

Spatial domain noise removal filtering for low-resolution digital images

Zaher Salah¹, Waleed T. Al-Sit², Kamal Salah³, Esraa Elsoud⁴

¹Department of Information Technology, Faculty of Prince Al-Hussein Bin Abdullah II for Information Technology, The Hashemite University, Zarqa, Jordan

²Department of Computer Engineering, Mutah University, Al-Karak, Jordan; Higher Colleges of Technology, Dubai, United Arab Emirates

³Deanship of Preparatory Year and Supporting Studies, Imam Abdulrahman Bin Faisal University, Dammam, Saudi Arabia

⁴Department of Computer Science, Faculty of Information Technology, Zarqa University, Zarqa, Jordan

Article Info

Article history:

Received Jan 20, 2024

Revised Feb 28, 2024

Accepted Mar 10, 2024

Keywords:

Digital image processing

Filtering

Low-resolution images

Noise detection

Noise removal

ABSTRACT

In this research work, six different filters are applied on a low resolution 8 b/pixel gray-scale images, which operate on small sub-images (windows of 3×3 to 11×11 pixels). The enhanced images are used to compare the efficiency of the different six filters using the peak signal to noise ratio (PSNR) image quality measure. Noise peak elimination filter (PSNR)=36.63) outperforms others, such as median filter (PSNR=36.61), while corruption estimation (PSNR=36.03) significantly cuts processing time by only processing the corrupted pixels while maintaining image details. Mean filter (PSNR=34.05) is sensitive to outliers, which cause the image's sharpness and fine features to be lost. By avoiding averaging across edges, bimodal-averaging filter (PSNR=35.30), which improves on the mean filter, chooses the mean of the biggest population. The median-mean filtering (PSNR=36.32), which combines median and mean filters and determines the output pixel by averaging the median and some nearby pixels, is another improvement above averaging.

This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.



Corresponding Author:

Zaher Salah

Department of Information Technology

Faculty of Prince Al-Hussein Bin Abdullah II for Information Technology, The Hashemite University

P.O. Box 330127, Zarqa 13133, Jordan

Email: zaher@hu.edu.jo

1. INTRODUCTION

Humans are primarily visual creatures; we rely heavily on our vision to understand the world around us because our eyes provide such an accurate description of the environment. However, vision is related to very important mental activities that could not be performed without it. We don't just look at things to identify and classify them; we can also scan for differences and quickly get a general impression of a scene. As a result of our highly developed visual abilities, humans have naturally developed the ability to recognize faces instantly, distinguish between colors, and process large amounts of visual data quickly. This clearly demonstrates the significance that our bodies have placed on visual information, so it only makes sense that we would act in this way. Human improvisation and dissection play a key factor in the decision of one technique over another, and this choice is frequently based on subjective visual judgments. Human beings have a tendency to examine visual information in line with other criteria. Thus, it's essential to gain a fundamental grasp of what makes human visual perception special [1]. The primary distinction between the lens of the eye and a regular optical lens is that the first one is more flexible, and its anterior surface's

curvature has a radius that is larger than its posterior surface's. Tension in the ciliary body's fibers determines the curvature of the lens. The regulating muscles make the lens somewhat flat so that it can focus on distant objects. In a similar way, these muscles enable the lens to thicken in order to concentrate on things close to the eye [2].

Images have always existed in our world, although digital images may be more recent. One of the earliest industries to employ digital images was the newspaper industry. With the development of the Bart Lane Cable Image Transmission System in the beginning of the 1920s, the time needed for sending an image over the Atlantic Ocean decreased dramatically, from more than a week down to fewer than three hours. This is regarded as an important time-saving innovation in the communication and image processing industries [3]. For a better understanding of how digital images were transmitted back then, consider how dedicated printing equipment encoded images for cable transmission before reconstructing them at the other end. But because computers had not yet developed, these are not regarded as digital images. With the advent of this innovation, a new need was created: one for an image format that could be easily handled for various operations to be applied on, could be stored on various and dependable media to ensure the ease of its transportation and sharing, could be stored for long periods of time (archiving) and to support an innovative level of picture clarity and detail precision that would allow images to emerge in crucial disciplines that will be discussed later, as well as for the digital image to assist the development and giant step in those fields and sciences as well. The beginning of what we now refer to as digital image processing may be connected to the availability of the machines and the start of the space program during that time. The first computer powerful enough to carry out crucial image processing tasks initially arrived in the early 1960s. The potential of digital image processing concepts was only realized through the combination of these two developments; the first development is significant because earlier computers could not handle the complexity and processing power required for digital image representation in a computer, and as a result, processing a digital image required a sufficient amount of space for the image handling utilization to run and carry out the most fundamental operations on the recently introduced digital image [3]. Computerized tomography (CT), for example Lung cancer [4], and astronomical observations are two major applications for digital imaging using different electromagnetic (EM) fields like gamma-ray, while X-ray imaging is also widely used in industry and in other fields, such as astronomy and medical diagnostics. Additionally, the ultra violet (UV) band is used for industrial inspection, lithography, microscopy, biological imaging and astronomical investigations.

Remote sensing may additionally be accomplished by employing the visible bands, infrared bands or the microwave bands. Radar and radio bands are the most frequently utilized instrument, where magnetic resonance imaging is mostly employed in astronomy and therapy [5]. Over time, images, particularly digital images, have encroached on every aspect of our lives, which has motivated us to take care of this field and show interest in its development. However, when the field of digital images entered extremely important areas of our lives, this added additional pressure on us to increase our interest in developing all fields and aspects of digital image processing and doing our best to increase the capabilities of the current digital image processing techniques, and the programs developed to enhance the digital image or remove any deformity (noise), and to attempt to design and implement new programs that can efficiently perform all of the anticipated tasks and perform everything significant to this extent as well as to be readily available and easily extended to meet the needs and requirements of the new life demands and the day to day continually evolving environment, though it can be highly challenging, it is a responsibility that should be taken regarding this important field. The term digital image processing (DIP) typically describes how a digital device or computer processes a 2-dimensional image. An array of complex integers or real values with a finite number of bits makes up a digital image. Wide range of business applications for digital image processing including medical equipment, robots, radar, sonar and remote sensing using spacecraft and satellites for image transmission and storage [6]. The term digital image, is referring to a 2-dimensional function $f(x,y)$ for light intensity, where x and y are the coordinates and f at any point (x,y) as shown in Figure 1 denotes the gray level value (brightness) at that point [7].

A digital image $f(x,y)$ is digitized in both brightness and spatial coordinates. A digital image may be interpreted as a matrix with row and column indices that represent points in the image and the associated matrix element values that specify the gray level at those points [7], with the gray-level variations defined over a rectangle or square lattice [8], [9]. It is important to characterize the quantity that each image element (also defined as a pixel or pel) in an image represents. An image might show the luminance of objects in a scene like in photos taken with a regular camera, the body's tissue's absorption properties in X-ray imaging, the cross sections of a targeted object in radar imaging or the region's temperature distribution in infrared imaging. Generally, a digital image can be considered as a two-dimensional function that holds information. A quantitative description for this function characteristics can be given by image models [6]. In general, Digital Image can be described as a record of the light falling on a sensor. Digital Images are quantized, and to this end we typically only record 256 different light levels when capturing digital images. Below a certain

level no light is recorded, above a certain level the sensor is saturated and between these levels only discrete values are recorded (usually 256 levels-1 byte), as shown in Figure 2. All of these mean our images are not quite accurate records.



Figure 1. Axis convention representation for a digital image

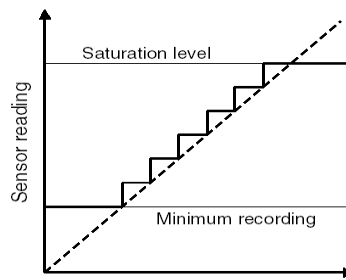


Figure 2. Amount of light falling on sensor

A basic requirement of digital image processing is that the digital image can be sampled and quantized easily. The sample rate (frequency of pixels per unit area) must be high enough to retain useful information in the image. Image quantization means the process of from analog to digital conversion of a sampled image into a finite number of gray levels. Another way to describe local properties of pixels can be is to describe each pixel by its relationship to neighboring pixels. For example, linear systems characterized by (low-order) difference equations and constrained by white noise or another random field with known power spectral density are useful approaches to represent ensembles [6]. The presence of noise is represented by unwanted information unrelated to the scene under study, which disturbs the information corresponding to the form observed in the image. It is translated into more or less cruel values that are added or subtracted to the proper gray level values of many pixels [10]. Noise originates from different sources, e.g. imperfect sensors introduce noise. Noise may be small black spots caused by fax transmission errors or dirty scanning machine. Various types of noise will be discussed in detail in the next section. Deviations from an ideal image can be modeled as additive noise as shown in Figures 3 and 4.

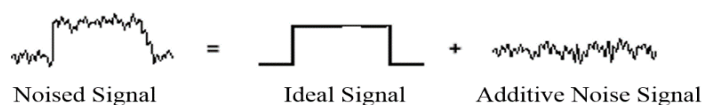


Figure 3. Noised signal

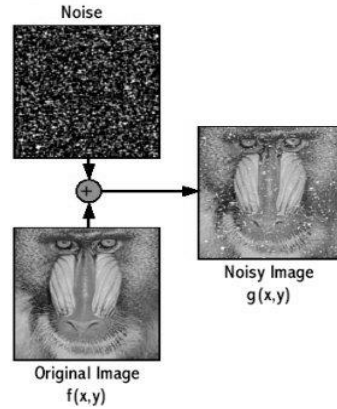


Figure 4. Model of noise in an image

Noise removal is a major concern in fields of image processing and computer vision. For instance, in many applications that apply operators based on image derivation, any noise in the image can lead to fatal errors. Noise can appear in an image from a variety of sources: during image acquisition, due to the quality and resolution of the camera in addition to the conditions of image acquisition such as illumination levels, calibration, and positioning. Furthermore, it can depend on the environment of the scene like: shape composition, the background, nature of material [6]. In this research paper, three types of noise will be discussed briefly: salt and pepper (impulse) noise, uniform noise and Gaussian noise. Salt and Pepper (Impulse) noise often harm images as a result of errors produced by noisy sensors or transmission channels. [11], [12]. In contrast to the noise that is typically added during acquisition, impulsive noise typically arises separately [13]. It randomly alters pixels [14], forcing values to diverge greatly from the actual values and frequently from values of neighboring pixels. As seen in Figures 5 and 6, impulsive noise appears as a dispersion of dark and light spots throughout the image. Faulty timing or erroneous memory locations issues that corrupt the analog to digital convertibility can also be a reason in addition to transmission failure or damaged pixels in the digital camera sensors [10]. Sometimes sensors saturate erroneously or respond excessively. Excessive saturation causes a white spot (salt) in the image, whereas lack of response causes a black spot (pepper). The parameters are: $a, b: -\infty < a, b < +\infty$, and the probability mass function is:

$$f(x) = \begin{cases} f(a) & \text{for } x = a \\ f(b) & \text{for } x = b \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

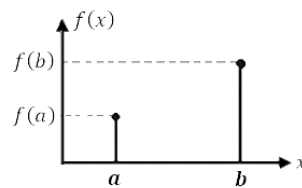


Figure 5. Impulse distribution

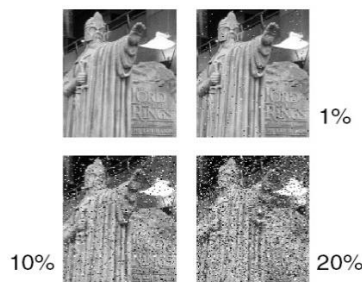


Figure 6. An image with varying amounts of salt and pepper noise added

In the case of the uniform noise, the intensity values of pixels are typically close to their actual values. We receive a range of values around the average value, which is equal to the actual value. As shown in Figure 7, the range forms a Probability Density Function (PDF) with a uniform distribution. Figure 8 displays an image that has been corrupted with different degrees of uniform noise. In the context of the uniform distribution PDF, errors lie within the range [a,b] and each value is equally likely, i.e. $f(x_1) = f(x_2) = \dots = f(x_n)$; $a < x_i < b$. The parameters are: $a, b: -\infty < a, b < +\infty$, and probability mass function is:

$$f(x) = \begin{cases} \frac{1}{b-a} & \text{for } a < x < b \\ 0 & \text{otherwise} \end{cases}, \text{mean} = (a + b)/2, \text{variance} = (b - a)^2/12 \quad (2)$$

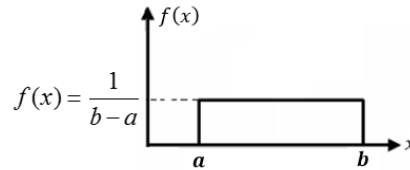


Figure 7. Uniform distribution

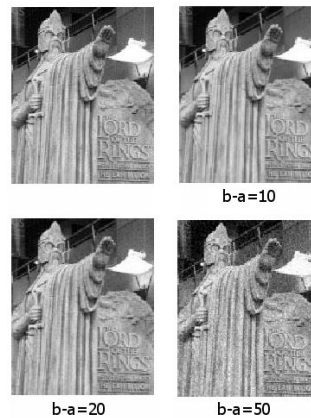


Figure 8. An image with varying degrees of uniform noise added

With respect to Gaussian noise, sensors tend to give measurements that are somewhat the actual value. They often give the actual value and tend to give values close to the true value rather than those that are far from it. The Parameters are: x which is the value for which you desire the distribution, μ which is the arithmetic mean for the distribution and σ which is the standard deviation for the distribution. $-\infty < x, \mu < +\infty$; $\sigma > 0$ and the Gaussian probability density is: $f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right]$.

Almost 68% of the data, 95% of the data, and 99.7% of the data, as shown in Figure 9, will be within 1 standard deviation of the mean, 2 standard deviations of the mean, and 3 standard deviations of the mean, respectively. An image with different degrees of Gaussian noise added is shown in Figure 10. The noise distribution within the digital image differs significantly between inter-image and intra-image. The image signal can contain noise in a variety of ways. With this signal, it can be both random and coherent. In this instance, it is included into the image spatial frequency domain and cannot be eliminated without first having a prior knowledge about the digital image. A reduction in spatial resolution results frequently from this. Periodic noise is not relevant information and can be readily removed without compromising data integrity and information lost. Last but not least, when the captor sends a repeating and redundant signal, which is typically the case for moving scenes, the noise that is there is incoherent and may be readily removed [10]. As expected, image data are frequently subject to noise (such as analog electrical transmission), so a variety of algorithms can be applied to enhance the image quality, as shown in Figure 11. Therefore, noise removal is a crucial step in the processing of digital images [15], [16].

Certain digital image processing techniques are noise-sensitive. Before progressing on, the noise reduction method aims to eliminate the damaging effects of noise. However, if doing so and the information cannot be gained, this will result in a loss of information. The majority of methods for eliminating noise are referred to as filters, and they are done to each point in an image utilizing information obtained from a tiny local window of pixels. The Gaussian noise Filtering Process is shown in Figure 12.

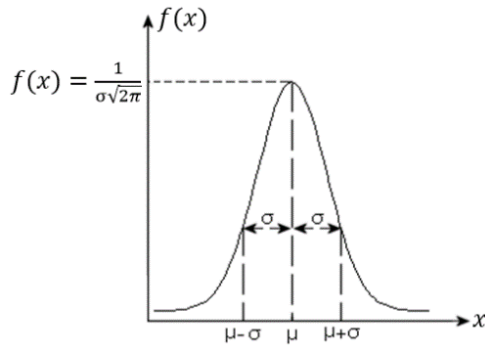


Figure 9. Gaussian distribution

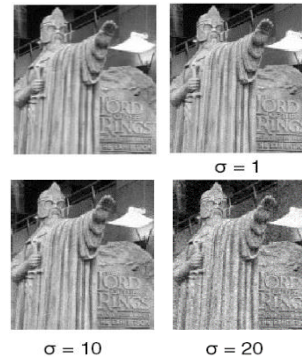


Figure 10. Image with varying degrees of Gaussian noise added

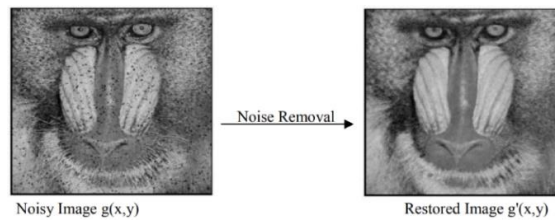


Figure 11. Simplified explanation for noise removal process

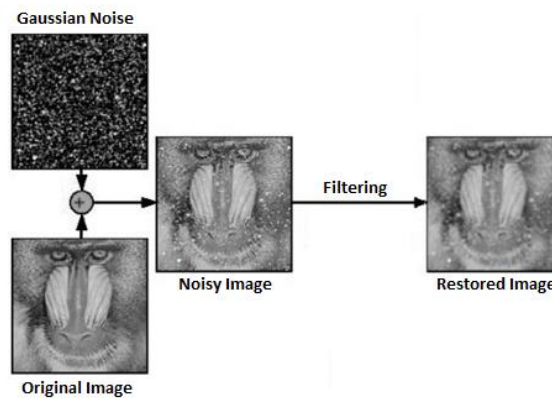


Figure 12. Gaussian noise removal (filtering) process

2. FILTERING

A technique called Filtering is used to enhance or modify a digital image such that the resulted image is better suited than the original image for certain applications [9]. For instance, you can filter a digital image to highlight particular features or elements while removing others. This process is a neighborhood operation in which the values of the pixels neighboring to the corresponding input pixel are used to calculate the value of any pixel in the output digital image. The neighborhood of a pixel is an arrangement of pixels that are identified by their location close to that pixel [17] as depicted in Figure 13. This means that a sub-

image region (mask or window) centered at (x,y) can be utilized for constructing a neighborhood around (x,y) [9].

The noise from the corrupted images can be removed using a variety of methods. The following is a general framework description for them. We establish an image window that is centered on each pixel and is often 3×3 , 5×5 , or 7×7 pixels in size. After applying a noise suppression scheme to substitute a pixel with a new value derived from the information in the window, the local window's predefined restrictions are met. The limitation of this framework, however, is that it can only see information contained within the window [17].

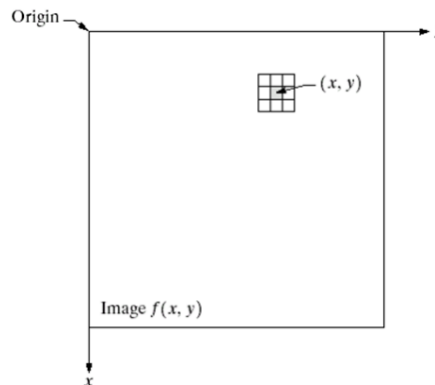


Figure 13. A 3×3 neighborhood mask centered at pixel (x,y) in a digital image

2.1. Which filter is optimal?

Noise in an image can be modeled with a Gaussian, uniform, or impulsive statistical distribution for the majority of common applications. Analytical descriptions of Gaussian noise can be made, and it has a recognizable bell shape. The gray level values of the noise are uniformly distributed across a certain range (0-255 levels), according to uniform distribution. Pixels produced by impulsive noise have gray level values that are inconsistent with their local neighborhood [17]. Linear filter combines neighborhood values from an image with a constant matrix to produce a linear combination. This can result in blur, poor feature localization, and insufficient noise suppression when impulsive noise is present. Nonlinear filters have been recommended to fix those shortcomings. The median filter is the most widely used nonlinear filter. Regardless of its straightforward concept, it has proven to be especially efficient at suppressing impulsive noise when applied to small neighborhoods. However, the median filter occasionally produces results that are unexpected and frequently falls short of sufficiently smoothing non-impulsive noise. Impulsive noise-intensive reduction demands more advanced techniques that deal with more data related to images. Using statistics and thresholds to support replacement values is made possible by the additional information, permitting filters to be adjusted to specific image regions. There is still some use of the median filter in several of the most advanced impulsive noise reduction filters. As a result, the median filter not only plays a crucial role in determining the efficiency baseline for all of those filters, but also has an impact on how well they function as a whole. In conclusion, an optimal impulsive noise filter must preserve edge information, smooth pixel variations in smoothly blending large regions, and not alter the proper (non-noisy) pixels [10].

2.2. Filter formalization

If the neighborhood is 1×1 , then enhancements using point processing (such as contrast stretching and thresholding) are solely based on the intensity of individual pixels, and the restored or enhanced image g only depends on the value of the original image f at (x,y) and T turns into a function for gray level transformation (or mapping): $s=T(r)$, where r, s are the corresponding gray levels of $f(x,y)$ and $g(x,y)$ at (x,y) . The symbols r and s stand for the pixel intensity before and after processing, respectively. Image Negatives (using a function $s=T(r)=L-1-r$ to reverse the sequence from black to white such that as the input image intensity increases, the output image intensity decreases) is one example of a simple intensity transformation. In order to create screen slides and medical images, image negatives are frequently employed. $[0, L-1]$ is the range of gray level values) and piecewise-linear transformation functions like contrast stretching (to expand the dynamic range of the processed image), gray-level slicing (to highlight a certain gray level in an image to improve particular features), and Bit-Plane Slicing (to expand the visibility of a particular bit's contribution to the appearance of the overall image). On the other hand, mask processing or filtering employs the image

neighborhood (mask or window) centered at (x,y), where the values of f in the neighborhood of (x,y) will determine the value of g at (x,y) by using of masks (kernels, windows or filters) [9]. The processing of each single pixel in a digital image typically looks like:

```
int w, h;
for (w = 0; w < image.Width(); w++) {
    for (h = 0; h < image.Height(); h++) {
        // do something with pixel at (w,h)
    }
}
```

The processing at (x,y) depends on a neighborhood, typically a square with radius r (3x3 has radius 1, 5x5 has radius 2). Figure 13 shows a 3x3 neighborhood about a pixel (x,y) in a digital image.

```
int w, h, dw, dh, r;
for (w = 0; w < image.Width(); w++) {
    for (h = 0; h < image.Height(); h++) {
        for (dw = -r; dw <= r; dw++) {
            for (dh = -r; dh <= r; dh++) {
                // do something with (w+dw, h+dh)
            }
        }
    }
}
```

2.3. The peak and valley filter

Peak-and-Valley filter, a nonlinear smoothing technique that is relatively new, is used to minimize changes to the image's gray levels while reducing impulsive-like noise. This technique maximizes the original data's preservation. It might serve as an outline for developing of more complex impulsive noise suppression filters, which would take the place of the median filter in the initial processing [10], [14]. Nonlinear filters are those that are unable to be modeled by convolutional algorithms. Impulsive noise can often be successfully eliminated by nonlinear filters [14]. The order filters are a good example of this kind of filters. The base of order filters is a specific type of image statistics known as order statistics. These filters typically replace the center pixel value in tiny sub-images (windows of 3x3 to 11x11 pixels), like the convolution process, in accordance with some specified criteria. Order statistics is a technique to select the replacement value by sequentially ordering all the pixels from smallest to largest gray level values [10]. The center pixel value in the ordered window of pixel values is chosen by the median filter. Since the filter is not adaptive to specific noise and image regions, it either fails to preserve true pixels or fails to remove impulse noise. This uncomplicated rule suggests that it will be easy to implement, but its performance is limited. In order to assess noise density, determine parameter settings, or direct the filtering process, a number of strategies have been presented that attempt to benefit from the median filter's average performance. Some examples of these techniques comprise the mask median, minimum-maximum, switching-based median, adaptive trimmed mean, center weighted median, and weighted median filters. These methods include identifying harmed pixels in advance in an effort to preserve the integrity of true pixels. The noise peak removal filter has been shown to outperform a number of other filters, including the conventional median filter, although it may take several iterations to get the optimal outcomes, which will negatively impact its efficiency. The min-max operators provide the base of the noise peak elimination filter [10].

As was previously mentioned, impulsive noise is seen in the image as a dispersion of dark and light spots. We may anticipate that the majority of single corrupted pixels will only be framed by pixels with proper gray levels if the probability of salt and pepper pixels is sufficiently low. Additionally, there is no basic reason to change the value of the majority of the non-noisy pixels when attempting to reduce impulsive noise because the number of noisy pixels is typically lower than the entire number of image pixels (making noise reduction achievable). Therefore, we merely need to change the gray levels of noisy pixels, replacing them with "real or proper gray level values". The image must have these values, or at least they must be both globally and locally coherent with it. In all cases, we must first distinguish actual gray levels apart from noise [10]. Due to the unique characteristics of each image region, thresholding techniques are inadequate and may even increase noise. As a result, only a small neighborhood for each image pixel must be examined in order to detect and remove impulsive noise (or attenuate it) [10]. As we will see in more detail, the nonlinear peak-and-valley filter is an n-dimensional filter that can remove impulsive-like noise from 2-dimensional images in addition to preserving the majority of the image. The following is this filter main expression:

$$\begin{array}{ll}
 \text{if } V(i, j, \dots) < \min(V(i+r, j+s, \dots)) & V'(i, j, \dots) = \min(V(i+r, j+s, \dots)) \\
 \text{else if } V(i, j, \dots) > \max(V(i+r, j+s, \dots)) & V'(i, j, \dots) = \max(V(i+r, j+s, \dots)) \\
 \text{else} & V'(i, j, \dots) = V(i, j, \dots)
 \end{array}$$

Where the original and new values are V and V' respectively, $r, s, \dots = -1, 0, 1$ and $r \neq s \neq \dots$ when $r = 0$ or $s = 0$ or \dots [6]. Applying the peak-and-valley filter on a profile as shown in Figure 14 signal will, in the case of the gray level profile shown in Figure 14(a), remove all "peaks" and "valleys" that are less dense than two pixels and fill them by performing several rounds of cutting/filling and filling/cutting actions while shifting along the profile vector. At each stage, the operator takes three pixels into account. Regarding the cutting process, if the middle pixel has a higher gray level than the outer two, see Figure 14(b), its gray level gets substituted by the highest gray level of the outer two pixels. Regarding the filling the process, see Figure 14(c), we substitute the smallest value for the middle pixel's value if its gray level is smaller than the other two. To ensure that no peaks and/or valleys are left, all of these processes are repeated [10]. It is simple to convert the peak-and-valley filter from one dimension (1-D) to two dimensions (2-D). In this instance, a 3x3 pixel neighborhood is taken into account. The peak-and-valley filter for the 1-D case can be represented by the following expressions:

$$\begin{aligned} \text{if } P(i) < P(i - 1) \text{ and } P(i) < P(i + 1) & \quad P'(i) = \min(P(i - 1), P(i + 1)) \\ \text{if } P(i) > P(i - 1) \text{ and } P(i) > P(i + 1) & \quad P'(i) = \max(P(i - 1), P(i + 1)) \\ \text{else} & \quad P'(i) = P(i) \end{aligned}$$

The filling and cutting rules now comprise the eight pixels neighborhood around the central point. The expressions of the peak-and-valley filter for the 2-D case becomes:

$$\begin{aligned} \text{if } I(i, j) < \min(I(i + r, j + s)) & \quad I'(i, j) = \min(I(i + r, j + s)) \\ \text{if } I(i, j) > \max(I(i + r, j + s)) & \quad I'(i, j) = \max(I(i + r, j + s)) \\ \text{else} & \quad I'(i, j) = I(i, j) \end{aligned}$$

where $r, s, \dots = -1, 0, 1$ and $r \neq s$ when $r = 0$ or $s = 0$. In its 2-D version, the peak-and-valley filter is applied iteratively as opposed to recursively and is not decomposed, but it is equivalent to the basic operator used in the noisy peak eliminator filter. Peak-and-valley filters and morphological filters share some similarities, but they are fundamentally distinct. The peak-and-valley filter is recursive and does not "grow" (dilatation) or "shrink" (erosion) the shape, among other characteristics. The peak-and-valley filter would not benefit from repeated use, unlike morphological filters, because of its recursive definition. The definition of the peak-and-valley filter might take on various forms. The min and max functions, for instance, can be swapped out with some sort of neighborhood mean function [10]. Figure 15 shows the Peak-and-Valley Filter results for three types of noise: i) salt and pepper noise, ii) uniform noise, and iii) Gaussian noise.

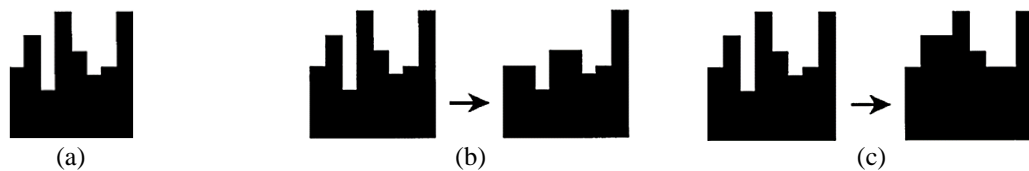


Figure 14. Peak-and-valley filter cutting and filling operations; (a) sample of gray level profile, (b) one pixel valley cutting, and (c) one pixel peak filling

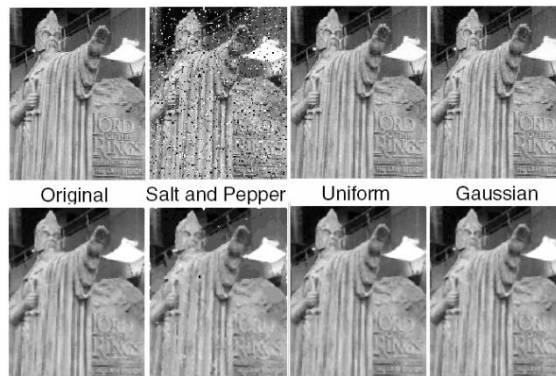


Figure 15. Peak-and-valley filter results

2.4. The mean (averaging) filter

The mean filter, illustrated in Figure 16, is a straightforward noise removal filter where each pixel is set to the mean (average) over a local window. This blurs the image, smoothing it down and erasing fine details. A linear filtering technique called a mean filter can be used to eliminate particular forms of noise. For this use, some filters, including Gaussian or averaging filters, are suitable. For instance, using an averaging filter to remove grain noise from an image where each pixel is set to the mean (average) of the pixels in its immediate surrounding area is useful [10]. This is illustrated in Figure 17.

$$\frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Figure 16. The mean filter 3×3 mask

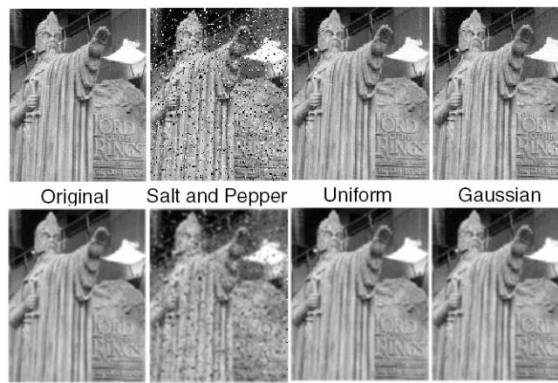


Figure 17. The mean filter results

2.5. The median filter

Because they are straightforward and may preserve image edges, median-based filters have gained a lot of attention over the last few decades [11], [18]. When utilizing a median filter, each output pixel is set to the average of the neighborhood pixels of the associated input, which is similar to how an averaging filter works. The value of each output pixel, however, is instead decided by the median of the nearby pixels when using median filtering [19]. The median is significantly less vulnerable to outliers (extreme values) than the mean is. As a result, median filtering, which has the advantage of preserving the image's edges, is frequently used to eliminate these outliers while still retaining the sharpness of the digital image. In summary, the median filter (also known as rank filtering) is a nonlinear function that replaces the value of each pixel in the resultant image with the middle value in the previously described order [15], [12]. It performs this by rearranging local window pixels by the magnitude of their intensity. In a local window, each pixel is set to the median value, which is statistically the middle value in a set. Figure 18 shows an example on Median Filter to explain its basic operation while, Figure 19 shows the Median Filter results for three types of noise: i) salt and pepper noise, ii) uniform noise and iii) Gaussian noise.

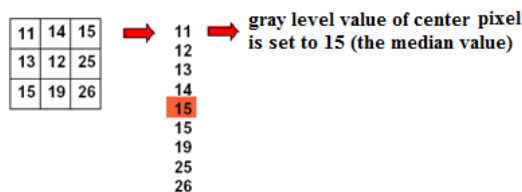


Figure 18. Median filter, example

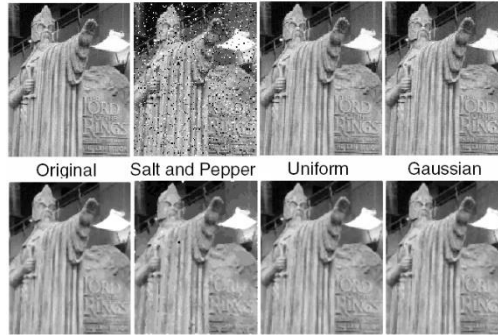


Figure 19. The median filter results

2.6. Bimodal averaging filter

Assume a bimodal distribution of gray values and choose the average value of the largest population to prevent averaging over edges. The following is this filter main expression:

$$\bar{g}_D = \frac{1}{|D|} \sum_{g_{mn} \in D} g_{mn}$$

$$\bar{g}_D = \frac{1}{|D|} \sum_{g_{mn} \in D} g_{mn}$$

$$A = \{g_k \text{ with } g_k \geq \bar{g}\}$$

$$B = \{g_k \text{ with } g_k < \bar{g}\}$$

$$g' = \begin{cases} \frac{1}{|A|} \sum_{g_k \in A} g_k & \text{if } |A| \geq |B| \\ \frac{1}{|B|} \sum_{g_k \in B} g_k & \text{otherwise} \end{cases}$$

where \bar{g} is the average of all pixels g_{mn} in the neighborhood of the corresponding input pixel over a local window D , $g_{mn} \in D$ and $m, n = -1, 0, 1$. A is the set of pixels ($g_k; k = 0 - 9$) within a window D which have a value higher than or equal to \bar{g} while B is the set of pixels ($g_k; k = 0 - 9$) within a window D which have a value less \bar{g} . $|D|$ is number of pixels within D (=9 here), $|A|$ is number of pixels in A while $|B|$ is number of pixels in B , and g' is the output pixel value. An example on Bimodal Averaging Filter is shown Figure 20 to explain its basic operation while, Figure 21 shows the Bimodal Averaging Filter results for three types of noise: (i) salt and pepper noise, (ii) uniform noise and (iii) Gaussian noise.

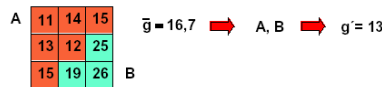


Figure 20. Bimodal averaging, example

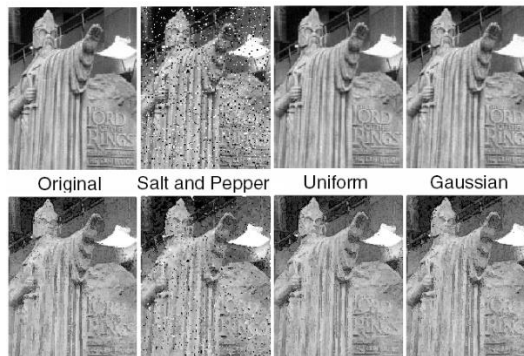


Figure 21. Bimodal averaging filter results

2.7. The median-mean filter

While each output pixel in median-mean filtering is set to the average of the pixel values in neighborhood of the associated input pixel, it functions similarly to averaging filter. But when using Median-Mean filter, the mean of the three median neighborhood pixels determines the value of the output pixel instead of the mean. Figure 22 shows an example on Median-Mean Filter to explain its basic operation while, Figure 23 shows the Median-Mean Filter results for three types of noise: i) salt and pepper noise, ii) uniform noise and iii) Gaussian noise.

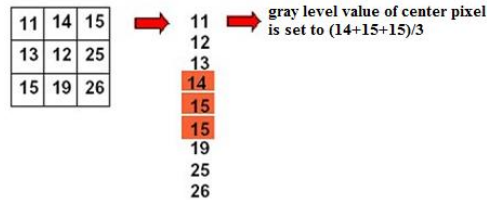


Figure 22. Median-mean, example

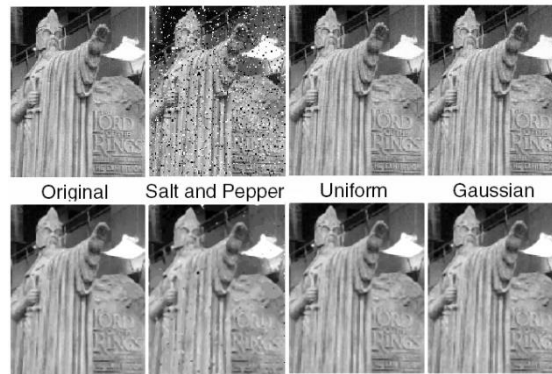


Figure 23. Median-mean filter results

2.8. The corruption estimation filter

This detection-estimation technique's detection phase involves assessing each pixel's local "energy" and comparing it to a locally adjustable threshold. The median value of the filtering window is used to replace outliers. Outliers are those pixels having energy exceed a given threshold [20]. The general expression of this filter is as follows:

$$\begin{aligned}
 E(m, n) &= \max(E_1(m, n), E_2(m, n)) \\
 E_1(m, n) &= |2 * (X(m, n) - \mu)^2 \\
 &\quad - (X(m - 1, n) - \mu) * (X(m + 1, n) - \mu) \\
 &\quad - (X(m, n - 1) - \mu) * (X(m, n + 1) - \mu)| \\
 E_2(m, n) &= |2 * (X(m, n) - \mu)^2 \\
 &\quad - (X(m - 1, n - 1) - \mu) * (X(m + 1, n + 1) - \mu) \\
 &\quad - (X(m + 1, n - 1) - \mu) * (X(m - 1, n + 1) - \mu)|
 \end{aligned}$$

where μ is the local mean of all pixels within the local window. The pixel value at (m, n) is denoted by $X(m, n)$ and the energy of the pixel at location (m, n) is denoted by $E(m, n)$.

The above generic expression can be improved in order to obtain various values for impulses of the same magnitude located in various backdrops [21]. The revised expression of this filter will be as follows:

$$\begin{aligned}
 E(m, n) &= 2 * X(m, n)^2 - X(m - 1, n) * X(m + 1, n) \\
 &\quad - X(m, n - 1) * X(m, n + 1)
 \end{aligned}$$

This "energy" is then compared against a locally calculated threshold: $Th = \frac{\alpha}{N} * \sum_{n=1}^N E_n$, where E_n is the energy of the n 'th pixel in the neighborhood window of N pixels, and α is a constant computed experimentally (here $\alpha = 1.9$). All pixels with energy exceeds the threshold Th are substituted at the output by the median of the pixels inside the filter window (mask). Otherwise, the pixel value is preserved. Advantages of Corruption Estimation Filter can be summarized as: i) removing impulses in addition to preserving image details, ii) reducing the processing time drastically by processing the corrupted pixels only, and iii) obviating time consuming operations such as calculating distances and angels [20]. Figure 24 shows the Corruption Estimation Filter results for three types of noise: i) salt and pepper noise, ii) uniform noise, and iii) Gaussian noise.

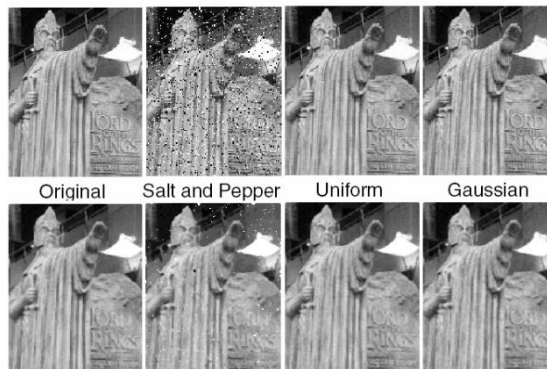


Figure 24. The corruption estimation filter results

3. RESULTS AND DISCUSSION

Image quality measurements (IQMs) are metrics used to assess the effectiveness of coding, processing, or systems for imaging. A good objective quality metric should accurately reflect the visual distortion brought on the image by for instance, blurring, compression, noise, and inadequate sensors. The effectiveness of vision-based algorithms including feature extraction, detection, tracking, segmentation image based measurements among others, is anticipated to be significantly influenced by such measurements [22]. The most commonly used measurements in the literature on image processing and computer vision are the deviations in mean square error (MSE) or signal to noise ratio (SNR) between the original and processed images. They are extremely common because of their analytical tractability and the ease with which it is frequently possible to create systems that reduce the MSE [9]. Raw error measures, like MSE, are capable of mathematically quantifying the error and perform well when there is an additive noise damage, but they do not always accurately reflect all elements of the observer's visual comprehension of the errors or structural coding artifacts [23]. This metric calculates the distortion between two pictures based on their pixel-wise differences or specific moments of the difference (error) image. The SNR or peak SNR (PSNR) is very frequently used to evaluate coding distortions based on pixel differences [19], [22], [24].

There are basically two methods for measuring the quantitative quality of images. The first, more useful measure focuses on distortion like the MSE, PSNR, and other comparable measurements. The quality metric is not necessarily connected with the subjective assessment for many different sorts of degradations, though, for this class of distortion measurements. Measures that are focused on human visual system (HVS) modeling represent the second class of evaluation methodologies [25]. Measures of signal-to-noise (SNR) indicate how well an image has been reconstructed in comparison to the original image. The fundamental concept is to calculate a single number that expresses the level of the reconstructed image quality. Better judgments are made of reconstructed images with higher metrics. However, signal-to-noise measures are simpler to calculate even though standard SNR measurements do not align with subjective comprehension of human. Just keep in mind that better quality does not always equate to higher measurements [26]-[28]. The actual metric used in the research described in this paper is the peak signal-to-noise ratio measure (PSNR). Assuming that we are given a source image $f(i, j)$ that contains N by M pixels and a reconstructed image $F(i, j)$ where F is reconstructed by decoding the encoded version of $f(i, j)$. Error measures are computed over all pixels using pixels intensity values $f(i, j)$ ranging between 0 and 255 [26]. First we calculated the mean squared error (MSE) of the filtered image as follows: $MSE = \frac{\sum [f(i, j) - F(i, j)]^2}{N * M}$. The root mean squared error (RMSE) is the square root of MSE, $RMSE = \sqrt{MSE}$. PSNR is calculated in decibels (dB) as follows:

$PSNR = -10 \log_{10} \left(\frac{255}{RMSE} \right)$. Between 20 and 40 is the typical PSNR value range. Usually, they are reported using two decimal places, like 32.63. Although the actual value is meaningless, a quality indicator can be determined by contrasting two values for various reconstructed images. Instead of 255/RMSE, some alternative PSNR definitions utilize 2552/MSE. We are focusing on the relative comparison rather than the absolute values, therefore any formulation will be effective. The fact that mean-squared error is heavily reliant on image intensity scaling is an issue. A MSE of 100.0 for an 8-bit image (with pixel values between 0 and 255) appears terrible, whereas an MSE of 100.0 for a 10-bit image (with pixel values between 0 and 1023) is hardly noticeable [26]. The six filters are implemented in Visual C++. The performance of these filters is tested on an image with gray level values coded on 8 Bits using a 3x3 neighborhood. The image was corrupted with impulsive noise with 20% noise level as the reference (baseline) for the tests. The test image corrupted with 20% impulsive noise is shown in Figure 25. The original and corrupted images are shown in Figures 25(a) and (b) respectively, while the results are summarized in Table 1.



Figure 25. The original image and the test image corrupted with 20% impulsive noise, (a) original image and (b) corrupted image

Table 1. Summary of the six filters results evaluated using PSNR in decibels (dB)

Filters	Result		
Filter:	Average (Mean)	Median	Median-Mean
PSNR (dB):	34.05	36.61	36.32
Filter:	Bimodal Averaging	Corruption Estimation	Peak and Valley
PSNR (dB):	35.30	36.03	36.63

4. CONCLUSIONS





Experimental results indicate that the noise peak elimination filter (PSNR=36.63) outperforms others including the typical median filter (PSNR=36.61), while corruption estimation (PSNR=36.03) reduces processing time significantly by only processing the damaged pixels, and preserves image details. Mean (average) filter (PSNR=34.05) is sensitive to extreme values (outliers) which reduce the sharpness of the image and remove fine details. Bimodal-Averaging filter (PSNR=35.30) which is an enhancement on mean filter by avoiding averaging across edges, assumes bimodal distribution and selects mean of the largest population. Another improvement over averaging is the median-mean filtering (PSNR=36.32) which works

as a combination of both: median and mean filters; here output pixel is determined by averaging the median and some pixels around the median (e.g. the median and two pixels: above and below the median).





REFERENCES

- [1] P. E. Hart, D. G. Stork, and R. O. Duda, "Vision in man and machine," *Computer Vision, Graphics, and Image Processing*, vol. 31, no. 1, p. 122, Jul. 1985, doi: 10.1016/S0734-189X(85)80079-7.
- [2] M. D. Levine, "Vision in man and machine," *Computer Vision, Graphics, and Image Processing*, vol. 31, no. 1, p. 122, Jul. 1985, doi: 10.1016/S0734-189X(85)80079-7.
- [3] E. Trucco, "Computer and Robot Vision," *AI Communications*, vol. 8, no. 1, pp. 50–51, 1995, doi: 10.3233/AIC-1995-8111.
- [4] MD. Ismail Hossain Sadhin, Methila Farzana Woishe, Nila Sultana, and Tamanna Zaman Bristy, "Identifying lung cancer using CT scan images based on artificial intelligence," *Malaysian Journal of Science and Advanced Technology*, pp. 31–35, Mar. 2022, doi: 10.56532/mjsat.v2i1.34.
- [5] A. Denniss, T. M. Lillesand, and R. W. Kiefer, *Remote sensing and image interpretation*, vol. 132, no. 2. New York: John Wiley & Sons, 1995, doi: 10.1017/S0016756800012024.
- [6] A. Vyas, S. Yu, and J. Paik, "Fundamentals of digital image processing," in *Signals and Communication Technology*, 2018, pp. 3–11, doi: 10.1007/978-981-10-7272-7_1.
- [7] A. Baskar, M. Rajappa, S. K. Vasudevan, and T. S. Murugesh, *Digital image processing*. Boca Raton: Chapman and Hall/CRC, 2023, doi: 10.1201/9781003217428.
- [8] E. Seeram and D. Seeram, "Image postprocessing in digital radiology—A primer for technologists," *Journal of Medical Imaging and Radiation Sciences*, vol. 39, no. 1, pp. 23–41, Mar. 2008, doi: 10.1016/j.jmir.2008.01.004.
- [9] S. K. Dewangan, "Importance and applications of digital image processing," *International Journal of Computer Science & Engineering Technology*, vol. 7, no. 07, pp. 2229–3345, 2016.
- [10] P. S. Windyga, "Fast impulsive noise removal," *IEEE Transactions on Image Processing*, vol. 10, no. 1, pp. 173–179, 2001, doi: 10.1109/83.892455.
- [11] Zhou Wang and D. Zhang, "Progressive switching median filter for the removal of impulse noise from highly corrupted images," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 46, no. 1, pp. 78–80, 1999, doi: 10.1109/82.749102.
- [12] M. Yildirim, "Analog circuit implementation based on median filter for salt and pepper noise reduction in image," *Analog Integrated Circuits and Signal Processing*, vol. 107, no. 1, pp. 195–202, Apr. 2021, doi: 10.1007/s10470-021-01820-3.
- [13] D. N. H. Thanh, V. B. S. Prasath, T. K. Phung, and N. Q. Hung, "Impulse denoising based on noise accumulation and harmonic analysis techniques," *Optik*, vol. 241, p. 166163, Sep. 2021, doi: 10.1016/j.ijleo.2020.166163.
- [14] R. Kosarevych, O. Lutsyk, and B. Rusyn, "Detection of pixels corrupted by impulse noise using random point patterns," *The Visual Computer*, vol. 38, no. 11, pp. 3719–3730, Nov. 2022, doi: 10.1007/s00371-021-02207-1.
- [15] A. Koschan and M. Abidi, "A comparison of median filter techniques for noise removal in color images," *German Workshop on Color Image Processing*, vol. 34, no. 15, pp. 69–79, 2001.
- [16] Y. Chen *et al.*, "Salt and pepper noise removal method based on stationary Framelet transform with non-convex sparsity regularization," *IET Image Processing*, vol. 16, no. 7, pp. 1846–1865, May 2022, doi: 10.1049/ipr2.12451.
- [17] Zhou Wang and D. Zhang, "Restoration of impulse noise corrupted images using long-range correlation," *IEEE Signal Processing Letters*, vol. 5, no. 1, pp. 4–7, Jan. 1998, doi: 10.1109/97.654865.
- [18] R. H. Chan, Chung-Wa, and M. Nikolova, "Salt-and-pepper noise removal by median-type noise detectors and detail-preserving regularization," *IEEE Transactions on Image Processing*, vol. 14, no. 10, pp. 1479–1485, Oct. 2005, doi: 10.1109/TIP.2005.852196.
- [19] K. Suriyan, N. Ramaingam, S. Rajagopal, J. Sakkarai, B. Asokan, and M. Alagarsamy, "Performance analysis of peak signal-to-noise ratio and multipath source routing using different denoising method," *Bulletin of Electrical Engineering and Informatics*, vol. 11, no. 1, pp. 286–292, Feb. 2022, doi: 10.11591/eei.v11i1.3332.
- [20] F. A. Cheikh, R. Hamila, M. Gabbouj, and J. Astola, "Impulse noise removal in highly corrupted color images," in *IEEE International Conference on Image Processing*, IEEE, 1996, pp. 997–1000, doi: 10.1109/icip.1996.559669.
- [21] M. Pompapathi, A. Sri Krishna, and B. Raveendra Babu, "An efficient approach for removal of impulse noise from highly corrupted images by preserving edge details," in *2010 International Conference on Signal and Image Processing*, IEEE, Dec. 2010, pp. 498–50, doi: 10.1109/ICSIP.2010.5697526.
- [22] B. Sankur, "Statistical evaluation of image quality measures," *Journal of Electronic Imaging*, vol. 11, no. 2, p. 206, Apr. 2002, doi: 10.1117/1.1455011.
- [23] M. Jiang, E. K. Wong, N. Memon, and X. Wu, "Steganalysis of degraded document images," in *2005 IEEE 7th Workshop on Multimedia Signal Processing*, IEEE, Oct. 2005, pp. 1–4, doi: 10.1109/MMSP.2005.248660.
- [24] U. Erkan and A. Kilicman, "Two new methods for removing salt-and-pepper noise from digital images," *ScienceAsia*, vol. 42, no. 1, pp. 28–32, 2016, doi: 10.2306/scienceasia1513-1874.2016.42.028.
- [25] A. Beghdadi and B. Pesquet-Popescu, "A new image distortion measure based on wavelet decomposition," in *Proceedings - 7th International Symposium on Signal Processing and Its Applications, ISSPA 2003*, IEEE, 2003, pp. 485–488, doi: 10.1109/ISSPA.2003.1224745.
- [26] T. L. Veldhuizen and M. E. Jernigan, "Grid filters for local nonlinear image restoration," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, IEEE, 1998, pp. 2885–2888, doi: 10.1109/ICASSP.1998.678128.
- [27] R. H. Altaie, "Restoration for blurred noisy images based on guided filtering and inverse filter," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 11, no. 2, pp. 1265–1275, Apr. 2021, doi: 10.11591/ijece.v11i2.pp1265-1275.
- [28] J. Gao, L. Li, X. Ren, Q. Chen, and Y. M. Abdul-Abbass, "An effective method for salt and pepper noise removal based on algebra and fuzzy logic function," *Multimedia Tools and Applications*, vol. 83, no. 4, pp. 9547–9576, Jan. 2024, doi: 10.1007/s11042-023-15469-9.





BIOGRAPHIES OF AUTHORS

Zaher Salah     received his PhD degree in Computer Science from the University of Liverpool, UK, in 2014, his MSc degree in Computer Science from Yarmouk University, Jordan, in 2004, and his BSc degree in Computer Science from University of Jordan, Jordan, in 2001. He is currently an Associate Professor in the Information Technology Department of The Hashemite University, Zarqa, Jordan. His research interests include machine learning, cyber security, information retrieval, opinion mining, sentiment analysis, biometrics, digital image and analysis, and pattern recognition. He can be contacted at email: zaher@hu.edu.jo.







Waleed Al-Sit     received his PhD degree in Electrical Engineering and Electronics from the University of Liverpool, UK, in 2015, his MSc degree in Electrical and Computer Engineering from New York Institute of Technology (NYIT), Jordan, in 2008, and his BSc degree in Computer Engineering from Mu'tah University, Jordan, in 2006. He is currently an Associate Professor in the Department of Computer Engineering, Mutah University, Al-Karak, Jordan; Higher Colleges of Technology, Dubai, UAE. His research interests include machine learning, cyber security, sentiment analysis, networking, and pattern recognition. He can be contacted at email: w_sitt@mutah.edu.jo.



Kamal Salah     received the B.Sc. degree in Physics from Yarmouk University, Jordan in 2003, his M.Sc. degree in Applied Physics (Experimental Atomic and Molecular Physics) from The Hashemite University Jordan in 2007. He is currently a Lecturer in the Deanship of Preparatory Year and Supporting Studies of the Imam Abdulrahman Bin Faisal University, P.O Box 1982, Dammam, Saudi Arabia. He can be contacted at email: kisalah@iau.edu.sa.



Esraa Elsoud     received the BSc degree in Electrical Engineering from The Hashemite University, Jordan, in 2013 and MSc in Cyber Security from The Hashemite University in 2023. Eng. Esraa's current research interests include cyber security, machine learning, big data and mobile network. She can be contacted at email: eabuelsoud@zu.edu.jo.