A Two-Step Algorithm for Denoising Peach Tree Leaf Images

***Abstract*—In the study, we investigate and analyze different denoising algorithms with specific application on peach tree leaf images with mixture of Gaussian and Poisson noise. Eleven diverse noise models and twenty-three distinct denoising filters are reviewed in depth. We propose a novel metric function based on three standard metrics to ensure a comprehensive evaluation of the filters, and develop a two-step procedure from the selected filters to reduce noise in the real image. The results demonstrated by both the artificial and real images show the effectiveness and viability of the approach.**

***Index Terms*—noise, denoising, filter, Gaussian, Poisson, peach, leaves**

I. Introduction

Agriculture significantly impacts the global economy. Digital agriculture emerged as a new scientific field uses data processing approaches to improve agricultural practice while minimizing labor-intensive cost and environmental impact. Image processing as one of the high-performance data processing technologies, will play an important role in gaining a deeper understanding by measuring the data, closely examining the data, and interpreting the results. Compared to traditional methods, image processing is more accurate and less time-consuming in agriculture areas [1], [2].

Image noise introduces unwanted information. Therefore, image noise processing poses a challenge in the application of image processing technology. It is one key data preprocessing step for any subsequent applications including tool design. Various image denoising models have been developed. The image denoising models are categorized into models shown in Fig. 1 [3]–[8]. Images captures from digital cameras and smart phones have noise originating from image sensors. The noise is further categorized into fixed pattern noise, banding noise, and random noise. The fixed pattern noise is from sensor measurements, which are the stochastic nature of photon counting. Banding noise is from a bank of analog-to-digital (A/D) converters. Random noise is from photon emission, intrinsic thermal and electronic fluctuations, and other sources. In the reality, the noise distribution in camera images follows Poisson and Gaussian distributions with intensity dependent variance [7], [9]. The various noise modes in image processing are listed below in Table I.

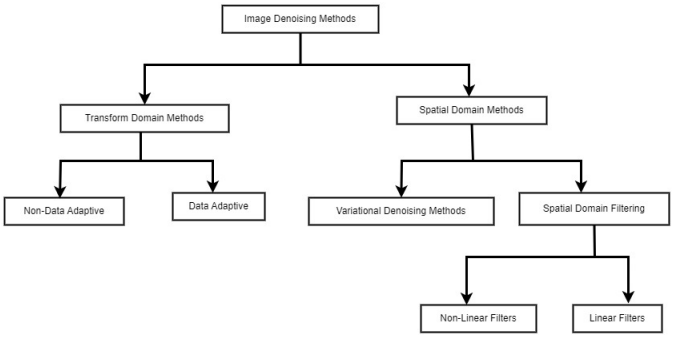


Fig. 1: The denoising categories.

TABLE I: Noise Models

Noise Model	Noise Source(s)/Characteristic(s)
Gaussian Noise	Noise generated by natural sources
White Noise	Constant noise power spectrum and zero auto-correlation
Fractal Noise	Brownian motion with nonstationary stochastic procedure
Salt and Pepper Noise	Noisy pixel value 0 or 255
Periodic Noise	Electronic interference
Quantization Noise	The conversion procedure from the analog data to digital data
Speckle Noise	Coherent imaging system
Poisson Noise	The statistical nature of electromagnetic waves
Poisson-Gaussian Noise	Magnetic resonance images
Gamma Noise	Laser images
Raleigh Noise	radar images

A. Gaussian noise model

The probability density function (PDF) is as follows.

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

where x is a variable representing the red(R), green(G), blue(B) or gray value, σ denotes the standard deviation, and μ represents the mean.

B. Poisson noise model

The PDF is the following.

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

λ is the mean of distribution. The mean and variance of the Poisson distribution are both equal to λ, which can be written as:

$$\mathrm{E}(x) = \mathrm{Var}(x) = \lambda$$

The contributions of this paper include:

1) Comparative analysis of state-of-arts denoising algorithms based on the real digital images noise models.
2) Propose a metric function to select the most effective denoising algorithm.
3) Propose a two-step method to address the real image noise
4) Application of obtained results to denoise images of the peach tree leaves

This paper is organized as follows. In section II, twenty-three denoising filters are reviewed with their respective formulas. In section III, we propose a novel metric function incorporating various metrics and real image noises. In section IV, we propose the two-step denoising algorithms with simulations.

II. Denoising filters

Denoising filters are required for images before they are used in applications, such as segmentation [5], [10]. Image filters are applied to reduce noise and preserve details including edges. In this research, twenty-three denoising filters used are discussed below. Their pros and cons are discussed in details with their formulas.

A. Ideal frequency filter

The ideal frequency filter removes frequency with a certain distance from the center of frequency domain. However, it has the limitation of causing ringing artifacts [4].

$$H(u,v) = \begin{cases} 1 & \text{if } D(u,v) \le D_0 - \frac{\omega}{2} \\ 0 & \text{if } D_0 - \frac{\omega}{2} < D(u,v) \le D_0 + \frac{\omega}{2} \\ 1 & \text{if } D(u,v) > D_0 + \frac{\omega}{2} \end{cases}$$

where u and v are the frequency coordinates, $H(u,v)$ is the ideal frequency filter response at (u,v), $D(u,v)$ is the distance between (u,v) and the cutoff frequency D_0, ω is the width of the transition region.

B. Ideal low-pass frequency filter

The ideal low-pass frequency filter in image processing passes all low frequency components with the radius D_0. It is also subject to the limitation causing ringing artifacts. Additionally, it has the sharp transition between the passband and stopband [11].

$$H(u,v) = \begin{cases} 1 & \text{if } D(u,v) \le D_0 \\ 0 & \text{if } D(u,v) > D_0 \end{cases}$$

C. Ideal high-pass frequency filter

Compared to the ideal low-pass frequency filter, the ideal high-pass frequency filter passes all the high-frequency components [12]. It has the same limitation as low-pass frequency filter.

$$H(u,v) = \begin{cases} 0 & \text{if } D(u,v) \le D_0 \\ 1 & \text{if } D(u,v) > D_0 \end{cases}$$

D. Ideal Notch filter

An ideal notch filter passes or rejects a narrow set of frequencies with the center frequency (u_0, v_0). The frequency centers can be more than one [13].

$$H(u,v) = \begin{cases} 0 & \text{if } D_1(u,v) \le D_0 \, or \, D_2(u,v) \le D_0 \\ 1 & \text{if } otherwise \end{cases}$$

where

$$D_1(u,v) = \left[(u - \omega_1/2 - u_0)^2 + (v - \omega_2/2 - v_0)^2 \right]^{1/2}$$

,

$$D_2(u,v) = \left[(u - \omega_1/2 + u_0)^2 + (v - \omega_1/2 + v_0)^2 \right]^{1/2}$$

, and ω_1 and ω_2 are band width.

E. Geometric mean filter

Geometric mean filter is a non-linear filter to reduce noise. It takes the geometric mean of the pixels in a neighborhood around each pixel. It effectively reduces multiplicative noise [14], [15].

$$G^{'}(x,y) = \left[\prod_{i=-a}^{a} \prod_{j=-b}^{b} G(x+i, y+j) \right]^{\frac{1}{mn}}$$

where $G^{'}(x,y)$ is the output pixel value at position (x,y), a,b,m,and n are the filter window sizes.

F. Median order-statistic filter

The median order-statistic filter is one non-linear filter. It replaces each pixel value by the median value in its surrounding area [16].

$$G^{'}(x,y) = \text{median}\left\{ G(x+i, y+j) : i = -a, \ldots, a, j = -b, \ldots, b \right\}$$

G. Adaptive noise smoothing filter

The adaptive noise smoothing (ANS) filter is a nonlinear filter used to reduce noise in an image while preserving its edges and details [17]. One example is shown below:

$$h(x,y) = \exp(\frac{-d(x,y)}{2})$$

where $h(x,y)$ is an ANS mask matrix coefficient.

$$d(x,y) = \sqrt{\begin{array}{l} \frac{1}{2}[G(x+1,y) - G(x-1,y)]^2 + \\ \frac{1}{2}[G(x,y+1) - G(x,y-1)]^2 \end{array}}$$

H. Adaptive median filter

The Adaptive Median Filter (AMF) is a nonlinear filter. AMF is similar to ANS in reducing image noise while preserving its edges and other details [18].

$$G^{'}(x,y) = \begin{cases} z_{med} & \text{if } z_{min} < z(x,y) < z_{max} \\ G(x,y) & \text{otherwise} \end{cases}$$

where z_{med} is the median of pixels in the local window. z_{min} is the minimum pixel value in the local window. z_{max} is the maximum pixel value in the local window. The local window size is adaptively increased from a minimum window size to a maximum window size until a pixel within the window is found whose value falls between z_{min} and z_{max}.

I. Neighborhood mean filer

The neighborhood mean file employs a modified window to compute the mean and compare the mean and central pixel [4]. The equation is shown below.

$$G^{'}(x,y) = \begin{cases} \frac{1}{N} \sum\limits_{(i,j)\in S_{x,y}} G(i,j) \\ \text{if } G(x,y) < (\frac{1}{N} \sum\limits_{(i,j)\in S_{x,y}} G(i,j) + T) \\ G(i,j) \\ otherwise \end{cases}$$

where n is the number of pixels in the window, $S_{x,y}$ is the neighborhood window, and T is the threshold.

J. Harmonic mean filter

The filter operates by replacing the pixel value at each location in the image with the harmonic mean of the pixel values in a surrounding neighborhood [19]. The equation is shown below.

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

K. Inverse harmonic mean filter

Inverse harmonic mean filter is a nonlinear filer. It effectively reduces impulsive noise or salt-and-pepper noise [4]. The equation is shown below.

$$G^{'}(x,y) = \frac{\sum_{(i,j)\in S_{x,y}} G(i,j)^{Q+1}}{\sum_{(i,j)\in S_{x,y}} G(i,j))^{Q}}$$

where Q is the order of the inverse harmonic mean filter and controls the amount of noise reduction. When $Q > 0$, the filter is effective for pepper noise. When $Q < 0$, the filter is effective for salt noise. When $Q = 0$, the inverse harmonic mean filter reduces to the arithmetic mean filter. When $Q = -1$, the inverse harmonic mean filter reduces to the harmonic mean filter.

L. Inverse filter

The inverse filter is able to recover the original image by undoing the blurring [20]. For the inverse filter, we have the complex blurring function $h(x,y)$ with the 2-D fast Fourier transform $H(u,v)$.

$$H(u,v) = \begin{cases} \frac{1}{H_0(u,v)} & \text{if } H_0(u,v) \neq 0 \\ 0 & \text{otherwise} \end{cases}$$

$$F(u,v) = G(u,v) \times H(u,v)$$

$$G^{'}(x,y) = \mathcal{F}^{-1}\{F(u,v)\}(x,y)$$

where $G^{'}$ is the recovering image.

M. Bilateral filter

The bilateral filter is a nonlinear image filter which smooths the image while preserving the edges and other details [21].

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

where $w(i,i) = w_s(i,j)w_r(i,j)$,
$w_s(i,j) = \exp(-\frac{|i-x|^2+|j-y|^2}{2\sigma_s^2})$
$w_r(i,j) = \exp(-\frac{|g(i,j)-g(x,y)|^2}{2\sigma_r^2})$
σ_s and σ_r are the spatial and range standard deviations.

N. Homomorphic filter

The homomorphic filter enhances the contrast of an image by adjusting its illumination($i(x,y)$) and reflectance($r(x,y)$) components [22]. $r(x,y)$ captures edges and other details. $i(x,y)$ represents illumination. The equation for homomorphic filtering is$G(x,y) = i(x,y)r(x,y)$. In homomorphic filtering, we first transform the multiplicative components to additive components by moving to the log domain: $ln(G(x,y)) = ln(I(x,y)) + ln(R(x,y))$. A high-pass filter ($H(u,v)$)is applied to the image in frequency domain with the equation: $H(u,v)G(u,v) = H(u,v)I(u,v) + H(u.v)I(u.v)$.The homomorphic filter is defined as:

$$G^{'}(x,y) = \mathcal{F}^{-1}H(u,v)G(u,v)$$

O. Morphological filter

The morphological filter is a non-linear filter with a morphological operation applied to the image [23]. Four common operations are:

- Erosion: The operation erodes away the boundaries of objects.
- Dilation: The operation expands the boundaries and shrinks the holes.
- Opening: The operation combines the erosion followed by dilation. It is used to remove small objects and smooth the boundaries of large objects.
- Closing: The operation combines the dilation followed by erosion. It is used to fill small holes in the image.

The erosion filter is defined as:

$$(G \ominus B)(x,y) = \begin{cases} 1 & \text{if B fits G} \\ 0 & \text{otherwise} \end{cases}$$

where G is the input image, B is the structuring element, and $\ominus$ denotes the erosion operation.
Dilation filter is defined as:

$$(G \oplus B)(x,y) = z|[(\hat{B})_z \bigcap G] \subseteq G$$

where $\hat{B}_z$ is the reflection of B about the origin and shift the reflection by z.
Opening filter is defined as:

$$G \circ B = (G \ominus B) \oplus B$$

Closing filter is defined as:

$$G \bullet B = (B \oplus B) \ominus B$$

P. Constrained least squares Filter

Constrained least squares filter reduces noise, enhances the quality of an image, and preserves key features and details [24]. The transfer function in frequency domain is the following.

$$G(u,v)^{'} = \left[\frac{H^*(u,v)}{|H(u,v)|^2 + \gamma|P(u,v)|^2}\right] G(u,v)$$

where $H(u,v)$ is the degradation function, $H^*(u,v)$ is its complex conjugate,γ is a parameter adjusted to get optimal visual result, and $P(u,v)$ is the Fourier transform of the Laplacian operator $p(x,y)$:

$$p(x,y) = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

Q. Nonlinear complex diffusion filter

The nonlinear complex diffusion filter removes noise and enhances details based on diffusion equation [25].

$$I_t = \frac{dI}{dt} = \nabla(D(\nabla I)\nabla I)$$

where I_t is the image change with time t, ∇ is the gradient operator, and D is the divergence operator.
$D = \frac{e^{i\theta}}{1+[\frac{Im(I)}{k\theta}]^2}$
where θ is a parameter that controls the amount of diffusion. When $k > 0$, θ approaches zero.

R. Gabor filter

The Gabor filter enhances edges and textures in images. It selectively passes frequencies within a certain band and attenuates frequencies outside band [26].
The equation for the one-dimensional Gabor filter is as follows:

$$G(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{x^2}{2\sigma^2}\right) (\cos(2\pi w_0 x) + i\sin(2\pi w_0 x))$$

where σ is the standard deviation of the Gaussian window, w_0 is the center frequency.
The two-dimensional Gabor filter is defined by the following equation.

$$G(x,y) = \frac{1}{\sqrt{2\pi}\sigma_x\sigma_y} \exp\left(-\frac{x^2}{2\sigma_x^2} - \frac{y^2}{2\sigma_y^2}\right)$$
$$(\cos(2\pi w_{x_0} + 2\pi w_{y_0} x) + i\sin(2\pi w_{x_0} + 2\pi w_{y_0} x))$$

S. Wiener filter

The Wiener filter is a linear filter that minimizes the mean squared error between the estimated signal and the original signal. It can be applied to images with additive noise and blurring [27]. The Wiener filter is defined as follows.

$$G(u,v) = \frac{H^*(u,v)}{|H(u,v)^2| + \frac{P_n(u,v)}{P_s(u,v)}}$$

where $H(u,v)$ is the Fourier transform of the point-spread function, $P_s(u,v)$ is the power spectrum of the signal process, and $P_n(u,v)$ is the power spectrum of the noise process.

T. Kuwahara filter

The Kuwahara filter reduces image noise and preserves edges and textures. It divides the image into four non-overlapping square regions and calculates the local mean and variance within each region. The filtered pixel value is then determined based on the region with the minimum variance [28]. The output of Kuwahara filter is as follows.

$$\phi(x_0,y_0) = \sum_i m_i(x_0,y_0) f_i(x_0,y_0)$$
$$f_i(x_0,y_0) = \begin{cases} 1 & s_i(x_0,y_0) \leqslant s_k(x_0,y_0), \forall k \\ 0 & \text{Otherwise} \end{cases}$$

where $i = 1,2,3,4$, m_i is the mean value of i_{th} local regions centered at (x_0,y_0), and s_i is the variance.

U. Beltrami filter

For Beltrami filter, the image is embedded. For gray images,

$$X^1 = x^1$$
$$X^2 = x^2$$
$$X^3 = \mathbf{I}(x^1 + x^1)$$

where x^1 and x^2 are the space coordinates and $\mathbf{I}$ is the intensity component [29].
The Beltrami flow is obtained by minimizing the area of the image manifold.
$P = \iint \sqrt{g} dx_1 dx_2$ where g is the metric element, given by: $g = det(G) = g_{11}g_{22} - g_{12}^2$

V. Lucy-Richardson filter

The Lucy–Richardson filter is an iterative method for recovering an image blurred by a known point spread function (PSF) [30].The equation for the Lucy-Richardson filter can be expressed as follows:

$$f^{(k+1)} = f^{(k)} \cdot \left[\frac{g}{(f^{(k)} \otimes P)} \otimes P^*\right]$$

where $\otimes$ is a 2D convolution, and P^* is the flipped point spread function.

W. Non-local means (NLM) filter

The non-local means filter preserves image edges and other important features while effectively reducing noise. It works by averaging the pixel values in similar patches across the image, instead of only considering local neighborhoods [31]. The equation for the non-local means filter can be represented as follows:

$$NLu(\bar{x}) = \frac{1}{C(\bar{x})} \int f(d(B(\bar{x}), B(\bar{y}))u(\bar{y})d\bar{y}$$

where $d(B(\bar{x}), B(\bar{y})$ is the Euclidean distance between the image patches centered at $\bar{x}$ and $\bar{y}$, f is a non-increasing function, and $C(\bar{x})$ is the normalizing factor.

III. Metric function to select denoising algorithms

In order to choose the 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 three standard metrics are used to evaluate the performance of denoising algorithms.

A. Mean squared error

Mean squared error is a metric used to evaluate the performance of a denoising algorithm by measuring the average discrepancy between the denoised image and the original noise-free image [32]. It quantifies the average squared difference between corresponding pixels in the two images. It provides a quantitative measure of denoising algorithm, with lower MSE indicating better performance. However, MSE alone does not capture perceptual quality without considering human visual perception, for example, changes in the structure of the image . The equation for MSE is represented as follows:

$$M1 = \frac{1}{N} \sum_{i=1}^{N} (G(i) - \hat{G}(i))^2$$

where, $M1$ represents the Mean Squared Error, N is the total number of pixels in the image, $G(i)$ represents the observed pixel value, and $\hat{G}(i)$ represents the predicted value.

B. Peak signal-to-noise ratio

The peak signal-to-noise ratio measures the ratio between the maximum possible power of a signal and the power of the distortion introduced by the denoising algorithm [33]. The equation for PSNR is represented as follows:

$$M2 = 10 \cdot \log_{10} \left(\frac{\text{MAX}^2}{\text{M1}} \right)$$

where $M2$ represents the PSNR, and MAX denotes the maximum possible pixel value of the image.
The PSNR is calculated as the logarithm (base 10) of the ratio between the squared maximum possible pixel value and MSE. The logarithmic scale is used to provide a more perceptually meaningful representation of the quality difference. Higher PSNR values indicate better quality, as they indicate a lower level of distortion introduced by the denoising process. However, PSNR is not always consistent with human visual perception, and higher PSNR values do not always guarantee better subjective image quality. When using PSNR as a metric, it is important to consider the limitations and drawbacks, such as its sensitivity to minor differences and inability to capture the structural information in the image.

C. Structural similarity index measure

The structural similarity index measure is used to assess the quality of a denoised image by comparing it to the original noise-free image [33]. SSIM includes both the structural information and the perceived similarity between the images. It measures the similarity between the local patches of the two images, considering luminance, contrast, and structure. The SSIM index ranges between -1 and 1, with a value of 1 indicating a perfect similarity, 0 indicating no similarity, and -1 indicating perfect anti-correlation. SSIM is more reliable than metrics based solely on pixel values. The equation for Structural Similarity Index Measure (SSIM) can be represented as follows:

$$\text{S1}(x, y) = \frac{(2\mu_x\mu_y + C_1) \cdot (2\sigma_{x,y} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1) \cdot (\sigma_x^2 + \sigma_y^2 + C_2)}$$

where, S1 represents SSIM, x and y are two measure windows with size $N \times N$, μ_x and μ_y denote the pixel sample means, σ_x^2 and σ_y^2 represent the standard deviations of the x and y, respectively, $\sigma_{x,y}$ represents the cross-covariance between the x and y, C_1 and C_2 are constants added to stabilize the division and avoid division by zero.

D. Metric function

Different metrics have their limitation. It is necessary to propose a metric function to evaluate the denoising algorithms. Based on three metrics, the general metric function is defined as follows.

$$Metric = f(w_1 M1, w_2 M2, w_3 S1) \tag{1}$$

where $w_i, i = 1, 2, \text{and} 3$ are weight values based on the actual application.
The general procedure to deal with the noise is by comparing the different denoising algorithms based on proposed standards with Gaussian and Poisson distributions, and the optimal denoising algorithm is selected, respectively. The general procedure is shown in Fig. 2.
In the flowchart, the values of $M1, M2$, and $S2$ are normalized between $[0, 1]$ based on the formula

$$normalized\_value = \frac{value - minimum\_value}{maximum\_value - minimum\_value}$$

In equation 1, w_3 is adjusted to 50% due to the similarity between metrics MSE and PSNR. A value of $f = 1$ represents

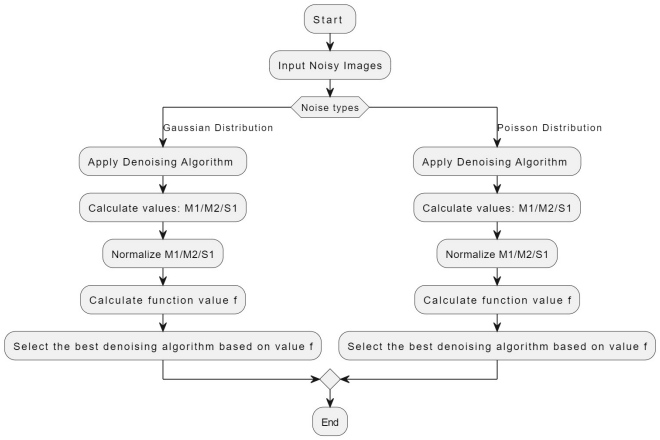


Fig. 2: The selection of denoising algorithm based on value f

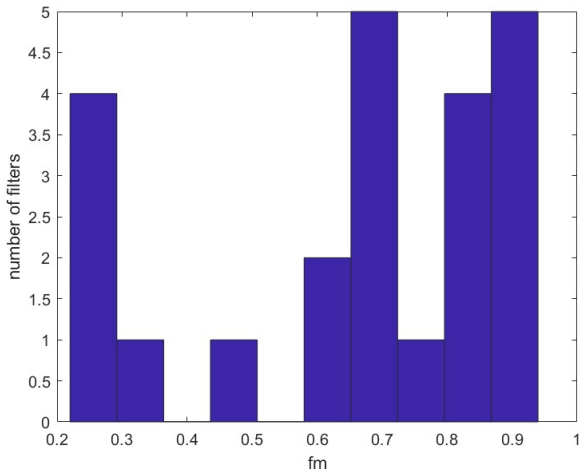


Fig. 3: The distribution of filters based on fm

perfect similarity.

In the simulation sections, a peach tree leave without noise is adopted to verify the procedure. The result is shown in Table II. $f1$ is the value of f with Gaussian noise, $f2$ is the value of f with Poisson noise, fm is the mean value of $f1$ and $f2$. Based on the result, bilateral filter is identified as the optimal denoising algorithm for the peach tree leaf with Gaussian noise, and inverse harmonic mean filter (0.969858048) is identified as the optimal algorithm with Poisson noise. The mean value of fm is 0.65. The distribution of fm is shown in Fig. 3, with the majority of values falling within the ranges of [0.65, 0.72] or [0.87, 0.94].

TABLE II: Denoising algorithm with f values

	Denoising algorithms	f1	f2	fm
1	Ideal frequency filter	0.70	0.64	0.67
2	Ideal low-pass frequency filter	0.93	0.85	0.89
3	Ideal high-pass frequency filter	0.34	0.30	0.32
4	Ideal Notch filter	0.70	0.64	0.67
5	Geometric mean filter	0.79	0.97	0.88
6	Median order-statistic filter	0.49	0.46	0.48
7	Adaptive noise smoothing filter	0.82	0.74	0.78
8	Adaptive median filter	0.87	0.79	0.83
9	The neighborhood mean filter	0.87	0.79	0.83
10	Harmonic mean filter	0.75	0.96	0.85
11	Inverse harmonic mean filter	0.90	0.97	0.93
12	Inverse filter	0.24	0.21	0.22
13	Bilateral filter	0.99	0.90	0.94
14	Homomorphic filter	0.72	0.66	0.69
15	Morphological filter	0.60	0.84	0.72
16	Constrained least squares Filter	0.24	0.21	0.22
17	Nonlinear complex diffusion filter	0.85	0.78	0.81
18	Gabor filter	0.66	0.60	0.63
19	Wiener filter	0.95	0.87	0.91
20	Kuwahara filter	0.24	0.21	0.22
21	Beltrami filter	0.24	0.21	0.22
22	Lucy-Richardson filter	0.66	0.60	0.63
23	Non-local means filter	0.70	0.64	0.67

The simulation results from a peach tree leaf are shown in Fig. 4.

IV. Two-step denoising algorithm design

Based on the simulation results from Table II, the bilateral filter and the inverse harmonic mean filter are identified as the optimal filters for Gaussian noise and Poisson noise , respectively. Based on the fm values, the inverse harmonic mean filter, the bilateral filter, and the Wiener filter are chosen to test the denoising effect on the images with those two types of noise using the procedure of Fig. 5.

The simulation results for three denoising filters with their corresponding f values are shown in Table III. The two-step procedure can preserve image details, reduce noise artifacts, and maintain structural integrity.

The denoising outcomes achieved for images of the peach tree leaf are shown in Fig. 6.

TABLE III: Denoising algorithms with f values and the real image noise

	Denoising algorithms	f value
1	Bilateral filter	0.9420
2	Inverse harmonic mean filter	0.2836
2	Wiener filter	0.8841

Based on obtained simulation results, two-step denoising procedure is proposed based on the bilateral filter and Wiener filter. Two-step denoising procedure is shown in Fig 7 on the real image and initially validated by the artificial image.

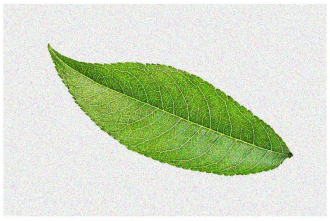
The denoising outcome is $f = 0.9449$ for the artificial image, and the figure is shown in Fig. 8.

The two-step denoising procedure is applied to the real images. The result is shown in Fig. 9.

V. Conclusion

Twenty-three denoising algorithms are studied under Gaussian noise and Poisson noise, which are encountered in real images. The optimal denoising algorithm are found based on

(a)



(b)

(c)

(d)

(e)

Fig. 4: a:The image of a peach tree leaf without noise. b: the image with Gaussian noise. c: the image with Poisson noise. d: the image with Gaussian noise denoising with the bilateral filter. e: the image with Poisson noise denoising with the inverse harmonic mean filter

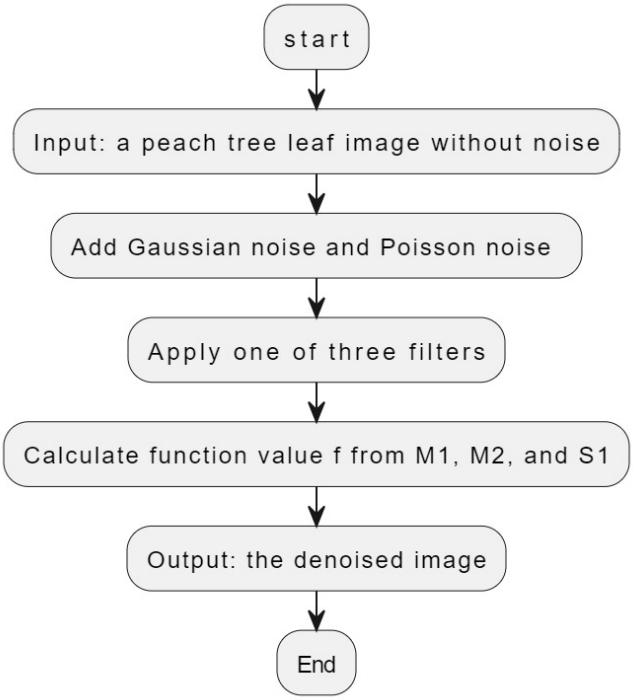


Fig. 5: The flow chart for the real image noise processing

each single noise with the proposed metric function based on MSE, PSNR, and SSIM. A two-step procedure is proposed to address the noise problem in the real image, especially for the peach tree leaves. The result is improved compared to noisy images, and demonstrates the effectiveness of the two-step procedure. As part of future work, deep learning will be integrated to enhance for the denoising procedure.

ACKNOWLEDGMENT

The research is supported by USDA 1890 Research Sabbatical Program.

(a)

(b)

(c)

(d)

Fig. 6: a:the image with Gaussian and Poisson noise. b: the image with both noises after the bilateral filter denoising. c: the image with both noises after the inverse harmonic mean filter processing. d: the image with both noises after Wiener filter processing

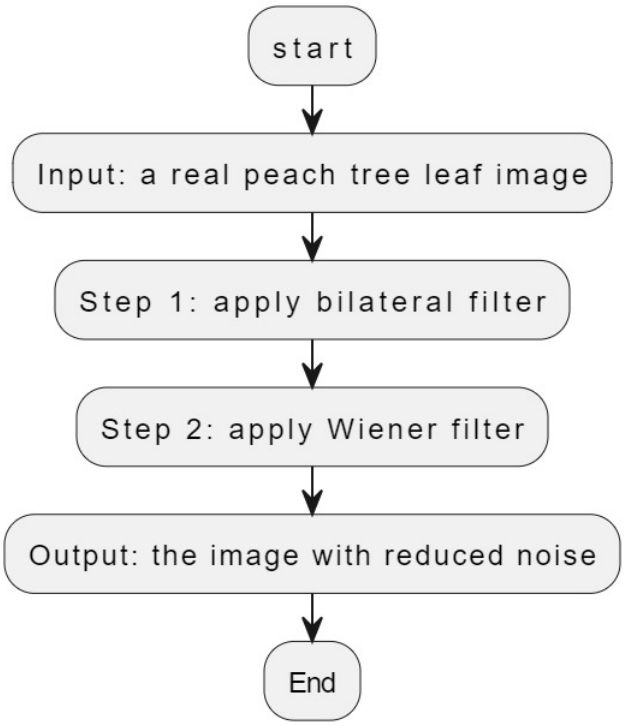


Fig. 7: The two-step denoising algorithm

Fig. 8: The image from two-step procedure

(a)

(b)

Fig. 9: a:the real image before two-step procedure. b: the real image after two-step procedure

REFERENCES

[1] A. Vibhute and S. K. Bodhe, "Applications of Image Processing in Agriculture: A Survey," *International Journal of Computer Applications*, vol. 52, no. 2, pp. 34–40, Aug. 2012.

[2] K. Liakos, P. Busato, D. Moshou, S. Pearson, and D. Bochtis, "Machine Learning in Agriculture: A Review," *Sensors*, vol. 18, no. 8, p. 2674, Aug. 2018.

[3] A. K. Boyat and B. K. Joshi, "A Review Paper : Noise Models in Digital Image Processing," *Signal & Image Processing : An International Journal*, vol. 6, no. 2, pp. 63–75, Apr. 2015.

[4] S. Xu, J. Ding, T. Wu, and J. Wei, *Image Filters and Denoising Analysis and Applications in Matlab*. Beihang University Press, 2015.

[5] L. Fan, F. Zhang, H. Fan, and C. Zhang, "Brief review of image denoising techniques," *Visual Computing for Industry, Biomedicine, and Art*, vol. 2, no. 1, p. 7, Dec. 2019.

[6] X. Jin and K. Hirakawa, "Approximations to camera sensor noise," in *IS&T/SPIE Electronic Imaging*, K. O. Egiazarian, S. S. Agaian, and A. P. Gotchev, Eds., Burlingame, California, USA, Feb. 2013, p. 86550H.

[7] F. Luisier, T. Blu, and M. Unser, "Image Denoising in Mixed Poisson–Gaussian Noise," *IEEE Transactions on Image Processing*, vol. 20, no. 3, pp. 696–708, Mar. 2011.

[8] M. Makitalo and A. Foi, "Optimal Inversion of the Generalized Anscombe Transformation for Poisson-Gaussian Noise," *IEEE Transactions on Image Processing*, vol. 22, no. 1, pp. 91–103, Jan. 2013.

[9] J. Anaya, "First steps towards Image Denoising under low-light conditions."

[10] B. Ahamed, D. Yuvaraj, and S. S. Priya, "Image Denoising With Linear and Non-linear Filters," in *2019 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE)*, Dec. 2019, pp. 806–810.

[11] I. W. Selesnick, H. L. Graber, D. S. Pfeil, and R. L. Barbour, "Simultaneous Low-Pass Filtering and Total Variation Denoising," *IEEE Transactions on Signal Processing*, vol. 62, no. 5, pp. 1109–1124, Mar. 2014.

[12] A. Dogra and P. Bhalla, "Image Sharpening By Gaussian And Butterworth High Pass Filter," *Biomedical and Pharmacology Journal*, vol. 7, no. 2, pp. 707–713, Dec. 2014.

[13] L. Tan and J. Jiang, *Digital Signal Processing Fundamentals and Applications*. Academic Press, 2013.

[14] H. Hiary, R. Zaghloul, A. Al-Adwan, and M. B. Al-Zoubi, "Image contrast enhancement using geometric mean filter," *Signal, Image and Video Processing*, vol. 11, no. 5, pp. 833–840, Jul. 2017.

[15] S. Suman, F. A. Hussin, A. S. Malik, N. Walter, K. L. Goh, I. Hilmi, and S. hooi Ho, "Image Enhancement Using Geometric Mean Filter and Gamma Correction for WCE Images," in *Neural Information Processing*, ser. Lecture Notes in Computer Science, C. K. Loo, K. S. Yap, K. W. Wong, A. T. Beng Jin, and K. Huang, Eds. Cham: Springer International Publishing, 2014, pp. 276–283.

[16] A. Stella and D. B. Trivedi, "Implementation of Order Statistic Filters on Digital Image and OCT Image: A Comparative Study," 2012.

[17] D. T. Kuan, A. A. Sawchuk, T. C. Strand, and P. Chavel, "Adaptive Noise Smoothing Filter for Images with Signal-Dependent Noise," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-7, no. 2, pp. 165–177, Mar. 1985.

[18] R. Ha, P. Liu, and K. Jia, "An Improved Adaptive Median Filter Algorithm and Its Application," in *Advances in Intelligent Information Hiding and Multimedia Signal Processing*, ser. Smart Innovation, Systems and Technologies, J.-S. Pan, P.-W. Tsai, and H.-C. Huang, Eds. Cham: Springer International Publishing, 2017, pp. 179–186.

[19] T. Pramoun, S. Srakaew, and T. Amornraksa, "Image Watermarking Based on Modified Harmonic Mean Filter," in *Proceedings of the 2019 2nd Artificial Intelligence and Cloud Computing Conference*, ser. AICCC '19. New York, NY, USA: Association for Computing Machinery, Feb. 2020, pp. 97–103.

[20] S. S. Goilkar and D. M. Yadav, "Defocused image restoration using Wiener and inverse filter in context of security application," *AIP Conference Proceedings*, vol. 2576, no. 1, p. 030013, Dec. 2022.

[21] S. Paris, P. Kornprobst, J. Tumblin, and F. Durand, *Bilateral Filtering: Theory and Applications*, 2009.

[22] K. Kaur, G. Singh, and A. Singla, "RESEARCH ON SPATIAL FILTERS AND HOMOMORPHIC FILTERING METHODS," 2010.

[23] P. Maragos, "Chapter 13 - Morphological Filtering," in *The Essential Guide to Image Processing*, A. Bovik, Ed. Boston: Academic Press, Jan. 2009, pp. 293–321.

[24] S. Wang, P. Xu, R. Song, P. Li, and H. Ma, "Development of High Performance Quantum Image Algorithm on Constrained Least Squares Filtering Computation," *Entropy*, vol. 22, no. 11, p. 1207, Nov. 2020.

[25] K. Saini, M. L. Dewal, and M. Rohit, "Modified nonlinear complex diffusion filter (MNCDF)," *Journal of Echocardiography*, vol. 10, no. 2, pp. 48–55, Jun. 2012.

[26] V. Dakshayani, G. R. Locharla, P. Pławiak, V. Datti, and C. Karri, "Design of a Gabor Filter-Based Image Denoising Hardware Model," *Electronics*, vol. 11, no. 7, p. 1063, Jan. 2022.

[27] S. Jadwaa, "Wiener Filter based Medical Image De-noising," *International Journal of Science and Engineering Applications*, vol. 7, pp. 318–323, Sep. 2018.

[28] J. Kuriakose and J. Joy, "Image Fusion Using Kuwahara Filter," vol. 5, 2014.

[29] G. Rosman, L. Dascal, A. Sidi, and R. Kimmel, "Efficient Beltrami Image Filtering via Vector Extrapolation Methods," *SIAM Journal on Imaging Sciences*, vol. 2, pp. 858–878, Jan. 2009.

[30] F. Zhao, "Image Restoration with Lucy-Richardson Algorithm," *Applied Mechanics and Materials*, vol. 462–463, pp. 343–346, Nov. 2013.

[31] X. Zhang, "Two-step non-local means method for image denoising," *Multidimensional Systems and Signal Processing*, vol. 33, no. 2, pp. 341–366, Jun. 2022.

[32] T. Hodson, T. Over, and S. Foks, "Mean Squared Error, Deconstructed," *Journal of Advances in Modeling Earth Systems*, vol. 13, no. 12, 2021.

[33] A. Horé and D. Ziou, "Image Quality Metrics: PSNR vs. SSIM," in *2010 20th International Conference on Pattern Recognition*, Aug. 2010, pp. 2366–2369.