Hindawi
Mathematical Problems in Engineering
Volume 2024, Article ID 8868837, 10 pages
https://doi.org/10.1155/2024/8868837

WILEY

Research Article

Flower Recognition Algorithm Based on Nonlinear Regression of Pixel Value

Xionghua Huang,[1] Tiaojun Zeng,[1] MinSong Li,[1] and Yunfei Pan[2]

[1]*School of Information Sciences and Engineering, Shaoguan University, Shaoguan, Guangdong Province, China*
[2]*College of Mechanical and Electrical Engineering, Shihezi University, Shihezi, Xinjiang, China*

Correspondence should be addressed to Tiaojun Zeng; zengtiaojun@163.com

Received 5 November 2023; Revised 23 March 2024; Accepted 5 April 2024; Published 31 May 2024

Academic Editor: Rohit Salgotra

An automated flower thinning system, when combined with machine vision, has the potential to reduce the labor force, improve efficiency, and lower costs. This combination represents the future of agricultural machinery development. The primary objective of automatic flower thinning is to determine the flowering density of fruit trees under natural light conditions. In this study, we introduce a flower recognition algorithm that uses pixel values as an independent variable to recognize flower categories by constructing a nonlinear regression model. Initially, the RGB pixel values of elements in the training set are extracted. Similar pixel values are clustered together to reduce the amount of computation, and representative elements are selected to construct a nonlinear classification function, known as the regression function. The coefficients in the classifier are determined by transforming the problem into an unconstrained optimization problem using the least square method. The optimal solution is then found as the coefficient value in the classifier. The classification function calculates the function value of the RGB pixel value for each input entity to determine whether it belongs to the flower entity. Finally, the developed algorithm is used to classify the flower graphic elements of the measured pictures, and the efficiency of the algorithm is verified.

1. Introduction

Flower thinning is an important operation in orchard production management as it helps to improve the fruit yield and quality [1–5]. This task is labor intensive and highly seasonal. Currently, there are three main methods of flower thinning: manual, chemical, and mechanical. Although manual flower thinning is effective, it faces challenges owing to agricultural labor scarcity and increasing costs. Chemical flower thinning indiscriminately removes flowers and can be harmful to the environment. Meanwhile, mechanical flower thinning is an efficient and automated method that has gained more attention than other methods in modern large-scale orchards [6, 7]. One of the major challenges in creating intelligent flower thinning systems is developing robust and accurate blossom detection algorithms using vision-based techniques. These algorithms must be able to detect flowers in complex environments with high noise and varying illumination conditions. Numerous researchers have made progress in developing machine vision systems for flower detection and positioning, using different sensors and image processing technologies to estimate flower density more accurately [8, 9].

Most existing methods rely on color thresholding, which is simple and easy to use. However, these methods are limited by varying lighting conditions and shading from leaves, stems, or other flowers. Recently, machine-learning and deep-learning techniques have shown great promise in this field [10–15]. For instance, some researchers have proposed an apple blossom-detection method based on convolutional neural network feature extraction, which can be used to detect apple flowers efficiently [10, 16]. However, deep-learning algorithms have two main drawbacks [17, 18]: (1) they require large datasets for training and (2) their training can be expensive and time-consuming owing to their complex data models. Additionally, while deep-learning models may perform well on training datasets, they may not work well on real-world test images. Although many researchers have tried to overcome these challenges, there is still no clear solution to make deep-learning algorithms work perfectly in

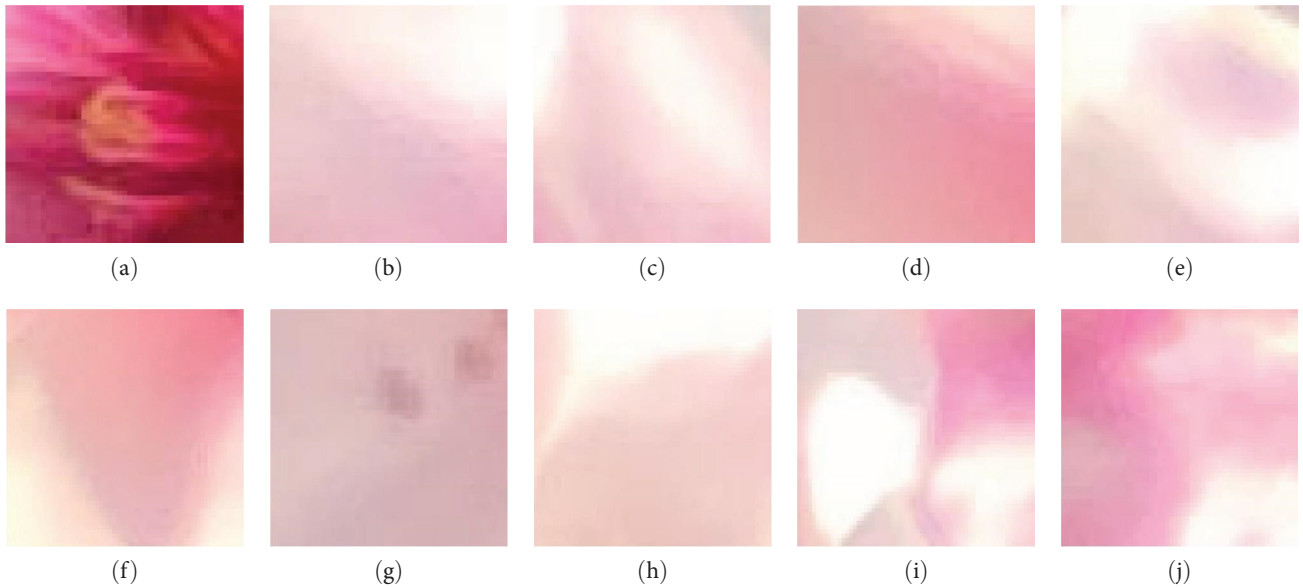


Figure 1: (a–j) Apple flower element.

complex engineering applications. To address these challenges, this study proposes a simpler algorithm suitable for engineering contexts. The proposed algorithm relies on a color recognition model trained on real-world images, enabling it to recognize different image elements such as fruit flowers, branches, and leaves. In the flower thinning system, each input image is segmented, and the recognition function is used to calculate the function value of each image block. This process enables the system to recognize the category of each image block and determine the position of all flowers in the image, making it possible to calculate flower density.

The rest of the paper is organized as follows: Section 2 presents the proposed flower recognition algorithm, Section 3 shows simulation results, and Section 4 presents the conclusions.

2. Proposed Approach

In this study, apple trees are considered as an example. Usually, marked difference is observed in the hues of the flowers, branches, and leaves of fruit-bearing trees. The graphic components of the flowers and branches and leaves are presented in Figures 1 and 2, respectively.

The RGB color mode is derived from the mixing principles of the three primary colors of light present in nature: red, green, and blue. Each pixel can represent 256 brightness levels for each color, resulting in over 16.7 million possible color combinations when the three-color channels are combined. This feature enables the accurate representation of any naturally occurring color. In the context of flower recognition functionality, a classification algorithm can be developed based on the color and pixel RGB value of the primitive serving as the independent variable. This process involves partitioning a large entity, subimage $\boldsymbol{K}$, into smaller entities, secondary subimage $\boldsymbol{k}$. The mean RGB value of each pixel of the small entity is then computed as the RGB value of the entity. Figure 3 provides a graphical representation of this process. The proposed algorithm is a feasible and effective means of accurate flower recognition based on color, because the pixel RGB value of the primitive serves as an independent variable: where $x_k = \frac{R_1+R_2+\ldots R_N}{N}$, $y_k = \frac{G_1+G_2+\ldots G_N}{N}$, and $z_k = \frac{B_1+B_2+\ldots B_N}{N}$ represent $\boldsymbol{g}_k = (x_k \quad y_k \quad z_k)$ as the RGB vector of small subgraph $\boldsymbol{k}$. Similarly, the image $\boldsymbol{K}$ has been segmented and processed to derive the RGB vector set $\boldsymbol{G}_K = \{\boldsymbol{g}_1, \boldsymbol{g}_2, \ldots, \boldsymbol{g}_n\}$. When the RGB values of multiple regions are very close, they are recognized as similar regions from a visual perspective. To reduce the amount of computation required, it is recommended to select a single representative area from among the similar areas and eliminate any redundant regions. As a result, ensuring that the elements in the set are nonredundant to minimize the volume of data. The measurement of similar areas can be determined using the following expression:

$$T\left(\boldsymbol{g}_i, \boldsymbol{g}_j\right) = \text{Max}\left\{\left|R_i - R_j\right|, \left|G_i - G_j\right|, \left|B_i - B_j\right|\right\} < \delta. \tag{1}$$

To identify the constituent parts of a subgraph, a numerical analysis of each element is performed. This analysis generates a function that utilizes an RGB value of the element as a variable. Based on established mathematical principles, the polynomial of the variable can be used to estimate the function. The preestablished constant, δ, is factored into this process:

$$\begin{aligned} F(x, y, z) = {} & a_0 + a_1 x + a_2 x^2 + \ldots + a_n x^n + b_0 + b_1 y \\ & + b_2 y^2 + \ldots + b_n y^n + c_0 + c_1 z + c_2 z^2 + \ldots + c_n z^n. \end{aligned} \tag{2}$$

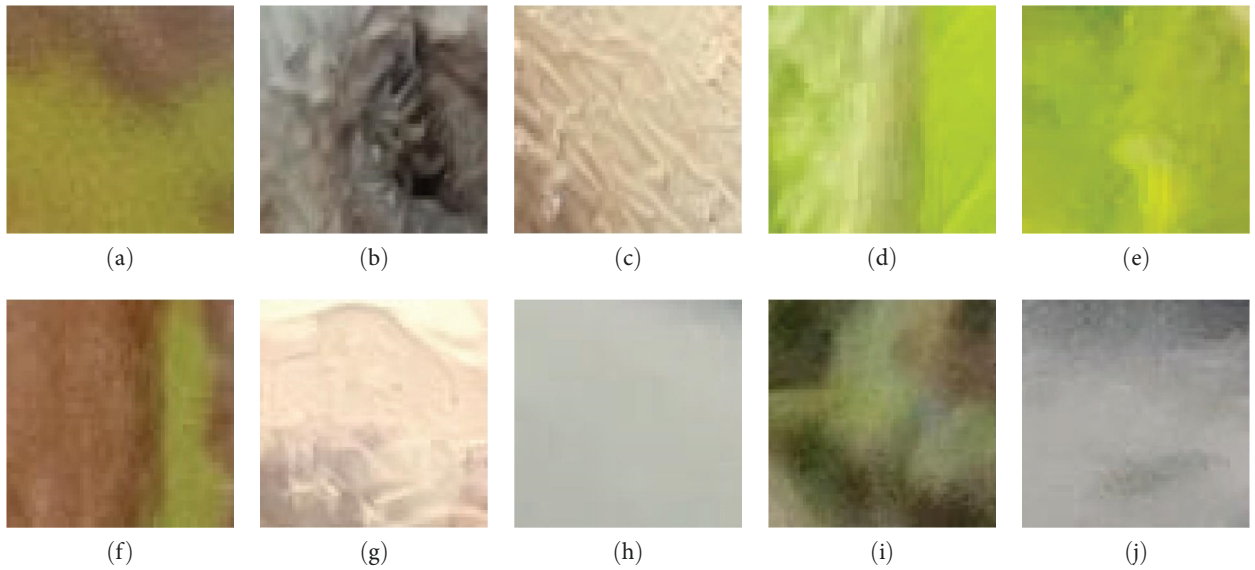


Figure 2: (a–j) Apple branch and leaf elements.

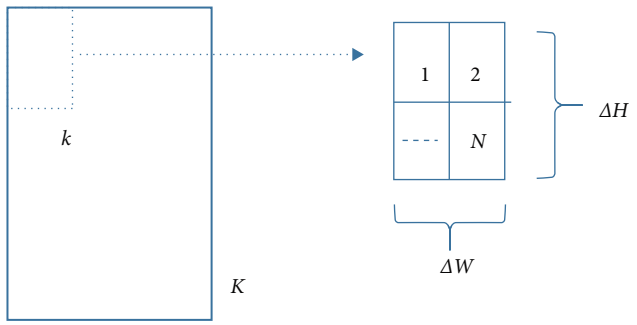


Figure 3: Diagram of element segmentation.

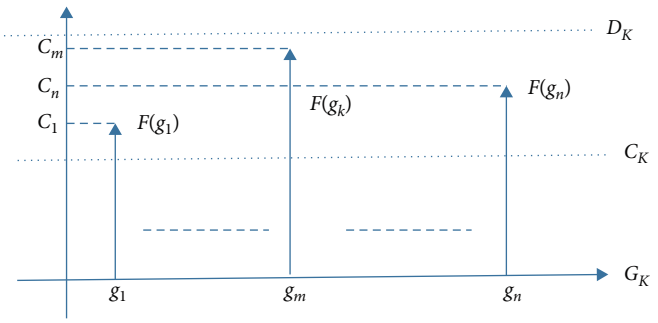


Figure 4: Mapping relationship between G_K and $[C_K, D_K]$.

The above equation is simplified as follows:

$$F(x, y, z) = \sum_{k=0}^{n} \alpha_k \cdot X^k, \tag{3}$$

where $\alpha_k = (a_k, b_k, c_k)$, $X^k = (x^k, y^k, z^k)^T$. Assuming that under the action of the function $F(x, y, z)$, the RGB vector set $G_K = \{g_1, g_2, \ldots, g_n\}$ of the graph elements is mapped to the positive real number interval $[C, D]$ (Figure 4) as follows:

$$F(g_k): \begin{pmatrix} g_1 \\ g_2 \\ \vdots \\ g_m \end{pmatrix} \longrightarrow [C, D]. \tag{4}$$

The function $F(g_k)$ incorporates an unspecified coefficient α_k. Once its value is determined, it will enable us to establish the relationship between the pixel RGB value and the subgraph element category. This, in turn, will allow us to determine whether the pixel value g'_k belongs to the entity G_K by calculating the value of the function $F(g_i)$. The attribute value coefficient c_i $(i = 1, 2, \ldots, n)$ is an unknown value. To derive the value of the coefficient α_k, we must minimize the overall mean square error of c_i and $F(g_i)$ using the least square method:

$$(\alpha_k^*, c_i^*) = \arg\ \min \sum_{\forall g_i \in G_k} (F(g_i) - c_i)^2. \tag{5}$$

To solve the issue, it must be converted into an unconstrained optimization problem:

$$\begin{aligned} P_K(\alpha_k, c_i) &= \sum_{\forall g_i \in G_k} \left((F_K(g_i) - c_i)^2 + w\left(\left(\frac{C_K}{c_i}\right)^2 + \left(\frac{c_i}{D_K}\right)^2 \right) \right) \\ &= \sum_{\forall g_i \in G_K} \left(\left(\sum_{k=0}^{n} \alpha_k \cdot X^k - c_i \right)^2 + w\left(\left(\frac{C_K}{c_i}\right)^2 + \left(\frac{c_i}{D_K}\right)^2 \right) \right). \end{aligned} \tag{6}$$

The abovementioned formula consists of constants w, C_K, and D_K, and penalty terms $(\frac{C_K}{c_i})^2$ and $(\frac{c_i}{D_K})^2$. These terms ensure that c_i converges to a constant within the range of the interval $[C_K, D_K]$ during the optimization process. This means that the value of c_i will not deviate from the

Step 1. For the flower graph element g_i to be tested, if there is a function F_j in the flower recognition function set $F = \{F_1, F_2, \ldots, F_N\}$ such that $F_j(g_i) \in [C_j, D_j]$, the flower graphic element can be correctly recognized and written as C_g.

Step 2. For the flower graph element g_i to be tested, if there is a function L_j in the branch and leaf recognition function set $L = \{L_1, L_2, \ldots, L_M\}$ such that $L_j(g_i) \in [A_j, B_j]$, the flower graphic element can be correctly recognized and written as M_g.

Step 3. For the flower graph element l_i to be tested, if there is a function L_j in the branch and leaf recognition function set $L = \{L_1, L_2, \ldots, L_M\}$ such that $L_j(l_i) \in [A_j, B_j]$, the flower graphic element can be correctly recognized and written as C_l.

Step 4. For the flower graph element l_i to be tested, if there is a function L_j in the branch and leaf recognition function set $F = \{F_1, F_2, \ldots, F_N\}$ such that $F_j(l_i) \in [C_j, D_j]$, the flower graphic element can be correctly recognized and written as M_l.

Algorithm 1: Single Entity Recognition Algorithm.

Table 1: Algorithm 1 test result.

Element type	Correct identification rate	Error recognition rate
Flowers	$C_g = 98.75\%$	$M_g = 36.15\%$
Branches and leaves	$C_l = 97.14\%$	$M_l = 41.21\%$

predetermined range. When $P_K(\alpha_k, c_i) \longrightarrow 0$, the obtained optimal solutions α_k^* and c_k^* $(k = 1, 2, \ldots, n)$ satisfy the following equation:

$$\begin{cases} \forall g_i \in G_K, \sum_{k=0}^{n} \alpha_k^* \cdot X^k \approx c_k^* \\ c_k^* \in [C_K, D_K] \end{cases}. \tag{7}$$

That is, for any small element g_i in the large figure element $G_{\boldsymbol{K}}$, $F_K(\alpha_k^*, g_i) \in [C_K, D_K]$ is satisfied. In this way, $F_{\alpha_k^*}$ can be used as the identification function of graph element $G_{\boldsymbol{K}}$. When it is necessary to determine whether unknown g'_k belongs to graph element $G_{\boldsymbol{K}}$, just calculate the function $F_K(\alpha_k^*, g'_k)$, and then determine according to its value. In the actual recognition of flower classes, we only need to select several flower class diagram elements and branch and leaf diagram elements, train them, respectively, and get the recognition function of each graph element. Using these functions, we can distinguish the classification of unknown graph elements.

Set the resolution of the segmented large image element to $\Delta W = 2, \Delta H = 2$. We first select 40 flower elements and 25 branch and leaf elements to train their classification functions, and then use 60 flower elements (4,685 small elements are obtained after segmentation) and 30 branch and leaf elements (2,842 small elements are obtained after segmentation) to test the classification efficiency, and define four test items. The Algorithm 1 (Single entity recognition algorithm) is as follows, and the test results are in Table 1.

Based on the results presented in Table 1, the forward recognition efficiency of the algorithm is very high. However, the reverse error rate is also quite high, which can be attributed to two main reasons. First, the optimization objective function converges to a constant that is not equal to zero, i.e., $P_K(\alpha_k, c_i) \longrightarrow \varepsilon \neq 0$. This function is determined based on the nature of the optimization algorithm used, whether it is numerical optimization algorithms such as the gradient descent algorithm or popular intelligent optimization algorithms such as the particle swarm optimization algorithm. The convergence value of the objective function is related to the parameters set in advance and the data used for training. Therefore, the obtained α_k may not be the optimal solution, leading to errors in the identification function F_j. Second, the model is trained using a single graphic element, either flowers or leaves, resulting in weak recognition function ability. This phenomenon, coupled with the existing errors, leads to a high reverse error rate.

To improve the accuracy of the classification function and minimize errors, mixed graph elements are utilized in the training process, resulting in the creation of two models: one consisting of a single flower graph element and multiple branch and leaf graph elements, and the other comprising single branch and leaf elements and multiple flower elements. The models are outlined as follows:

$$\mathrm{M}_K(\alpha_k, c_i, d_i) = P_K(\alpha_k, c_i) + \sum_{\forall T_j \in \mathrm{T}} Q_i(\alpha_k, d_j), \tag{8}$$

where $T = \{T_1, T_2, \ldots\}$ represents the collection of branch and leaf graph components, $Q_j(\alpha_k, d_j) = \sum_{\forall t_j \in T_i} ((\sum_{k=0}^{n} \alpha_k \cdot X^k - d_j)^2 + w((\frac{E_K}{d_j})^2 + (\frac{d_j}{F_K})^2))$. The vector consisting of small branches and leaves is part of the training data. $[E_K, F_K]$ is the range of mapping intervals for the graph components. Because branch and leaf elements exhibit considerable color variation, the mapping interval set $[C_K, D_K]$ for a single flower element is shorter. Because these sets do not overlap and their element values differ significantly, the proposed algorithm is able to converge more easily and enhance its classification capabilities. When $\mathrm{M}_K(\alpha_k, c_i, d_j) \longrightarrow 0$, the desired outcome is achieved:

$$\begin{cases} P_K(\alpha_k, c_i) \longrightarrow 0 \\ Q_j(\alpha_k, d_j) \longrightarrow 0 \; j = 1, 2, \ldots \end{cases}, \tag{9}$$

Step 1. Each flower graphic element is segmented, and the recognition function is obtained after training to form a flower graphic element recognition function set $H = \{F_1(g) \quad F_2(g) \quad \ldots \quad F_k(g)\}$. Similarly, the recognition function set $Z = \{f_1(g) \quad f_2(g) \quad \ldots f_n(g)\}$ of branch and leaf elements can be obtained by dividing branch and leaf elements and training.

Step 2. Train the recognition function of single flower element and multibranch-leaf element to form set $M = \{M_1(g) \; M_2(g) \ldots M_i(g)\}$. Similarly, the recognition functions of single branch-leaf and multiflower class elements are trained to form set $N = \{N_1(g) \quad N_2(g) \quad \ldots \quad N_j(g)\}$.

Step 3. The following algorithm is used to identify any unknown element:

(1) If $\exists F_l(g) \in H$, $F_l(g_i) \in [C_l \, D_l]$, $\forall f_l(g) \in Z$, and $f_l(g_i) \notin [P_l \, Q_l]$, it can be determined that element g_i belongs to flower and the algorithm ends. Correspondingly, if $\exists f_l(g) \in Z, f_l(g_i) \in [P_l \, Q_l]$, $\forall F_l(g) \in H$, and $F_l(g_i) \notin [C_l \, D_l]$, it can be determined that element g_i belongs to branches and leaves and the algorithm ends. If neither is satisfied, turn to (2).

(2) Using a single identification function set H and Z, the category of entity g_i cannot be determined, so the mixed function sets M and N are used to identify:

1. If there is a function M_k in the set M so that $M_k(g_i)$ meets the recognition conditions, it is determined that the element g_i is a flower class and the algorithm ends; if there is a function N_k in the set N so that $N_k(g_i)$ meets the recognition conditions, it is determined that the entity g_i is a branch and leaf class, and the algorithm ends. Otherwise, turn to (2).

2. Calculate the proportion of element g_i in function set $H = \{F_1(g) \quad F_2(g) \quad \ldots \quad F_k(g)\}$ and set $Z = \{f_1(g) \quad f_2(g) \quad \ldots \quad f_n(g)\}$ that meet the recognition conditions, respectively. If the proportion is high, g_i belongs to this class, and the algorithm ends.

Algorithm 2: Hybrid Element Recognition Algorithm.

i.e.,

$$\begin{cases} \forall g_i \in G_K, \sum_{k=0}^{n} \alpha_k^* \cdot X^k \approx c_k^*, \; c_k^* \in [C_K, D_K] \\ \forall g_j \in T_j \in \mathrm{T}, \sum_{k=0}^{n} \alpha_k^* \cdot X^k \approx d_j^*, \; d_k^* \in [E_K, F_K] \end{cases}. \tag{10}$$

The abovementioned formula demonstrates two types of graph element data that can be used for training the classification function. Upon achieving the optimal value, the function denoted as $F_{\alpha^*}(g_i) = \sum_{k=0}^{n} \alpha_k^* \cdot X^k$ can effectively map RGB values of varying graph element types to distinct intervals, enabling the classification process. Furthermore, the subsequent model is obtained through the application of single branch-leaf and multiple flower element RGB data during the training phase:

$$N_K(\alpha_k, c_i, d_i) = Q_K(\alpha_k, d_j) + \sum_{\forall g_i \in G} P_i(\alpha_k, c_i). \tag{11}$$

In principle, the use of flower and branch elements to train the function yields higher separation efficiency. However, when multiple elements are present, the number of separation functions can become excessive, increasing calculation requirements. Lastly, the algorithmic steps for recognizing graph elements are outlined in Algorithm 2.

The new algorithm is used to test the above small graph elements, and the results are $M_g = 4.25$ and $M_l = 3.82$, respectively. This shows that the new hybrid model can effectively improve the recognition of graphic elements and reduce the error recognition rate.

3. Simulation Results

To demonstrate the effectiveness of our flower recognition algorithm, we conducted a series of simulations. In this study, the iterative optimization process of the objective function was performed using randomly generated parameters. The simulations were performed with the intention of showcasing the performance of the algorithm in various scenarios.

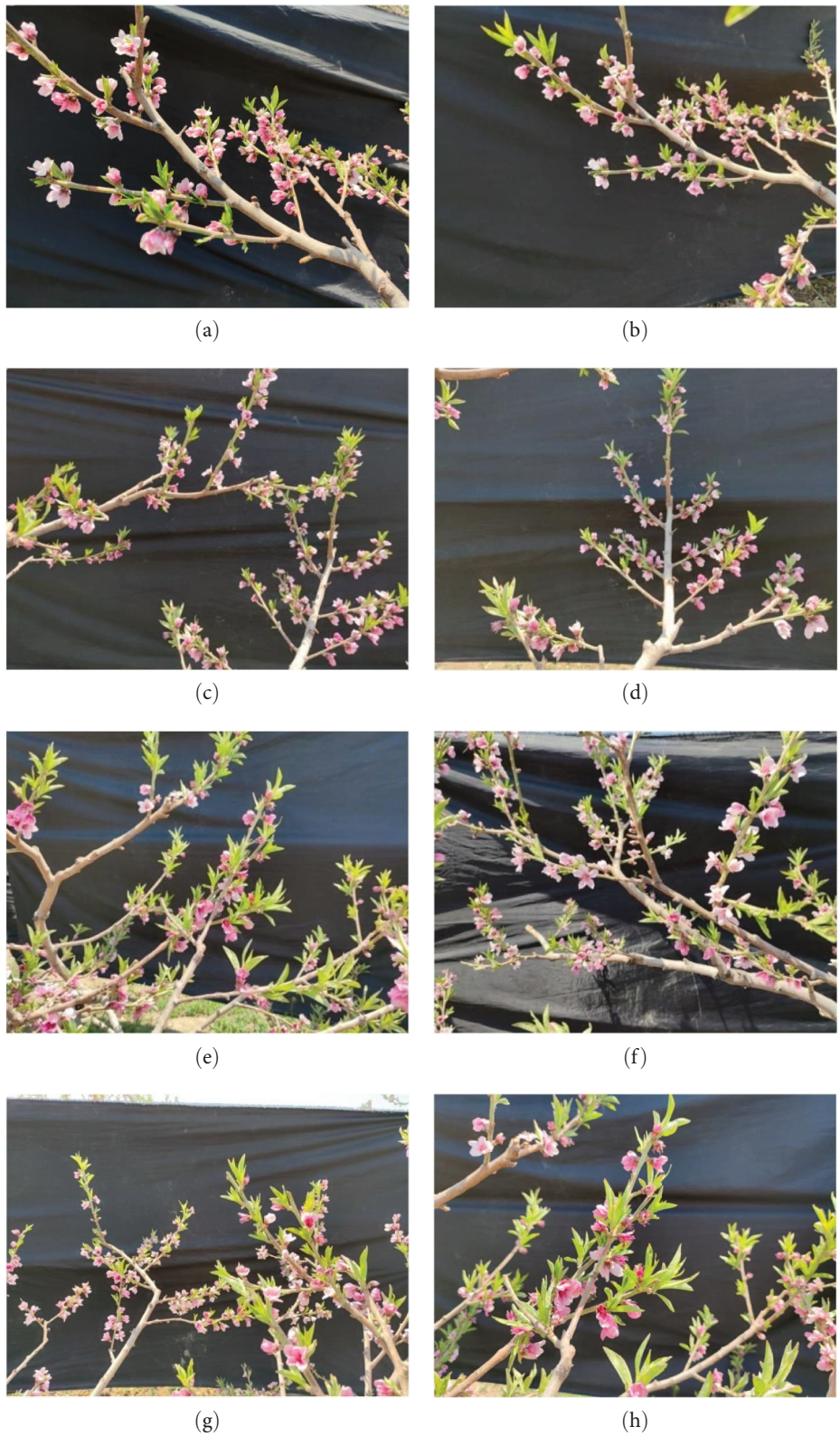
3.1. Simulation 1. In Simulation 1, 10 authentic apple flower images (Figure 5), each with a resolution of $5{,}792 \times 4{,}344$ pixels, were tested. The large pixel cutting resolution was set to $\Delta W = 3$ and $H = 3$, resulting in a segmentation of 2,794,640 small image elements from every image.

The analysis and processing of the given algorithm afforded the proportion of flowers present in the test images. The obtained data are presented in Table 2.

The images corresponding to recognition processing are presented in Figure 6. To improve observation and facilitate comparison, the nonflower graphic elements were adjusted to black.

Based on the analysis of the 10 recognized images, the algorithm is observed to proficiently identify a wide range of flowers and leaves, with most branches being accurately recognized. However, in some cases, certain branches have been misidentified as flowers, owing to their color being similar to white flowers under strong light. This observation underscores the fact that the algorithm is influenced by illumination. In future, efforts will be focused on addressing the limitation, along with the aim of enhancing the accuracy and reliability of the algorithm.

3.2. Simulation 2. In Simulation 2, the accuracy of the algorithm in recognizing apple flowers in images captured from real orchards under natural conditions was assessed. The dataset comprised 200 images, each with a resolution of $3{,}024 \times 4{,}032$ pixels. The entire dataset was labeled using the label box annotator, without any preprocessing or background manipulations. Multiple polygonal masks were used to annotate each image at the pixel level, indicating clusters of apple blossoms, as illustrated in Figure 7.



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

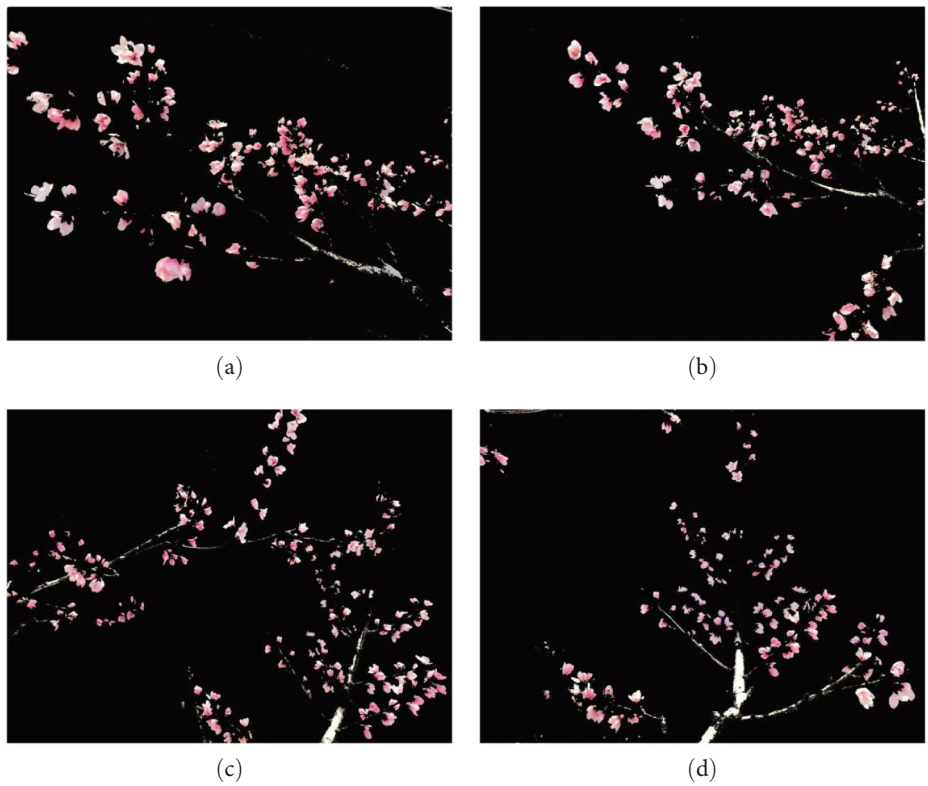
Figure 5: Continued.

(i) (j)

Figure 5: (a–j) Ten test apple flower images.

Table 2: Algorithm 2 test result.

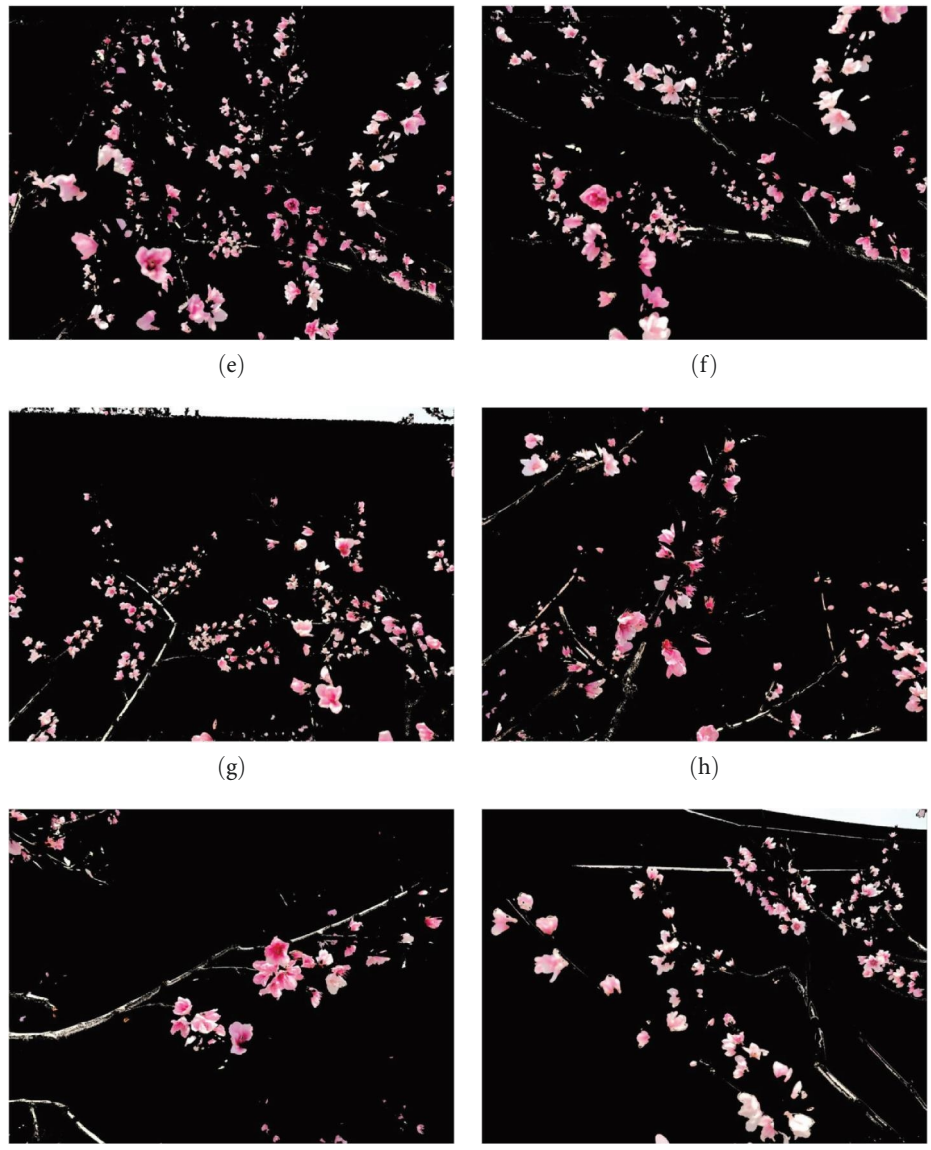
Number	1	2	3	4	5	6	7	8	9	10
Proportion of flowers (%)	6.92	5.09	5.58	4.95	10.42	9.39	9.40	6.84	5.58	9.22



(a) (b)

(c) (d)

Figure 6: Continued.



(e) (f) (g) (h) (i) (j)

Figure 6: (a–j) Ten recognized images.

The rectangular labeling box may contain nonflower elements depending on its nature. The nonsystematic error caused by these elements theoretically has little impact on the recognition accuracy of the algorithm, if the area of these boxes is small enough and enough boxes are present. The annotation software provides the geometric information of each rectangle, including its position, length, and width, allowing us to determine whether the recognition algorithm correctly identifies a real fruit flower, as illustrated in the following equation::

$$\begin{cases} \text{if } (i_0 + m,\ j_0 + n) \in (i + \Delta w,\ j + \Delta h) \\ 0 \le m \le \Delta W,\ 0 \le n \le \Delta H \end{cases}. \tag{12}$$

By identifying pixel $(i_0 + m,\ j_0 + n)$ as an authentic fruit flower, its position $(i,\ j)$, width (Δw), and height (Δh) within a rectangular box are determined. By categorizing each pixel, the subsequent formula is applied to measure the recognition accuracy of the algorithm:

$$P = \frac{\sum_k (i_0 + m,\ j_0 + n)_k}{\sum_i S_f^i}. \tag{13}$$

The proposed algorithm was tested on a sample of 200 images to evaluate its recognition accuracy. The results

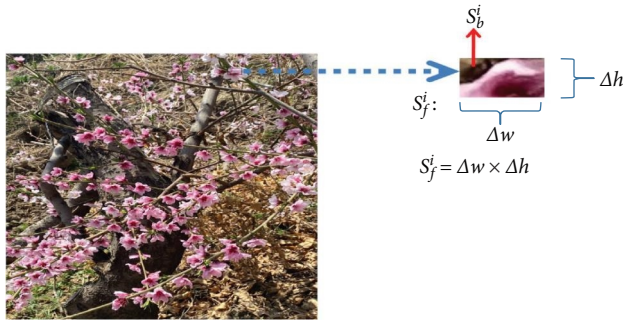


Figure 7: Labeling apple blossom with rectangle box.

indicate that the algorithm achieves an 88.56% (success rate), despite the images being captured in actual orchard scenarios under natural lighting conditions. These findings suggest that the algorithm exhibits high recognition efficiency, favorable robustness, and suitability for engineering applications.

4. Conclusions

In this study, we employed a nonlinear regression model to develop an algorithm that can accurately identify flower types. Pixel values were used as an independent variable, with color descriptions forming the foundation of the algorithm. To achieve the desired results, a standard trial-and-error training algorithm was used. The problem was adapted into an unconstrained optimization problem by applying the least square method. Through optimization calculations, the optimal solution was obtained, allowing the numerical color description to be used as the coefficient value of the classifier. This method provides a dependable and consistent means of precisely describing a color. Tests conducted on images captured in actual orchards and in natural lighting conditions confirm that the algorithm has a remarkable recognition efficiency for fruit flowers. Additionally, the algorithm provides position data for the flowers within the image, enabling the calculation of fruit flower density. This information can be leveraged for an intelligent flower thinning system. Despite its simple structure, the algorithm is supported by explicit mathematical theory, which allows it to work efficiently using only a small amount of training data.

Data Availability

All relevant data are included in the manuscript.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by High Level Talents Program of Shaoguan University under Grant 441-9900064602, Characteristic Innovation Projects of Guangdong Universities in 2021 under Grant 2021KTSCX121, Shaoguan Science and Technology Planning Project in 2021 under Grant 210808234530883, and Shaoguan Science and Technology Planning Project in 2022 under Grant 220607094530516.

References

[1] M. Penzel, M. Pflanz, R. Gebbers, and M. Zude-Sasse, "Tree-adapted mechanical flower thinning prevents yield loss caused by over-thinning of trees with low flower set in apple," *European Journal of Horticultural Science*, vol. 86, no. 1, pp. 88–98, 2021.

[2] K. Sun, X. Wang, S. Liu, and C. H. Liu, "Apple, peach, and pear flower detection using semantic segmentation network and shape constraint level set," *Computers and Electronics in Agriculture*, vol. 185, Article ID 106150, 2021.

[3] J. Lin, J. Li, Z. Yang, H. Lu, Y. Ding, and H. Cui, "Estimating litchi flower number using a multicolumn convolutional neural network based on a density map," *Precision Agriculture*, vol. 23, pp. 1226–1247, 2022.

[4] M. Piani, G. Bortolotti, and L. Manfrini, "Apple orchard flower clusters density mapping by unmanned aerial vehicle RGB acquisitions," in *2021 IEEE International Workshop on Metrology for Agriculture and Forestry (MetroAgriFor)*, pp. 92–96, IEEE, Trento-Bolzano, Italy, 2021.

[5] M. Simpson, S. E. Parks, S. Morris, and D. Joyce, "Use of thinners can increase the fruit size of blueberries in an evergreen system," *New Zealand Journal of Crop and Horticultural Science*, vol. 51, no. 2, pp. 188–197, 2021.

[6] Z. Wang, J. Underwood, and K. B. Walsh, "Machine vision assessment of mango orchard flowering," *Computers and Electronics in Agriculture*, vol. 151, pp. 501–511, 2018.

[7] H.-H. Lee and K.-S. Hong, "Automatic recognition of flower species in the natural environment," *Image and Vision Computing*, vol. 61, pp. 98–114, 2017.

[8] K. Kapach, E. Barnea, R. Mairon, Y. Edan, and O. Ben-Shahar, "Computer vision for fruit harvesting robots – state of the art and challenges ahead," *International Journal of Computational Vision and Robotics*, vol. 3, no. 1/2, pp. 4–34, 2012.

[9] A. R. Jimenez, R. Ceres, and J. L. Pons, "A survey of computer vision methods for locating fruit on trees," *Transactions of the ASAE*, vol. 43, no. 6, pp. 1911–1920, 2000.

[10] P. A. Dias, A. Tabb, and H. Medeiros, "Apple flower detection using deep convolutional networks," *Computers in Industry*, vol. 99, pp. 17–28, 2018.

[11] M. Cıbuk, U. Budak, Y. Guo, M. C. Ince, and A. Sengur, "Efficient deep features selections and classification for flower species recognition," *Measurement*, vol. 137, pp. 7–13, 2019.

[12] Y. Tian, G. Yang, Z. Wang, E. Li, and Z. Liang, "Instance segmentation of apple flowers using the improved mask R–CNN model," *Biosystems Engineering*, vol. 193, pp. 264–278, 2020.

[13] S. Cao and B. Song, "Visual attentional-driven deep learning method for flower recognition," *Mathematical Biosciences and Engineering*, vol. 18, no. 3, pp. 1981–1991, 2021.

[14] U. Bhattarai, S. Bhusal, Y. Majeed, and M. Karkee, "Automatic blossom detection in apple trees using deep learning," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 15810–15815, 2020.

[15] T. Abbas, A. Razzaq, M. A. Zia et al., "Deep neural networks for automatic flower species localization and recognition," *Computational Intelligence and Neuroscience*, vol. 2022, Article ID 9359353, 9 pages, 2022.

[16] D. Wu, S. Lv, M. Jiang, and H. Song, "Using channel pruning-based YOLO v4 deep learning algorithm for the real-time and accurate detection of apple flowers in natural environments," *Computers and Electronics in Agriculture*, vol. 178, Article ID 105742, 2020.

[17] A. Shrestha and A. Mahmood, "Review of deep learning algorithms and architectures," *IEEE Access*, vol. 7, pp. 53040–53065, 2019.

[18] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *2016 IEEE European symposium on security and privacy (EuroS &P)*, pp. 372–387, IEEE, Saarbruecken, Germany, 2016.