



Image Analysis and Processing of an Ant Using MATLAB

Hamed F alsalhin Saleh ^{1*}, Fathalla I. Solman ², Abdulbasit Abdulla AB Ali ³,
Zead Hamed Abdulkarim ⁴

^{1,2,3,4} Department of Electrical and Electronic Engineering, College of Engineering
Technologies Alqubba, Libya

تحليل ومعالجة صورة لنملة باستخدام برنامج الماتلاب

حامد فضل الله الصالحين صالح ^{1*}، فتح الله إبراهيم سليمان ²، عبد الباسط عبد الله عبد المولى علي ³،
زياد حمد عبد الكريم ⁴
^{1,2,3,4} قسم الهندسة الكهربائية والإلكترونية، كلية التقنيات الهندسية القبة، ليبيا

*Corresponding author: hlibya60@gmail.com

Received: July 28, 2024

Accepted: October 19, 2025

Published: November 01, 2025

Copyright: © 2025 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Abstract:

This research paper focuses on image processing using the MATLAB design tool. The primary objective is to analyze the histogram distribution of a given image and determine a threshold for object recognition, employing the variance growth methodology. The selected threshold is then justified. Next, noise is introduced into the image, and an effective noise-removal technique is developed, evaluating its impact on the image, threshold, and distribution. Following this, histogram equalization is applied to adjust the image contrast, enhancing the recognition process and achieving a more uniform distribution. Finally, the initial steps are repeated, and an analysis is conducted to assess the noise-removal method's effectiveness, determining the maximum noise density at which it remains effective.

Keywords: Image processing, Histograms, Thresholding, Grey-scale, Noise removal.

الملخص

تركز هذه الورقة البحثية على معالجة الصور باستخدام أداة التصميم (MATLAB). الهدف الأساسي هو تحليل توزيع المدرج التكراري لصورة معينة وتحديد عتبة للتعرف على الأجسام، باستخدام منهجية نمو التباين. يتم بعد ذلك تبرير اختيار العتبة المحددة. بعد ذلك، يتم إدخال ضوضاء (تشويش) في الصورة، وتُطوّر تقنية فعالة لإزالة الضوضاء، مع تقييم تأثيرها على الصورة والعتبة والتوزيع. بعد ذلك، يتم تطبيق تسوية المدرج التكراري لضبط تباين الصورة، مما يُحسن عملية التعرف ويُحقق توزيعاً أكثر تجانساً. أخيراً، يتم تكرار الخطوات الأولية، ويتم إجراء تحليل لتقييم فعالية طريقة إزالة الضوضاء، وتحديد أعلى كثافة ضوضاء تظل عندها الطريقة فعالة.

الكلمات المفتاحية: معالجة الصورة، المدرجات التكرارية، تحديد العتبة، المدرج الرمادي، إزالة الضوضاء.

Introduction

An image, in its fundamental sense, is a visual representation of an object, entity, or scene, encompassing elements such as people, animals, and items. More precisely, images can be classified as either continuous or discrete signals, depending on their form and application [1]. A digital image

represents a discrete signal that conveys spatial intensity information, composed of picture elements—commonly referred to as pixels. Each pixel occupies a specific spatial position defined by the m -th row and n -th column in a two-dimensional coordinate system [2-4].

Digital image processing is the field concerned with the computational manipulation and analysis of digital images to enhance their visual quality or to extract meaningful information. The major domains within digital image processing include image enhancement, which aims to improve visual quality by emphasizing critical features such as edges, adjusting contrast, and reducing noise; image restoration, which involves correcting degradations caused by motion blur, noise, or sensor imperfections; and image segmentation, which focuses on isolating and identifying specific objects or regions within an image for subsequent analysis [4].

These techniques play a crucial role across a wide range of disciplines, including medical imaging, agriculture, industrial automation, remote sensing, and law enforcement. Digital images can be categorized into various types based on their pixel characteristics. Binary images consist of two possible pixel values, typically 0 (black) and 1 (white). Grayscale images are 8-bit representations with pixel values ranging from 0 to 255, corresponding to varying shades of gray. True-color (RGB) images comprise three color channels, red, green, and blue, which combine to produce a full spectrum of colors. Other specialized forms include indexed and intensity images [5-7].

One of the fundamental techniques employed in digital image processing is thresholding, which is utilized to separate objects of interest from the background. This is achieved by assigning a threshold value (T) to the image's intensity levels, whereby each pixel is classified as part of the object or the background depending on whether its intensity exceeds or falls below the threshold. This method provides a simple yet effective means of object extraction and is often a preliminary step in more advanced image analysis procedures.

Research objective

This study focuses on the processing of a specific object of interest (an ant) within a given image. The objective is to convert the image to its gray-scale form, generate its histogram distribution, and determine the image threshold using the variance growth method. Additionally, the image is intentionally degraded with salt and pepper noise, followed by the development of an effective noise removal technique. The impact of the applied filter on the image's histogram and threshold is also examined. The entire process is then repeated after performing histogram equalization to enhance the contrast of the object of interest.

Literature review

1. Biological Image Processing and Object Recognition

Previous studies have demonstrated the use of digital image processing to analyze insect morphology. For example, in [8] employed image segmentation techniques to isolate insect bodies from complex backgrounds, enabling automated species identification. In similar research, edge detection and region-based segmentation were applied to ant images for recognizing specific anatomical parts such as antennae and legs [9]. MATLAB has been central in these efforts due to its strong support for image filtering, segmentation, and statistical analysis.

2. Histogram Analysis and Thresholding Techniques

Histogram-based image analysis is a widely used approach for distinguishing foreground from background in grayscale images. Otsu's method [10], a popular thresholding algorithm available in MATLAB, has been employed in several studies to separate ants from their backgrounds. The use of variance-based thresholding, particularly the variance growth method, allows for more adaptive recognition in non-uniform lighting conditions, as shown by Ahmed and Khan [11]. These techniques are vital for isolating the ant in noisy or cluttered environments.

3. Noise Introduction and Denoising Techniques

Images of ants captured in natural habitats often suffer from noise due to environmental factors or imaging conditions. Researchers have explored various denoising algorithms including median filtering, Gaussian filtering, and wavelet-based methods. In their comparative study, [12] found that adaptive median filters in MATLAB are particularly effective for salt-and-pepper noise, preserving edges while reducing background noise. The choice of denoising technique significantly affects the quality of subsequent segmentation and analysis.

4. Histogram Equalization and Contrast Enhancement

Histogram equalization is a key step in enhancing image contrast, especially when images are poorly lit or have low dynamic range. Several studies, including that of applied contrast stretching and histogram equalization to insect images to improve visibility of key features [13].

Simulation Results

The entire image processing procedure is carried out using the MATLAB software package.

Step 1: The image is imported into MATLAB using the command '`f=imread('Ants.jpg')`'.

The imported image is converted to a grayscale format using the function '`g=rgb2gray(f)`'.

The figure below displays the grayscale version of the original image.

Step 3: The histogram of the grayscale image is generated with the command '`h=imhist(g)`'.

Both the grayscale image and its corresponding histogram are illustrated in Figure 1.

The histogram of a grayscale image represents the frequency distribution of pixel intensity values. It helps in understanding the characteristics and quality of the image

A histogram with pixel intensities clustered toward the lower end of the range indicates a dark image.

A histogram with values spreads across the entire range signifies a well-contrasted image.

If the values are concentrated at the upper end, the image has high brightness and may be overexposed. A histogram with values centered in the middle suggests a poorly contrasted image

From Figure 1, it is evident that the grayscale values span almost the entire range of the histogram. This indicates that the image has high contrast, which is visually confirmed.

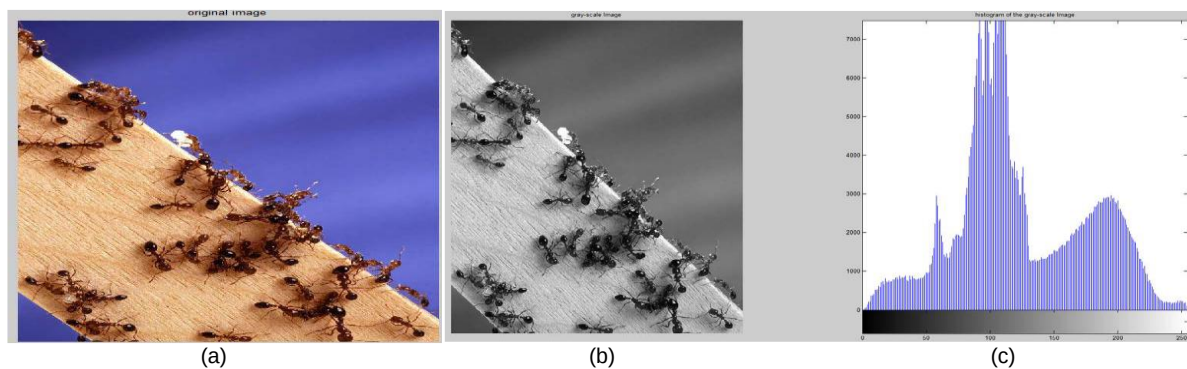


Figure 1 (a,b,c): illustrates original image, its gray-scale and its histogram in MATLAB.

Step 4: Applying Variance Growth Methodology for Image Thresholding

Thresholding is a technique used to convert a grayscale image into a binary image by assigning pixel values to either the foreground (object) or background based on a specified intensity threshold. In this report, the Variance Growth Methodology is employed to determine the optimal threshold for segmenting the object of interest from the background. In MATLAB, thresholding can be implemented using the command: '`T=im2bw(g,[ ])`' where, the square brackets '`[ ]`' are replaced with the desired threshold value, which must be a normalized grayscale value between 0 and 1. To calculate the threshold using the Variance Growth Method, the following MATLAB commands are used.

```
C = cumsum(h);    % Computes the cumulative sum of the grayscale histogram
Var = max(C) - C;  % Calculates the variance based on cumulative sum
Horz = 0:1/255:1; % Defines the x-axis values for plotting
plot(Horz, Var);   % Plots the variance growth curve
```

The resulting curve is shown in the figure 2.

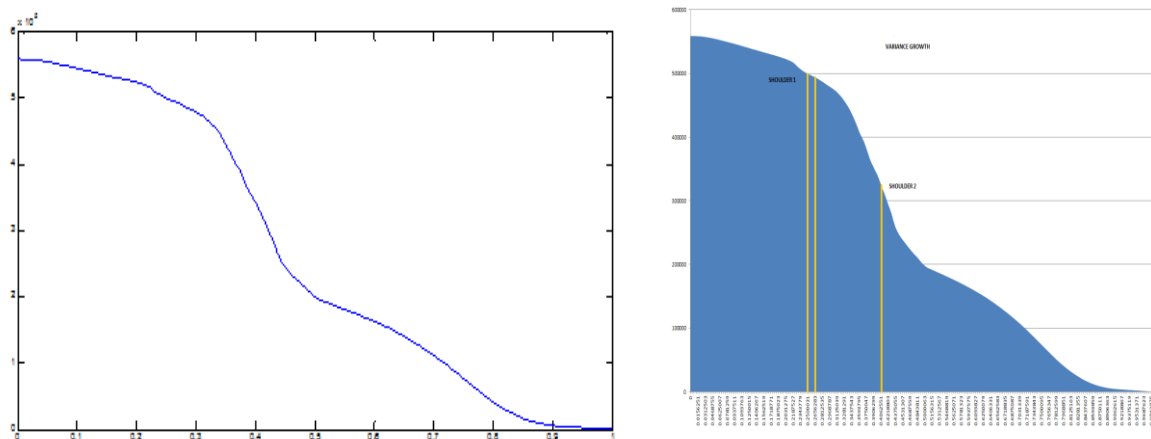


Figure 2: shows variance growth in MATLAB and Ms Excel

To better visualize and analyze the pixel values, the output from MATLAB was exported and plotted in Microsoft Excel, as illustrated in the Figure 3. From the variance growth curve shown in Figure 2, it can be observed that the region between Shoulder 1 and Shoulder 2 provides a good estimate for the optimal threshold values. Several threshold values within this range were tested, and the resulting binary images were evaluated to determine which value produced the most accurate segmentation of the object of interest.

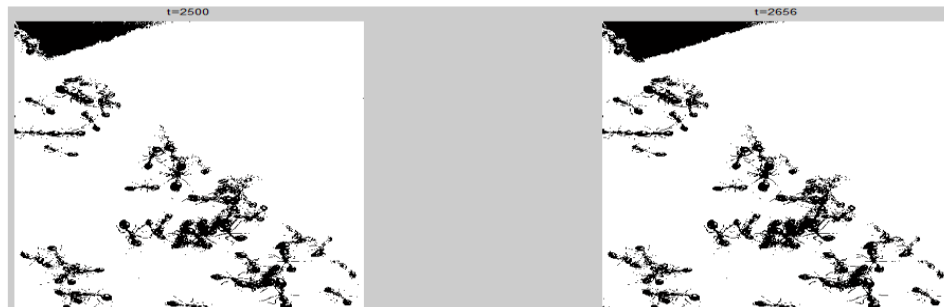


Figure 3: displays selected values for the threshold.

From the selected threshold values within the two shoulder points on the variance growth curve, it can be concluded that the threshold value of 0.2500 provides a more accurate representation of the object of interest in the image. The foreground clearly highlights the object, while the remaining areas are identified as the background. It is also evident that any threshold values outside these two shoulder points result in undesirable outcomes, yielding poor image segmentation.

CONTAMINATION WITH NOISE

Noise refers to random errors in image processing, often caused by imperfections during the image acquisition process. These errors can degrade the image, leading to a loss in both quality and intensity. There are several types of noise in image processing, including salt and pepper noise, Gaussian noise, speckle noise, Poisson noise, and others. Each type of noise manifests in different ways.

In MATLAB, noise can be added to an image using the command:

`matlab imnoise(g, 'type of noise'([ ]),'`

Where `[ ]` defines the intensity of the noise to be introduced. For this task, salt and pepper noise is used.

Salt and pepper noise occur when random high and low intensity pixel values (black and white) are introduced into the image. This type of noise is also referred to as impulse or binary noise. The impact of salt and pepper noise on the image, along with its effect on the histogram distribution, is shown in Figure 4.

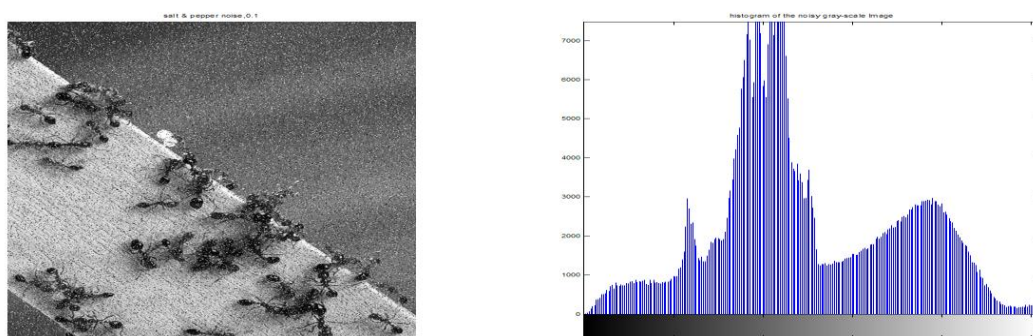


Figure 4: Image Contaminated with Salt and Pepper Noise and Its Corresponding Histogram Distribution

The introduction of noise into the image, as illustrated above, has minimal visual impact due to the low noise intensity used. At lower levels, the noise does not significantly distort the image, resulting in a histogram that remains relatively consistent with the original.

Noise-Removal Methodology

Noise contamination in images leads to degradation and loss of critical visual information, posing a major challenge in image processing. To address this, effective noise removal techniques, also known as filtering, are applied to reduce or eliminate noise while preserving essential image details. Filtering involves applying mathematical operations to the pixel values of an image using specialized algorithms. Different types of filters are tailored to target specific types of noise. Common filtering techniques include:

Median filter, Wiener filter, Rank filter, Gaussian filter, and Mean filter.

When noise is added to an image, it leads to loss of vital information in the image which is a major concern in image processing. As a result of this, an effective noise-removal methodology is needed to help clear out or reduce the effect of the noise in the image.

Noise removal methodology is also known as filtering in image processing which involves removing or reducing the effect of noise in an image. Different filters are employed for different noise.

There are various filters which includes; median filter, wiener filter, rank filter, Gaussian filter, mean filters etc. In this task, salt and pepper noise was introduced to the image, and three filters were applied to remove it:

1. Median. Filter. 2. Wiener. Filter. 3. Rank Filter.

The output images for each filtering method are displayed in the figures below.

MATLAB Implementation

The filtering techniques were implemented in MATLAB using the following commands, where `r` represents the noisy image:

Rank filter = `ordfilt2(r,9,ones(3));` and `ordfilt2(r,25,ones(5));` where `r` is the noisy image, `ones(3)` and `ones(5)` displays the 3x3 and 5x5 mask respectively.

Wiener filter = `wiener2(r,[3 3]);` and `wiener2(r,[5 5]);`

Median filter = `medfilt2(r,[3 3]);` and `medfilt2(r,[5 5]);`.

From the results, it is evident that the median filter performs best in removing salt and pepper noise, especially at low noise intensities. Compared to the Wiener and Rank filters, the median filter more effectively restores image quality with minimal loss of information.

However, as the noise intensity increases (e.g., to 0.3), the median filter's performance degrades. At this level, significant image details may be lost, indicating that the median filter has limitations when handling high-intensity salt and pepper noise.

Overall, the median filter remains the most suitable method for removing salt and pepper noise at moderate intensities. The effect of applying the median filter to an image with a 0.3 noise intensity is shown in the figure 5.

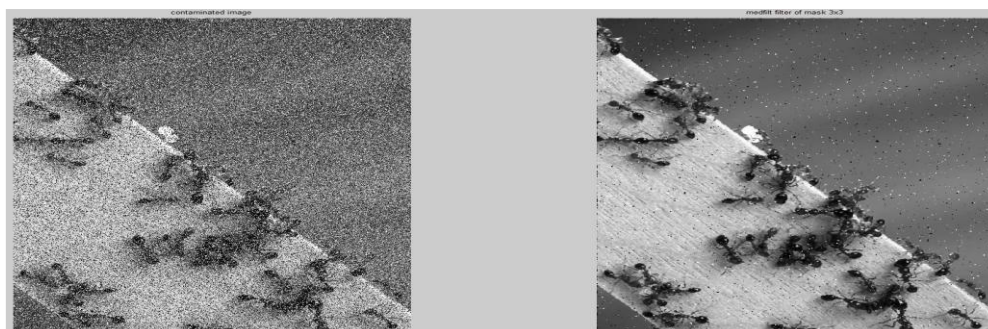


Figure 5: indicates filtered image with its histogram.

It is evident from the figure above that the median filter has effectively restored the image to a form very close to its original state. When comparing the histogram distribution of the filtered image to that of the original, only minor differences are observed. However, the histogram indicates a slight reduction in pixel intensity values, suggesting that some loss of information occurs during the noise removal process -even when the noise is minimal. Although this loss may not be immediately noticeable, it highlights an inherent limitation of filtering techniques.

Histogram equation

Histogram equalization is a technique used to enhance the contrast of an image by redistributing its pixel intensity values across the full range of possible values. This process results in a more balanced and visually improved image. In MATLAB, histogram equalization can be performed using the following command: `matlab histeq(g)`

Where `g` is the gray-scale image to be equalized. This function automatically adjusts the image to produce a more uniform histogram, thereby improving its contrast.

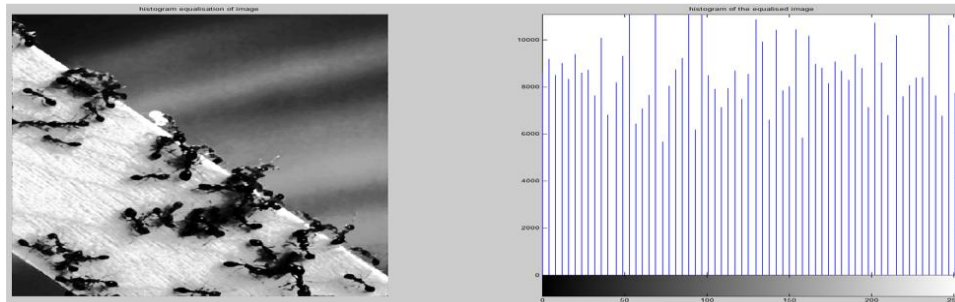


Figure 6: demonstrates histogram equalization of the filtered image.

From the figure 6, it is evident that the pixel values of the image have been effectively spread across the entire intensity range. This indicates a significant improvement in the image contrast. The impact of this enhancement on the histogram distribution is clearly noticeable when compared to the histogram of the original image, with a marked difference between the two.

REPETITION OF THE ABOVE STEPS

The next step is to find the threshold value of the equalised image using the variance growth methodology.

The command function in MATLAB is listed as follows;

The next step involves determining the threshold value for the equalized image using the variance growth methodology. The following MATLAB commands are used to implement this process:

`C=cumsum(he); % computes the cumulative sum of the histogram`

`Var=max(C)-C; % computes the variance`

`Horz=0:1/255:1; % specifies the x-axis`

`Plot(horz,var); %generates the resulting curve.`

Where 'he' is the histogram of the equalised image.

The resulting output of the plot in MATLAB is displayed in the figure 7;

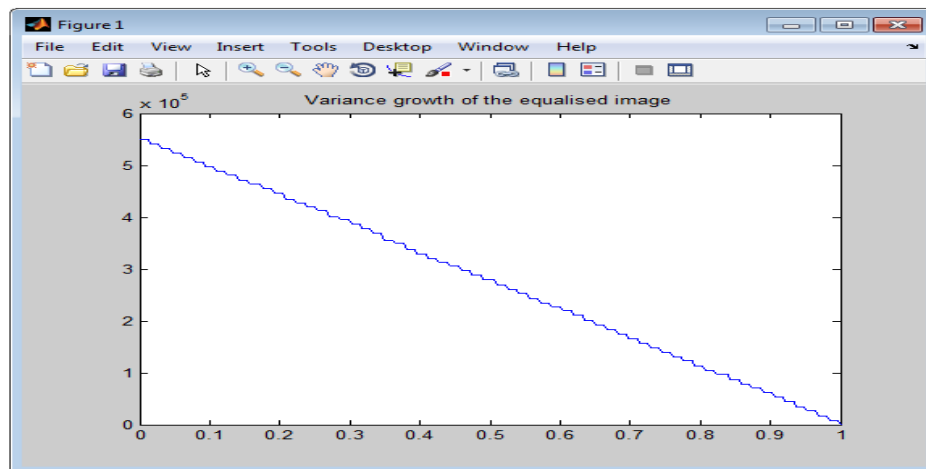


Figure 7: displays variance growth of the equalized image.

To determine the threshold value for the object of interest, we select different values that lie between the two shoulder points indicated in Figure 7 above. The chosen threshold values are 0.250 and 0.2656. The resulting images for each threshold are shown in figure 8:



Figure 8: Resulting images for the selected threshold values.

The output images clearly depict the object of interest. Appreciations to the efficient median filter applied to remove noise from the equalized image, the noise's effect on the threshold is minimal. The filtered image closely resembles the original, though the histogram equalization process has caused a slight reduction in the threshold value when compared to the pre-equalized image. For this analysis, we will use the image with a threshold value of 0.3437 as the final threshold for the object of interest.

Addition of Noise to The Equalised Image

Next, salt and pepper noise with an intensity of 0.8 will be added to the equalized image. Following this, the median filter will be applied to remove the noise. The result of adding this noise to the image is shown in the figure 9, and 10:

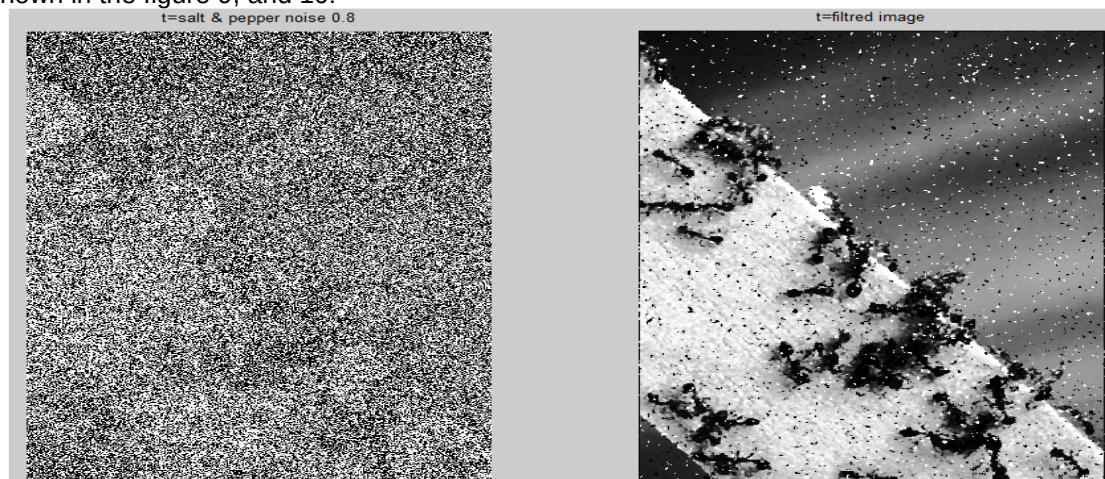


Figure 9: illustrates salt and pepper noise with an intensity of 0.8 and filtered image the noise

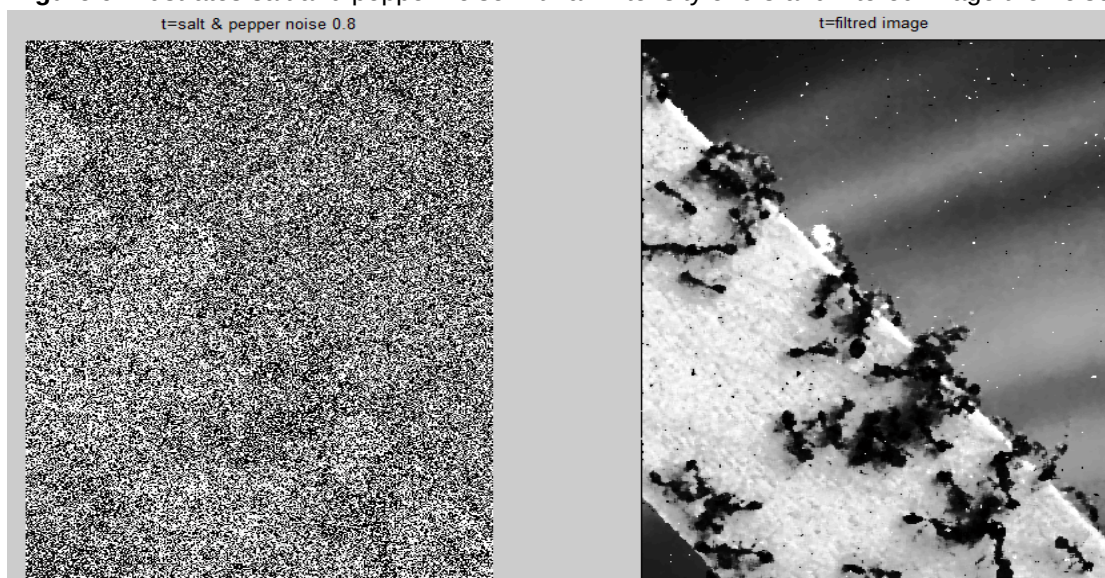


Figure 10: shows salt and pepper noise with an intensity of 0.8 and the best filtered image

The intensity of the added salt and pepper noise has been increased to 0.8. As observed in Figures 9 and 10 above, the 7x7 mask filter proved to be highly effective, as there are no visible noise particles in the filtered image. This is in contrast to the 5x5 mask, which still shows some residual noise. While the 7x7 filter results in a slightly blurred image compared to the 5x5 mask, it was preferred due to the significantly greater reduction in noise. It is important to note that the introduction of noise to an image inevitably leads to loss of information. Even with the most suitable filter, achieving a perfect restoration of the original image is impossible, as some effects of the noise remain.

Discussion and Analysis

The results from the analysis clearly demonstrate the impact of noise on the histogram distribution of an image. Noise reduces the sharpness and quality of the image, even when the appropriate filter is applied. The intensity of the image is diminished, and although efforts to restore the image bring it closer to its original form, some information loss is inevitable.

The median filter, used to remove salt and pepper noise in this task, proved to be highly effective compared to other filters. It can be concluded that the median filter is the best choice for removing salt and pepper noise. However, a key limitation of this filtering method is that its effectiveness decreases as the noise intensity increases. Therefore, for higher-intensity noise, more advanced filtering techniques will be required to achieve optimal results in future applications.

Conclusion

The image processing task was successfully completed using MATLAB. The histogram distribution of the image was generated and thoroughly analyzed. An appropriate threshold for segmenting the object of interest (the ant) was determined and its selection was justified. The impact of introducing salt and pepper noise was evaluated, particularly in relation to changes in the histogram distribution. Various filtering techniques were applied to remove the noise, and their effectiveness was assessed and compared. While the applied methods showed promising results, further research is ongoing to identify the most efficient filtering techniques. Overall, the project demonstrated a solid understanding of key digital image processing concepts. The literature demonstrates a strong foundation for using MATLAB in processing and analyzing insect images, particularly ants. Techniques such as histogram-based thresholding, denoising, and contrast enhancement are critical in developing accurate, efficient, and reproducible workflows. As computational entomology advances, MATLAB continues to be a valuable platform for researchers aiming to automate and refine the analysis of biological imagery.

References

- [1] Charbit, M., & Blanchet, G. (2006). Digital signal and image processing using MATLAB (ISTE).
- [2] Marques, O. (2011). *Practical image and video processing using MATLAB*. John Wiley & Sons.
- [3] McAndrew, A. (2004). An introduction to digital image processing with matlab notes for scm2511 image processing. *School of Computer Science and Mathematics, Victoria University of Technology*, 264(1), 1-264.
- [4] Shi, C. Y. (2014). MATLAB-based digital image processing technology research. *Applied Mechanics and Materials*, 687, 3769-3772.
- [5] Solomon, C., & Breckon, T. (2011). Fundamentals of digital image processing: A practical approach with examples in MATLAB. Chichester, West Sussex: Wiley-Blackwell
- [6] Pitas, I. (2000). Digital image processing algorithms and applications. Wiley.
- [7] Pratt, W. K. (2007). Digital image processing: PIKS Scientific inside (4th ed.). Wiley-Interscience.
- [8] Smith, B., Hermesen, M., Lesser, E., Ravichandar, D., & Kremers, W. (2021). Developing image analysis pipelines of whole-slide images: Pre-and post-processing. *Journal of Clinical and Translational Science*, 5(1), e38.
- [9] Park, H., Son, J. H., Jung, H. S., Kweon, K. E., Lee, K. D., & Kim, T. (2020). Development of the precision image processing system for CAS-500. *Korean Journal of Remote Sensing*, 36(5), 881-891.
- [10] Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9 (1), 62–66. <https://doi.org/10.1109/TSMC.1979.4310076>
- [11] Rahman, S., Uddin, J., Zakarya, M., Hussain, H., Khan, A. A., Ahmed, A., & Haleem, M. (2023). A comprehensive study of digital image steganographic techniques. *IEEE Access*, 11, 6770-6791.
- [12] Wang, W., Li, H., Zhao, P., Zhang, Z., Zang, D., Wang, C., ... & Lu, Y. (2019). Advanced digital signal processing for reach extension and performance enhancement of 112 Gbps and beyond direct detected DML-based transmission. *Journal of Lightwave Technology*, 37(1), 163-169.
- [13] Patel, R. K., Alimi, I. A., Muga, N. J., & Pinto, A. N. (2020). Optical signal phase retrieval with low complexity DC-value method. *Journal of Lightwave Technology*, 38(16), 4205-4212.