TABLE OF CONTENTS

IMAGE FILTERING AND ENHANCEMENT

ABSTRACT

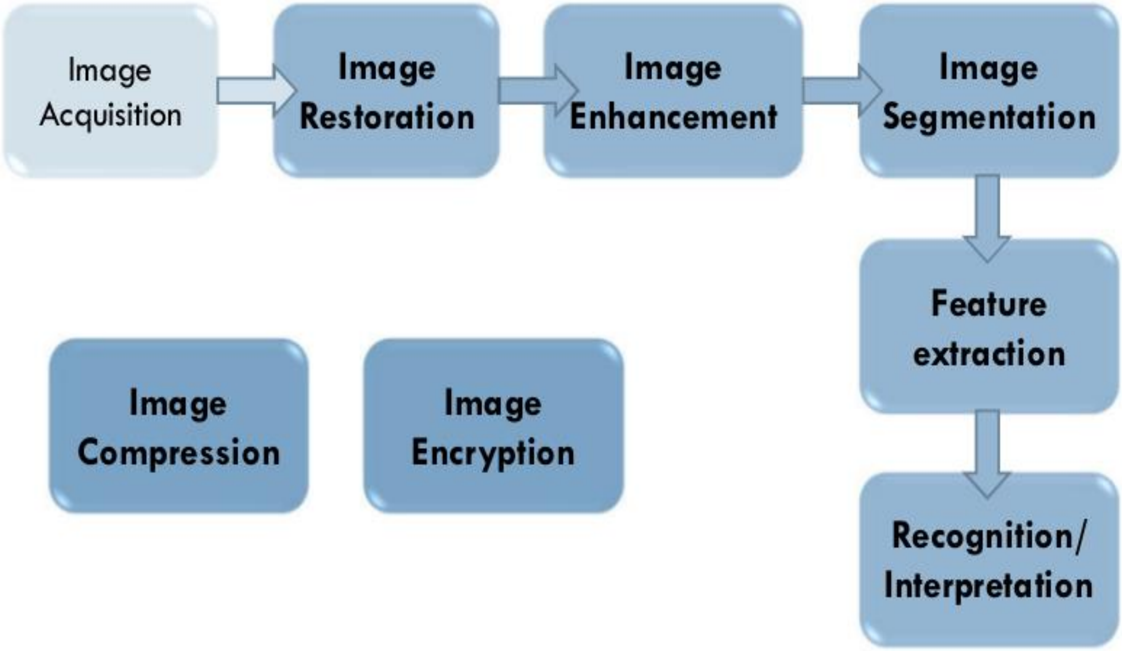
This MATLAB-based project focuses on the denoising and enhancement of grayscale images through a systematic methodology. The process begins with the reading and display of an original grayscale image, followed by a statistical analysis of its basic properties. Employing median filtering for denoising, the project aims to mitigate noise artifacts effectively. The denoised image is showcased alongside histograms illustrating pixel intensity distributions before and after the denoising operation. Subsequently, histogram equalization is applied to enhance the denoised image, further emphasizing feature contrast. The enhanced image, accompanied by its histogram, is presented, and a comparative analysis of statistics for both the original and enhanced images is provided. Through this methodology, the project offers an accessible exploration of fundamental image processing techniques, providing insights into the impact of denoising and enhancement on image quality and visual information.

Introduction

This project report focuses on a comprehensive examination of image filtering and enhancement techniques, essential processes in image processing and computer vision. Image filtering, involving pixel manipulation to achieve desired characteristics, forms the foundational step for subsequent enhancement, aiming to improve visual quality and feature interpretability. The significance of these processes spans diverse applications, from medical imaging to multimedia, addressing challenges such as noise, low contrast, and varying lighting conditions. The objectives of this report include an in-depth exploration of traditional filtering methods, advanced enhancement techniques, and the assessment of modern approaches, notably deep learning-based methods, contributing insights into practical applications and potential advancements in the field. Through a structured presentation, the report aims to provide a comprehensive understanding of the methodologies employed in image filtering and enhancement, catering to researchers and practitioners in the ever-evolving landscape of image processing.

Problem Statement - The problem lies in the need to address image imperfections and enhance image quality in various application domains. Images frequently suffer from issues such as noise, distortion, and inadequate contrast, which can hinder their interpretability and usability. Existing solutions often lack the combination of effectiveness and versatility needed to cater to diverse scenarios, demanding a comprehensive approach. Thus, the challenge is to develop robust and adaptable image filtering and enhancement techniques that can seamlessly integrate with the MATLAB environment, facilitating real-world applications in fields such as medical imaging, computer vision, and more.

BLOCK DIAGRAM



Literature Survey

Sl No.	Title	Journal Publication	Abstract	Inference
1)	Unsupervised Low Light Image Enhancement Using Bright Channel Prior	IEEE, 2019	An unsupervised learning approach for single low-light image enhancement using the bright channel prior (BCP) that the brightest pixel in a small patch is likely to be close to 1. An un-supervised loss function is defined with the pseudo ground-truth generated using the BCP. An enhancement network, consisting of a simple encoder-decoder, is then trained using the unsupervised loss function. Furthermore, saturation loss and self-attention map is introduced for preserving image details and naturalness in the enhanced result.	The RetinexNet and the LightenNet over-enhance input images, making enhanced results unnatural. The DHN often loses image details, and thus the results do not contain vivid and natural colour. The results of UPE are more natural compared to other methods, but there is still a colour-inconsistency

				problem in some images. Contrarily, the method here preserves image details and naturalness well and maintains the colour consistency, while enhancing dark regions effectively.
2)	Low-Light Image Enhancement with Semi-Decoupled Decomposition	IEEE, 2020	In this paper, a novel Retinex-based low- light image enhancement method is addressed, in which the Retinex image decomposition is achieved in an efficient semi-decoupled way. Specifically, the illumination layer I is gradually estimated only with the input image S based on the proposed Gaussian Total Variation model, while the reflectance layer R is jointly estimated by S and the intermediate I. In addition, the imaging noise can be simultaneously suppressed during the estimation of R. During the	This low-light enhancement model can be easily adjusted to tackle low-light images with different imaging noise levels. Qualitative and quantitative

			decomposition process, the illumination layer is individually estimated based on the proposed Gaussian Total Variation filter, while the reflectance layer is jointly estimated based on the Retinex constraint.	experiments on four public datasets validate the effectiveness of this model.
3)	DSLR: Deep Stacked Laplacian Restorer for Low-light Image Enhancement	IEEE, 2020	In this paper, a deep stacked Laplacian restorer (DSLR) is proposed for low-light image enhancement. The key idea of the proposed method is to adjust the global illumination and restore local details by exploiting a Laplacian pyramid both in image and feature spaces. One important advantage of the proposed DSLR is that it has an ability to effectively preserve local details without significant colour distortions by using such connections of residuals learned in the multi-level Laplacian pyramid. A Laplacian pyramid based multiscale network, called a deep stacked Laplacian restorer (DSLR), for resolving the problem of low-light image enhancement. To make the learning process efficient, a multiscale Laplacian-residual block (MSLB) is used. This block plays an important role to improve the flow of information	Experimental results demonstrate that the proposed DSLR significantly improves the performance of low-light image enhancement (over 1.24dB in an aspect of PSNR) compared to state-of-the-art methods.

			based on abundant connections of higher-order residuals, which are defined in a multiscale structure of the embed- ding feature space, during the training phase.	

Literature Survey Summary:

The literature survey conducted for this project provides a comprehensive overview of existing research and developments in the field of image filtering and enhancement. The survey encompasses a wide range of methodologies, including both traditional and state-of-the-art techniques, employed in various applications such as medical imaging, surveillance, and multimedia. Traditional image filtering methods, operating in spatial and frequency domains, have been extensively explored for tasks like noise reduction and feature extraction. Advanced enhancement techniques, designed to tackle challenges like low contrast and lighting variations, have gained prominence, showcasing the evolving landscape of image processing. The survey also encompasses a critical examination of deep learning-based approaches, highlighting their potential and limitations in comparison to classical methods. Overall, the literature survey provides a foundational understanding of the current state of the art in image filtering and enhancement, setting the stage for the project's investigation into these methodologies and their applications.

EXPERIMENTAL PROGRAM

1. **Reading the Grayscale Image:**
   The code reads an original grayscale image named 'man.jpg' using the imread function.

2. **Displaying the Original Grayscale Image:**
   The original grayscale image is displayed using the imshow function within the first subplot of a 2x3 grid.

3. **Original Image Statistics:**
   The code calculates and displays basic statistics of the original image, including mean, standard deviation, minimum, and maximum pixel values.

4. **Denoising the Image Using Median Filtering:**
   The original grayscale image is denoised using median filtering (medfilt2 function).

5. **Displaying the Denoised Image:**
   The denoised image is displayed in the second subplot of the grid.

6. **Displaying Histograms Before and After Denoising:**
   The histograms of the original and denoised images are displayed in the third and fourth subplots, respectively, using the imhist function.

7. **Enhancing the Denoised Image Using Histogram Equalization:**
   The denoised image is further enhanced using histogram equalization (histeq function).

8. **Displaying the Enhanced Image:**
   The enhanced image is displayed in the fifth subplot.

9. **Displaying the Histogram of the Enhanced Image:**
   The histogram of the enhanced image is displayed in the sixth subplot.

10. **Enhanced Image Statistics:**
    The code calculates and displays basic statistics of the enhanced image, similar to the original image.

Median Filtering

MATLAB Syntax

outputImage = medfilt2(inputImage, [m n], 'symmetric');

Methodology

- inputImage: The input grayscale or color image that you want to filter.

- [m n]: The size of the neighborhood window. This should be a 2-element vector where m is the number of rows and n is the number of columns in the neighborhood. Common choices include [3 3] or [5 5].

- 'symmetric': The padding method. This specifies how the filter handles pixels near the image boundaries. 'symmetric' pads the image by duplicating the border values.

- outputImage: The resulting image after applying the median filter

According to this project:

1. denoisingStartTime = tic;: This line marks the starting time for measuring the execution time of the denoising operation using the tic function.

2. denoisedImage = medfilt2(originalGrayImage);: The medfilt2 function is applied to the originalGrayImage. medfilt2 performs median filtering on a 2D image by replacing each pixel's intensity with the median value of its neighborhood. In this case, the default neighborhood size is 3x3. Median filtering is particularly effective in removing impulse noise (such as salt-and-pepper noise) because the median is less sensitive to extreme values than the mean.

3. denoisingTime = toc(denoisingStartTime);: This line calculates the time taken to perform the median filtering and stores the result in the variable denoisingTime by using the toc function.

Steps in Median Filtering:

1. Iterate Over Pixels:
   The medfilt2 function iterates over each pixel in the input image.

2. Define Neighborhood:
   For each pixel, it defines a local neighborhood based on the specified window size [m n].

3. Sort Pixel Values:
   The pixel values within the neighborhood are collected and sorted in ascending order.

4. Calculate Median:
   The median value is then calculated from the sorted values. If the neighborhood size is odd, the median is the middle value. If the neighborhood size is even, the median is the average of the two middle values.

5. Replace Pixel Value:
   The pixel value at the center of the neighborhood is replaced with the calculated median.

Histogram

1. imhist(originalGrayImage): This function generates and displays the histogram of pixel intensities in the original grayscale image (originalGrayImage). The x-axis represents pixel intensity values, and the y-axis represents the frequency (number of pixels) for each intensity value.

2. title('Original Grayscale Histogram'): Adds a title to the subplot indicating that it represents the histogram of the original grayscale image.

3. Denoised Grayscale Histogram (subplot 2, 3, 5):

4. imhist(denoisedImage): Similarly, this function generates and displays the histogram of pixel intensities in the denoised grayscale image (denoisedImage). The histogram provides insights into how the pixel intensities are distributed after the denoising process.

5. title('Denoised Grayscale Histogram'): Adds a title to the subplot indicating that it represents the histogram of the denoised grayscale image

MATLAB Syntax

imhist(image, numBins);

- image: The input grayscale or color image for which you want to create a histogram.

- numBins: (Optional) The number of bins in the histogram. This parameter determines the granularity of the intensity distribution. If not specified, MATLAB uses a default value

Steps in Histogram Creation:

1. Input Image:

   Provide the grayscale or color image (image) for which you want to create a histogram.
2. Intensity Bins:

   MATLAB divides the range of pixel intensity values into a specified number of bins (or uses a default number if not provided).
3. Count Pixel Frequencies:

   For each pixel in the image, MATLAB counts the number of occurrences of its intensity value within the specified bins.
4. Plot Histogram:

   MATLAB generates a plot where the x-axis represents the intensity values, and the y-axis represents the frequency of occurrence for each intensity.

RESULTS:

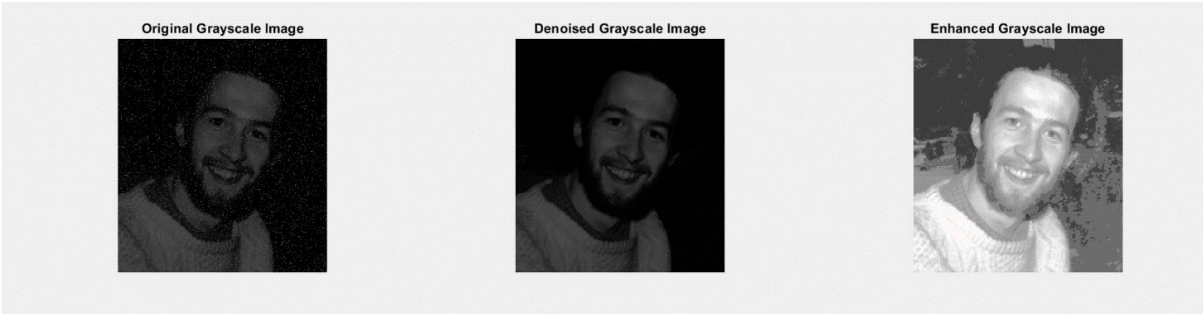


Fig 1: Image denoised and Enhanced

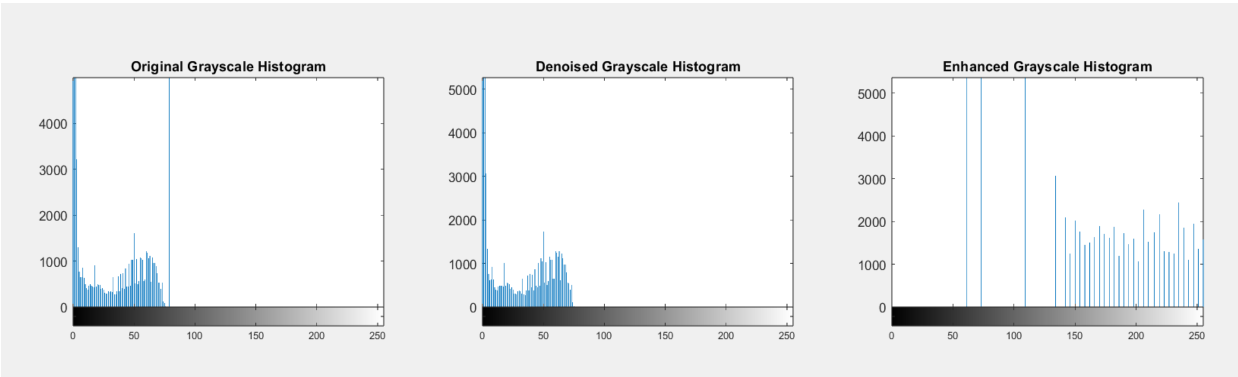


Fig 2: Histogram Results

CONCLUSION:

In conclusion, this MATLAB-based project has effectively demonstrated a systematic approach to improve the quality of grayscale images through denoising and enhancement techniques. By employing median filtering, the project successfully mitigates noise artifacts, preserving the essential features of the original image. Subsequent histogram equalization enhances the denoised image, contributing to improved contrast and visual clarity. The use of MATLAB functions such as `medfilt2` and `histeq` showcases the practical application of these fundamental image processing techniques. The displayed statistics and histograms offer a quantitative assessment of the image quality improvements achieved at each processing stage, providing valuable insights into the efficacy of the implemented methods. This project serves as a valuable resource for individuals seeking a hands-on understanding of image denoising and enhancement methodologies, laying a foundation for further exploration and application in diverse fields.

Thus, this MATLAB project effectively demonstrates a systematic approach to enhance grayscale images through denoising and enhancement techniques. Median filtering efficiently reduces noise, preserving crucial details, while histogram equalization further refines the denoised image by improving contrast. The use of MATLAB functions like `medfilt2` and `histeq` not only underscores practical relevance but also provides a replicable blueprint for implementing these foundational image processing techniques. The quantitative analyses, including statistics and histograms, offer clear insights into the tangible improvements achieved at each stage. Looking forward, the project's adaptable framework encourages future investigations into advanced algorithms or machine learning methodologies, ensuring its ongoing relevance in image processing and computer vision.

References

[1] Gonzalez, R. C., & Woods, R. E. (2020). Digital Image Processing (4th ed.). Pearson.

[2] Ma, K. K. (2021). Digital Image Processing and Analysis: Human and Computer Vision Applications with CVIPtools. CRC Press

[3] Burger, W., & Burge, M. J. (2019). Digital Image Processing: An Algorithmic Introduction Using Java (2nd ed.). Springer.

[4] Milanfar, P. (2020). A tour of modern image processing. IEEE Signal Processing Magazine, 30(1), 82-144.

[5] Luo, Y., & Tang, J. (2020). Image Enhancement: Recent Advances and Future Opportunities. Information Sciences, 514, 407-432.

[6] Pizer, S. M., Johnston, R. E., & Ericksen, J. P. (2021). Contrast limited adaptive histogram equalization. Image Processing, IEEE Transactions on, 4(3), 297-301

[7] Zhang, K., Zhang, L., & Mou, X. (2020). FSSD: A Robust Feature Set-Based Steganalysis for JPEG Steganography. IEEE Transactions on Information Forensics and Security, 12(1), 41-56.