CAMES, **29**(1–2): 161–175, 2022, doi: 10.24423/cames.419

This article belongs to the *Special Issue on Advanced Optimization Methods for Uncertainties in Intelligent Industrial Systems* edited by Dr. I.J. Jacob, Dr. B.H. bin Ahmad and Dr. Z.F. Khan

Intelligent Agrobots for Crop Yield Estimation using Computer Vision

D. BINI[1),2)]*, D. PAMELA[3)], Thusnavis Bella MARY I.[4)], D. SHAMIA[5)], Shajin PRINCE[4)]

[1)] *Department of Electronics and Instrumentation Engineering*
[2)] *Department of Robotics Engineering*
[3)] *Department of Biomedical Engineering*
[4)] *Department of Electronics and Communication Engineering*
Karunya Institute of Technology and Sciences
Coimbatore, India
*Corresponding Author e-mail: binivlsies@gmail.com

[5)] *Department of Electronics and Communication Engineering*
V.S.B. College of Engineering Technical Campus
Coimbatore, India

The machine vision-based autonomous intelligent robots perform precise farm tasks such as robot harvesting, weeding, pest or fertilizer spraying, monitoring, and pruning. Estimating crop yield is an essential assignment on a regional or federal scale. For a long time the estimation measures were based on the statistics from manual counting of plants in a specific zone. The computer vision algorithms have addressed the technical drawbacks of the conventional image processing techniques and established an autonomous discipline and yielded new approaches to crop planning. A method for quantitative assessment of a tomato crop has been developed in this research using color thresholding in MATLAB using the RGB color model. Converting an RGB image to a grayscale image is one of the steps involved in detecting red color in a taken image. After subtracting the two images, a median filter is employed to filter the noisy pixels to produce a two-dimensional black and white image. The bounding boxes are used to label the binary digital images to detect related components, and the parameters of the labeled regions are computed to measure the number of tomatoes in a crop. The obtained R^2 correlation coefficient between the tomato berry counting algorithm and human counting was 0.98. Furthermore, the color of each pixel in the acquired image is evaluated by examining RGB values for pixel intensities in the obtained image. The performance of the berry counting algorithm was evaluated, and the technique was determined to have a high precision and recognition ratio of 96%. The research indicates that this technique may be used to estimate the crop yield, which is helpful information for forecasting yields, planning harvest plans, and generating prescription maps for field-specific management strategies. The proposed model performed exceptionally well in estimating yield with each tomato (Solanum lycopersicum) crop.

Keywords: computer vision, real-time images, autonomous agricultural robots, digital image processing, object detection, crop yield estimation.

1. Introduction

Precision agriculture necessitates constant innovation to improve the quality and quantity of crop yield production. The detection and counting of fruitlets at various budding phases are enforced for yield management such as harvesting, and transporting fruit load in connection with import and export decisions. Agronomy helps nearly 58% of the Indian population. In the state of Tamil Nadu, almost 63 lakh ha area has been used for plantation beyond 1.3 lakh sq km from its all-inclusive area. The conventional method of crop yield estimation otherwise known as crop cutting experiments is time-consuming, inaccurate and labor-intensive.

Computer vision-based berry detection and counting algorithms for crop yield estimation and robot harvesting applications emerge in shipping stations. The tomato plants are sowed in rows with a line spacing of 24–36 inches and are mostly vines. The bloom to fruit-set period takes around 60–80 days from the day of sowing. The crop yield estimation is required at the fruit-set stage before harvesting ripened tomato berries. The human-based estimations are inaccurate and unfeasible. The machine vision approaches perpetuate an automatic analysis and digital images to guide and accelerate the agrobots to solve various agricultural tasks [1].

The detection of fruit areas from an image and removal of background pixels in various color models such as grayscale, CIELAB, red-green-blue (RGB) and HSV was determined with diverse image processing techniques. The detection of apple fruit to estimate the fruit yield was based on the R-B and G-R color features [2, 3]. A machine vision-based algorithm was developed to detect the immature citrus considering the features such as contour, color and texture. The fast normalized cross correlation (FNCC) was used to detect the fruit regions and the circular hough transform (CHT) detected circles from image after background removal based on color features [4]. An automatic red apple detection algorithm was designed for robot harvesting based on the histogram equalization applied to diverse color models [5, 6].

Several studies addressed the fruit detection algorithm based on multi-scale feature extraction and classification techniques. Diverse conventional image processing approaches and machine learning and deep learning models have emerged for various agricultural applications. A pixel-based image segmentation with processing techniques such as linear color space segmentation, threshold-based segmentation, Mahalanobis distance, histogram segmentation and Bayesian classifier were used to detect the reddish grapes in the vineyard by [7].

A vision-based automaton system, in accordance with cognitive computing, enables autonomous ground vehicles and unmanned aerial vehicles to commute the pathway and carry out farming activities in order to reduce labour and increase food quality [8, 9]. Compared to the method of manual berry counting, the machine vision-based technology could aid farm owners in investigating vast tomato fields more quickly and efficiently. The goal of this work is to quantify ripened tomato berries in nature-based RGB images.

In order to detect the berry in an RGB color space, a methodology is formulated with a different combination of image segmentation techniques for computer vision. The phenology of tomato proceeds from a stage of immature green tomato to a state of tomato of amber color and into a red ripened berry ready to be harvested. The performance evaluation for a combination of diverse image segmentation techniques is analyzed to detect the fruit pixel in a real-time image.

Agricultural acreages, as well as sample assessments based on crop cutting experiments, are involved in the traditional method for crop yield estimation. Crop surveys are still accompanied by small zone crop data from a specific location, necessitating a larger number of crop cutting trials due to the high costs involved. Crop cutting experiments are carried out by government personnel in various regions and districts of the state to kick-start national agricultural programs. The current industrial yield estimation methods rely on automated computer vision technologies [10] to detect and estimate the number of harvests. An automation methodology for agricultural yield estimation techniques using virtual analysis is discussed in this research. Farmers can use agricultural sensors to gather data and the meteorological conditions in their agricultural areas [11].

Conventional machine vision approaches have advanced with constraints that must be investigated in order to improve automated precision farming, particularly during the current COVID-19 pandemic. These approaches execute a detection algorithm [12] in each frame in real time and then associate the results across frames. Multiple views of the same fruit can be used in our scenario. We demonstrate a statistical method for combining data from several viewpoints in order to increase the fruit detection rate.

The berry is identified in a tomato field by a combination of the following characteristics: (i) ovoid shape (or curving edge, if part of the fruit is concealed), (ii) smooth surface, and (iii) skin color (reddish blush, if present, or a green paler than leaves). However, because the tomato berry is shaped like a basic sphere or ovoid, the 2D projection of the fruit in a camera image can take on a variety of shapes, and the fruit's skin color can change, as previously stated. As a result, the imagery processing challenge of recognizing tomato berries within a canopy image is more complex than, say, recognizing ripened berries.

The focus of this research is to use image processing to provide a strategy for counting the number of berries in an RGB image. A machine vision technique for detecting tomato berries in images of the entire crop field was based on image segmentation based on color and texture, conducted in previous work. For the field covered in this research, additional work was performed to examine the association between machine count and manual count.

2. Materials and methods

The objective of this paper is to develop and review a system for tomato yield estimation prior to the harvest. A Canon EOS 1500D Sony digital camera was used to capture the real-time images from the tomato fields. The hardware used includes Intel Core i5 Dual-core processor laptop, and NVIDIA GeForce GTX. Figure 1 depicts the overall framework employed in this research. The outline of the research is as follows. Subsection 2.1 emphasizes the materials and methods in which the agricultural tasks were performed. Subsection 2.2 discusses the mode of capture and data acquisition techniques. Subsection 2.3 deals with the image preprocessing techniques involved in enhancing the image and the image segmentation approaches for fruit and background separation. Section 3 elucidates the results and discussions of the algorithm used for crop yield estimation. Section 4 provides the conclusions and the future work connected with this research.

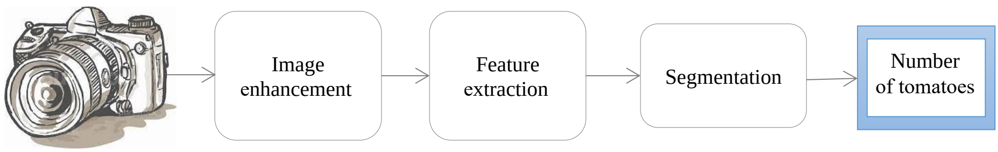

Fig. 1. Framework for the berry detection algorithm.

2.1. Proposed methodology

The estimation of tomato yield can be evaluated with the high-resolution RGB images. The real-time RGB images were collected from the nearby fields in Coimbatore, India during the months of September, October and November. The berry detection and counting can be based on the color and shape features. The image enhancement techniques remove the noise and improve the real-time image quality. The images were acquired such that no overlapping occurs with the sequential images. The images were resized with the bicubic method to ease the processing technique. The RGB components are obtained from the resized image to form a 2D matrix. Now, the RGB image is transformed into a grayscale image along with a weighted summation of R, G, B values.

The gray-level image is now transformed to a binary image, with the input image being assigned value 0 and the background pixels being assigned value 1. Using the blob counting method, the binary picture obtained is now used to count the quantity of berries. The fruit clusters may be clustered under the same region during binarization, resulting in inaccuracy in counting the number of fruits. Individual berries were quickly computed using the proposed method. By extracting the fruit with the Laplacian-Gaussian technique of segmentation, the regions with clusters of berries are enumerated. The image processing techniques are aimed at a picture that is followed by a count of the number of tomatoes. The estimation findings are kept in a database and displayed in a graphical user interface.

2.2. Image acquisition

The photos were taken in a tomato field planted in rows with an optimal spacing of 75×45 cm and a repeated temperature range of 16°C to 25°C. During the fruit-set time, the tomato plant was kept in direct sunshine so that dark red berries could form. Ripe berries are medium-sized, round-shaped, and reddish-colored. Direct sunshine, which produces saturated objects with no color information, and shadows, which lead traditional segmentation algorithms to break the tomato surfaces into multiple fragments, significantly impact detection feasibility.

Obtaining images in diffusive sunlight circumstances can help to reduce such undesired effects (for example, portraying on a partly overcast day). During the harvesting phase, a traditional digital camera with a high-level resolution of 24.2 MP was employed to capture photographs in natural daylight. Within the RGB color model, the photos were saved in JPG format with a high-level resolution of 2448×2048 pixels. The recorded images were scaled to 256×256 pixels for computational accessibility. The photos of tomato plants were taken in a certain order so that the samples did not overlap. Various image segmentation approaches were studied for image processing and counting the number of yields. The general framework employed in this paper is presented in Fig. 2. The flowchart shows how real-time images are acquired, scaled, and thus further enhancement and segmentation algorithms are applied.

2.3. Pre-processing of images

The most challenging and important task in image processing is segmentation, which involves extracting the desired items from the image. For the abstraction of an image's contours, regions, and boundaries, segmentation techniques are still available [13]. In digital image segmentation, binary or gray-level thresholding remains the most basic technique. Because the characteristics were unimodal, thresholding for the same shades of hue failed to separate leaf and citrus fruits from the backdrop [14]. Segmentation was used in this paper to distinguish the tomato berry from the tomato plant as well as the background. MATLAB R2020a software was used to progress the method. The amount of time

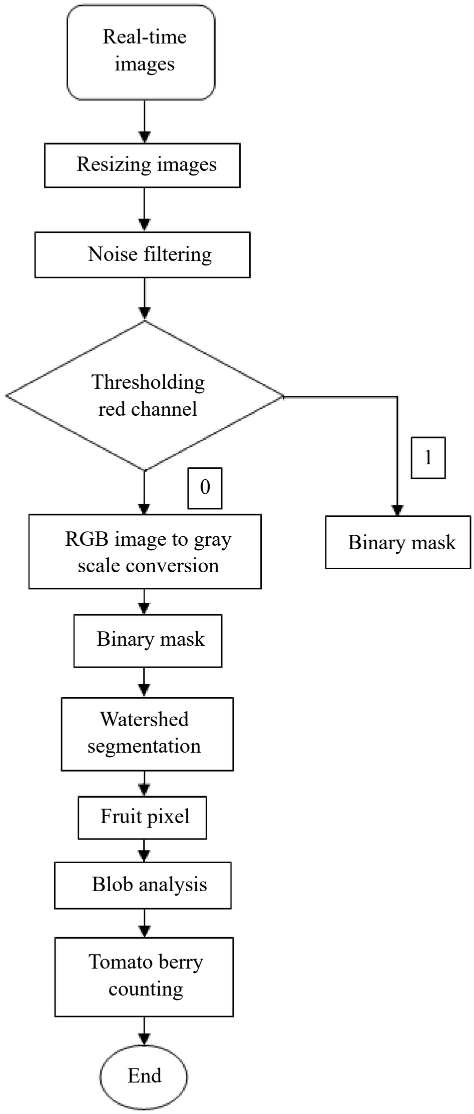

FIG. 2. Flowchart for the berry detection and counting.

it took to process the image was influenced by its size. The procedure was time-consuming because of the larger size of the acquired image, so the images were scaled before the segmentation procedure. Figure 3 depicts the RGB color model. The scaled RGB image is converted to a grayscale image, which necessitates noise filtering. Filtering techniques are used to highlight specific characteristics while also neutralizing noise [15]. Figures 4 and 5 show the input RGB image of a real-time tomato crop and resized input image, respectively. In the cap-

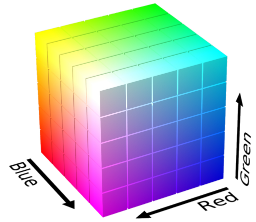

FIG. 3. RGB color model.

FIG. 4. Image of the tomato plant.

FIG. 5. Resized image.

tured images, the median filter efficiently retains edges and reduces impulsive noise [16]. The grayscale image's Gaussian white noise is counteracted by the Wiener filter (WF). The filter reduces the mean square inaccuracies between the desired and estimated random processes.

2.3.1. Split and convert RGB channel. By the use of methodical qualities, the ripe tomatoes in the real-time image may be separated from the background pixels. The features of the image are evaluated for each pixel in order to segment and classify it. The selection of pertinent pixels will aid in determining image features for color thresholding-based separation of the foreground and background layers. Each pixel in a multispectral image is assigned an R, G, and B magnitude within a given range. In a 3D cubical Cartesian coordinate system, the RGB color model is an additive color space of three fundamental chromaticities. Figure 6 illustrates the red, green, and blue channel splitting for the input images used in this article.

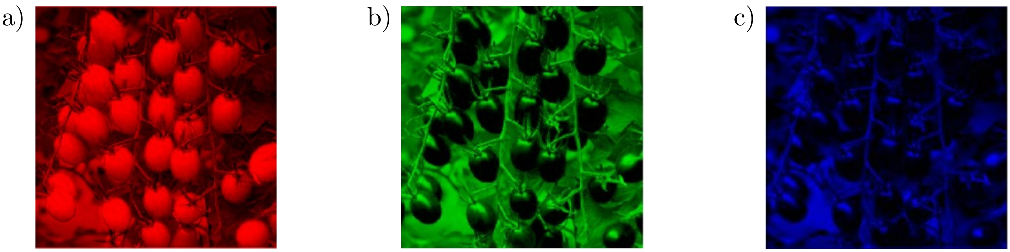

FIG. 6. Splitting of the RGB image (a) red channel, (b) green channel, (c) blue channel.

2.3.2. Red berry detection by thresholding. In RGB color space patterns, the berry and background pixels are examined for pixels inside the berry's range, with most other pixels being discarded as background. There were few obvious colour distinctions between the tomato, which was directly lighted by sunshine, and the sky, ground, as well as some foliage, which were likewise directly exposed to daylight (background). To determine the value of the threshold $t(x, y)$:

$$t(x,y) = \begin{cases} 0, & f(x,y) < T, \\ 1, & f(x,y) \geq T, \end{cases} \tag{1}$$

where $f(x, y)$ – pixel's initial value, and T – threshold value.

Since ripened tomato berries signify the red region, a red color threshold is set, and the berries are trimmed from the surrounding pixels. Green tomatoes, foliage, and twigs enclose the image's background. Consider that the gray-level histogram correlates to an image $f(x, y)$ consisting of dark objects against a light backdrop, with gray levels sorted into two dominating modes for both fruit and background pixels.

The red ripened tomatoes were retrieved from the RGB color model of the image using a threshold parameter, which removed the background as seen in relationships (1) and (2):

$$\mathrm{R} > \mathrm{G} \qquad \text{and} \qquad \mathrm{R} > \mathrm{B}. \tag{2}$$

Selecting a threshold T that divides these modes is one obvious technique to isolate the objects from the backdrop. Each pixel (x, y) in the image when $f(x, y) > T$ is referred to as an object point; otherwise, the point is referred to as a background point. With computer vision, thresholding is used to segment a digital image into smaller segments or bits, by defining their edges with at least one color or grayscale component [17]. In comparison to other picture segmentation algorithms, histogram-based methods are particularly efficient since they usually only require one pass. The peaks and troughs in the histogram are used to find clusters in the image in this technique, which involves computing a histogram for any pixels in an image. The technique used to convert RGB color images to grayscale is frequently thought to have very few impacts on detection accuracy in image processing. Figures 7a and 7b characterize a grayscale image with a histogram of intensity levels.

2.3.3. RGB to grayscale conversion. The technique used to convert RGB color images to grayscale is frequently thought to have very few impacts on detection accuracy in image processing. The image values are supposed to be in the range of 0 to 1. The R, G, and B define red, green, and blue linear channels,

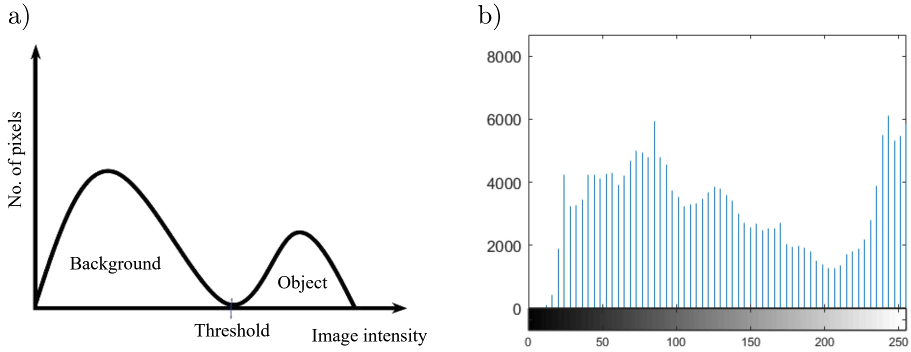

Fig. 7. a) Threshold selection in gray-level image, b) histogram of grayscale values.

respectively. Each grayscale method produces a result that ranges from 0 to 1. In general, there are various algorithms for converting color images to grayscale. To obtain a single gray value, one effective method is to aggregate the RGB pixel intensity:

$$\text{gray value} = \frac{1}{3}(\mathrm{R} + \mathrm{G} + \mathrm{B}). \tag{3}$$

A watershed algorithm was used to distinguish any linked tomato berry. The red color was altered to a gray color to achieve a good watershed performance. Equation (3) was employed to change the RGB image into a gray image in this research.

2.3.4. Noise removal. A few of the foliage and twigs were misdiagnosed, resulting in undesired noise. To reduce this undesirable noise, we used median filtering and the subsequent image processing techniques. In image recognition, median filtering is a nonlinear technique that is frequently used to minimize "salt and pepper" distortion [18]. So, when the goal is to minimize noise while preserving edges, a median filter is much more successful than convolution. With certain synthetic distortion, the tomato berries are separated from the background. Filtering techniques are employed to reduce noise and the erosion of obsolete pixels at the edges. The median filter examines each pixel in the image individually, comparing it to its neighbors to determine whether it is representative of its surroundings.

This median filter supplemented the pixel value with the median of the nearby pixel values rather than the mean of those values. The median was assessed by sorting all the pixels from the surrounding neighborhood region into numerical order, then changing the pixel in consideration with the median pixel value (whereas if the neighborhood under evaluation had an even pixel count, the average values of the two middle image pixels were used).

2.3.5. Construct a binary mask. Binary images are images in which each pixel has just two significant intensities, often shown as black and white. In terms of numbers, the two values are frequently 0 for black and 1 or 255 for white. Thresholding a grayscale or color image to isolate an object from the background is a classic way to create binary images. The binary images are valuable to recognize an object from its background. The foreground colour refers to the object's color (usually white). The rest of the color (typically black) is known as the background color. However, depending on the image that will be thresholded, this polarity may be transposed. The item is presented with 0 when the polarity is reversed, whereas the backdrop has a non-zero value. The binary image proficiently conceals the red berry regions in the image.

2.3.6. Watershed segmentation. The feature extraction of objects in an image is one of the most difficult image processing procedures. Among the various segmentation techniques that segment images with ambiguous boundaries, the watershed transform is one of the most efficient [19]. This method provides a robust morphological tool for segmenting textured images into interest areas. A binary image's distance-based watershed transforms are the distances between each black pixel and the nearest white pixel in the object component. There are just two gray levels in binary images: 0 and 1, with 0 indicating black and 1 indicating white. Only if two black blobs join together only one catchment basin will emerge in the topography of a binary picture surface.

The marker-controlled watershed transformation has the advantage of being classic, instinctive information that can be automated. The over-segmentation caused by the presence of several local minima seems to be the method's fundamental flaw. Marker-controlled watershed modifications have been proposed to mitigate the effects of extreme over-segmentation. The result of the erosion process is determined by the input pixel's neighborhood and also the indeterminate shape and size. The pixel values are joined in a random way to describe the tomato estimation. Individual berries, two to three berries in a bunch, and some berries partially obscured by foliage are all possibilities for the berry pattern. The color contrast between the leaf and the fruit, and their medium size, distinguish mature berries. By crumbling out the undesired surrounding pixels, the erosion approach made it easier to count aggregated berries. Because the thresholding of red pixels was problematic and phrased as under-estimation, the overlapped berries resulted in incorrect counting in some cases.

2.3.7. Berry detection and blob counting. After then, the count of blobs in the segmentation process was measured. Both the lower and higher limits of the amount of pixel in a blob were used to confine the assessed blobs. The lower limit was used to reduce image noises and had the side effect of rejecting certain red tomato particles that had been separated due to occlusion. From the outcomes of the tomato berry detection and the counting technique, a yield estimate was assessed before they were harvested. The yield estimations derived by observers and the computer-vision counting algorithm were then compared before the berries were harvested. Due to the sections of the fruit being split by branches or foliage, a few of them were recognized twice. When numerous berry regions were overlapping, the upper bound had the impact of eliminating some of the fruit regions.

3. Results and discussions

The main purpose of this research is to create a computer vision system to detect and count the ripened red tomato berries using the hybrid watershed transformation to estimate the crop yield under various natural illumination conditions and compare yield estimation results acquired using various methodologies. The real-time image of the tomato plant is shown in Fig. 8a. The images

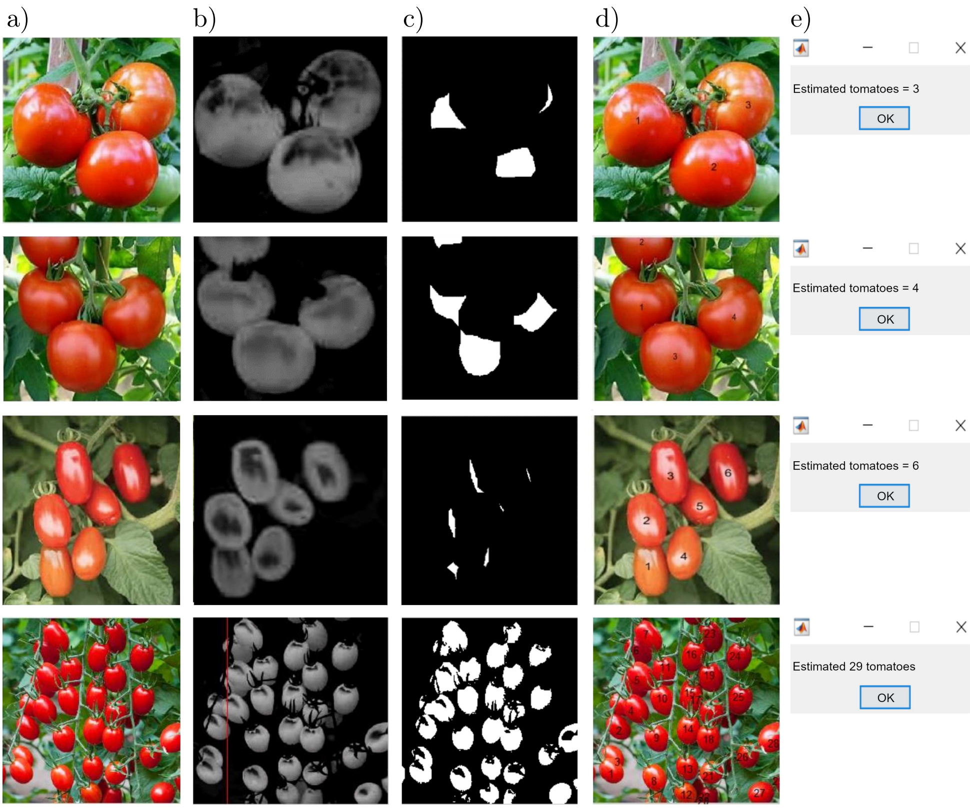

Fig. 8. a) Real-time RGB image, b) grayscale thresholded image, c) binary and eroded image by watershed segmentation, d) labeling the fruit count, e) estimated tomato berries.

in Figs. 8b and 8c have been resized, and the ripened berries have been given thresholds for identifying tomatoes. Then, for overlapping, binary mask outlines the berries and degrades the nearby pixels. Figure 8d shows the blob analysis, which counts the pixels recognized as berries and labels the tomatoes. Figure 8e provides the total estimated count of tomato berries. The proposed model has a lot of potential in terms of crop yield estimation during the harvesting phase.

A regression analysis was conducted for each image. The amount of berry per plant estimated by a fruit counting algorithm was correlated to manual berry counts based on the images. The performance of the fruit counting algorithm employed in this study was evaluated using this method. Due to the relatively large size and the more recognizable color difference between the red berry and the foliage, ripening tomatoes can be distinguished more precisely. The berry counting algorithm detected a discrepancy between the berry numbers detected by the algorithm and those manually counted in images with a large number of berries obscured by foliage or overlapping tomatoes in a fruit cluster.

The performance analysis of the proposed segmentation techniques over the conventional methods is given in Table 1. The performance metrics by watershed segmentation in terms of accuracy is 96%, which is higher than the other conventional segmentation methods. Similarly, the proposed segmentation approach regarding sensitivity and specificity is 61% and 91%, respectively. The precision of the proposed approach is 82%, which is better than the other conventional algorithms.

TABLE 1. Performance measures on conventional image segmentation methods.

Measures	Gray thresholding	Binary mask	Graph cut segmentation [20]	Watershed segmentation
Accuracy	0.57	0.79	0.62	0.96
Sensitivity	0.59	0.58	0.76	0.61
Specificity	0.49	0.48	0.81	0.91
Precision	0.32	0.41	0.46	0.82
Average processing time [ms]	30	32	26	52

Use of texture and color filters minimized false positives in tomato identification. Adjusting filter parameters to reduce false – positive led to an increase in the inability to recognize berry in the image, which was expected. Increased particle size reduced detected background tomatoes (i.e., fewer false positives), but it also removed deeply hidden berries from the image (decreased true positives). The filter values reported were intended to keep false positives to less than 2.2% of all berry counts.

As shown in Figs. 9 and 10, a regression analysis was performed for the tomato berry ripening time. When comparing actual harvested berries to a set of berries recognized by the fruit counting algorithm, a closer correlation ($R^2 = 0.98$) was achieved, as shown in the linear graph in Fig. 10. The closer time to berry harvest and the obvious color and diameter changes in tomato berry, which could be recognized more easily using colour features of image processing, were related to the ripening period. Another set of sampling data was used to validate the model and regression analysis between the number of tomato berries per plant predicted by the model and the number of tomato berries actually gathered.

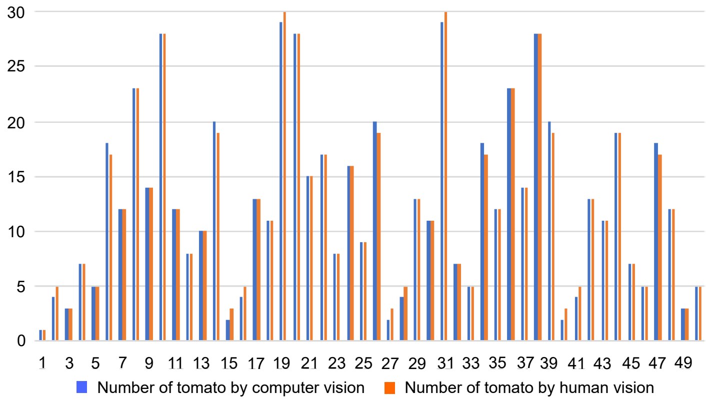

FIG. 9. Comparison of the crop yield estimation with computer vision counting algorithm and manual counting.

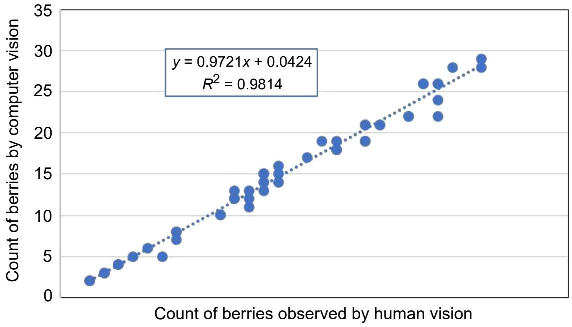

FIG. 10. Regression analysis between the computer vision counting algorithm and human vision.

The crop yield of a tomato field is determined by a variety of parameters. These parameters include: (1) the number of crops in the field, (2) meteorological conditions in both the growing and fruiting seasons, (3) pest and disease prevalence, (4) soil properties, and (5) number of flowers that produce berries. There are various approaches to measure the percent of berry set. Counting the number of blooms on a given tomato plant is one of them.

4. Conclusions

Under various situations, the estimation technique for ripe tomatoes based on RGB color attributes was improved. A digital camera with a fine resolution of 2448×2048 was used to capture photographs of vivid red berries even during the fruit-set phase in natural daylight. To avoid merging of consecutive frames, the images were acquired in an order. The number of berries examined by the estimation method was highly linked with the growth and yield empirical count. The R^2 correlation coefficient between the tomato berry counting vision system and human counting was 0.98. The proposed system's harvest estimation provides a useful report for reap planning and insurance claim accuracy and ease. Fruits such as peaches, red apples and pears, cherries, red peppers, and other red berries can be deducted using this method.

In future work, we will improve the algorithm to deal with larger crop fields, necessitating more exact and complex algorithms for estimating tomato berries within the image. Further, we will enhance the identification and counting algorithm, so that tomato berries recognized on opposite sides of a row can be appropriately combined.

References

1. J. Massah, K. Asefpour Vakilian, M. Shabanian, S.M. Shariatmadari, Design, development, and performance evaluation of a robot for yield estimation of kiwifruit, *Computers and Electronics in Agriculture*, **185**(April), p. 106132, 2021, doi: 10.1016/j.compag.2021.106132.
2. R. Zhou, L. Damerow, Y. Sun, M.M. Blanke, Using colour features of cv. 'Gala' apple fruits in an orchard in image processing to predict yield, *Precision Agriculture*, **13**(5): 568–580, 2012, doi: 10.1007/s11119-012-9269-2.
3. R. Xiang, Y. Ying, H. Jiang, Research on image segmentation methods of tomato in natural conditions, *2011 4th International Congress on Image and Signal Processing*, 2011, pp. 1268–1272, doi: 10.1109/CISP.2011.6100424.
4. H. Li, W.S. Lee, K. Wang, Immature green citrus fruit detection and counting based on fast normalized cross correlation (FNCC) using natural outdoor colour images, *Precision Agriculture*, **17**(6): 678–697, 2016, doi: 10.1007/s11119-016-9443-z.
5. D.M. Bulanon, T. Kataoka, Y. Ota, T. Hiroma, AE–Automation and emerging technologies: A segmentation algorithm for the automatic recognition of Fuji apples at harvest, *Biosystems Engineering*, **83**(4): 405–412, 2002, doi: 10.1006/bioe.2002.0132.
6. S. Bargoti, J.P. Underwood, Image segmentation for fruit detection and yield estimation in apple orchards, *Journal of Field Robotics*, **34**(6): 1039–1060, 2017, doi: 10.1002/rob.21699.
7. D. Font, M. Tresanchez, D. Martínez, J. Moreno, E. Clotet, J. Palacín, Vineyard yield estimation based on the analysis of high resolution images obtained with artificial illumination at night, *Sensors*, **15**(4): 8284–8301, 2015, doi: 10.3390/s150408284.

8. D. Bini, D. Pamela, S. Prince, Machine vision and machine learning for intelligent agrobots: A review, *2020 5th International Conference on Devices, Circuits and Systems (ICDCS)*, pp. 12–16, 2020, doi: 10.1109/ICDCS48716.2020.243538.

9. M. Tripathi, Analysis of convolutional neural network based image classification techniques, *Journal of Innovative Image Processing*, **3**(2): 100–117, 2021, doi: 10.36548/jiip.2021.2.003.

10. R. Sharma, A. Sungheetha, An efficient dimension reduction based fusion of CNN and SVM model for detection of abnormal incident in video surveillance, *Journal of Soft Computing Paradigm*, **3**(2): 55–69, 2021, doi: 10.36548/jscp.2021.2.001.

11. R. Dhaya, R. Kanthavel, A. Ahilan, Developing an energy-efficient ubiquitous agriculture mobile sensor network-based threshold built-in MAC routing protocol (TBMP), *Soft Computing*, **25**(18): 12333–12342, 2021, doi: 10.1007/s00500-021-05927-7.

12. K. Kottursamy, Multi-scale CNN Approach for Accurate Detection of Underwater Static Fish Image, *Journal of Artificial Intelligence and Capsule Networks*, **3**(3): 230–242, 2021, doi: 10.36548/jaicn.2021.3.006.

13. A. Gong, J. Yu, Y. He, Z. Qiu, Citrus yield estimation based on images processed by an Android mobile phone, *Biosystems Engineering*, **115**(2): 162–170, 2013, doi: 10.1016/j.biosystemseng.2013.03.009.

14. P. Annamalai, T.F. Burks, F.C. Laurier, Color Vision System for Estimating Citrus Yield in Real-time Written for presentation at the 2004 ASAE/CSAE Annual International Meeting Sponsored by ASAE/CSAE, 2004, https://elibrary.asabe.org/abstract.asp?JID=5&AID=16714&CID=can2004&T=1.

15. X. Liu *et al.*, Monocular camera based fruit counting and mapping with semantic data association, *IEEE Robotics and Automation Letters*, **4**(3): 2296–2303, 2019, doi: 10.1109/LRA.2019.2901987.

16. B. Darwin, P. Dharmaraj, S. Prince, D.E. Popescu, D.J. Hemanth, Recognition of bloom/yield in crop images using deep learning models for smart agriculture: A review, *Agronomy*, **11**(4): 1–22, 2021, doi: 10.3390/agronomy11040646.

17. A.D. Aggelopoulou, D. Bochtis, S. Fountas, K.C. Swain, T.A. Gemtos, G.D. Nanos, Yield prediction in apple orchards based on image processing, *Precision Agriculture*, **12**(3): 448–456, 2011, doi: 10.1007/s11119-010-9187-0.

18. M.A. Shaikh, S.B. Sayyad, Color image enhancement filtering techniques for agricultural domain using Matlab, [in:] *ISRS Proceeding Papers on Sort Interactive Session. ISPRS TC VIII International Symposium on "Operational Remote Sensing Applications: Opportunities, Progress and Challenges"*, Hyderabad, India, December 9–12, 2014, no. 224, 6 pages.

19. A.L. Tabb, D.L. Peterson, J. Park, Segmentation of apple fruit from video via background modeling, *American Society of Agricultural and Biological Engineers*, 2016, no. December, doi: 10.13031/2013.20873.

20. F. Yi, I. Moon, Image segmentation: A survey of graph-cut methods, *International Conference on Systems and Informatics (ICSAI 2012)*, Yantai, China, 2012, pp. 1936–1941, doi: 10.1109/ ICSAI.2012.6223428.

Received November 11, 2021; revised version January 19, 2022; accepterd February 10, 2022.