

RESEARCH ARTICLE

Artificial Intelligence and Applications
2025, Vol. 00(00) 1-18
DOI: [10.47852/bonviewAIA52025928](https://doi.org/10.47852/bonviewAIA52025928)

BON VIEW PUBLISHING

Estimation of Physical Characteristics of Noisy Peach Leaves Using a Unified Algorithm

Chunxian Chen¹ and Haixin Wang^{2,*}¹ Southeastern Fruit and Tree Nut Research Laboratory, USDA Agricultural Research Service, USA² Department of Mathematics and Computer Science, Fort Valley State University, USA

Abstract: Peach production and breeding benefit from digital agriculture and image processing technology. However, there is little knowledge about the optimal image denoising and processing algorithms for estimating the physical attributes of peach leaves from images. The objective of this study was to evaluate different image denoising and processing techniques to obtain optimal methods and establish a unified approach for estimating the physical characteristics of healthy peach leaves, including length, width, perimeter, and area. Twenty denoising filters were evaluated, and the bilateral filter was determined as the optimal filter for peach leaf image processing with Gaussian and Poisson noise. Twenty-four color space models, including twenty vegetation indices, were evaluated, among which the CIELAB (L*a*b*) color space performed best in the segmentation task using K-means clustering. Seven segmentation algorithms were evaluated and K-means clustering was found to be the optimal method based on a custom metric function. The unified algorithm including optimal denoising, color space selection, and segmentation techniques successfully estimated the physical characteristics of peach tree leaves. The results show that a unified approach is reliable and accurate for estimating the physical characteristics of peach leaves from images. These techniques are crucial for efficient orchard management and other digitalization-based applications in peach production and breeding.

Keywords: bilateral filter, CIELAB color space, Gaussian noise, K-means clustering, peach leaves, physical characteristics, Poisson noise

1. Introduction

Agriculture is a major component of the economy and a crucial factor in national food security. In 2023, a USDA report stated that agriculture, food, and related industries accounted for 5.6% of the GDP and 10.4% of employment in the United States (U.S.) [1]. As an agricultural product, peaches provide nutrients such as vitamins, minerals, and antioxidants that enhance the national diet. In 2021, the total value of peach production in the U.S. was \$624 million [2]. Research in the peach industry, including peach genetics, insect control, and sustainable production practices, contributes to agriculture in general and to sound agricultural practices.

The application of advanced computer technologies in agriculture, known as digital agriculture, has transformed traditional agriculture by promoting efficiency, sustainability, and profitability. Because of its advantages of precise data measurement and analysis, image processing technology simplifies the laborious work in agriculture and makes it more efficient and accurate. The applications of image processing in digital agriculture include peach disease management, automated weed detection in orchards, precise yield estimation, and early detection of plant diseases and pest infestations [3–10].

1) Precision agriculture: With the help of images from satellites, drones, and ground sensors, precision agriculture checks the health of crops, pests and diseases, and the soil.

- 2) Monitoring crops and diseases: Processing images of crops helps farmers track their crops from the beginning to the end of the growing season. It locates diseases and nutrient problems by analyzing information from images, such as color, texture, and growth pattern.
- 3) Weed control: Processing images helps to detect and control weeds in crop fields. It assists in building weed control systems and reduces the need for chemicals and their environmental impact.
- 4) Prediction of crop yields and quality: Processing images of historical data and imagery helps to predict the yield of crops and quality of the fruits.
- 5) Control of resources: Processing images helps save resources such as water, fertilizer, and energy by providing real-time images of crop conditions.

Image segmentation is an important step in image processing. It divides an image into clear and meaningful parts or objects. There are many image segmentation methods, and each has its own advantages and disadvantages. The selection of an appropriate approach depends on the characteristics of an image and the requirements of an application. The following are some commonly used image segmentation methods [11–14].

- 1) Threshold segmentation: Each pixel is assigned to a segment based on its intensity or color. It is fast and efficient when objects and the background have clear separation.
- 2) Edge segmentation: This method finds edges in an image. It is effective when objects have sharp intensity changes.

*Corresponding author: Haixin Wang, Department of Mathematics and Computer Science, Fort Valley State University, USA. Email: wangh@fvsu.edu

- 3) Region segmentation: This method groups pixels into regions based on color, texture, or intensity. It works well for segmenting objects that appear similar on their entire surface.
- 4) Clustering segmentation: This method clusters pixels based on similar features and uses K-means clustering, mean-shift, or fuzzy C-means clustering. It facilitates automatic segmentation with minimal prior information.
- 5) Deep learning segmentation: These approaches are trained on large datasets of labeled images. They are promised to achieve automated and accurate segmentation. But they require robust hardware support with sufficient computational resources and extensive training data.
- 6) Other methods: Semantic segmentation, instance segmentation, panoptic segmentation, local segmentation, and global segmentation. One technique may belong to multiple categories.

Image segmentation results are influenced by the choice of color space models and the noise in images. In image processing, color space models represent colors using mathematical models. Each color space model standardizes color representation and has its benefits and disadvantages for image segmentation. The selection of an optimal model depends on the applications used because different models give different segmentation results [15–17].

Another factor affecting image segmentation is the noise present in images. Images captured by digital cameras and smartphones contain noise originating from image sensors. This noise can be categorized into fixed pattern noise, banding noise, and random noise. Fixed pattern noise arises from sensor measurements due to the stochastic nature of photon counting. Banding noise results from a bank of analog-to-digital (A/D) converters. Random noise is generated by photon emission, intrinsic thermal and electronic fluctuations, and other sources. Table 1 lists the different types of noise encountered in image processing. Image noise introduces unwanted information, posing a significant challenge in the application of image processing technology. A critical step in image processing is the denoising of images, which is fundamental

Table 1
Noise models

Noise Model	Noise Source(s)/Characteristic(s)
Gaussian noise	From natural sources
White noise	Constant power spectrum and zero auto-correlation
Fractal noise	Brownian motion with a non-stationary stochastic process
Salt-and-pepper noise	Pixel values are set to 0 or 255
Periodic noise	From electronic interference
Quantization noise	From the process of converting analog data into digital data
Speckle noise	From coherent imaging systems
Poisson noise	From the statistical nature of electromagnetic waves
Gaussian-Poisson noise	From the combination of Gaussian noise and Poisson noise
Gamma noise	From laser images
Rayleigh noise	From radar images
Impulse noise	Random occurrence of white and black pixels
Uniform noise	Uniformly distributed in a range of values
Multiplicative noise	Signal-dependent scaling noise

for improving the quality of subsequent segmentation tasks. Various image denoising models have been developed and are categorized as shown in Figure 1 [18–23]. The physical characteristics of plants are indicators of their health. Physical characteristics of leaves such as leaf color, texture, shape, and size reflect the physiological condition of the plant. They help us detect plant diseases, nutrient deficiencies, and other stress factors. Understanding the physical characteristics is important for understanding plant growth and productivity, especially in agriculture where plant health affects yield and quality.

Image segmentation technology is a tool for estimating and analyzing these physical characteristics. This process involves partitioning an image into multiple segments, each matching a distinct component of the plant, such as leaves, stems, and flowers. In the work of Ashwinkumar et al. [24], image segmentation was used to detect early signs of disease in tomato plants, and it showed high accuracy rates in distinguishing healthy and infected leaves. In the work of Fuentes et al. [25], convolutional neural networks were used to segment and classify coffee leaf images, helping to find leaf rust disease. Paithane and Wagh [26] developed a novel fuzzy c-means algorithm for cotton leaf spot detection, further highlighting the utility of clustering methods in agricultural imaging.

Accurate estimation of leaf characteristics is crucial for precision agriculture. Recent studies span denoising, color-space choices, and segmentation or trait extraction. Fan et al. [27] developed a two-stream deep framework which performs plant segmentation and leaf counting on Computer Vision Problems in Plant Phenotyping (CVPPP) images. Li et al. [28] built an automated, non-destructive monitoring system for rapeseed leaves and reported strong UNet-Attention segmentation performance. For color space selection in field imagery, Hernández et al. [29] proposed a principled method for choosing optimal channels for plant or soil segmentation. As a preprocessing step, Bhujade et al. [30] introduced an optimization-assisted cascaded filtering pipeline that improves denoising quality for crop images. For classical pipelines, Jamjoom et al. [31] combined K-means segmentation with SVM classification for leaf-disease analysis on PlantVillage. These works highlight gaps in noise robustness, color representation, and computational efficiency. Our unified algorithm employs bilateral filtering, CIELAB, and K-means clustering to improve robustness and accuracy relative to these approaches. Table 2 summarizes and clarifies the scope and metrics.

Unlike deep models that require training data, GPU resources, and task-specific labels, our workflow is end-to-end interpretable: denoising (20 filters), color space selection (24 transformations, including 20 vegetation indices), and segmentation (7 algorithms) are compared quantitatively on the same imagery. The analysis isolates the noise effects (Gaussian vs. Poisson) and motivates the final choices (bilateral filtering and K-means in CIELAB) that maximize our composite scores. The output is a set of physical traits (length, width, perimeter, area) directly usable in orchard management, which is outside the scope of prior works. In contrast, our contribution is a unified workflow that (i) quantitatively selects the denoiser, color space, and segmentation method under mixed Gaussian-Poisson noise; (ii) demonstrates noise robustness and explains failure modes; and (iii) outputs physical leaf traits (length, width, perimeter, area) directly from noisy images. Within our study, bilateral filtering and CIELAB with K-means emerge as the optimal choices, and we quantify that Gaussian noise, not Poisson noise, is the dominant factor affecting accuracy. This positions our method as a resource-light and reproducible alternative to learning-based systems while addressing a different end goal: trait estimation.

The contributions of the study are listed as follows.

- 1) An approach is proposed to estimate the physical characteristics of healthy peach leaves.

Figure 1
Categories of denoising models

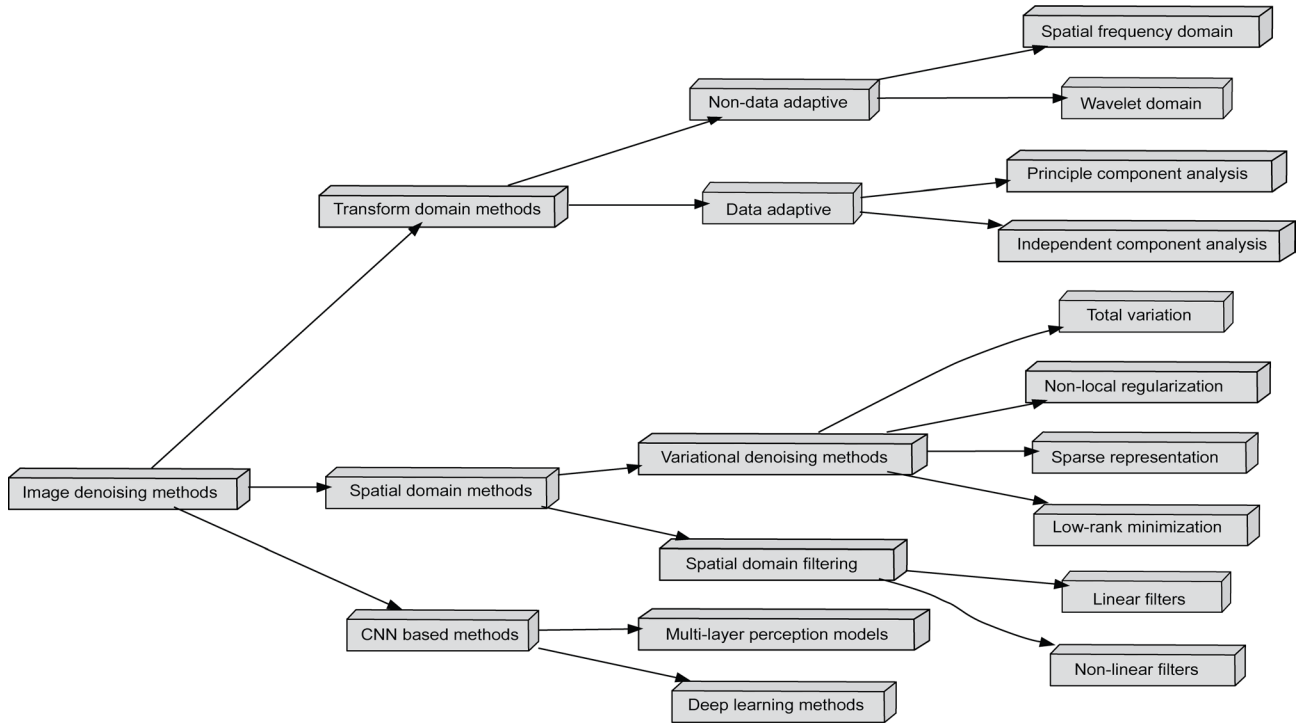


Table 2
Comparison with recent work: tasks, data, metrics, and differentiators

Study	Task	Data	Reported metric(s)	Key differentiators
Fan et al. [27]	Leaf counting guided by segm.	CVPPP17 (<i>Arabidopsis</i> , tobacco)	Segm. IoU $\approx$ 0.98; counting DiC 0.11, ADiC 0.36, MSE 0.42	Deep two-stream model; task = counting; requires training
Li et al. [28]	Leaf-area monitoring	<i>Brassica napus</i>	Segm. acc. 96.77%; area error down to 1.25% (splint)	UNet-Attention + device; task = area on rapeseed; training/hardware
Hernández-Hernández et al. [29]	Plant/soil segm. (color space selection)	Field imagery	Accuracy > 95.8%	Focus on channel selection; no trait estimation
Bhujade et al. [30]	Denoising for crop images	Soybean, cotton	PSNR/SSIM vs. baselines	Denoising only; no segmentation/traits
Jamjoom et al. [31]	Disease segm. and cls.	PlantVillage	SVM acc. 97.2% (KNN 80.2%, Ensemble 83.6%)	Disease classification; training and labels required
This work	Trait estimation from noisy images	Peach leaves (PlantVillage subset)	Unified: selects optimal denoiser, color space, segmentation; reports length/width/area/perimeter	Noise-robust; broad comparison: 20 filters, 24 color transforms, 7 segmentation algorithms

- 2) Twenty denoising filters are compared to identify the optimal filter for peach leaf images.
- 3) The robustness of noise effects is examined on both synthetic data and real leaf images.
- 4) Twenty-four color space models are evaluated to determine the optimal color space for peach leaf images.
- 5) Within the $L^*a^*b^*$ color space, 360 PlantVillage peach leaf images, and a fixed preprocessing and postprocessing, K-means clustering

is identified as the optimal segmentation algorithm for peach leaf images in the study, among seven segmentation algorithms.

2. Proposed Unified Methodology

The physical characteristics of peach leaves, such as length, width, perimeter, and area, are indicators of peach leaf health and vigor. Therefore, a comprehensive analysis of these characteristics can

provide important information about possible problems that affect the health of the leaves and the productivity of peach trees. Studying these characteristics can help in the early detection of problems so that timely actions can be taken and good management strategies can be adopted to solve these problems [50].

Based on the noise analysis of real images, the comparison of color spaces, and the comparison of segmentation algorithms in Section 2, this study proposes a unified approach to find the physical characteristics of peach leaves. The unified approach includes the bilateral filter for denoising, the CIELAB color space for feature representation, and the K-means clustering algorithm for segmentation. The process is illustrated in Figure 2.

2.1. Bilateral filter

The bilateral filter is a nonlinear, edge-preserving, and noise-reducing smoothing filter. The equation is given as follows.

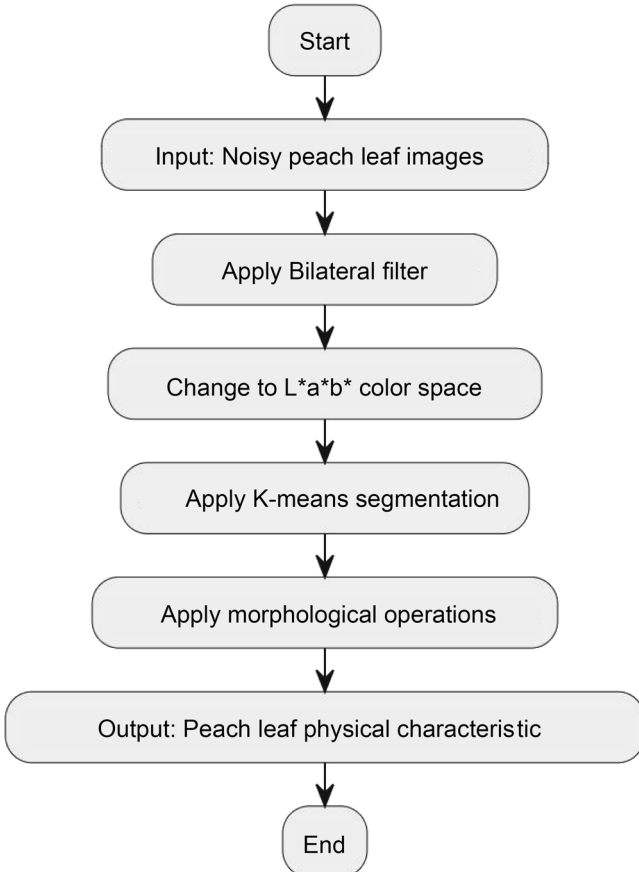
$$G'(x, y) = \frac{\sum_{(i,j) \in S_{x,y}} w(i,j) G(i,j)}{\sum_{(i,j) \in S_{x,y}} w(i,j)} \quad (1)$$

where

- 1) $G'(x, y)$: Output pixel value at position (x, y) after applying the filter
- 2) $S_{x,y}$: The neighborhood window centered around (x, y)
- 3) $w(i, j)$: Weights used in the bilateral filter, with $ws(i, j)$ as spatial weights and $wr(i, j)$ as range weights

Figure 2

A unified approach to estimate the physical characteristics of peach leaves



2.2. RGB to L*a*b*

The CIELAB (or L*a*b*) color space is identified as the optimal color space for this study. The input RGB values must first be converted to the CIE XYZ color space. Then the L*, a*, and b* are obtained by a nonlinear transformation. The conversion process is described as follows.

RGB to XYZ Conversion:

$$\begin{aligned} X &= 0.4124564 \cdot R' + 0.3575761 \cdot G' + 0.1804375 \cdot B' \\ Y &= 0.2126729 \cdot R' + 0.7151522 \cdot G' + 0.0721750 \cdot B' \\ Z &= 0.0193339 \cdot R' + 0.1191920 \cdot G' + 0.9503041 \cdot B' \end{aligned}$$

XYZ to LAB Conversion:

$$\begin{aligned} L &= 116 \cdot f\left(\frac{Y}{Y_n}\right) - 16 \\ a &= 500 \cdot \left[f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right) \right] \\ b &= 200 \cdot \left[f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right) \right] \end{aligned}$$

where $f(t)$ is defined as

$$f(t) = \begin{cases} t^{\frac{1}{3}} & \text{if } t > \left(\frac{6}{29}\right)^3 \\ \frac{1}{3} \cdot \left(\frac{29}{6}\right)^2 \cdot t + \frac{4}{29} & \text{otherwise} \end{cases}$$

2.3. K-means clustering

K-means clustering is an unsupervised machine learning algorithm which splits a dataset into K non-overlapping clusters [44]. The goal of the algorithm is to minimize the sum of distances between data points and their corresponding cluster centers. K-means clustering has advantages such as simplicity, scalability, speed, and flexibility to different data types. However, it also has limitations, including the need to prespecify the number of clusters K, the need to randomly initialize the centroids and being sensitive to outliers, being affected by the shape of the clusters, and being likely to reach suboptimal solutions. K-means clustering is selected as the segmentation method based on the comparative evaluation in Section 2.3. The general procedure consists of the following steps:

Step 1: Initialization

Choose K centroids from the dataset.

Step 2: Assignment Step

Assign each data point to the closest centroid based on Euclidean distance:

$$S_i^{(t)} = \left\{ x_p : \|x_p - \mu_i^{(t)}\| \leq \|x_p - \mu_j^{(t)}\| \forall j, 1 \leq j \leq K \right\} \quad (2)$$

where μ_i is the centroid for the i^{th} cluster, x_p is a data point, and $S_i^{(t)}$ is the set of all points that are closer to the i^{th} centroid than any other centroid.

Step 3: Update Step

Calculate the new centroids as the mean of all data points in the cluster:

$$\mu^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_i \in S_i^{(t)}} x_i$$

Step 4: Output Step

Output: The final K clusters and their centroids until the centroids do not change significantly or the maximum number of iterations is reached.

3. Experimental Evaluation

Image noise, color space models, and segmentation algorithms influence the quality of image analysis. In this section, we perform a comparative analysis of different denoising filters, color space models, and segmentation algorithms to determine the optimal algorithm. This study applies these methods to investigate the physical characteristics of peach leaves.

3.1. Dataset

The dataset used in this study is publicly available from the PlantVillage dataset on Kaggle, which includes 360 images of healthy peach leaves. The input images are shown in the RGB color space.

3.2. Denoising process for real digital images

Knowing the types of noise in digital images helps in developing denoising algorithms and improving image quality. Digital images captured by cameras or smartphones contain noise that follows Poisson and Gaussian distributions. Those noises affect quality of image processing results [21, 32]. See Appendix A for a description of Poisson and Gaussian noise.

3.2.1. Denoising filters

Denoising filters are crucial for processing images before they are applied in tasks such as segmentation [19, 33]. These filters reduce noise and preserve details, including edges. This study evaluates twenty denoising filters, listed in Table 3, to determine the optimal filter for peach leaf images.

3.2.2. Custom noise metric function

To select appropriate denoising algorithms for real images with Poisson and Gaussian distributions, three standard metrics are considered: mean squared error (MSE), peak signal-to-noise ratio (PSNR), and structural similarity index measure (SSIM). These metrics are used to evaluate the performance of denoising algorithms and are listed in Table 4.

Table 3
List of 20 denoising filters

Filter Name	Filter Name
Ideal notch filter	Homomorphic filter
Geometric mean filter	Morphological filter
Median order-statistic filter	Constrained least squares filter
Adaptive noise smoothing filter	Nonlinear complex diffusion filter
Adaptive median filter	Gabor filter
Neighborhood mean filter	Wiener filter
Harmonic mean filter	Kuwahara filter
Inverse harmonic mean filter	Beltrami filter
Inverse filter	Lucy-Richardson filter
Bilateral filter	Non-local means (NLM) filter

Table 4
Summary of image quality metrics for noise analysis

Name of Image Quality Metric	Formula
Mean squared error (MSE)	$MSE = \frac{1}{N} \sum_{i=1}^N (G(i) - \widehat{G}(i))^2$
Peak signal-to-noise ratio (PSNR)	$PSNR = 10 \cdot \log_{10} \left(\frac{MAX^2}{MSE} \right)$
Structural similarity index measure (SSIM)	$SSIM = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$

where

- 1) N is the total number of pixels in the image.
- 2) $G(i)$ is the observed pixel value.
- 3) $\widehat{G}(i)$ is the predicted pixel value.
- 4) MAX is the maximum possible pixel value of the image.
- 5) x and y are two measurement windows with size $N \times N$.
- 6) μ_x and μ_y are the pixel sample means of the two measurement windows x and y.
- 7) σ_x^2 and σ_y^2 are the standard deviations of x and y.
- 8) σ_{xy} is the cross-covariance between x and y.
- 9) C_1 and C_2 are constants added to stabilize the division and avoid division by zero.

While each metric has its limitations, it is necessary to formulate a comprehensive metric function for assessing denoising algorithms. In this paper, we design a new metric function to evaluate the performance of denoising methods on Poisson and Gaussian noisy images. This function integrates three common metric functions: MSE, PSNR, and SSIM, to create a comprehensive evaluation metric. The formula for the custom metric is as follows.

$$\text{Metric}_n = f(w_1^n M1, w_2^n M2, w_3^n S1) \quad (3)$$

where

- 1) n stands for noise. This function is denoted as Metric_n to evaluate the performance of denoising algorithms.
- 2) w_1^n , w_2^n , and w_3^n are weights. The weights can be adjusted to emphasize the importance of a specific metric in applications.
- 3) M1 is the MSE. It measures the average squared difference between the original and denoised images. It is used to measure the reconstruction quality.
- 4) M2 is the PSNR. It compares the maximum possible signal to the noise and assesses the quality of image reconstruction in terms of signal fidelity.
- 5) S1 is the SSIM. It measures the quality of the image by comparing the structural similarity between the original and denoised images.

In Equation (3), w_3^n is adjusted to 50% due to the similarity between metrics MSE and PSNR. After normalizing each component to [0,1], we enforce $w_1^n, w_2^n, w_3^n \geq 0$, and $w_1^n + w_2^n + w_3^n = 1$. Because MSE and PSNR quantify closely related fidelity aspects, we allocate half of the weight to SSIM and split the remainder equally between MSE and PSNR. In all reported results, we set $w_1^n = 0.25$ (MSE), $w_2^n = 0.25$ (PSNR), and $w_3^n = 0.50$ (SSIM). For robustness checks, we considered $w_1^n, w_2^n \in [0.20, 0.30]$, and $w_3^n \in [0.40, 0.60]$; the nominal (0.25, 0.25, 0.50) setting is used for tables and figures. This metric

function is made to balance the evaluation by using the strengths of the three main metrics. MSE shows pixel-level accuracy. PSNR shows the noise level in the reconstructed image. SSIM shows the structural similarity based on human visual perception.

3.2.3. General procedure for comparing denoising filters and results

The general procedure to deal with noise, including Gaussian and Poisson noise, is to compare different denoising algorithms using a custom metric and select the optimal denoising algorithm. Figure 3 illustrates this general procedure.

To standardize the values of M1, M2, and S1 within the range [0, 1], the following equation is used for normalization.

$$\text{normalized}_{\text{value}} = \frac{\text{value} - \text{minimum}_{\text{value}}}{\text{maximum}_{\text{value}} - \text{minimum}_{\text{value}}} \quad (4)$$

A value of $f = 1$ indicates perfect similarity.

In the simulation sections, a noise-free image of a peach tree leaf is used to verify the procedure. Figure 4 demonstrates an example of the procedure. The result is shown in Table 5. f_1 represents the value of f with Gaussian noise, f_2 represents the value of f with Poisson noise, and f_m is the mean value of f_1 and f_2 . The results show that the bilateral filter is identified as the optimal denoising algorithm for the peach tree leaf with Gaussian noise and the geometric mean filter as the optimal algorithm for Poisson noise. The mean value of f_m is 0.65. The

distribution of f_m is depicted in Figure 5, and the majority of f_m values lie within the ranges [0.65, 0.72] or [0.87, 0.94]. The bilateral filter is identified as the top-performing filter for images affected by both noise types. The Wiener filter and inverse harmonic filter are also deemed optimal based on the values of f_m .

The example images shown were generated by adding Gaussian noise with a variance of 0.5 and Poisson noise with a scale factor of 6. The visual differences between images are small and may appear imperceptible to the human eye. However the measurement using MSE, PSNR, and SSIM shows variations in noise reduction and image detail preservation. This simulation is similar to the real digital images. Noise exists in images, but we cannot see it, so quantitative simulation must be performed.

3.2.4. Robustness analysis of noise

Noise impacts image processing results. The covariance of Gaussian noise ranges from 0 to 2.5. The Poisson noise scale factors range from (1 to 10) for different levels of Poisson noise. The general procedure is illustrated in Figure 6.

Figures 7, 8, and 9 provide a visual representation of how accuracy varies with different levels of Gaussian noise covariance and Poisson noise scale factors. The standard deviation of accuracy across different noise levels is 0.136. These variations depend on the Gaussian noise covariance and Poisson noise scale factor. Image processing results show fluctuations in accuracy when these noise parameters

Figure 3
Selection of denoising algorithm based on value f

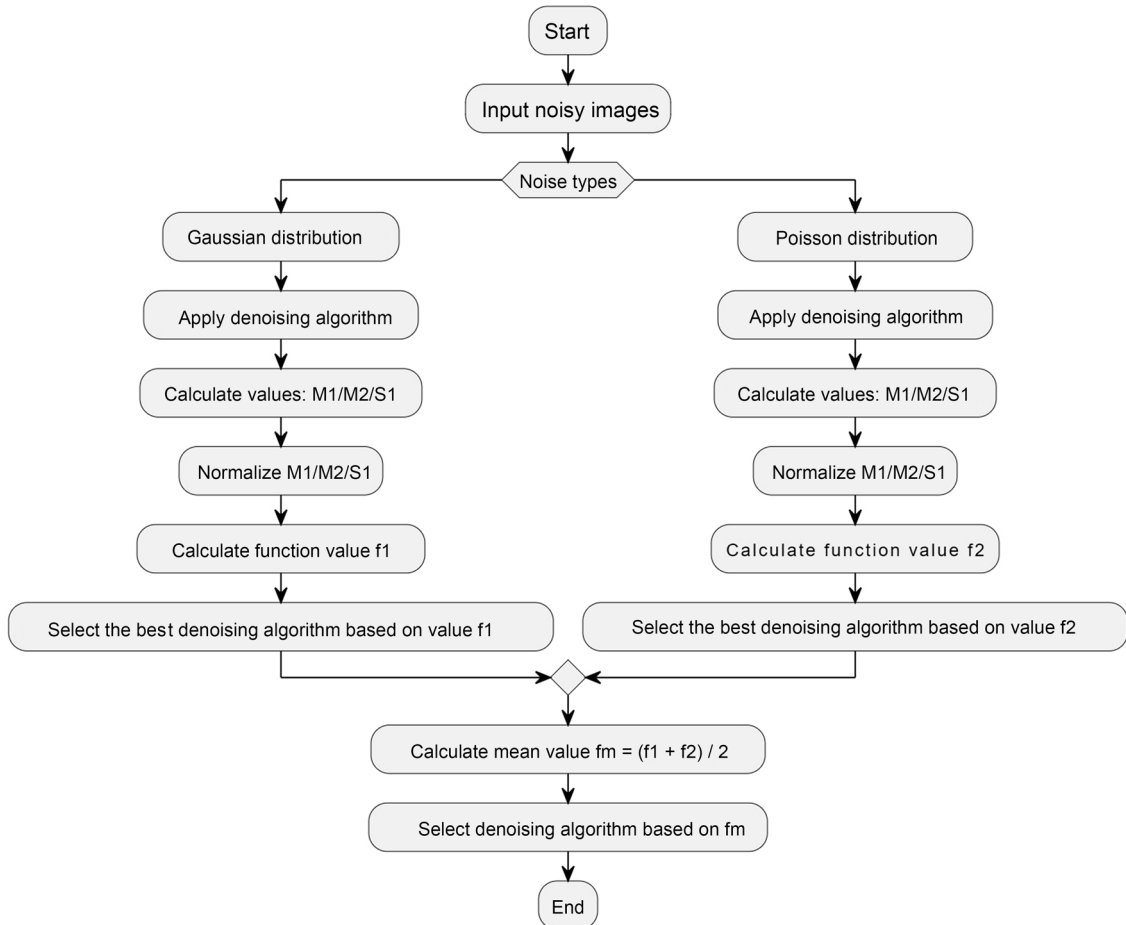


Figure 4

A peach leaf image affected by Gaussian noise and Poisson noise

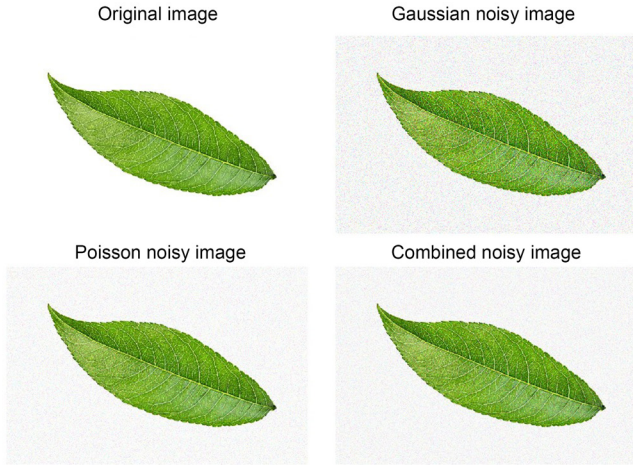


Table 5

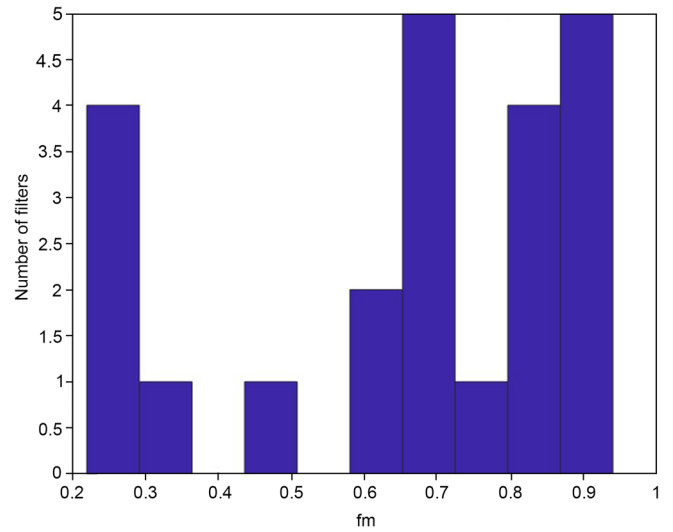
Denoising filters and their performance metrics denoising algorithms f1, f2, fm

Denoising Algorithms	f1	f2	fm
Ideal notch filter	0.4218	0.7275	0.5747
Geometric mean filter	0.7825	0.9720	0.8773
Median order-statistic filter	0.6311	0.5916	0.6113
Adaptive noise smoothing filter	0.8647	0.8023	0.8335
Adaptive median filter	0.9112	0.8431	0.8771
Neighborhood mean filter	0.9053	0.8391	0.8722
Harmonic mean filter	0.8144	0.9609	0.8876
Inverse harmonic mean filter	0.9197	0.9699	0.9448
Inverse filter	0.3373	0.3107	0.3240
Bilateral filter	0.9915	0.9151	0.9533
Homomorphic filter	0.7857	0.7300	0.7578
Morphological filter	0.7086	0.8750	0.7918
Constrained least squares filter	0.3366	0.3102	0.3234
Nonlinear complex diffusion filter	0.8966	0.8305	0.8635
Gabor filter	0.7098	0.6648	0.6873
Wiener filter	0.9635	0.8923	0.9279
Kuwahara filter	0.3372	0.3104	0.3238
Beltrami filter	0.3374	0.3106	0.3240
Lucy-Richardson filter	0.7526	0.6987	0.7256
Non-local means filter	0.7824	0.7264	0.7544

change. The correlation between Gaussian noise covariance and accuracy is -0.77, which means a strong negative relationship. When the Gaussian noise level increases, the accuracy of image segmentation will decrease. The correlation between the Poisson noise scale factor and accuracy is 0.057, indicating Poisson noise has little impact on the accuracy of image segmentation. Therefore, Gaussian noise is the primary factor that affects overall accuracy. Reducing Gaussian noise can improve image processing performance. In the contour plot, some regions show a steep gradient, indicating that accuracy changes significantly when the Gaussian covariance changes.

Figure 5

Distribution of filters based on fm



Regions with wide spaces between contours show areas where accuracy stays steady even when parameters change. In the heatmap, certain columns or rows show higher or lower accuracy. This means that Gaussian covariance has a greater effect on the accuracy than other factors.

3.3. Comparison of color space models

This study analyzes the physical characteristics of peach leaves using different segmentation algorithms. The optimal color space model depends on the specific application, as different models give different results.

3.3.1. Color space models

Previous research [34, 35] investigated twenty-four red, green, blue (RGB) color transformations, including twenty vegetation indices. These transformations are listed in Table 6.

3.3.2. The custom metric function for segmentation

Metrics are used to assess the performance of segmentation algorithms by comparing an algorithm's output and the established ground truth annotations. These metrics help measure the accuracy, completeness, and consistency of the segmented regions. The selection of metrics depends on specific tasks and desired evaluation criteria. Common metrics include pixel accuracy, sensitivity, precision, recall, dice score (or F1-score), intersection over union (IoU) (or Jaccard index), and specificity. Each metric returns a value between 0 and 1, where higher values indicate greater overlap and better segmentation performance [36, 37].

Calculating pixel accuracy requires identifying the true positive (TP), true negative (TN), false positive (FP), and false negative (FN).

For each class:

- 1) TP: A pixel is correctly classified as belonging to a target class.
- 2) TN: A pixel is correctly classified as not belonging to a target class.
- 3) FP: A pixel is incorrectly classified as belonging to a target class.
- 4) FN: A pixel is incorrectly classified as not belonging to a target class.

Selecting a metric depends on the specific application because each has its own advantages and limitations. For instance, metrics such as pixel accuracy, sensitivity, precision, and recall are not suitable for

Figure 6
Flowchart of robustness analysis of noise

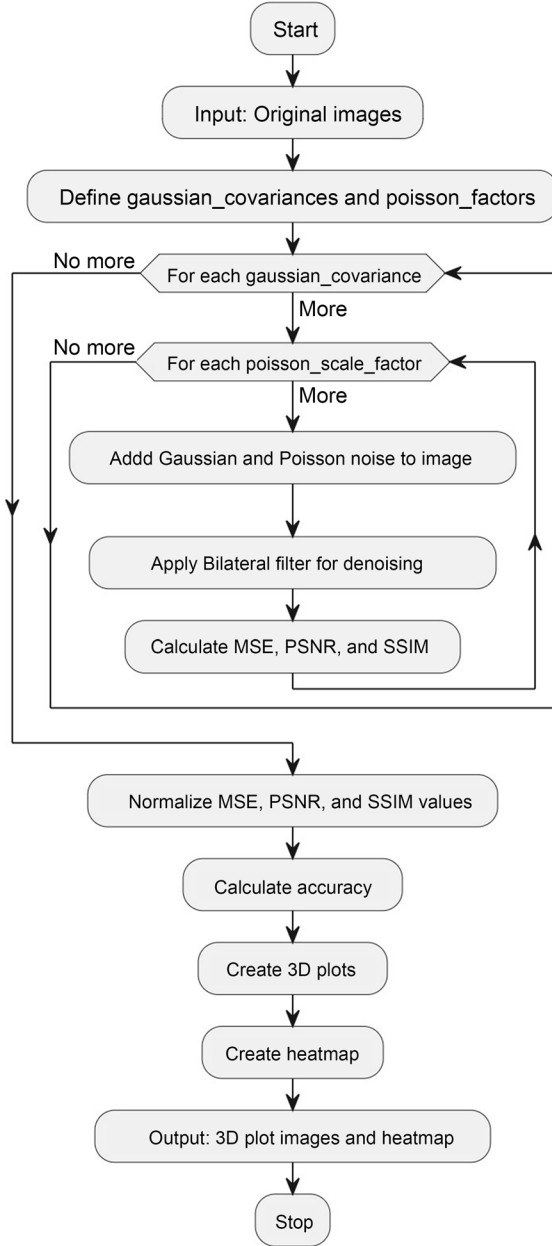


Figure 7
Synthetic image accuracy with different noise levels: 3D surface plot

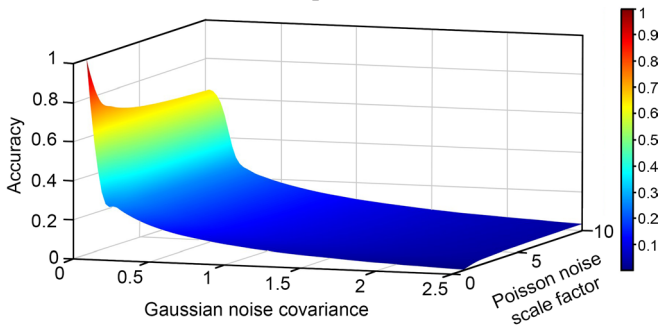


Figure 8
Synthetic image accuracy with different noise levels: contour plot

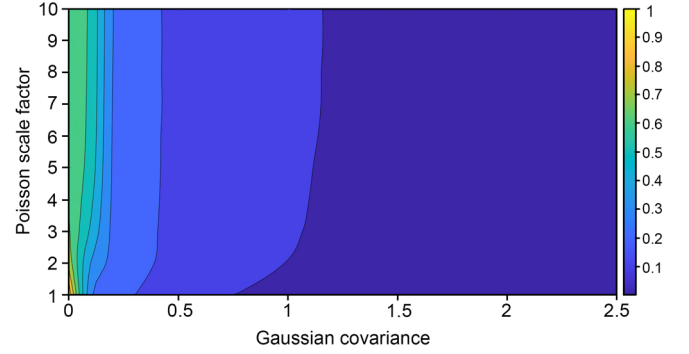
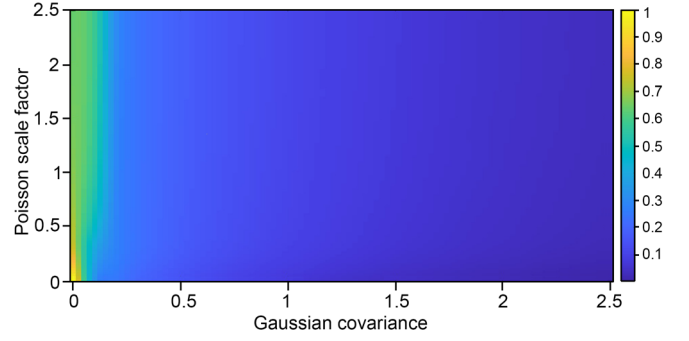


Figure 9
Accuracy of synthetic images at different noise levels shown in a heatmap



class imbalance, while dice score and Jaccard index are suited for class imbalance at the expense of computational cost. Pixel accuracy weights all pixels equally. Sensitivity does not take false positives into account. Precision misses false negatives. Specificity does not count false negatives. The Jaccard index is affected by object sizes. The metric function shown in Equation (5) evaluates the performance of image segmentation algorithms by using six common evaluation metrics, each designed to emphasize different aspects of segmentation quality. The function is given by

$$\text{Metric}_s = f(w_1^s p_1, w_2^s s_1, w_3^s p_2, w_4^s d_1, w_5^s j_1, w_6^s s_2) \quad (5)$$

where

- 1) w_i^s (for $i=1,2,\dots,6$) are weights that can be tuned based on the application and the requirements of the segmentation task. These weights allow the function to focus on certain evaluation metrics depending on the type of images or the segmentation process.
- 2) p_1 is pixel accuracy. It calculates the ratio of correctly classified pixels (true positives and true negatives) to the total number of pixels.
- 3) s_1 is sensitivity or Recall. It measures how well the algorithm finds pixels belonging to the target class.
- 4) p_2 is precision. It measures the number of correctly classified pixels compared to all pixels classified as the target class.
- 5) d_1 is the dice coefficient or F1-score. It balances precision and recall and provides a single score to quantify segmentation performance.
- 6) j_1 is the Jaccard index or IoU. It measures the overlap between the ground truth and the segmented region.
- 7) s_2 is specificity. It measures the ability to correctly find pixels that do not belong to the target class.

Table 6
Color space models

Name	Description
Original	
Red/Green/Blue	R/G/B channels from the RGB color space
Theoretical transformation	
X/Y/Z	X/Y/Z channels from the XYZ color space
L*/a*/b*	L*/a*/b* channels from the CIELAB color space
Hue/Saturation/Value	H/S/V channels from the HSV color space
Y/Cb/Cr	Y/Cb/Cr channels from the YCbCr color space
Empirical transformation	
Normalized Red	$NR = \frac{R}{R+G+B}$
Normalized Green	$NG = \frac{G}{R+G+B}$
Normalized Blue	$NB = \frac{B}{R+G+B}$
Excess Red	$ExR = \frac{1.4R-G}{R+G+B}$
Excess Blue	$ExB = \frac{1.4B-G}{R+G+B}$
Excess Green Red	$ExGR = \frac{3G-2.4R-B}{R+G+B}$
Green Blue Difference	$GBD=G-B$
Red Blue Difference	$RBD=R-B$
Red Green Difference	$RGD=R-G$
Green Red Ratio	$GRR = \frac{G}{R}$
Green Blue Ratio	$GBR = \frac{G}{B}$
Normalized Green Red Difference	$NGRD = \frac{G-R}{G+R}$
Normalized Green Blue Difference	$NGBD = \frac{G-B}{G+B}$
Modified NGRD	$MNGRD = \frac{G^2-R^2}{G^2+R^2}$
Visible Band Difference	$VD = \frac{2G-B-R}{2G+B+R}$
RGB Vegetation Index	$RGBVI = \frac{G^2-B-R}{G^2+B-R}$
Crust Index	$CI = \frac{2B}{R+B}$
Color Index of Vegetation Extraction (CIVE)	$CIVE = 0.441R - 0.811G + 0.385B + 18.78745$
Triangular Greenness Index	$TGI = 95G - 35R - 60B$
Modified Excess Green	$MExG = 1.262G - 0.884R - 0.311B$

Table 7
Summary of image segmentation metrics

Name	Formula
Pixel accuracy	$P_1 = \frac{TP+TN}{TP+TN+FP+FN}$
Sensitivity (Recall)	$s_1 = \frac{TP}{TP+FN}$
Precision	$p_2 = \frac{TP}{TP+FP}$
Dice coefficient	$d_1 = \frac{2 \times TP}{2 \times TP+FP+FN}$
Jaccard index	$j_1 = \frac{TP}{TP+FP+FN}$
Specificity	$s_2 = \frac{TN}{TN+FP}$

depends on the image properties and the requirements of the application. An optimal choice of color space for this work enhances the accuracy and reliability of the segmentation results. Twenty-four different color transformations, including twenty vegetation indices, are implemented [34]. The K-means clustering algorithm is employed for image segmentation. Each color space's metric function score is computed. A detailed illustration of this process is shown in Figure 10.

An analysis of the metric function scores provided in Table 8, and the six metric scores shown in Figure 11 across different color spaces concludes that the L*a*b* color space excels in pixel accuracy, sensitivity, precision, dice coefficient, Jaccard index, specificity, and the custom metric function score. It also exhibits high sensitivity. Both TGI and YCbCr also yield positive results across most metrics. Considering these outcomes, the L*a*b* color space has been selected as the most suitable choice for this study. It is important to note that the conclusion regarding the performance of the L*a*b* color space is based on the K-means clustering algorithm. Different segmentation

Figure 10
Proposed methodology for color space selection in image segmentation

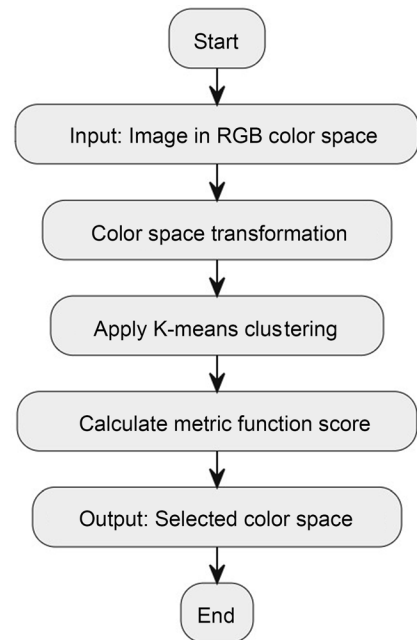


Table 7 shows the formulas for the six metrics. This metric function uses most common metrics when evaluating the segmentation results and gives a balanced, accurate, and full evaluation.

3.3.3. Comparative evaluation of color spaces using the metric function

Choosing the color space has an impact on the outcomes of image segmentation. The effect of color space on segmentation

Table 8
Metric function scores for each color space

Set A		Set B		Set C	
Color space	Score	Color space	Score	Color space	Score
CI	0.6183	L*a*b*	0.9840	RGB	0.9112
CIVE	0.4952	MExG	0.8644	RGBVI	0.5176
ExB	0.9560	MNGRD	0.2619	RBD	0.2812
ExGR	0.6452	NB	0.6986	RGD	0.9715
ExR	0.9409	NG	0.8460	TGI	0.9808
GBD	0.9674	NGBD	0.7047	VD	0.8427
GBR	0.2997	NGRD	0.8406	XYZ	0.7941
GRR	0.9300	NR	0.4512	YCbCr	0.9817
HSV	0.8556				

algorithms may yield different results when used in conjunction with the various color spaces. Previous studies [38, 39] have shown that a comparative analysis of color spaces across various segmentation methods reveals that results are relatively similar and the segmented regions of the resulting images using HSV, HSI, L*a*b*, and L*u*v* color spaces are more uniform.

3.4. Comparative analysis of segmentation algorithms within the CIELAB (L*a*b*) color space

The selection of the segmentation algorithm is important because each algorithm has its own strengths and limitations that affect the quality of the segmentation results [40–43]. The comparison of the seven algorithms follows a standard procedure, as shown in Figure 12. The quality is evaluated using a custom metric function score. Algorithmic details of energy forms, thresholds, neighborhood rules, and hyperparameters for segmentation algorithms: Active Contour, Region Growing, Multilevel/Global/Adaptive Thresholds, and KNN, are provided in Appendix B.

3.4.1. Performance analysis of segmentation algorithms

This study uses the custom metric function in Equation (5) with six metrics: pixel accuracy, sensitivity, precision, dice score, Jaccard index, and specificity to compare the performance of different segmentation algorithms. From Figures 13 and 14, the K-means clustering algorithm gets the highest score based on the custom metric function. The global threshold and adaptive threshold algorithms follow the K-means clustering algorithm. The best algorithm depends on the application requirements. In this study, the K-means clustering algorithm is selected for peach leaf segmentation because it gives the best performance according to the custom metric function score.

Figure 15 outlines the general procedure of segmenting a peach leaf from its background, and Figure 16 illustrates an example of the segmentation results.

4. Results and Discussion

4.1. Component selection results

Our experimental evaluation of 20 denoising filters confirmed that the Bilateral filter provided the best performance for images with mixed Gaussian and Poisson noise, achieving the highest mean score of 0.9533.

Among the 24 color space transformations evaluated, the CIELAB color space was identified as optimal, yielding a top metric score of 0.9840 in K-means segmentation tasks.

The comparison of seven segmentation algorithms demonstrated that K-means clustering was the most effective for isolating peach leaves within the CIELAB space, outperforming other methods on our custom metric function.

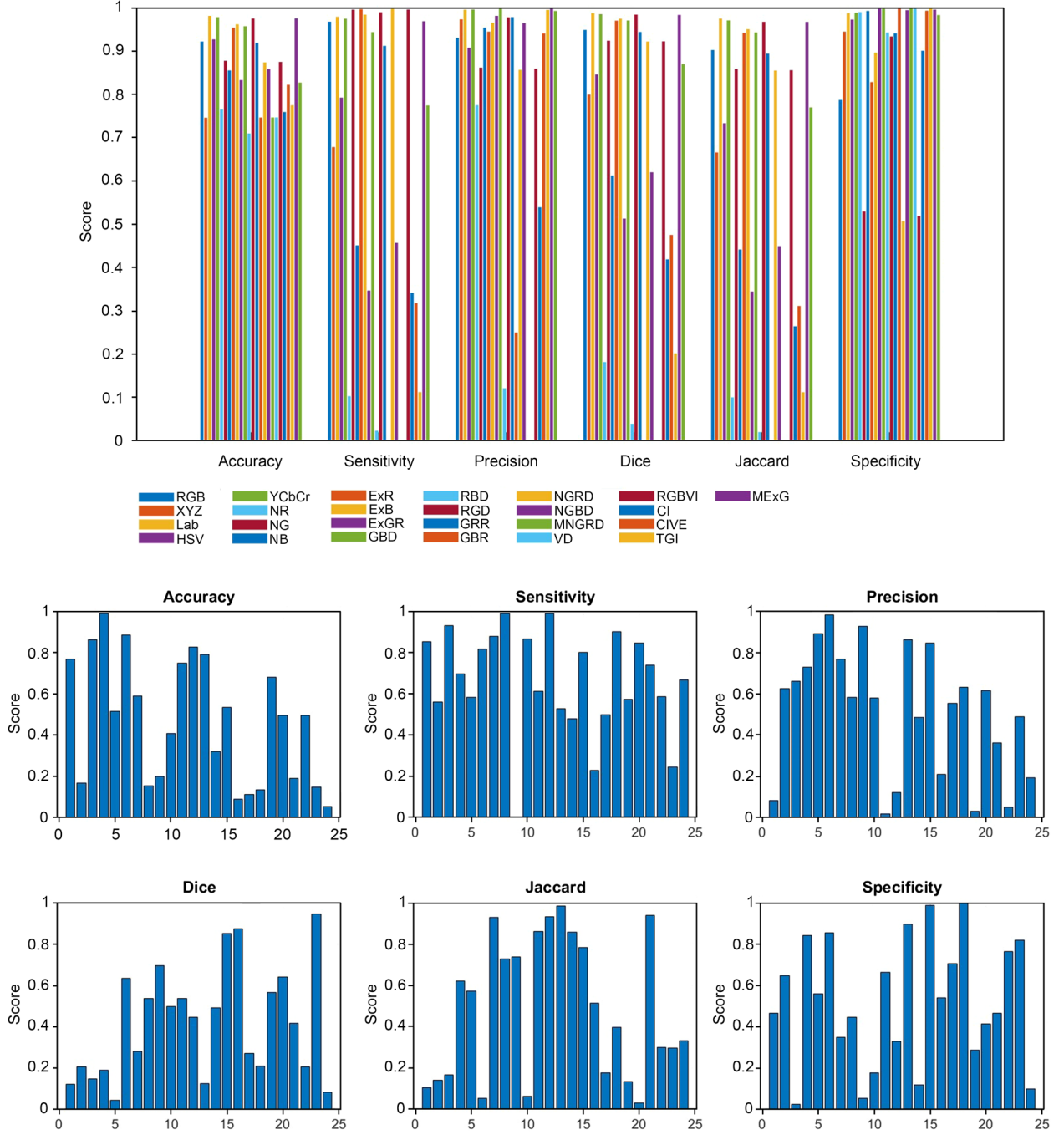
4.2. Noise robustness and implications

Real images inherently contain noise, which impacts the performance of segmentation algorithms. The noise robustness of the proposed segmentation algorithm was evaluated by introducing additional noise covariance. To evaluate the performance of the algorithm under realistic mixed-noise conditions, we created a three-dimensional (3D) parameter space. The following figures visually illustrate the accuracy of the system, where one axis represents the Gaussian noise covariance and the other axis the Poisson noise scale factor. This approach allows for a clear analysis of how the algorithm responds to each noise type individually and to their combined effect. Figures 17 to 19 illustrate the results generated by the flowchart procedure in Figure 6.

Unlike the analysis of synthetic data, real digital images are sensitive to Gaussian noise variance, which is between [0, 0.05]. Figures 17 and 18 show 3D surface and contour plots visually illustrating the accuracy of the image segmentation algorithm when the Gaussian noise covariance and Poisson noise scale factor are varied. The results show that as the Gaussian noise covariance increases, the segmentation accuracy drops significantly. This drop becomes very steep when the covariance exceeds 0.05. On the other hand, Poisson noise has a much less impact on accuracy, and accuracy decreases much slowly as the scale factor varies. The plots show that for getting high-accuracy segmentation, controlling Gaussian noise is more important than handling Poisson noise.

Figure 19 shows more details with a heatmap that maps segmentation accuracy across different Gaussian and Poisson noise levels. Gradient of intensity in the heatmap visualizes levels of the image where accuracy drops dramatically. The largest color changes happen when Gaussian noise covariance is high, supporting the finding that Gaussian noise is the main factor. The relatively stable regions of the heatmap along the axis of increasing Poisson noise suggest that the segmentation algorithm exhibits greater resilience to Poisson noise compared to Gaussian noise.

Figure 11
Six metric scores for each color space except for the custom metric score



The analysis of Figures 17–19 demonstrates that while both Gaussian and Poisson noise can degrade segmentation accuracy, the system is notably more sensitive to variations in Gaussian noise covariance. The accuracy decline is steep in regions of higher Gaussian noise, indicating that in real-world applications, efforts to mitigate Gaussian noise are crucial for preserving segmentation quality. On the other hand, the system shows robustness against Poisson noise, with accuracy only gradually decreasing as the Poisson noise scale factor increases. This suggests that while Poisson noise may be present in imaging scenarios, its impact on segmentation performance is less than

that of Gaussian noise. Therefore, in practical applications requiring high-precision segmentation, strategies should prioritize reducing Gaussian noise to maximize accuracy.

4.3. Physical characteristics estimation

The measurement of the physical characteristics of peach leaves such as length, width, area, and perimeter is achieved using a systematic procedure that uses segmentation and measurement methods. Below is a step-by-step breakdown of the process:

Figure 12
Comparative evaluation process for segmentation algorithms

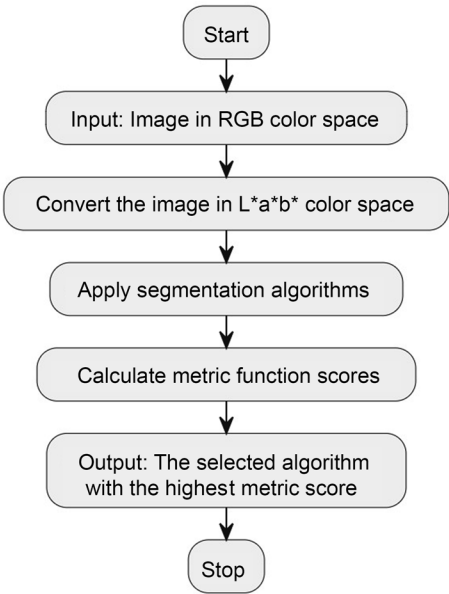


Figure 15
General procedure to segment a leaf from its background

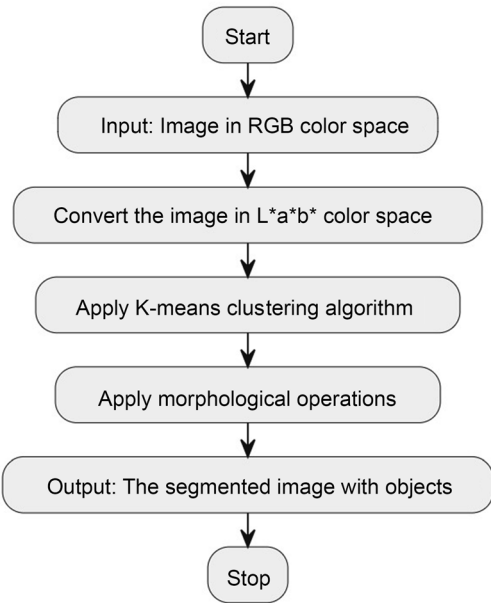


Figure 13
Performance of segmentation algorithms based on six metrics

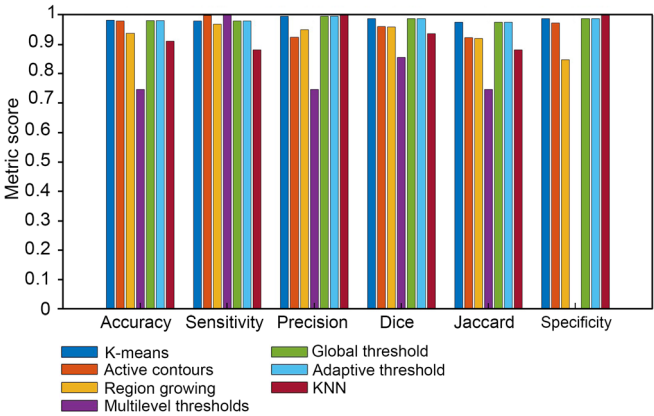


Figure 16
(a) Original image, (b) Leaf image segmented from its background for better readability and formatting

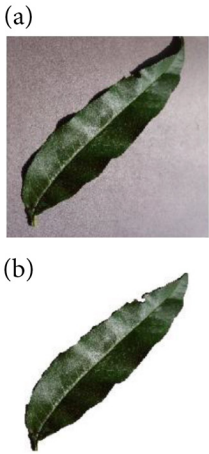
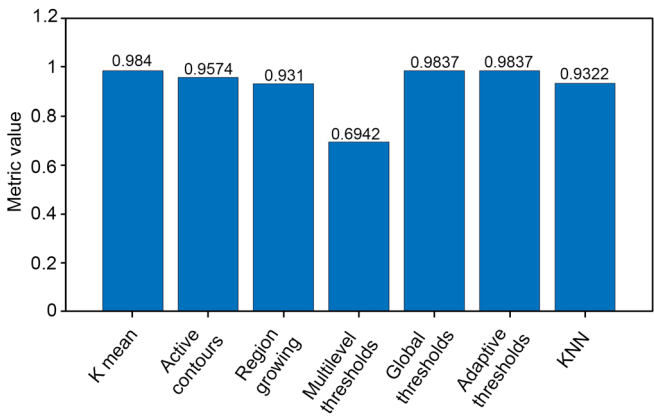


Figure 14
Comparison of custom metric scores of segmentation algorithms



- 1) Image acquisition and preprocessing: A peach leaf image is preprocessed using the bilateral filter to eliminate noise while retaining edges.
- 2) Segmentation: The leaf is segmented from the background using K-means clustering in the L*a*b* color space.
- 3) Measurement:
 - (a) Length: The length is measured from the base to the tip of the leaf in the major axis of an ellipse that fits the segmented leaf.
 - (b) Width: It is measured as the minor axis of the same ellipse and the shortest diameter.
 - (c) Area: It is measured by counting the number of pixels inside the segmented leaf.
 - (d) Perimeter: It is measured by the sum of the distances between adjacent pixels on the leaf edge.

Visual results are shown in Figure 20.

Figure 17

Accuracy of real image segmentation with different noise levels:
3D surface plot

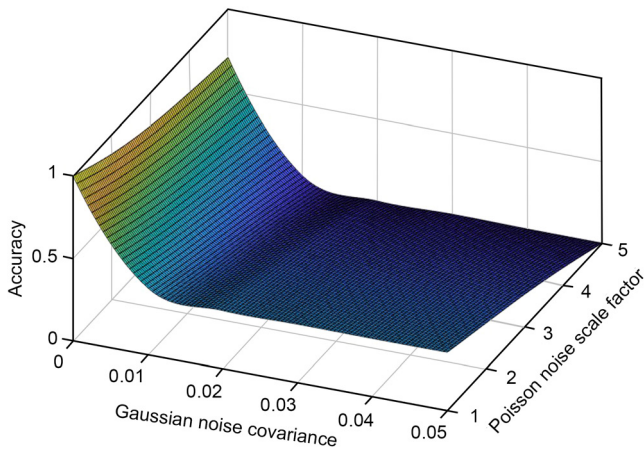


Figure 18

Accuracy of real image segmentation with different noise levels:
contour plot

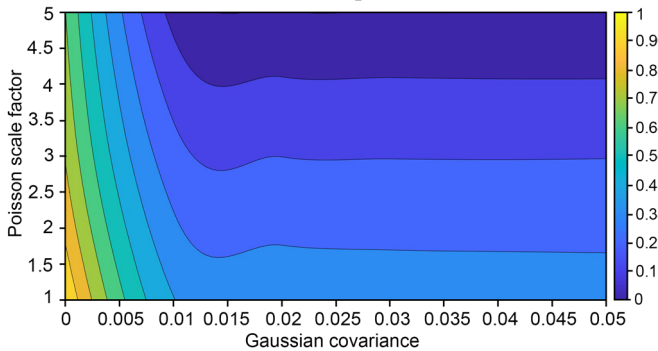


Figure 19

Accuracy of real image segmentation with different noise levels:
heatmap

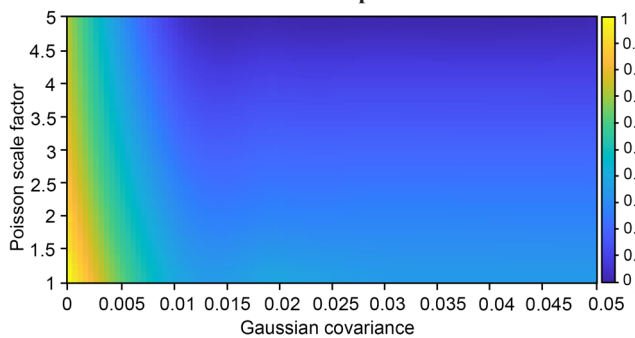


Table 9 presents the pixel-based measurements of the physical characteristics of a peach leaf. This table includes measurements associated with a peach leaf image, specifically aimed at evaluating and determining properties related to a leaf's physical characteristics.

Table 2 contrasts our study with recent research. Two clarifications apply: (1) tasks and datasets differ (e.g., leaf counting on

Figure 20

A peach leaf example demonstrating the unified algorithm: (a) original image of a peach leaf, (b) denoised image after applying bilateral filter, (c) illustration of width and length of a peach leaf, (d) segmented image using K-means clustering

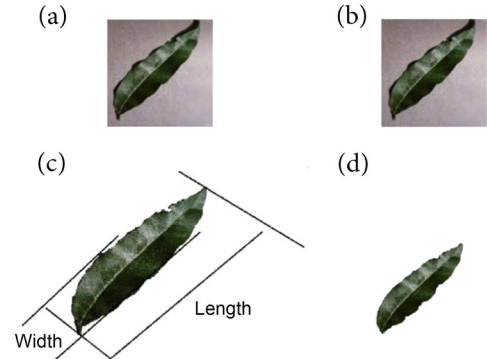


Table 9

Physical characteristics measurements of a peach leaf in pixels

Area	Length	Width	Perimeter
16,213.0	278.3	75.5	641.7

CVPPP, disease classification on PlantVillage, or rapeseed leaf-area monitoring) and (2) several works target a single pipeline component (denoising or segmentation), whereas our goal is an end-to-end trait estimation workflow. In our study, we provide quantitative evidence that (i) bilateral filtering is optimal across Gaussian/Poisson noise when evaluated by a composite metric, outperforming the next best filters, inverse harmonic mean and Wiener filters, in terms of the mean score fm and (ii) K-means in the CIELAB space yields the highest segmentation score among seven candidates in our evaluation. We also find that Gaussian noise strongly and negatively correlates with accuracy (-0.77), whereas Poisson noise has a minimal effect, and we visualize robustness across different noise levels. Consequently, our unified pipeline is competitive and capable of reporting peach leaf physical traits required for orchard applications.

5. Conclusion

The study fills the gap in digital agriculture by designing a unified image processing algorithm to find the physical characteristics of peach leaves, such as length, width, perimeter, and area, from noisy images. This study falls under digital agriculture, where image processing helps with crop monitoring, disease detection, and resource management.

Peach trees are important in agriculture, and the ability to automatically evaluate leaf health is important for orchard management and breeding programs. However, the noise in real agricultural images makes it difficult to correctly segment and measure leaf features. Therefore, different image denoising filters, color space models, and segmentation algorithms were applied and compared to find the appropriate method for processing peach leaf images with different levels of noise.

The comprehensive analysis shows that the bilateral filter is the best for dealing with Gaussian and Poisson noise in peach leaf images. It preserves the edges and details of the leaf during the denoising process. The $L^*a^*b^*$ color space model is the suitable color space model that provides better performance in the segmentation process using the

K-means clustering algorithm due to its robust color separation property. Among the seven segmentation methods, K-means clustering was the most reliable for segmenting the leaf region and extracting the physical dimensions of the leaf.

The algorithm developed in this study integrates optimal denoising, color space selection, and segmentation methods to obtain a reliable and accurate method for finding the physical characteristics of peach leaves. The reliability was demonstrated by the high accuracy of the leaf blade length, width, area, and perimeter measurements from noisy images. These measurements can be used to monitor the health of peach trees and aid in insect control and the efficient use of resources in the orchard.

This work provides a strong and flexible image processing method for peach leaf analysis in digital agriculture. This technique is also applicable to other crops and agricultural use cases where image noise and segmentation are a challenge. Future work will explore the application of the method to images of peach leaves that are affected by disease and pests, and explore the application of deep learning approaches in such study.

Funding Support

This work is sponsored by the USDA 1890 Research Sabbatical Program under the project “Development of New Stone Fruit Cultivars and Rootstocks for the Southeastern United States” (Agreement No. 58-6042-2-014; Project No. 6042-21000-006-002S).

Ethical Statement

This study does not contain any studies with human or animal subjects performed by any of the authors.

Conflicts of Interest

The authors declare that they have no conflicts of interest to this work.

Data Availability Statement

The data that support the findings of this study are openly available in the PlantVillage Dataset on Kaggle at <https://www.kaggle.com/datasets/abdallahalidev/plantvillage-dataset>.

Author Contribution Statement

Chunxian Chen: Conceptualization, Methodology, Validation, Investigation, Resources, Data curation, Writing – review & editing, Supervision, Project administration, Funding acquisition. **Haixin Wang:** Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Writing – original draft, Writing – review & editing, Visualization, Supervision, Project administration, Funding acquisition.

References

- [1] USDA Economic Research Service (ERS). (2024). Ag and Food Statistics: Charting the Essentials. Retrieved from: https://ers.usda.gov/sites/default/files/_laserfiche/publications/108461/AP-121.pdf
- [2] Agricultural Marketing Resource Center. (2024). Peaches. Retrieved from: <https://www.agmrc.org/commodities-products/fruits/peaches>
- [3] Nowak, B. (2021). Precision agriculture: Where do we stand? A review of the adoption of precision agriculture technologies on field crops farms in developed countries. *Agricultural Research*, 10(4), 515–522. <https://doi.org/10.1007/s40003-021-00539-x>
- [4] Domingues, T., Brandão, T., & Ferreira, J. C. (2022). Machine learning for detection and prediction of crop diseases and pests: A comprehensive survey. *Agriculture*, 12(9), 1350. <https://doi.org/10.3390/agriculture12091350>
- [5] Liu, J. & Wang, X. (2021). Plant diseases and pests detection based on deep learning: A review. *Plant Methods*, 17(1), 22. <https://doi.org/10.1186/s13007-021-00722-9>
- [6] Esposito, M., Crimaldi, M., Cirillo, V., Sarghini, F., & Maggio, A. (2021). Drone and sensor technology for sustainable weed management: A review. *Chemical and Biological Technologies in Agriculture*, 8(1), 18. <https://doi.org/10.1186/s40538-021-00217-8>
- [7] Khaki, S., Wang, L., & Archontoulis, S. V. (2020). A CNN-RNN framework for crop yield prediction. *Frontiers in Plant Science*, 10, 1750. <https://doi.org/10.3389/fpls.2019.01750>
- [8] Hou, Y. (2023). Optimization model of digital agricultural resources allocation based on pso algorithm. In *2023 IEEE 5th Eurasia Conference on IOT, Communication and Engineering*, 661–664. <https://doi.org/10.1109/ECICE59523.2023.10383140>
- [9] Wang, H. & Chen, C. (2024). Estimation of physical characteristics of unhealthy peach leaves using a two-step algorithm. In *2024 2nd International Conference on Intelligent Perception and Computer Vision*, 14–20. <https://doi.org/10.1109/CIPCV61763.2024.00013>
- [10] Shamshiri, R. R. & Shafian, S. (2022). Digital Agriculture, Methods and Applications. *Austri: Intech Open*. <https://doi.org/10.5772/intechopen.98141>
- [11] Upadhyay, P. & Chhabra, J. K. (2021). Multilevel thresholding based image segmentation using new multistage hybrid optimization algorithm. *Journal of Ambient Intelligence and Humanized Computing*, 12(1), 1081–1098. <https://doi.org/10.1007/s12652-020-02143-3>
- [12] Burke, J. & King, S. (2021). Edge tracing using Gaussian process regression. *IEEE Transactions on Image Processing*, 31, 138–148. <https://doi.org/10.1109/TIP.2021.3128329>
- [13] Mittal, H., Pandey, A. C., Saraswat, M., Kumar, S., Pal, R., & Modwel, G. (2022). A comprehensive survey of image segmentation: clustering methods, performance parameters, and benchmark datasets. *Multimedia Tools and Applications*, 81(24), 35001–35026. <https://doi.org/10.1007/s11042-021-10594-9>
- [14] Wang, H. & Chen, C. (2023, July). A two-step algorithm for denoising peach tree leaf images. In *2023 8th International Conference on Image, Vision and Computing*, 277–285. <https://doi.org/10.1109/ICIVC58118.2023.10270603>
- [15] Khattab, D., Ebied, H. M., Hussein, A. S., & Tolba, M. F. (2014). Color image segmentation based on different color space models using automatic GrabCut. *The Scientific World Journal*, 2014(1), 126025. <https://doi.org/10.1155/2014/126025>
- [16] Perez, F. & Koch, C. (1994). Toward color image segmentation in analog VLSI: Algorithm and hardware. *International Journal of Computer Vision*, 12(1), 17–42. <https://doi.org/10.1007/BF01420983>
- [17] Wu, J., Wang, X., Liu, Y., & Fang, C. (2024). Adaptive sparse regularized fuzzy clustering noise image segmentation algorithm based on complementary spatial information. *Expert Systems with Applications*, 256, 124943. <https://doi.org/10.1016/j.eswa.2024.124943>

- [18] Boyat, A. K. & Joshi, B. K. (2015). A review paper: noise models in digital image processing. *arXiv Preprint:1505.03489*. <https://doi.org/10.48550/arXiv.1505.03489>
- [19] Fan, L., Zhang, F., Fan, H., & Zhang, C. (2019). Brief review of image denoising techniques. *Visual Computing for Industry, Biomedicine, and Art*, 2(1), 7. <https://doi.org/10.1186/s42492-019-0016-7>
- [20] Jin, X. & Hirakawa, K. (2013). Approximations to camera sensor noise. In *Image Processing: Algorithms and Systems XI*, 8655, 149–155. <https://doi.org/10.1117/12.2019212>
- [21] Luisier, F., Blu, T., & Unser, M. (2010). Image denoising in mixed Poisson–Gaussian noise. *IEEE Transactions on Image Processing*, 20(3), 696–708. <https://doi.org/10.1109/TIP.2010.2073477>
- [22] Makitalo, M. & Foi, A. (2012). Optimal inversion of the generalized Anscombe transformation for Poisson–Gaussian noise. *IEEE Transactions on Image Processing*, 22(1), 91–103. <https://doi.org/10.1109/TIP.2012.2202675>
- [23] Jiang, B., Li, J., Lu, Y., Cai, Q., Song, H., & Lu, G. (2025). Efficient image denoising using deep learning: A brief survey. *Information Fusion*, 103013. <https://doi.org/10.1016/j.inffus.2025.103013>
- [24] Ashwinkumar, S., Rajagopal, S., Manimaran, V., & Jegajothi, B. J. M. T. P. (2022). Automated plant leaf disease detection and classification using optimal MobileNet based convolutional neural networks. *Materials Today: Proceedings*, 51, 480–487. <https://doi.org/10.1016/j.matpr.2021.05.584>
- [25] Fuentes, A., Yoon, S., Kim, S. C., & Park, D. S. (2017). A robust deep-learning-based detector for real-time tomato plant diseases and pests recognition. *Sensors*, 17(9), 2022. <https://doi.org/10.3390/s17092022>
- [26] Paithane, P. M. & Wagh, S. J. (2023). Novel modified kernel fuzzy c-means algorithm used for cotton leaf spot detection. <https://doi.org/10.20535/SRIT.2308-8893.2023.4.07>
- [27] Fan, X., Zhou, R., Tjahjadi, T., Das Choudhury, S., & Ye, Q. (2022). A segmentation-guided deep learning framework for leaf counting. *Frontiers in Plant Science*, 13, 844522. <https://doi.org/10.3389/fpls.2022.844522>
- [28] Li, M., Liao, Y., Lu, Z., Sun, M., & Lai, H. (2023). Non-destructive monitoring method for leaf area of Brassica napus based on image processing and deep learning. *Frontiers in Plant Science*, 14, 1163700. <https://doi.org/10.3389/fpls.2023.1163700>
- [29] Hernández-Hernández, J. L., García-Mateos, G., González-Esquiva, J. M., Escarabajal-Henarejos, D., Ruiz-Canales, A., & Molina-Martínez, J. M. (2016). Optimal color space selection method for plant/soil segmentation in agriculture. *Computers and Electronics in Agriculture*, 122, 124–132. <https://doi.org/10.1016/j.compag.2016.01.020>
- [30] Bhujade, V. G., Sambhe, V., & Banerjee, B. (2024). Digital image noise removal towards soybean and cotton plant disease using image processing filters. *Expert Systems with Applications*, 246, 123031. <https://doi.org/10.1016/j.eswa.2023.123031>
- [31] Jamjoom, M., Elhadad, A., Abulkasim, H., & Abbas, S. (2023). Plant leaf diseases classification using improved k-means clustering and SVM algorithm for segmentation. *Computers, Materials & Continua*, 76(1). <https://doi.org/10.32604/cmc.2023.037310>
- [32] Foi, A., Trimeche, M., Katkovnik, V., & Egiazarian, K. (2008). Practical Poissonian–Gaussian noise modeling and fitting for single-image raw-data. *IEEE Transactions on Image Processing*, 17(10), 1737–1754. <https://doi.org/10.1109/TIP.2008.2001399>
- [33] Ahamed, B. B., Yuvaraj, D., & Priya, S. S. (2019). Image denoising with linear and non-linear filters. In *2019 International conference on computational intelligence and knowledge economy*, 806–810. <https://doi.org/10.1109/ICCIKE47802.2019.9004429>
- [34] Yuan, W., Wijewardane, N. K., Jenkins, S., Bai, G., Ge, Y., & Graef, G. L. (2019). Early prediction of soybean traits through color and texture features of canopy RGB imagery. *Scientific Reports*, 9(1), 14089. <https://doi.org/10.1038/s41598-019-50480-x>
- [35] Armano, G. & Farmani, M. R. (2014). Clustering analysis with combination of artificial bee colony algorithm and k-means technique. *International Journal of Computer Theory and Engineering*, 141–145. <https://hdl.handle.net/11584/134250>
- [36] Müller, D., Hartmann, D., Meyer, P., Auer, F., Soto-Rey, I., & Kramer, F. (2022). MISeval: A metric library for medical image segmentation evaluation. In *Challenges of Trustable AI and Added-Value on Health*, 33–37. <https://doi.org/10.3233/SHTI220391>
- [37] Wang, H. & Chen, C. (2023). Estimation of physical characteristics of peach leaves using k-means clustering in the L* a* b* Color Space. In *Proceedings of the 2023 12th International Conference on Computing and Pattern Recognition*, 475–482. <https://doi.org/10.1145/3633637.3633712>
- [38] Abdelsadek, D. A., Al-Berry, M. N., Ebied, H. M., & Hassaan, M. (2022). Impact of using different color spaces on the image segmentation. In *International Conference on Advanced Machine Learning Technologies and Applications*, 456–471. https://doi.org/10.1007/978-3-031-03918-8_39
- [39] Garcia-Lamont, F., Cervantes, J., López, A., & Rodríguez, L. (2018). Segmentation of images by color features: A survey. *Neurocomputing*, 292, 1–27. <https://doi.org/10.1016/j.neucom.2018.01.091>
- [40] Kaur, A. (2012). A review paper on image segmentation and its various techniques in image processing. *International Journal of Science and Research*, 3(12), SUB14203.
- [41] Hua, L., Gu, Y., Gu, X., Xue, J., & Ni, T. (2021). A novel brain MRI image segmentation method using an improved multi-view fuzzy c-means clustering algorithm. *Frontiers in Neuroscience*, 15, 662674. <https://doi.org/10.3389/fnins.2021.662674>
- [42] Sammouda, R. & El-Zaart, A. (2021). An optimized approach for prostate image segmentation using K-means clustering algorithm with elbow method. *Computational Intelligence and Neuroscience*, 2021(1), 4553832. <https://doi.org/10.1155/2021/4553832>
- [43] Li, H., He, H., & Wen, Y. (2015). Dynamic particle swarm optimization and k-means clustering algorithm for image segmentation. *Optik*, 126(24), 4817–4822. <https://doi.org/10.1016/j.ijleo.2015.09.127>
- [44] Wang, Q., Du, W., Ma, C., & Gu, Z. (2021). Gradient color leaf image segmentation algorithm based on meanshift and kmeans. In *2021 IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference*, 1609–1614. <https://doi.org/10.1109/IAEAC50856.2021.9391029>
- [45] Cheng, K., Xiao, T., Chen, Q., & Wang, Y. (2020). Image segmentation using active contours with modified convolutional virtual electric field external force with an edge-stopping function. *Plos One*, 15(3), e0230581. https://doi.org/10.1007/978-981-15-8411-4_133
- [46] Wang, L., Zhu, Y., & Wang, C. (2020). A recognition algorithm based on region growing. In *International Conference in Communications, Signal Processing, and Systems*, 1013–1019. https://doi.org/10.1007/978-981-15-8411-4_133
- [47] Bangare, S. L., Dubal, A., Bangare, P. S., & Patil, S. (2015). Reviewing Otsu’s method for image thresholding. *International Journal of Applied Engineering Research*, 10(9), 21777–21783.

- [48] Fan, W. (2022). Research on adaptive segmentation algorithm of sports image based on support vector machine. In *Proceedings of the 3rd Asia-Pacific Conference on Image Processing, Electronics and Computers*, 920-924. <https://doi.org/10.1145/3544109.3544381>
- [49] Syriopoulos, P. K., Kalampalikis, N. G., Kotsiantis, S. B., & Vrahatis, M. N. (2025). KNN classification: A review. *Annals of mathematics and artificial intelligence*, 93(1), 43-75. <https://doi.org/10.1007/s10472-023-09882-x>
- [50] Abenina, M. I. A., Maja, J. M., Cutulle, M., Melgar, J. C., & Liu, H. (2022). Prediction of potassium in peach leaves using hyperspectral imaging and multivariate analysis. *AgriEngineering*, 4(2), 400-413. <https://doi.org/10.3390/agriengineering4020027>

How to Cite: Chen, C., & Wang, H. (2025). Estimation of Physical Characteristics of Noisy Peach Leaves Using a Unified Algorithm. <i>Artificial Intelligence and Applications</i>. https://doi.org/10.47852/bonviewAIA52025928
--

Appendix A. Gaussian and Poisson noise models

1) Gaussian noise model

Gaussian noise is a common type of noise in digital images. It comes from sensor imperfections, the environment, and the electronic devices during the image-capturing process. It also follows a normal distribution, so the probability of a pixel intensity changing from its expected value. Gaussian noise is added to each pixel, resulting in a corrupted image with pixel intensities varying around their true values. Gaussian noise is modeled by a probability density function (PDF):

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

where x represents the pixel value, μ is the mean (the average noise level), and σ is the standard deviation (the noise variance).

Gaussian noise is additive and independent of the signal strength. It can obscure fine image details, making effective denoising techniques crucial for image processing tasks.

2) Poisson noise model

Poisson noise, also called shot noise, is caused by the statistical nature of photon detection during image acquisition. As observed in low-light conditions, fewer photons were captured, resulting in random fluctuations in image intensities. Poisson noise depends on the signal. Therefore, its variance increases with the signal intensity. This makes brighter parts of the image appear noisier. The Poisson noise is modeled for a pixel value x as follows:

$$P(x = k) = \frac{e^{-\lambda} \lambda^k}{k!} \quad \text{for } k = 0, 1, 2, \dots$$

where λ is the mean and variance of the distribution. It represents the expected number of photon arrivals at a pixel. The statistical nature of Poisson noise makes it more difficult to remove than Gaussian noise, requiring special denoising methods to maintain image quality.

Appendix B. Segmentation algorithms

For each pixel p , we construct a feature vector in the CIELAB space $x_p = [L_p, a_p, b_p]^T \in \mathbb{R}^3$, which is obtained by RGB to $L^*a^*b^*$ mapping. All segmentation algorithms in this section operate on x_p .

1) Active contour model

The active contour model, also known as a snake, involves the dynamic evolution of a curve in 2D or a surface in 3D, guided by both internal and external forces. Its objective is to adjust the contour to align precisely with the boundaries of objects under consideration. Despite its advantages of flexibility, user interaction facilitation, and ability to handle complex shapes, it is sensitive to initialization, tends to converge to local minima, and incurs high computational costs [45].

2) Region growing algorithm

The region growing algorithm is a pixel-based method. It groups pixels based on predefined similarity criteria. In this study, the pixels with the maximum variation from the mean value of the initially selected seed region are added to the growing regions. This method is simple and easy to adapt and tune parameters. However the initial seed selection can affect the segmentation results and may lead to over-segmentation or under-segmentation. It is also sensitive to noise [46]. The general procedure for implementing active contours is as follows:

Step 1: Initial Contour Selection

Start with a chosen initial contour within a dataset.

Step 2: Energy minimization

Minimize the energy function:

$$E_{\text{snake}}^* = \int_0^1 E_{\text{snake}}(x_s) ds \\ = \int_0^1 (E_{\text{internal}}(x_s) + E_{\text{image}}(x_s) + E_{\text{con}}(x_s)) ds$$

where E_{internal} is internal energy, E_{image} is external energy, E_{con} is energy from the constraint forces, and x_s is a data point.

Step 3: Output Step

Output: The final contour once it stabilizes or the maximum number of iterations is reached.

3) Multilevel image thresholds

The method of multilevel thresholding uses Otsu's method and assumes that an image has pixels that belong to the foreground or the background. Its objective is to identify an optimal threshold that minimizes intra-class variance or maximizes inter-class variance. This method is inherently simple and efficient for multimodal images, and it autonomously determines threshold values. However, it is constrained by the assumption of a bimodal distribution and is sensitive to noise [11]. The general process is outlined below:

Step 1: Compute the histogram and probabilities of each intensity level in an image.

Step 2: Set up the inter-class variance formula.

Step 3: Compute the inter-class variance.

Step 4: Choose thresholds that maximize inter-class variance.

Step 5: Output: the segmented image with minimum intra-class variance.

4) Global image threshold

Global image threshold, another method based on Otsu's idea, finds the optimal global threshold of a grayscale image [47]. It gives each pixel a binary value (0 or 1) based on the threshold.

5) Adaptive image threshold

The adaptive image threshold sets the threshold for each pixel based on the local mean intensity in its neighborhood [48]. This method has the advantages of adaptability and detail preservation. But it also has disadvantage such as noise sensitivity, computational complexity and dependence on the homogeneity assumption. The procedure is as follows:

Step 1: Define the neighborhood size.

Step 2: Compute the local mean intensity within the neighborhood.

For a pixel at location (x, y) , the adaptive thresholding can be defined as follows:

$$T(x, y) = \frac{1}{M \times N} \sum_{i=-a}^a \sum_{j=-b}^b I(x+i, y+j)$$

where M and N are the sizes of the neighborhood, $I(x+i, y+j)$ is the intensity of the image at location $(x+i, y+j)$, and a and b are the distances from the central pixel to the edges of the neighborhood in the x and y directions.

Step 3: Set the threshold. If the pixel at location (x, y) is $I(x, y) \geq T(x, y)$, then it is set to white; otherwise, it is set to black.

Step 4: Output: the segmented image based on the threshold.

K-nearest neighbors

The K-nearest neighbors (KNN) algorithm is used for pixel-by-pixel classification tasks. Each pixel is classified based on

its similarity to a set of pre-labeled training data samples. KNN is simple, requires few parameters to tune, and is capable of handling multiple class segmentation. However, its drawbacks include high computational intensity and sensitivity to parameter K, which may lead to over-segmentation or under-segmentation [49]. The general process can be outlined as follows:

Step 1: Select a value for K.

Step 2: Compute the distance between the new input and all training samples.

Step 3: Select K samples with the smallest distance to the input.

Step 4: Assign the most frequent class label from K neighbors to its input.

Step 5: Output: the segmented image with each pixel assigned a predicted class label.