experiment analyzed the precision and recall of Faster R-CNN for ripe tomato detection; the AP achieved was approximately 80% despite the complexity of the greenhouse tomato images used, which include adjacent, overlapping and obscured tomatoes. The second experiment presented the contour segmentation process of tomatoes in four different states. The results show the applicability of our proposed method for tomato segmentation. The last experiment validated the tomato localization performance. We conducted comparison experiments to analyze the parameter errors for a single tomato using the proposed method and Faster R-CNN with the Canny operator, compared to manual measurement. The proposed method is able to accurately calculate the width, height and center position of a single tomato in an image; the RMSE values for the width and height are 2.996 and 3.306 pixels, respectively. The mean relative error percent (MRE%) for the shifted horizontal and vertical distances of the center positions are 0.261% and 1.179%, respectively. If we use Faster R-CNN without further contour detection to detect the tomato, the RMSE values for the width and height are 7.915 and 8.436 pixels, respectively. These results demonstrate that the proposed method can localize the tomato center more accurately than Faster R-CNN alone.

ACKNOWLEDGMENT

The authors would like to thank the Jiangsu Academy of Agricultural and Sciences, of Nanjing, China for their assistance with data collection and technical support.

REFERENCES

[1] X. Wei, K. Jia, J. Lan, Y. Li, Y. Zeng, and C. Wang, "Automatic method of fruit object extraction under complex agricultural background for vision system of fruit picking robot," *Optik*, vol. 125, no. 19, pp. 5684–5689, 2014.

[2] R. Zhou, L. Damerow, Y. Sun, and M. M. Blanke, "Using colour features of cv. 'Gala' apple fruits in an orchard in image processing to predict yield," *Precis. Agricult.*, vol. 13, no. 5, pp. 568–580, 2012.

[3] A. Arefi, A. M. Motlagh, K. Mollazade, and R. F. Teimourlou, "Recognition and localization of ripen tomato based on machine vision," *Austral. J. Crop Sci.*, vol. 5, no. 10, pp. 1144–1149, 2011.
[4] M. TeixidÃş, D. Font, T. Pallejà, M. Tresanchez, M. Nogués, and J. Palacín, "Definition of linear color models in the RGB vector color space to detect red peaches in orchard images taken under natural illumination," *Sensors*, vol. 12, no. 6, pp. 7701–7718, 2012.
[5] A. Ostovar, O. Ringdahl, and T. Hellström, "Adaptive image thresholding of yellow peppers for a harvesting robot," *Robotics*, vol. 7, no. 1, p. 11, 2018.
[6] L. Luo, Y. Tang, X. Zou, C. Wang, P. Zhang, and W. Feng, "Robust grape cluster detection in a vineyard by combining the Adaboost framework and multiple color components," *Sensors*, vol. 16, no. 2, p. 2098, 2016.
[7] Z. Xie, C. Ji, X. Guo, and S. Ren, "Open access an object detection method for quasi-circular fruits based on improved Hough transform," *Trans. Chin. Soc. Agricult. Eng.*, vol. 26, no. 7, pp. 157–162, 2010.
[8] T.-H. Liu, R. Ehsani, A. Toudeshki, X.-J. Zou, and H.-J. Wang, "Detection of citrus fruit and tree trunks in natural environments using a multi-elliptical boundary model," *Comput. Ind.*, vol. 99, pp. 9–16, Aug. 2018.
[9] E. K. Nyarko, I. Vidović, K. Radočaj, and R. Cupec, "A nearest neighbor approach for fruit recognition in RGB-D images based on detection of convex surfaces," *Expert Syst. With Appl.*, vol. 114, pp. 454–466, Dec. 2018.
[10] J. Wu, B. Zhang, J. Zhou, Y. Xiong, B. Gu, and X. Yang, "Automatic recognition of ripening tomatoes by combining multi-feature fusion with a bi-layer classification strategy for harvesting robots," *Sensors*, vol. 19, no. 3, p. 612, 2019.
[11] K. Yamamoto, W. Guo, Y. Yoshioka, and S. Ninomiya, "On plant detection of intact tomato fruits using image analysis and machine learning methods," *Sensors*, vol. 14, no. 7, pp. 12191–12206, Jul. 2014.
[12] R. Fernández, C. Salinas, H. Montes, and J. Sarria, "Multisensory system for fruit harvesting robots. Experimental testing in natural scenarios and with different kinds of crops," *Sensors*, vol. 14, no. 12, pp. 23885–23904, 2014.
[13] H. Gan, W. S. Lee, V. Alchanatis, R. Ehsani, and J. K. Schueller, "Immature green citrus fruit detection using color and thermal images," *Comput. Electron. Agricult.*, vol. 152, pp. 117–125, Sep. 2018.
[14] M. Monta and K. Namba, "Three-dimensional sensing system for agricultural robots," in *Proc. AIM*, Kobe, Japan, Jul. 2003, pp. 1216–1221.
[15] H. N. Patel, R. K. Jain, and M. V. Joshi, "Fruit detection using improved multiple features based algorithm," *Int. J. Comput. Appl.*, vol. 13, no. 2, pp. 1–5, 2011.
[16] H. Li, W. S. Lee, and K. Wang, "Immature green citrus fruit detection and counting based on fast normalized cross correlation (FNCC) using natural outdoor colour images," *Precis. Agricult.*, vol. 17, no. 6, pp. 678–697, 2016.
[17] W. C. Seng and S. H. Mirisaee, "A new method for fruits recognition system," in *Proc. ICEEI*, Bangi, Malaysia, Aug. 2009, pp. 130–134.
[18] J. Senthilnath, A. Dokania, M. Kandukuri, K. N. Ramesh, G. Anand, and S. N. Omkar, "Detection of tomatoes using spectral-spatial methods in remotely sensed RGB images captured by UAV," *Biosyst. Eng.*, vol. 146, pp. 16–32, Jun. 2016.
[19] E. Barnea, R. Mairon, and O. Ben-Shahar, "Colour-agnostic shape-based 3D fruit detection for crop harvesting robots," *Biosyst. Eng.*, vol. 146, pp. 57–70, Jun. 2016.
[20] I. Sa, Z. Ge, F. Dayoub, B. Upcroft, T. Perez, and C. Mccool, "DeepFruits: A fruit detection system using deep neural networks," *Sensors*, vol. 16, no. 8, p. 1222, 2016.
[21] G. Lin, Y. Tang, X. Zou, J. Xiong, and J. Li, "Guava detection and pose estimation using a low-cost RGB-D sensor in the field," *Sensors*, vol. 19, no. 2, p. 428, 2019.
[22] M. Stein, S. Bargoti, and J. Underwood, "Image based mango fruit detection, Localisation and yield estimation using multiple view geometry," *Sensors*, vol. 16, no. 11, p. 1915, 2016.
[23] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. CVPR*, Columbus, OH, USA, Jun. 2014, pp. 580–587.
[24] R. Girshick, "Fast R-CNN," in *Proc. ICCV*, Santiago, Chile, Dec. 2015, pp. 1440–1448.
[25] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, Sep. 2015.
[26] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
[27] O. Korostynska, A. Mason, and P. J. From, "Electromagnetic sensing for non-destructive real-time fruit ripeness detection: Case-study for automated strawberry picking," in *Proc. Eurosens. Conf.*, Graz, Austria, Sep. 2018, pp. 980-1–980-5. [Online]. Available: https://www.mdpi.com/2504-3900/2/13/980
[28] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2012, pp. 1097–1105.
[29] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.
[30] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, "Selective search for object recognition," *Int. J. Comput. Vis.*, vol. 104, no. 2, pp. 154–171, Apr. 2013.
[31] C.-H. Hu and P. Li, "Edge detection for hardwood seedlings leaves based on Intuitionistic fuzzy set," in *Proc. ICITA*, Nov. 2013, pp. 76–80.
[32] P.-L. Shui and W.-C. Zhang, "Noise-robust edge detector combining isotropic and anisotropic Gaussian Kernels," *Pattern Recognit.*, vol. 45, no. 2, pp. 806–820, Feb. 2012.
[33] W. Zhang, Y. Zhao, T. P. Breckon, and L. Chen, "Noise robust image edge detection based upon the automatic anisotropic Gaussian Kernels," *Pattern Recognit.*, vol. 63, pp. 193–205, Mar. 2017.
[34] K. T. Atanassov, "Intuitionistic fuzzy sets," *Fuzzy Sets Syst.*, vol. 20, pp. 87–96, Aug. 1986.
[35] T. Chaira, "A rank ordered filter for medical image edge enhancement and detection using intuitionistic fuzzy set," *Appl. Soft Comput.*, vol. 12, no. 4, pp. 1259–1266, 2012.
[36] P. Melo-Pinto, P. Couto, H. Bustince, E. Barrenechea, M. Pagola, and J. Fernandez, "Image segmentation using Atanassov's intuitionistic fuzzy sets," *Expert Syst. With Appl.*, vol. 40, no. 1, pp. 15–26, 2013.
[37] X. Fu, Y. Sun, M. LiWang, Y. Huang, X.-P. Zhang, and X. Ding, "A novel retinex based approach for image enhancement with illumination adjustment," in *Proc. ICASSP*, May 2014, pp. 1190–1194.
[38] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst., Man, Cybern.*, vol. 9, no. 1, pp. 62–66, Jan. 1979.
[39] J. Gené-Mola, V. Vilaplana, J. R. Rosell-Polo, J.-R. Morros, J. Ruiz-Hidalgo, and E. Gregorio, "Multi-modal deep learning for Fuji apple detection using RGB-D cameras and their radiometric capabilities," *Comput. Electron. Agricult.*, vol. 162, pp. 689–698, Jul. 2019.
[40] L. Fu, Y. Feng, Y. Majeed, X. Zhang, J. Zhang, M. Karkee, Q. Zhang, "Kiwifruit detection in field images using Faster R-CNN with ZFNet," *IFAC-PapersOnLine*, vol. 51, no. 17, pp. 45–50, 2018.
[41] J. Xiong, Z. Liu, R. Lin, R. Bu, Z. He, Z. Yang, and C. Liang, "Green grape detection and picking-point calculation in a night-time natural environment using a charge-coupled device (CCD) vision sensor with artificial illumination," *Sensors*, vol. 18, no. 4, p. 969, 2018.
[42] P. Wan, A. Toudeshki, H. Tan, and R. Ehsani, "A methodology for fresh tomato maturity detection using computer vision," *Comput. Electron. Agricult.*, vol. 146, pp. 43–50, Mar. 2018.
[43] D. Whittaker, G. E. Miles, O. R. Mitchell, and L. D. Gaultney, "Fruit location in a partially occluded image," *Trans. ASAE*, vol. 30, no. 3, pp. 591–596, 1987.

CHUNHUA HU received the B.S. degree in mechanical design and automation and the M.S. degree in agricultural electrification and automation from Jiangsu University, China, in 2001 and 2004, respectively, and the Ph.D. degree in control theory and control engineering from Southeast University, China, in 2008. She was a Teacher with the Department of Electrical Automation, Changzhou College of Science and Engineering, China, from 2008 to 2012. Since 2012, she has been an Associate Professor with the School of Information Science and Technology, Nanjing Forestry University, China. Her main research interests include computer vision, artificial intelligence, and 3D visualization.

XUAN LIU received the B.S. degree in electrical engineering and automation, Nanjing Forestry University, China, in 2018, where he is currently pursuing the M.S. degree in instrument engineering. His research interests include computer vision and machine learning.

ZHOU PAN received the B.S. degree in electrical engineering and automation from the Huaian Institute of Technology, China, in 2016. She is currently pursuing the M.S. degree in control theory and control engineering with Nanjing Forestry University, China. Her research interests include computer vision and machine learning.

PINGPING LI received the B.S. and M.S. degrees in agronomy from Zhejiang Agricultural University, China, in 1982 and 1985, respectively, and the Ph.D. degree in agronomy from Nanjing Agricultural University, China, in 1995. She was a Teacher with the Department of Agriculture, Nanjing Agricultural University, China, from 1985 to 1997. She was a Research Professor with the College of Mechanical Engineering, Jiangsu University, from 1997 to 2011. Since 2011, she has been with the School of Biology and Environment, Nanjing Forestry University, China, as a Professor. Her main research interests include computer vision, agricultural information engineering, and facility horticulture.