IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received November 2, 2020, accepted November 11, 2020, date of publication November 16, 2020,
date of current version November 30, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3038184

Using Fuzzy Mask R-CNN Model to Automatically Identify Tomato Ripeness

YO-PING HUANG[1,2], **(Fellow, IEEE), TZU-HAO WANG** [1], **(Graduate Student Member, IEEE), AND HAOBIJAM BASANTA**[1]

[1]Department of Electrical Engineering, National Taipei University of Technology, Taipei 10608, Taiwan
[2]Department of Information and Communication Engineering, Chaoyang University of Technology, Taichung 41349, Taiwan

Corresponding author: Yo-Ping Huang (yphuang@ntut.edu.tw)

This work was supported in part by the Ministry of Science and Technology, Taiwan, under Grant MOST108-2221-E-027-111-MY3, Grant MOST108-2321-B-027-001-, and Grant MOST109-2622-E-027-001-CC3.

ABSTRACT Manual inspection and harvesting of ripening tomatoes is time consuming and labor intensive. Smart agriculture can emphasize the use of digital horticultural resources for farming and can increase farm sustainability; to that end, we proposed a fuzzy Mask R-CNN model to automatically identify the ripeness levels of cherry tomatoes. First, to annotate the images automatically, a fuzzy c-means model was used to maintain the spatial information of various foreground and background elements of the image. Then, a Hough transform method was applied to locate the specific geometric edge positions of the tomatoes. Each data point of the image space was annotated to a JavaScript Object Notation file. Second, annotated images were trained with Mask R-CNN to identify each tomato precisely. Finally, to prevent preharvest abscission of tomatoes, a hue–saturation–value color model and fuzzy inference rules were used to predict the ripeness of the tomatoes. A trigonometric function with Euclidian distance was calculated from the origin of calyx and stem to the bottom of the tomato to obtain the position of the pedicle head and dissect the fruit in a timely manner. For detection of 100 tomato images, Mask R-CNN achieved an accuracy of 98.00%. The ripeness classification of tomatoes achieved overall weighted precision and recall rates of 0.9614 and 0.9591, respectively. Thus, automatic tomato harvesting applications can empower farmers to make better decisions and enhance overall production efficiency and yield.

INDEX TERMS Automatic annotation, detection of tomato ripeness, fuzzy c-means, Mask Region-based Convolutional Neural Network (Mask R-CNN), hue–saturation–value (HSV) color model.

I. INTRODUCTION

Tomato cultivation is one of the most globalized horticultural industries, as tomatoes are extensively consumed worldwide. Relative to other crops grown worldwide, tomato cultivation quantities are three times higher than those of potatoes and six times more than rice [1]. On a global scale, The Food and Agriculture Organization of the United Nations estimated that the world annual production for tomatoes in 2016 was 179 508 401 metric tons. However, in 2017, tomato production grew by 1.6%, with an estimated production of approximately 182 301 395 metric tons [2]. Cultivation of tomatoes is economically crucial, especially in rural and suburban areas of most developing countries [3]. Additionally, it is true that quality–yield measurement not only benefits consumers but also economically benefits the farmers who toil diligently, day and night, to produce yields of the highest possible quality. Harvesting is an essential task in horticultural activity. Maturity at harvest is a vital factor that determines the storage life and final fruit quality, flavor, juiciness, and texture.

When immature fruits are harvested, they are of poor quality and are often incapable of ripening; immature fruits are eventually susceptible to internal deterioration and decay. Conversely, delayed harvesting of fruits and vegetables can markedly increase chances of fruit damage, resulting in drastic postharvest loss. To control the quantitative or qualitative losses of preharvest and postharvest vegetables, it is therefore crucial to understand the delicateness of vegetables, physiological maturity conditions, and methods of timely harvesting, as well as other factors. The loss of quality of tomatoes constitutes a major challenge for tomato cultivators. Monitoring the growth of tomatoes and harvesting them at the breaker

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

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

stage reduce the chance of cracking or damage and also help the farmers to control the ripening progression. Typically, detection of any type of disease and evaluation of the stage of ripeness in fruits or vegetables are conducted through manual inspection and evaluation according to the farmer's personal experience. At the same time, farm production entails a number of challenges and must overcome unfavorable agroclimatic conditions such as soil degradation, lack of water, climate change, and natural calamities (droughts, floods, and hailstorms) that destabilize farms. Thus, the refinement of various horticultural practices with innovative technologies can intensify the strategic advantages of agricultural production. To overcome such obstacles to farming, this paper proposes an innovative system for perceiving different stages of maturation in tomatoes that are cultivated in open fields. To implement a feasible real-time system, tomato images acquired from open fields are identified and segmented using fuzzy c-means method. Then, acquired images are automatically annotated to filter the images of interest. Mask R-CNN is used to estimate the precise position of each tomato, thereby improving the obtained segmentation for more useful experimental results. Finally, a hue–saturation–value (HSV) color model is used to predict the ripeness of the tomato and the appropriate schedule for timely harvesting.

This method can help growers to discover optimally mature tomatoes and determine whether they should be picked or not. In this manner, the quality of harvested tomatoes can be enhanced.

The structure of the remainder of this paper is as follows. Section II explains related image processing techniques that can assist in tomato quality assessment and ripeness detection. Section III describes the materials and method adopted in the proposed system. Section IV presents the experiment results and analyzes the data sets. Section V provides conclusions and directions for future study.

II. RELATED WORK

Digitalization in agriculture technology significantly transforms farming, overcomes farming challenges, and raises farming efficiency with better environmental, social, and economic sustainability by lowering production costs and retaining better yields of production with high efficiency.

In the early twenty-first century, the employment of computer vision with object detection has provided a new method for precision farming that has enabled farmers to accurately perform soil mapping, crop scouting, disease detection, visual inspection of fruits, fruit grading, fruit counting, and yield estimation without human intervention. Scholars have published numerous studies intended to assist in the quality assessment of fruits and vegetables based on texture and color feature extraction. Wan *et al.* [4] used color feature values with a backpropagation neural network classification technique to classify the maturity levels of Roma and Pear tomato varieties, yielding an accuracy of 99.31%. Zhao *et al.* [5] used L*a*b* color space and luminance in-phase quadrature-phase (YIQ) color space with wavelet transformation image fusion features to recognize mature tomatoes. Arefi *et al.* [6] designed an algorithm for harvester robots to recognize and localize ripe tomatoes using combinations of morphological features and RGB, HSI, and YIQ spaces. Their accuracy rate was 96.36%, but their harvester robot did not recognize and localize occluded tomatoes. To reduce the influence of illumination and occlusion in tomato detection, Liu *et al.* [7] introduced histograms of oriented gradients with support vector machine (SVM). Their method achieved 90.00% accuracy, 94.41% precision, and 92.15% F1 metrics. However, their method is unsuitable when more than 50% of the blocked area has overlapped and occluded tomatoes. Numerous researchers have tried various sensors and have applied machine learning or deep learning techniques to overcome the challenges of recognizing tomatoes under varying illumination, overlapping, and occlusion conditions. Nyalala *et al.* [8] combined SVM and Bayesian-ANN to estimate mass and volume values of cherry tomatoes based on depth images from both two-dimensional (2D) and three-dimensional (3D) images.

Yuan *et al.* [9] designed a robust cherry tomato detection algorithm based on a single-shot multibox detector. Moreover, the method was compared with various base networks of VGG16, MobileNet, and InceptionV2 networks. InceptionV2 achieved an average precision of 98.85%. Hu *et al.* [10] combined Faster R-CNN and intuitionistic fuzzy sets for automatic detection of a single ripe tomato on a plant. Wu *et al.* [11] adopted a bilayer classification strategy with multiple-feature analysis and a weighted related vector machine classifier to recognize ripening tomatoes. However, markedly limited numbers of tomato samples have been used in most relevant deep learning experimentation. With classifiers (and especially neural networks), it would become problematic if correct classification of tomato ripeness entailed increases in classification error rates. To overcome the existing challenges, we developed a system to enhance recognition efficiency in complex environments.

1. Unlike other studies, the present study accomplished automatic annotation of examined tomatoes.
2. Relative to competitor systems, our system more accurately detected different physiologically levels of tomatoes from the immature green to mature stage of harvesting, namely the immature (green), breaker (green to tannish yellow), preharvest (orange), and harvest (red) stages.
3. Our model can detect tomatoes efficiently despite environment challenges such as varying illumination, overlapping fruits, or occlusion of leaves and branches.
4. Our platform can streamline tomato picking by detecting the right position of the pedicle head.

III. MATERIALS AND METHODS

This section describes the data acquisition process, materials, and methods (including contour detection, feature extraction, and segmentation) that enable the effect of the classifier.

A. DATA COLLECTION AND PREPROCESSING

Tomato images with dimensions of 1108 × 1478 pixels per image were collected from a greenhouse cooperative farm located in Tainan, Taiwan. All collected images were screened carefully; we selected 900 images after excluding defective items. To extract highly relevant features and to overcome any overfitting of the data set, images were augmented as follows:

- Translation: Images were randomly shifted -10 to 10 pixels.
- Flipping: Images were horizontally (mirror image) flipped.
- Gaussian filtering: Images were blurred for effective smoothing of noise.

We generated 2000 data items after data augmentation; the obtained data set was divided into a training set and a validation set at a ratio of 80:20. Finally, another 20 test samples depicting 100 tomatoes were used for testing.

B. FUZZY C-MEANS SEGMENTATION

Fuzzy c-means (FCM) is an unsupervised method developed by Dunn in 1973 [12] that was further improved by Bezdek in 1981 [13]. This fuzzy-logic-based clustering algorithm is widely used for solving multiclass and ambiguous problems. FCM is an iterative optimization method in which one sample can be assigned to more than one cluster. It is directly implemented on a data matrix to generate a membership function that represents the degree of association of the samples with each cluster. For instance, each image pixel has a specific membership degree associated with each of the cluster centroids. Then, the membership of each pixel is calculated and represented by a membership value between 0 and 1. This specifies the strength of the association between that image pixel and a particular cluster centroid.

FCM partitions every image pixel into a collection of the M fuzzy cluster centroids with respect to certain given conditions. Initially, let N be the total number of pixels in a given image and let m be the exponential weight of the membership degree. The minimization of objective function O_m of the FCM is defined as [13]

$$O_m(U, V) = \sum\nolimits_{j=1}^{N} \sum\nolimits_{k=1}^{M} u_{\mathrm{jk}}^{m} d_{\mathrm{jk}}^{2} \tag{1}$$

where u_{jk} is the degree of membership of the jth pixel in the kth cluster, d_{jk} is the distance between the jth pixel and the kth cluster center, and $O_m(U,V)$ is the performance index that measures the weighted sum of distance d_{jk} between the jth pixel and the kth cluster center. The membership degree of the jth pixel to the kth cluster center indicates the membership value u_{jk}, where $u_{jk} \in [0, 1]$. A membership value close to 1 represents that the pixel belongs to the corresponding cluster. If $U_j = (u_{j1}, u_{j2}, \ldots, u_{jM})^T$ is the set of membership degrees of the jth pixel associated with each cluster center, x_j is the jth pixel in the image, and v_k is the kth cluster center, then $U = (U_1, U_2, \ldots, U_N)$ is the membership degree matrix and $V = (v_1, v_2, \ldots, v_M)$ is the set of cluster centers.

The FCM algorithm can be explained as follows:

Step 1: Set the initial parameters, such as the number of clusters V, convergence error ε, and the number of iterations s to 0.

Step 2: Calculate $U^{(s)}$ according to $V^{(s)}$, defined as

$$u_{jk} = \frac{1}{\sum_{g=1}^{M} \left(\frac{d_{jk}}{d_{gk}}\right)^{2/(m-1)}}, \quad 1 \leq \mathrm{j} \leq \mathrm{N} \tag{2}$$

If $d_{jk} = 0$, then $u_{jk} = 1$; then, set the other membership degrees of this pixel to 0.

Step 3: Calculate $V^{(s+1)}$ according to $U^{(s)}$, defined as

$$V_k = \frac{\sum_{j=1}^{N} u_{jk}^{m} x_j}{\sum_{j=1}^{N} u_{jk}^{m}}, \quad 1 \leq \mathrm{k} \leq \mathrm{M} \tag{3}$$

Step 4: Update $U^{(s+1)}$ according to $V^{(s+1)}$ using (3).

Step 5: Finally, compare $U^{(s+1)}$ with $U^{(s)}$. If $||U^{(s+1)} - U^{(s)}|| \leq \varepsilon$, then stop execution. Otherwise, repeat step 2.

C. HOUGH TRANSFORM

The Hough transform (HT) [14] is an efficient method for extracting geometric shapes by independently considering geometric data composed of edge points from a digital image [15]. The key notion of the standard HT is to define a mapping between an image space and a parameter space such that every edge point in the edge map of a tomato is transformed to all possible lines that could pass through that point. In the case of circle detection, if a circle is in the image, then it is defined as

$$(x - c_x)^2 + \left(y - c_y\right)^2 = r^2 \tag{4}$$

where (c_x, c_y) are the coordinates of the circle center, and r is the radius. To transform a 2D input edge image $I(x,y)$ to a 3D accumulator matrix $A(c_x, c_y, r)$, the HT for the circle is described as follows [16]:

Algorithm 1 Pseudocode for Hough circle

```
**Input:** Image $I(x, y)$
**Output:** Detect circle
**Initialize:** Accumulator array $A(c_x, c_y, r)$ to zeros
**for** all $x$:
    **for** all $y$:
        **If** $I(x, y)$:
            **for**all $c_x$:
                **for** all $c_y$:
                    $r = \sqrt{(x - c_x)^2 + \left(y - c_y\right)^2}$
                    $A(c_x, c_y, r) \leftarrow A(c_x, c_y, r) + 1$
                **end for**
            **end for**
    **end for**
**end for**
```

For ellipse shape detection, we adopted the methods of [17], [18]. An ellipse consists of five unknown parameters, which can be denoted as follows. Let point c be the center position of the ellipse; (c_x, c_y) denotes the coordinates of point c, α and β represent the half-lengths of the major and minor axes, respectively, and θ denotes the angle between the major axis and the x-axis. Consider an arbitrary ellipse with points p and q as the foci of the ellipse and with point c as the center position. For each pixel, (x_1, y_1) and (x_2, y_2) can be used to calculate the four parameters $\{c_x, c_y, \alpha, \theta\}$ for the assumed ellipse as follows:

$$c_x = \frac{x_1 + x_2}{2} \tag{5}$$

$$c_y = \frac{y_1 + y_2}{2} \tag{6}$$

$$\alpha = \frac{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}}{2} \tag{7}$$

$$\theta = \tan^{-1}\left(\frac{y_2 - y_1}{(x_2 - x_1)}\right) \tag{8}$$

Fig. 1 shows an arbitrary ellipse with foci p and q with center c. To calculate the half-length of the minor axis β, let d be the arbitrary point on the contour of the ellipse. Because p and q are the foci of an ellipse, the sum of the line segments $l_{p,d}$ and $l_{q,d}$ can be estimated as follows [17]:

$$\begin{aligned} &\sqrt{(d_x - p_x)^2 + (d_y - p_y)^2} \\ &\quad + \sqrt{(d_x - q_x)^2 + (d_y - q_y)^2} \\ &= 2\alpha \end{aligned} \tag{9}$$

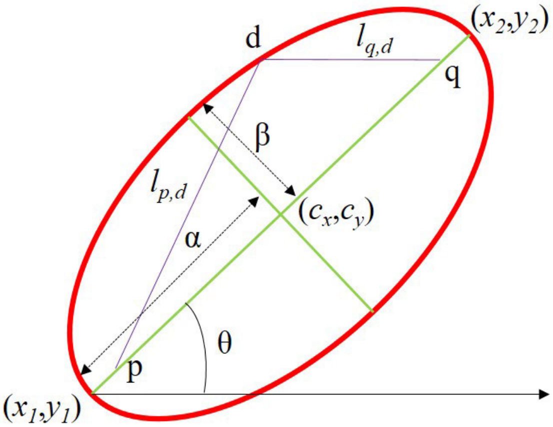


FIGURE 1. Arbitrary ellipse with foci *p* and *q*, and center *c*.

where

$$p_x = c_x - \cos|\theta|\sqrt{\alpha^2 - \beta^2} \tag{10}$$

$$p_y = c_y - \sin|\theta|\sqrt{\alpha^2 - \beta^2} \tag{11}$$

$$q_x = c_x + \cos|\theta|\sqrt{\alpha^2 - \beta^2} \tag{12}$$

$$q_y = c_y + \sin|\theta|\sqrt{\alpha^2 - \beta^2} \tag{13}$$

Thus, for a given arbitrary point on the contour of the ellipse, the value of β can be derived using (9) to (13):

$$\beta = \sqrt{\frac{\alpha^2\delta^2 - \alpha^2\gamma^2}{\alpha^2 - \gamma^2}} \tag{14}$$

where

$$\delta = \sqrt{(d_x - c_x)^2 + (d_y - c_y)^2} \tag{15}$$

$$\gamma = \cos|\theta|\,(d_x - c_x) + \sin|\theta|(d_y - c_y) \tag{16}$$

D. MASK R-CNN

In this section, a scheme for tomato instance segmentation based on Mask R-CNN is explained. Mask R-CNN [19], [20] is an intuitive extension of Faster R-CNN with additional object segmentation for a manageable number of candidate object regions of interest, enabling locations and shapes of object instances to be attained accurately. Mask R-CNN can accurately mark object regions with bounding boxes and can extract object regions from the background at the pixel level. Additionally, the picking points of a tomato pedicle can be localized easily by analyzing the shape and edge features of the mask images generated from Mask R-CNN [21], [22].

Fig. 2 displays the architecture of Mask R-CNN tomato instance segmentation.

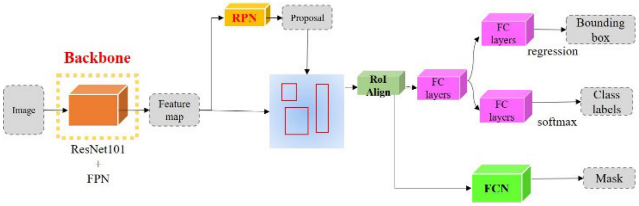


FIGURE 2. Architecture of Mask R-CNN.

Specifically, for feature extraction over an entire tomato image, we compared both the backbone networks of Mask R-CNN, ResNet-50 and ResNet-101 feature pyramid network (FPN) models. For comparison, we considered the prediction rate and computational time of the Mask R-CNN model. To validate the performance, we used 20 images with 100 tomatoes. ResNet-50 FPN shows an accuracy of 97% with an average time interval of 1.45 sec in each tomato detection whereas the ResNet-101 FPN shows an accuracy of 98% with time interval of 1.65 sec in each tomato detection. Since accuracy is more concerned, we adopted ResNet-101 architecture that includes stacked convolutional layers, a pooling layer, and residual connections. The model comprises five convolutional blocks; the first block uses a convolutional layer size of 7 × 7, and the second to fifth blocks are confined to convolutional layers of sizes 1 × 1, 3 × 3, and 1 × 1. We employed FPN methods to improve the network backbone to extract relevant semantic and spatial information for tomatoes of various sizes. Then, feature maps generated from the backbone were sent to the region proposal network (RPN) to create regions of interest for each feature map with anchors. This defined the scores and position coordinates of the foreground and background of the tomato image. RPN predictions helped the anchor to select the best bounds for the target tomato and to fine-tune its position and size. If multiple anchors overlap each other, the anchor with the highest foreground score is recorded and the rest are discarded, after which we obtain the final regional proposal and pass it on to RoIAlign. Our system computes the value of each sampling point of the feature map through bilinear interpolation to reduce the feature losses that might be caused by the spatial quantization.

Finally, regions of interest generated from the RPN layer are sent to the fully connected layer to create bounding boxes and segmentation masks for the target tomatoes. Fig. 3 depicts the architecture of Mask R-CNN tomato instance segmentation.

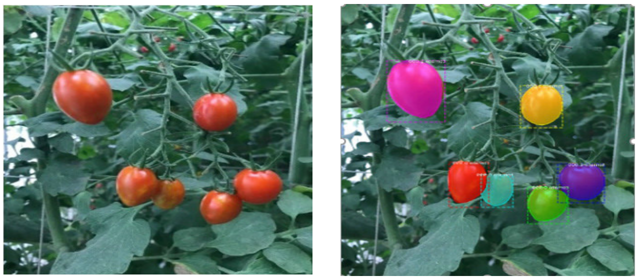

(a)
(b)

FIGURE 3. (a) Original tomato images; (b) segmentation masks and bounding boxes for the target tomatoes.

E. RIPENESS DETECTION

Tomatoes often ripen from the bottom up; through a gradual process, color changes first for a small portion of the skin, and eventually, the color of the entire skin indicates ripeness. However, tomato harvesting is influenced by the climate and varies by the variety that is grown. Tomatoes that have been harvested at the right time taste excellent and yield flavor far superior to the flavor of fruit that has been picked early. Overripe tomatoes typically ripen off the vine, which can lead to untimely decay that can destroy a large portion of the harvest. To judge various stages and harvesting periods of tomatoes, tomatoes are divided into four categories according to harvestability as defined as follows:

- Immature (completely green).
- Breaker (green to tannish).
- Preharvest (surface is light red).
- Harvest (fully colored tomato).

To accurately predict the level of tomato maturity in the bounding boxes obtained from the Mask R-CNN, tomato ripening classification is mainly completed through the following steps.

Step 1: Convert RGB images to HSV color space to accurately identify the color features of the tomato.

Step 2: Determine the color channels of red, orange, yellow, and green from HSV to extract and analyze the maturity of varied tomato colors. The HSV color range distribution is shown in Table 1.

TABLE 1. HSV Color Space Distribution Range.

	Red		Orange	Yellow	Green
H_{min}	0	156	11	26	35
H_{max}	10	180	25	34	77
S_{min}	43		43	43	43
S_{max}	255		255	255	255
V_{min}	46		46	46	46
V_{max}	255		255	255	255

Step 3: Calculate the pixel ratio distribution for color channel intensity based on HSV color range from the original tomato.

Step 4: Implement the fuzzy inference rule to designate different levels of tomato maturity.

A classical flowchart of tomato maturity classification is presented in Fig. 4.

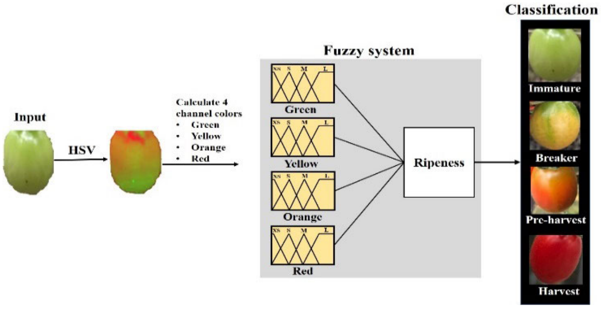


FIGURE 4. Classification of tomato maturity.

The established fuzzy system includes four membership function inputs to represent ratios of red, orange, yellow, and green channels from the HSV color space; these map to an output classifying various levels of tomato maturity. From the color intensity of the tomato surface, each attribute function has four semantic linguistic variables—XS, S, M, L—that represent extra small, small, medium, and large, respectively.

Fuzzy inference rules were automatically generated from the derived color ratio of HSV to mark the best relationship between the color intensity distribution present on the surface of the tomato and the output variables levels of tomato maturity. Since there are four input variables and each has four linguistic terms, this may come up to $4^4 = 256$ fuzzy rules in total. To quantify the maturity level of tomatoes, only 19 fuzzy rules were considered based on the real color intensity distribution of the tomato by omitting the others that have minimal distribution of colors in prediction of tomato ripeness. Some of the linguistic hedges are defined as follows:

Rule 1: If (Green is L) and (Orange is XS) and (Red is XS) and (Yellow is XS) then (classification is Immature).

Rule 2: If (Green is L) and (Orange is XS) and (Red is XS) and (Yellow is S) then (classification is Immature).

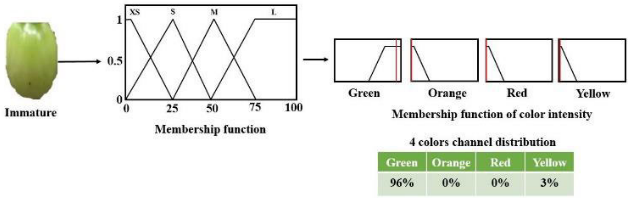
Rule 3: If (Green is S) and (Orange is M) and (Red is XS) and (Yellow is S) then (classification is Breaker)....

Rule 17: If (Green is XS) and (Orange is S) and (Red is L) and (Yellow is XS) then (classification is Preharvest).

Rule 18: If (Green is XS) and (Orange is XS) and (Red is L) and (Yellow is XS) then (classification is Harvest).

Rule 19: If (Green is XS) and (Orange is XS) and (Red is M) and (Yellow is XS) then (classification is Harvest).

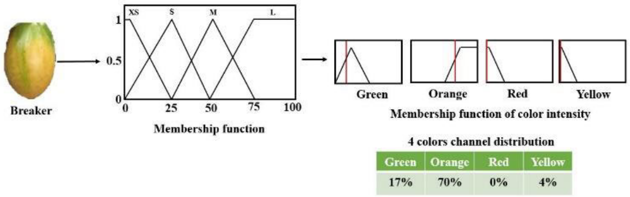
The membership functions of different color intensity distributions for the four relevant tomato maturity stages are illustrated in Fig. 5 to Fig. 8.



4 colors channel distribution

Green	Orange	Red	Yellow
96%	0%	0%	3%

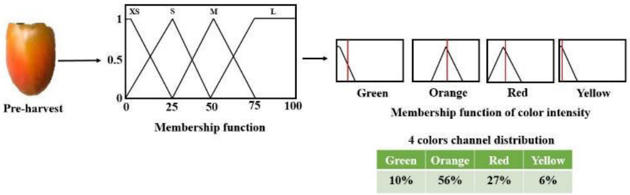
FIGURE 5. Membership function of Immature tomato.



4 colors channel distribution

Green	Orange	Red	Yellow
17%	70%	0%	4%

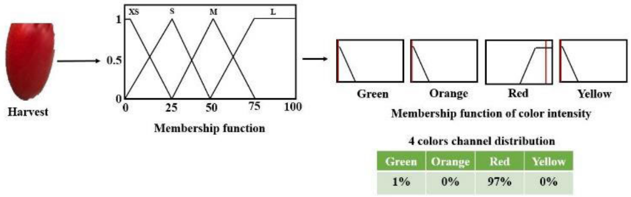
FIGURE 6. Membership function of Breaker tomato.



4 colors channel distribution

Green	Orange	Red	Yellow
10%	56%	27%	6%

FIGURE 7. Membership function of Preharvest tomato.



4 colors channel distribution

Green	Orange	Red	Yellow
1%	0%	97%	0%

FIGURE 8. Membership function of Harvest tomato.

F. LOCALIZATION OF TOMATO STALK POINT

The localization of tomato stalk points for harvesting was based on instance segmentation. After obtaining the contour of any tomato from Mask R-CNN, the system can determine the ripeness of that tomato. The coordinate values of tomato picking points were calculated using elliptical long- and short-axis information to obtain the right position of the pedicle. The localization steps for tomato stalk points are as follows:

Step 1: Locate the long-axis position from the tomato contour interest points and record the long-axis coordinates of those two points as (x_m, y_m), (x_d, y_d).

Step 2: Calculate the long-axis point using (17).

$$\text{Red\_d} = \sqrt{(x_m - x_d)^2} + \sqrt{(y_m - y_d)^2} \tag{17}$$

where *Red_d* is the Euclidian distance of long-axis.

Step 3: Locate the intersection of the tomato stalk from (x_m, y_m) to the pedicle (x_t, y_t), as illustrated in Fig. 9, and cut off the tomato stalk at 1/5 of the length of the tomato (i.e., approximately 1 cm, independent of the size of tomatoes based on our measurement on more than 100 tomatoes) away from (x_m, y_m), as in (18).

$$\text{Green\_d} = \text{Red\_d}/5 \tag{18}$$

where *Green_d* is the Euclidian distance from (x_m, y_m) to pedicle (x_t, y_t).

Step 4: Calculate the θ angle to obtain the cutting position of the pedicle head (x_t, y_t), as in (19) to (21).

$$\cos\theta = (y_m - y_d)/\text{Red\_d} \tag{19}$$

$$\cos\theta = |(y_t - y_m)|/\text{Green\_d} \rightarrow y_t \tag{20}$$

$$\sin\theta = |(x_t - x_m)|/\text{Green\_d} \rightarrow x_t \tag{21}$$

Fig. 9 illustrates the localization of tomato stalk points.

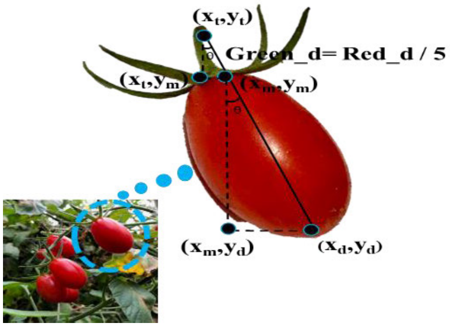


FIGURE 9. Schematic of tomato pedicle estimation.

IV. RESULTS

The results were obtained from the models using a set of 20 samples with 100 tomato images that had been excluded from the training set; they are presented in Table 2.

The approach system was executed with TensorFlow on a GPU workstation with an Intel Xeon 8 CPU, 32 GB of memory, and an Nvidia GeForce 11 GB GTX 1080 Ti graphics card.

TABLE 2. Accuracy of tomato detection.

Sample	No. of tomatoes	Correctly classified	Misclassified	Accuracy (%)
1	6	6	0	100.00
2	1	1	0	100.00
3	6	6	0	100.00
4	6	5	1	83.33
5	6	6	0	100.00
6	3	3	0	100.00
7	2	2	0	100.00
8	7	7	0	100.00
9	4	4	0	100.00
10	3	3	0	100.00
11	12	11	1	91.66
12	6	6	0	100.00
13	7	7	0	100.00
14	5	5	0	100.00
15	3	3	0	100.00
16	4	4	0	100.00
17	4	4	0	100.00
18	6	6	0	100.00
19	4	4	0	100.00
20	5	5	0	100.00
Total	100	98	2	98.00

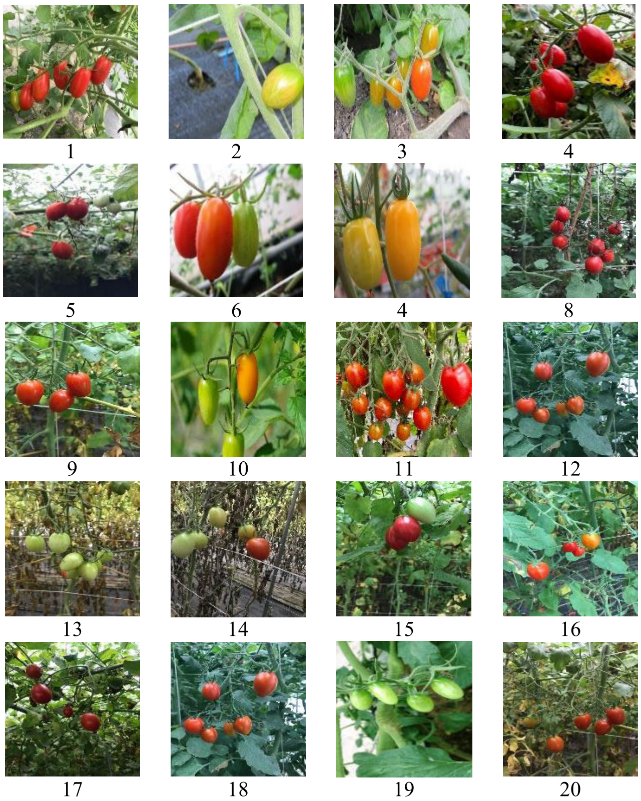


FIGURE 10. Original tomato test samples.

A. MASK R-CNN ACCURACY

The final task of the research was to discover the position of the pedicle, and thus, it was vital to locate the tomato accurately. To evaluate and localize the tomato with the trained Mask R-CNN, accuracy matrices were implemented, where accuracy means the ratio between the number of correct predictions of tomatoes and the total number of predictions. Mathematically, accuracy is defined as

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{22}$$

where TP, FP, TN, and FN signify true positive, false positive, true negative, and false negative rates, respectively. Fig. 10 presents original images from the 20 test samples. After localization of the tomato with Mask R-CNN, the target tomatoes are again presented in their original pixel display, and the background of the tomato image is rendered in grayscale as presented in Fig. 11. Testing for 20 test samples with 100 tomatoes yielded an average accuracy rate of 98.00%. Table 2 lists the numbers of correctly and misclassified tomatoes.

B. RIPENESS PERFORMANCE EVALUATION

After the recognition of tomatoes by Mask R-CNN, 98 tomatoes were classified into the four ripeness stages: immature, breaker, preharvest, and harvest. The classification process comprised color feature representation from RGB to HSV and fuzzy inference categorization. Furthermore, to delineate the model performance levels, we evaluated the confusion matrix, precision, recall, weighted precision, and weighted recall.

Precision quantifies the probability that the tomato maturity class retrievals reflect the tomato maturity stages. Recall quantifies the proportion of all positive categories of tomato maturity that are correctly recognized as tomato maturity stages. Precision and recall are calculated with (23) and (24).

$$\text{Precision} = \frac{TP}{(TP + FP)} \tag{23}$$

$$\text{Recall} = \frac{TP}{(TP + FN)} \tag{24}$$

where true positives (TP) are cases in which the model correctly predicted particular stages of tomato maturity correctly, true negatives (TN) are cases in which the model correctly predicted the tomato that does not belong to particular maturity stages, false positives (FP) are cases in which the model predicted particular maturity stages but the tomato did not actually belong to those model maturity stages, and false negatives (FN) are cases in which the model made no prediction of tomato maturity stage but the tomato actually belonged to some maturity stage. Regarding the number of correctly classified samples in each class, we calculated weights for the original values of tomato maturity, weighted precision, and recall. We used (25) and (26):

$$\text{Weighted} - \text{precision} = \frac{\sum_1^n W_n \times P_n}{n} \tag{25}$$

$$\text{Weighted} - \text{recall} = \frac{\sum_1^n W_n \times R_n}{n} \tag{26}$$

where n denotes the tomato maturity stages, W_n represents the proportion of the number of tomatoes of the nth class to the total number tomato images, and P_n and R_n are the

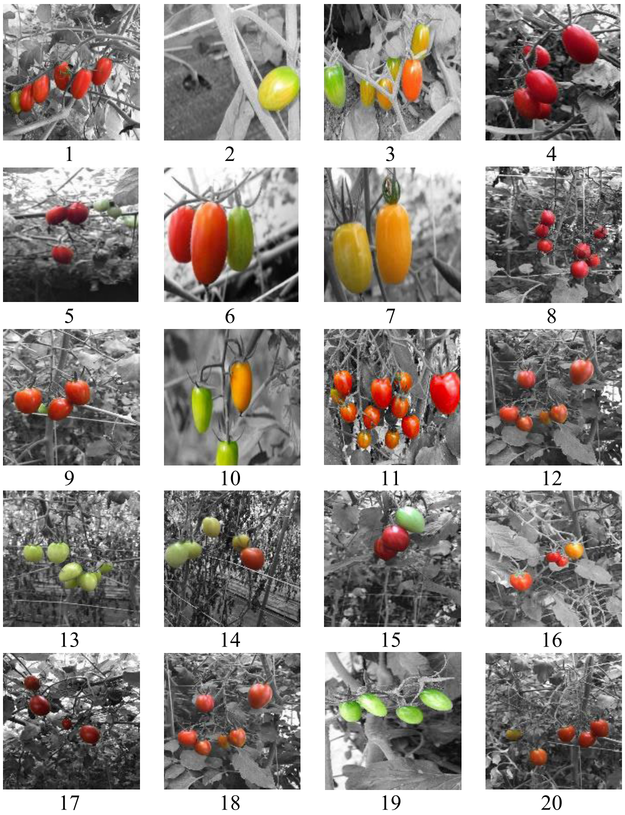


FIGURE 11. Target tomatoes restored to the original pixel.

precision and recall values of the *n*th class of tomato maturity, respectively. Table 3 presents the confusion matrix visualization of the algorithm. Each row of this confusion matrix denotes the instances in a predicted class, whereas each column signifies the instances in an actual class. The immature maturity stage generated the highest value of precision, and the breaker maturity stage obtained the highest value of recall. Overall weighted precision and weighted recall values were 0.9614 and 0.9591, respectively. The results are shown in Table 4.

TABLE 3. Confusion matrix of tomato ripeness.

Actual	Prediction			
	Immature	Breaker	Preharvest	Harvest
Immature	24	1	0	0
Breaker	0	8	0	0
Preharvest	0	0	13	1
Harvest	0	0	2	49

Varying stages of tomato maturity classification (immature, breaker, preharvest, and harvest), as predicted by the proposed model, are illustrated in Fig. 12.

C. DISCUSSION

Many fruit detection and recognition approaches have been based on multiple features, such as color [23], [24], shape [25], texture [26], edge, and orientation [27], [28]. Color image segmentation [29] is based on intensities of image pixels; such segmentation can separate similar fruits from the background according to some homogeneity of color features in the image. This technique identifies the range of color intensities based on a color threshold. The target objects that lie outside the predefined range eliminate the unwanted pixels of the image. Region-based extraction of the geometric features of fruits, (including edge contour and whole region features) can be executed with the strength of neighboring pixels that have similar values in a specified local region of interest. However, in some instances, these approaches fail to identify specific fruits or locate them correctly, especially in the presence of varying illumination, overlapping fruits, or occlusion of leaves and branches.

TABLE 4. Performance evaluation of tomato detection.

Class	Precision	Recall	Weighted precision	Weighted recall
Immature	1.0000	0.9600	--	--
Breaker	0.8888	1.0000	--	--
Preharvest	0.8666	0.9285	--	--
Harvest	0.9800	0.9607	--	--
Overall	--	--	0.9614	0.9591

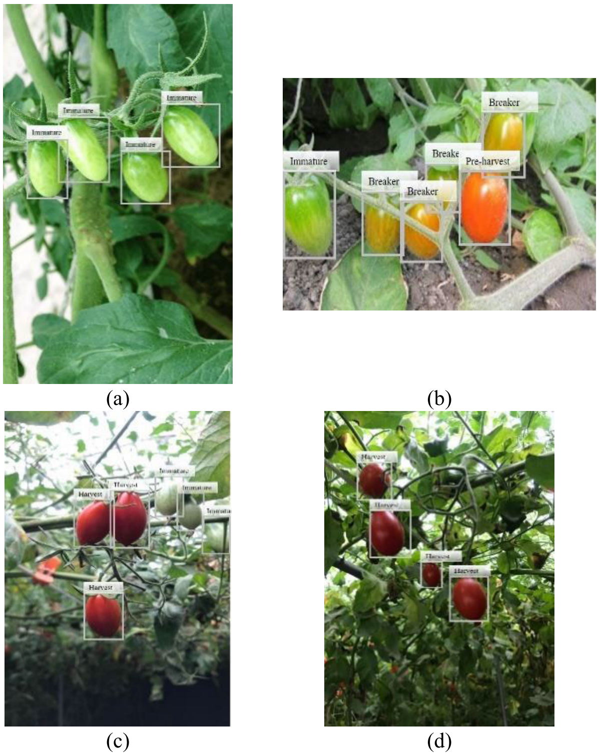


FIGURE 12. Visualization of various tomato ripening stages as detectable in the test data set. (a) Immature tomato stage, (b) immature, breaker, and preharvest tomato stages, (c) harvest and immature tomato stages, and (d) harvest tomato stage.

To increase robustness and meet the needs of practical applications in tomato detection, we adopted a fuzzy approach with Mask R-CNN [7] to combine information regarding multiple pixel features, such as the integrity of color intensity, shape detection, edge orientation, and contour segmentation. Subsequently, the system was able to detect target tomatoes in real-world environments subject to challenges such as multiple overlapping fruits, occlusion of leaves or stems, and shaded tomatoes of uneven illumination.

The proposed model was tested on 20 samples with 100 tomatoes. The proposed method efficiently detected the fruits that had varied intensity, color, edges, and orientation. It was also able to identify plants that contained dense regions and plants that had one or more fruit regions that were different from the background, as presented in Fig. 13. Input images are presented in Fig. 13(a), (c), (e), and (g), whereas the detected tomatoes are depicted in Fig. 13(b), (d), (f), and (h). The proposed model is robust for different environmental conditions and unstructured scenes such as overlapping, as depicted in the Fig. 13(a) and (c). Occlusion, inadequate illumination, and shading conditions are shown in Fig. 13(e) and (g). However, in Fig. 13(a), the model efficiently identified a group of overlapping tomatoes (encircled with blue dots) and two separate tomatoes (encircled with pink dots) but failed to detect some fruits (encircled with yellow dots) because more than 50% of the fruit was concealed by stems.

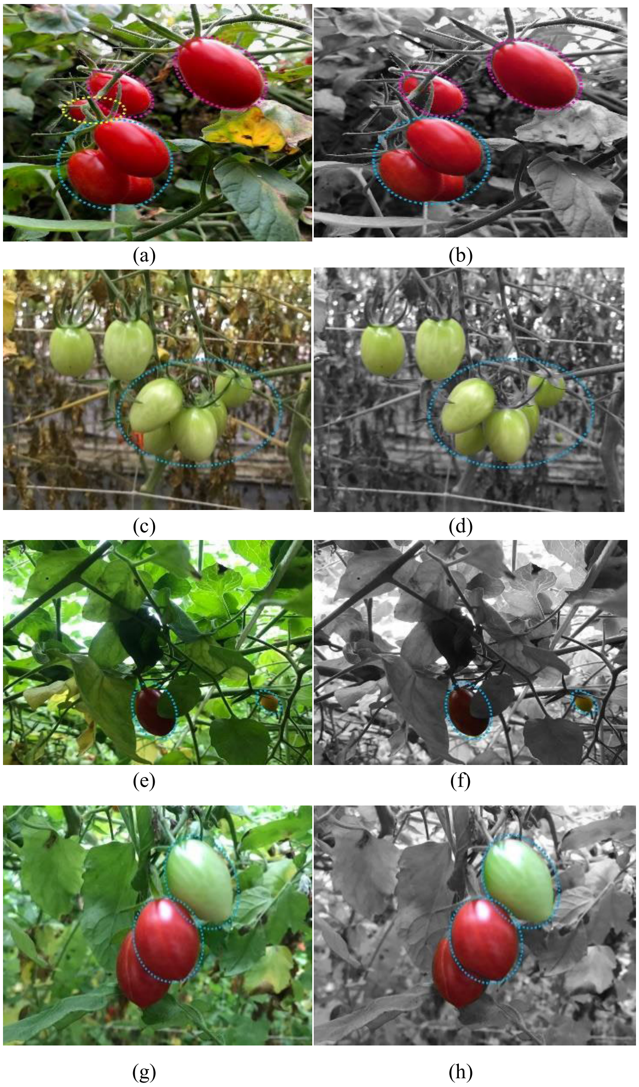

(a) (b) (c) (d) (e) (f) (g) (h)

FIGURE 13. (a) Overlapping and vine occlusion, (b) detection of overlapping tomatoes but failure to detect vine occlusion (which covered more than 50% of the fruit region), (c) overlapping and immature tomatoes, (d) detected overlapping and immature tomatoes, (e) shaded and occluded by leaf, (f) successful detection despite shade, occlusion by leaves, and different stages of maturity, (g) sunlight variation condition, and (h) detection under varied illumination.

To validate the proposed model in real time application, we further compared metrics from tomato detection and ripeness detection methods with other exiting approaches. Table 5 shows the performance comparison with other existing methods. Fig. 14 and Fig. 15 show the histograms of performance comparison for tomato detection and ripeness detection, respectively.

TABLE 5. Performance comparison of different existing methods.

Existing approach	Tomato detection			Tomato ripeness detection		
	P (%)	R (%)	MC (%)	P (%)	R (%)	MC (%)
YOLO v3 [30]	93.09	94.75	5.25	-	-	-
SVM+HoG [7]	-	-	-	90.00	94.41	5.59
RGB+ML [31]	88.00	80.00	20.00	93.00	100.00	0.00
SVM+CIE Lab [32]	-	-	-	83.11	83.06	16.94
RGB+GLCM [33]	-	-	-	93.40	100	0.00
Our model	100.00	98.00	2.00	96.14	95.91	4.09

**P: precision; R: recall, MC: Misclassified; YOLOv3: You look only once, version 3; SVM: support vector machine; HoG: histogram of oriented gradients; RGB: red, green and blue; CIELab: a device-independent color space defined by the International Commission on Illumination (CIE); GLCM: gray level co-occurrence matrix.

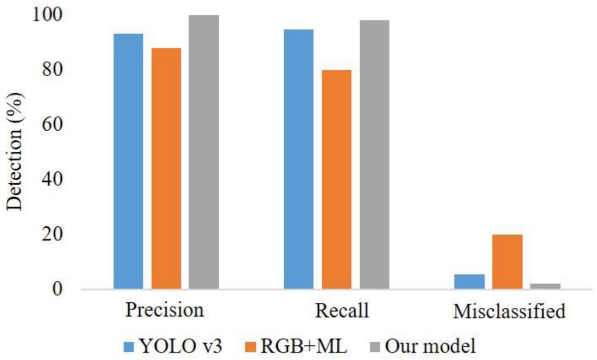


FIGURE 14. Histograms of performance comparison. (a) tomato detection, and (b) tomato ripeness detection.

Overall, our proposed system shows quite efficient performance with respect to tomato detection and tomato ripeness detection. In the tomato detection, our system achieved 100% precision and 98.00% recall. In tomato ripeness detection, our model achieved 96.14% precision and 95.91% recall. Based on the result obtained, our proposed model indicates superiority of other existing proposed method. Thus, it can be applied in real practical applications efficiently.

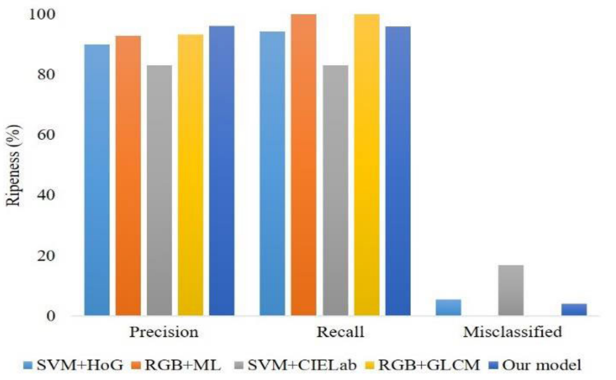


FIGURE 15. **Histograms of performance comparison for tomato ripeness detection.**

V. CONCLUSION

In this article, we present a method based on deep learning and fuzzy systems for automatic identification of the maturity stages of tomatoes. Our system can locate the right picking point of the tomato stalk so that each tomato can be harvested at the right time. The proposed approach includes four main phases: first, automatic annotation of tomatoes is performed through the combination of FCM and HT in a manner that maintains the spatial information of the image and locates the specific geometric edges positions of the tomatoes, followed by annotation of each data point of the image space to a JavaScript Object Notation file. Second, localization of tomatoes is conducted using Mask R-CNN in which the localization of a tomato can be automatically mapped back onto the corresponding positions in the original tomato images to attain target-area masks of the tomato. Third, color feature representation from RGB to HSV (which maintains color integrity of the tomato surface colors and prevents minimum loss of color information) followed by the fusion of color scores with fuzzy rules to detect different maturity stages of the tomatoes is performed. Finally, pedicle harvesting points are localized on the mask image output from Mask R-CNN so that the mature tomatoes can be harvested quickly and conveniently.

The analysis of 100 test images showed that the recognition accuracy rate, weighted precision, and weighted recall were 98.00%, 0.9614, and 0.9591, respectively. These detection results prove the system can feasibly recognize ripeness, ensuring that each tomato can be harvested at the right time.

Future work will focus on early detection and identification of tomato diseases on the basis of deep learning and object detection approaches. If such systems can be delivered, then tomato plants can deliver higher yields, better quality of sustainable agricultural production, and greater safety for human health.

REFERENCES

[1] Background. (Jun. 2020). *The Global Tomato Processing Industry*. [Online]. Available: http://www.tomatonews.com/en/background_47.html

[2] *Global Tomato Production in 2012*, FAO, Rome, Italy, Nov. 2014.

[3] S. Grandillo, D. Zamir, and S. D. Tanksley, "Genetic improvement of processing tomatoes: A 20 years perspective," *Euphytica*, vol. 110, no. 2, pp. 85–97, Apr. 1999.

[4] 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.

[5] Y. Zhao, L. Gong, Y. Huang, and C. Liu, "Robust tomato recognition for robotic harvesting using feature images fusion," *Sensors*, vol. 16, no. 2, pp. 1–12, Jan. 2016.

[6] 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.

[7] G. Liu, S. Mao, and J. H. Kim, "A mature-tomato detection algorithm using machine learning and color analysis," *Sensors*, vol. 19, no. 9, pp. 1–19, May 2019.

[8] I. Nyalala, C. Okinda, L. Nyalala, N. Makange, Q. Chao, L. Chao, K. Yousaf, and K. Chen, "Tomato volume and mass estimation using computer vision and machine learning algorithms: Cherry tomato model," *J. Food Eng.*, vol. 263, pp. 288–298, Dec. 2019.

[9] T. Yuan, L. Lv, F. Zhang, J. Fu, J. Gao, J. Zhang, W. Li, C. Zhang, and W. Zhang, "Robust cherry tomatoes detection algorithm in greenhouse scene based on SSD," *Agricult.*, vol. 10, no. 5, pp. 1–14, May 2020.

[10] C. Hu, X. Liu, Z. Pan, and P. Li, "Automatic detection of single ripe tomato on plant combining faster R-CNN and intuitionistic fuzzy set," *IEEE Access*, vol. 7, pp. 154683–154696, Oct. 2019.

[11] 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, pp. 1–22, Feb. 2019.

[12] J. C. Dunn, "A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters," *J. Cybern.*, vol. 3, no. 3, pp. 32–57, Sep. 1973.

[13] J. C. Bezdek, "Pattern recognition with fuzzy objective function algorithms," in *Advanced Applications in Pattern Recognition*. New York, NY, USA: Plenum Press, Jul. 1981.

[14] D. H. Ballard, "Generalizing the Hough transform to detect arbitrary shapes," *Pattern Recognit.*, vol. 13, no. 2, pp. 111–122, Jan. 1981.

[15] J. Illingworth and J. Kittler, "A survey of the Hough transform," *Comput. Vis., Graph., Image Process.*, vol. 44, no. 1, pp. 87–116, 1988.

[16] O. F. Tuna. (Jan. 2019). *Detecting Number of Circles in a Binary Input Image*. [Online]. Available: https://medium.com/merfaruktuna/detecting-number-of-circles-in-a-binary-input-image-d1163ba6d57f

[17] A. Y. S. Chia, M. K. H. Leung, H.-L. Eng, and S. Rahardja, "Ellipse detection with Hough transform in one dimensional parametric space," in *Proc. IEEE Int. Conf. Image Process.*, Nov. 2007, pp. 333–336.

[18] Y. Xie and Q. Ji, "A new efficient ellipse detection method," in *Proc. Object Recognit. supported Interact. Service Robots*, Quebec City, QC, Canada, 2002, pp. 957–960.

[19] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Venice, Italy, Oct. 2017, pp. 2961–2969.

[20] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask R-CNN," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 2, pp. 386–397, Feb. 2020.

[21] L. Zhang, J. Jia, G. Gui, X. Hao, W. Gao, and M. Wang, "Deep learning based improved classification system for designing tomato harvesting robot," *IEEE Access*, vol. 6, pp. 67940–67950, 2018.

[22] Y. Yu, K. Zhang, L. Yang, and D. Zhan, "Fruit detection for straw berry harvesting robot in non-structural environment based on mask-RCNN," *Comput. Electron. Agricult.*, vol. 163, pp. 1–9, Aug. 2019.

[23] W. Castro, J. Oblitas, M. De-La-Torre, C. Cotrina, K. Bazan, and H. Avila-George, "Classification of cape gooseberry fruit according to its level of ripeness using machine learning techniques and different color spaces," *IEEE Access*, vol. 7, pp. 27389–27400, 2019.

[24] J. A. Cortes-Osorio, J. B. Gomez-Mendoza, and J. C. Riano-Rojas, "Velocity estimation from a single linear motion blurred image using discrete cosine transform," *IEEE Trans. Instrum. Meas.*, vol. 68, no. 10, pp. 4038–4050, Oct. 2019.

[25] X. Liu, D. Zhao, W. Jia, W. Ji, and Y. Sun, "A detection method for apple fruits based on color and shape features," *IEEE Access*, vol. 7, pp. 67923–67933, 2019.

[26] W. Zhang, J. Hu, G. Zhou, and M. He, "Detection of apple defects based on the FCM-NPGA and a multivariate image analysis," *IEEE Access*, vol. 8, pp. 38833–38845, 2020.

[27] F. Garcia, J. Cervantes, A. Lopez, and M. Alvarado, "Fruit classification by extracting color chromaticity, shape and texture features: Towards an application for supermarkets," *IEEE Latin Amer. Trans.*, vol. 14, no. 7, pp. 3434–3443, Jul. 2016.

[28] C. Wang, T. Luo, L. Zhao, Y. Tang, and X. Zou, "Window zooming–based localization algorithm of fruit and vegetable for harvesting robot," *IEEE Access*, vol. 7, pp. 103639–103649, 2019.

[29] X. Liu, D. Zhao, W. Jia, W. Ji, C. Ruan, and Y. Sun, "Cucumber fruits detection in greenhouses based on instance segmentation," *IEEE Access*, vol. 7, pp. 139635–139642, 2019.

[30] G. Liu, J. C. Nouaze, P. L. T. Mbouembe, and J. H. Kim, "YOLO-tomato: A robust algorithm for tomato detection based on YOLOV3," *Sensors*, vol. 20, no. 7, pp. 1–20, Apr. 2020.

[31] 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.

[32] M. B. Garcia, S. Ambat, and R. T. Adao, "Tomayto, tomahto: A machine learning approach for tomato ripening stage identification using pixel-based color image classification," in *Proc. IEEE 11th Int. Conf. Humanoid, Nanotechnol., Inf. Technol., Commun. Control, Environ., Manage. (HNICEM)*, Laoag, PA, USA, Nov. 2019, pp. 1–6.

[33] M. P. Arakeri and Lakshmana, "Computer vision based fruit grading system for quality evaluation of tomato in agriculture industry," *Procedia Comput. Sci.*, vol. 79, pp. 426–433, 2016.

YO-PING HUANG (Fellow, IEEE) received the Ph.D. degree in electrical engineering from Texas Tech University, Lubbock, TX, USA. He was a Professor and the Dean of the Research and Development, the Dean of the College of Electrical Engineering and Computer Science, and the Department Chair with Tatung University, Taipei. He is currently a Chair Professor with the Department of Electrical Engineering, National Taipei University of Technology, Taipei, Taiwan, where he served as the Secretary General. His current research interests include fuzzy systems design and modeling, deep learning modeling, intelligent control, medical data mining, and rehabilitation systems design. He is an IET Fellow, CACS Fellow, and an International Association of Grey System and Uncertain Analysis Fellow. He was the President of the Taiwan Association of Systems Science and Engineering, the Chair of the IEEE SMCS Taipei Chapter, the Chair of the IEEE CIS Taipei Chapter, and the CEO of the Joint Commission of Technological and Vocational College Admission Committee, Taiwan. He serves as the IEEE SMCS BoG, the Chair of the IEEE SMCS Technical Committee on Intelligent Transportation Systems, and the Chair of the Taiwan SIGSPATIAL ACM Chapter.

TZU-HAO WANG (Graduate Student Member, IEEE) received the M.Sc. degree from the Department of Electrical Engineering, National Taipei University of Technology, Taipei, Taiwan, in 2018, where he is currently pursuing the Ph.D. degree in electrical engineering. His current research interests include AOI panel defect detection and inspection, the Internet of Things (IoT), deep learning, and image processing.

HAOBIJAM BASANTA received the master's degree in computer application (MCA) from the University of Jamia Millia Islamia, Delhi, India, and the Ph.D. degree in electrical engineering and computer science from the National Taipei University of Technology, Taipei, Taiwan. He currently holds a postdoctoral position with the National Taipei University of Technology. His current research interests include the Internet of Things (IoT) for elderly healthcare systems, big data analytics, deep learning, and image processing.