IMAGE DENOISING

A project report submitted in partial fulfillment of the requirements for the degree of

MASTER OF COMPUTER APPLICATION (MCA)
OF
TEZPUR UNIVERSITY
2023

Submitted by

DONALD MAHANTA (CSM22027)
MD. AYUBALI, (CSM22028)
SWATI CHANCHAL, (CSM22044)

Guided by

DR. SWARNAJYOTI PATRA,
ASSISTANT PROFESSOR

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
TEZPUR UNIVERSITY
TEZPUR 784028
ASSAM.

TEZPUR UNIVERSITY

Certificate

This is to certify that the minor project report entitled **Image Denoising**, submitted to the Department of Computer Science and Engineering, Tezpur University, in partial fulfillment for the award of the Postgraduate degree in Master of Computer Application, is a record of bonafide work carried out by **Donald Mahanta**(CSM22027), **Md. AyubAli**(CSM22028) & **Swati Chanchal** (CSM22044).

Place: Tezpur
Date:

Dr. S. Saharia
[Head of Department]
Department of CSE
Tezpur University

DEPARTMENT OF

COMPUTER SCIENCE & ENGINEERING

TEZPUR UNIVERSITY

TEZPUR 784028

ASSAM.

TEZPUR UNIVERSITY

CERTIFICATE

This report entitled **“Image Denoising”** submitted by **Donald Mahanta, Md. AyubAli, Swati Chanchal** to the Department of Computer Science and Engineering in partial fulfillment of requirements for the degree of Master of Computer Application (MCA) of Tezpur University has been examined.

Internal Examiner	External Examiner
Date:	Date:
Place:	Place:

TEZPUR UNIVERSITY

CERTIFICATE

This is to certify that the minor project report on Image Denoising, submitted to the Department of Computer Science and Engineering, Tezpur University in the partial fulfillment for the award of the degree in Master of Computer Application, is a record of bonafide work carried out by Donald Mahanta (CSM22038), Md AyubAli (CSM22041) and Swati Chanchal (CSM22044) has been Examined.

Place: Tezpur Examiner
Date:

ACKNOWLEDGEMENT

I express my deep and sincere gratitude to Dr.SWARNAJYOTI PATRA, Assistant Professor, Department of Computer Science and Engineering. Tezpur University who has been my supervisor since the beginning of this study as well the project. This project has been far beyond completion without his exquisite support. I am greatly indebted for his valuable, constructive and critical guidance throughout the tenure of this project.

I would also like to express my solemnized gratitude and thanks to my group members Donald Mahanta (CSM22027), Md AyubAli(CSM22028) and Swati Chanchal (CSM22044) and my friends who have been by my side and pouring their valuable advice and encouragement.

Donald Mahanta(csm22027)
Md AyubAli(csm22028)
Swati Chanchal(csm22044)
Master Of Computer Application
Dept.Of Computer Science and Engineering
Tezpur University

Table of Contents

1. INTRODUCTION……………………………………5

1.1 Motivation

1.2 Significance of Image Denoising

1.3 Objectives of the Project

2. RELATED WORK……………………………………...8

3. BACKGROUND……………………………………..8

4. PROCESS……………………………………………...3

4.1 Noise

4.2 Local Smoothing Filters

4.2.1 Mean Filter

4.2.2 Median Filter

4.2.3 Gaussian Filter

4.2.4 Drawbacks

4.3. Non local Mean Algorithm

4.3.1 Overview

4.3.2 Algorithm

5. EXPERIMENTS ……………..……………………..43

6. RESULTS & OBSERVATIONS……………………..11-13

7. CONCLUSION……………………………………..14

8. REFERENCES……………………………………..15

Abstract

While many algorithms have been proposed for image de-noising, the problem of image noise suppression remains an open challenge. This paper presents a Non-local Means algorithm that stands out as a powerful and widely used technique for achieving this goal. Unlike traditional denoising techniques that rely on local information, the Non-Local Mean algorithm takes advantage of similar patches present in natural images. By considering similar patches throughout the image, the algorithm computes weighted averages, effectively suppressing noise while preserving important image structures. We delve into the core concepts of the Non-Local Mean algorithm, exploring its theoretical basis and practical implications. We also perform a comparative study of the algorithm with local smoothing filters. Local Means algorithm stands out as a powerful tool, contributing significantly to the enhancement of image quality and the extraction of meaningful information from noisy data.

Chapter 1

Introduction

The field of digital image processing refers to processing digital images by means of digital computers. Digital image is composed of a finite number of elements, each of which has a particular location and value. These elements are called picture elements, image elements, pels and pixels. Pixel is the term used most widely to denote the elements of digital image.

In the world of digital pictures, making them super clear can be a bit tricky because of something called "noise". Noises are nothing but unwanted changes in values of pixels. Whether it's the graininess in low-light photographs or distortions in medical imaging, the presence of noise can compromise the quality of digital images. Addressing this challenge, our project delves into the intricate domain of image denoising, aiming to unravel an innovative technique that enhances visual clarity and elevates the standards of image quality. Also, it is important to note that all the judgements hereby made given in the conclusion is based on the visual quality of the output results.

2.2 Significance of Image Denoising:

Image denoising, the process of removing unwanted noise from images, is of paramount importance in ensuring the desired quality of visual data. Whether in diagnostic medical imaging where precision is critical, or in the aesthetic realms of photography and cinematography, the ability to effectively denoise images is a game-changer. It not only enhances the visual appeal but also aids in accurate analysis and interpretation.

2.3 Objectives of the Project:

Exploration of Denoising Algorithms: Delve into a comprehensive study of existing image denoising algorithms, ranging from traditional methods to state-of-the-art deep learning approaches. Analyze their strengths, weaknesses, and applicability across diverse domains.

Development of Innovative Solutions: Propose and implement novel image denoising techniques, leveraging advancements in artificial intelligence and machine learning. Strive to surpass current benchmarks and address specific challenges posed by different types of noise.

Chapter 2

Related Work

Image denoising is a well-researched area in computer vision and image processing. Various algorithms and techniques have been proposed to address the challenge of removing noise from images while preserving important details. Below are some notable image denoising algorithms that have been explored in related work:

Improved Non-Local Means Algorithm Based on Dimensionality Reduction
Golam Morshed Maruf, The University of Western Ontario

In this thesis, they have proposed an improvement in computational time for Non-Local Means consumed to measure patch similarity. The image patches are projected into a global feature space and then reduced the dimensionality of this feature space. Denoising is achieved based on this reduced feature space

Survey Paper on Image Denoising Non- Local Means framework.
Department Of Computer Engineering, RMD Sinhgad School of Engineering, Savitribai Phule Pune University,,Pune,India

The paper introduces a novel framework called Nonlocal Means based Framework (NMF) for image denoising. The goal is to reduce noise in digital images without compromising important image features. This framework addresses the challenge of images corrupted by a mixture of additive white Gaussian noise (AWGN) or impulse noise (IN).

A New Method for Nonlocal Means Image Denoising Using Multiple Images.
Wang X, Wang H, Yang J, Zhang Y (2016)

The first contribution is that they use two images to denoise the pixel. These two noised images are with the same noise deviation. Instead of using only one image, they calculate the weight from two noised images.

Chapter 3

Background

As digital technology continues to advance, the acquisition and sharing of visual information have become integral to various fields, including photography, medical imaging, satellite imagery, and more. However, the inherent imperfections in imaging sensors, coupled with external factors such as environmental conditions or hardware limitations, contribute to the introduction of unwanted artifacts known as noise.

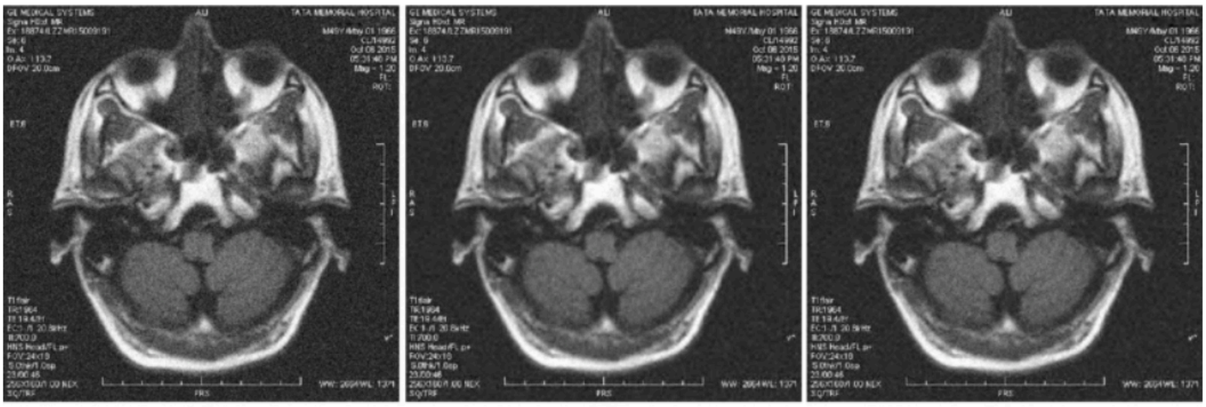
The background of image denoising techniques spans various methodologies and approaches that have evolved over the years. Here's a brief overview of the historical development and key concepts in the background of image denoising:

1960s: Pioneering work by Wiener and Lee laid the foundation for statistical image denoising, modeling noise as an additive component to the true image.

1970s-80s: Averaging and median filters gained popularity for their simplicity, albeit at the cost of blurring image details.

1990s: Bilateral filters emerged, incorporating spatial and intensity similarities for noise reduction while preserving edges.

In 2005, Buades et al. introduced the non-local means (NLM) algorithm, revolutionizing image denoising. NLM's brilliance lies in its ability to exploit image redundancy: similar textures and patterns often appear in different locations.



(a) (b) (c)

a Noisy image NLM, b denoised MRI using BM3D, c denoised MRI using NLM

Calculating weights for Non-Local Pixels
For every image, we define iterate for each (x) pixel. x determines the position of the pixel w.r.to the image. We also define R as operator which returns the pixel value of the (x).

$$\mathbf{R : x \mapsto R(x) = (I(x_1), \ldots, I(x_d))^T \in \mathbb{R}^d.}$$

After defining the Pixel operator, we define distance between two pixels x and y as d(R(x),R(y)). This distance is the euciledian distance of pixel values of pixels at x and y.

$$d\,(\mathbf{R(x), R(y)}) = |\mathbf{R(x) - R(y)}|_2^2 = \sum_{\mathbf{i=1}}^{\mathbf{d}} (\mathbf{R(x)_i - R(y)_i})^2.$$

The similarity of two pixel locations are determined by their distance. Two pixel locations are considered similar if their distance is less. This distance is normalized using the exponential filtering.

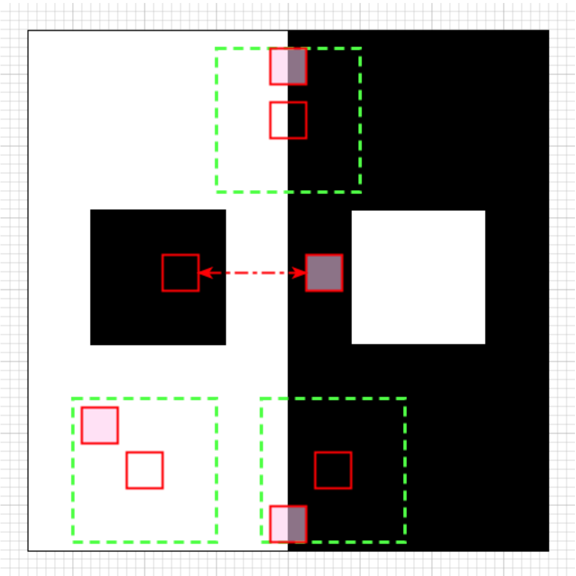
$$w(\mathbf{x}, \mathbf{y}) = \exp - \frac{\mathbf{d(x, y)}}{\mathbf{h^2}} = \exp - \frac{|\mathbf{R(x) - R(y)}|_2^2}{\mathbf{h^2}}.$$

Denoising using weights
We use the weights to obtain an average over the centered area around x: where

$$NL(\mathbf{x}) = \frac{\mathbf{1}}{\mathbf{Z(x)}} \sum_{\mathbf{y}:|\mathbf{x}-\mathbf{y}|\leq\rho} \mathbf{w(x, y) I(y)}.$$

p is our search window:

Here Z(x) is the sum of all the weights.



Chapter 4

Process

4.1 NOISE

Noises are unwanted changes in the pixel values in the digital images. This change may occur while image acquisition, processing or transferring the images. Let's have a look at certain types of noises more common to us.

Gaussian Noise: This arises during acquisition. The sensor or charge-coupled device (CCD) generates inherent noise due to the level of illumination and its own temperature. In addition to this, the electronic circuits connected to the sensor also inject their own level of electronic circuit noise.

Code to add Gaussian noise

```
import cv2
from skimage.util import random_noise
o_img = cv2.imread('image.jpg', cv2.IMREAD_GRAYSCALE)
img = random_noise(o_img, mode='gaussian', var=0.005)
```

Salt and pepper noise: Immediately recognizable by dark pixels in bright areas and light pixels in dark areas. This type of noise is a result of converting videos from analog to digital or other errors in pixel interpretation, such as dead sensor elements.

Code to add Salt & Pepper noise

```
import cv2
from skimage.util import random_noise
o_img = cv2.imread('image.jpg', cv2.IMREAD_GRAYSCALE)
img = random_noise(o_img, mode='s&p', amount=0.005)
```

Speckle noise: It is a multiplicative noise which is shown in all coherent images like radar and sonar images, and mainly in medical imaging modalities, which are based on ultrasound and laser imaging. The noise components are multiplied to each pixel of the original image. The speckle noise does not follow normal distribution.

Code to add speckle noise

```
import cv2
from skimage.util import random_noise
o_img = cv2.imread('image.jpg', cv2.IMREAD_GRAYSCALE)
img = random_noise(o_img, mode='speckle', var=0.05)
```

4.2 LOCAL SMOOTHING FILTERS

Local smoothing filters work by applying a mathematical operation to each pixel in an image based on the values of neighboring pixels. As the name suggests, local smoothing filters use only the pixels that locally surround a certain pixel to perform operations. The fundamental idea is to average or blend the colors of nearby pixels to reduce sharp transitions and enhance the smoothness of the image. Here's a simplified explanation of how it generally works:

1. **Selecting a Pixel**: The filter starts by selecting a pixel in the image, one at a time.

2. **Defining a Neighborhood**: A small region around the selected pixel, known as a neighborhood, is considered. This region includes the pixel itself and its surrounding pixels.

3. **Calculating a New Value**: The filter calculates a new value for the selected pixel based on the values of the pixels in its neighborhood. This calculation often involves taking an average or weighted average of these pixel values.

4. **Applying the Filter across the Image**: This process is repeated for every pixel in the image. The filter moves through the entire image, applying the smoothing operation to each pixel based on its local neighborhood.

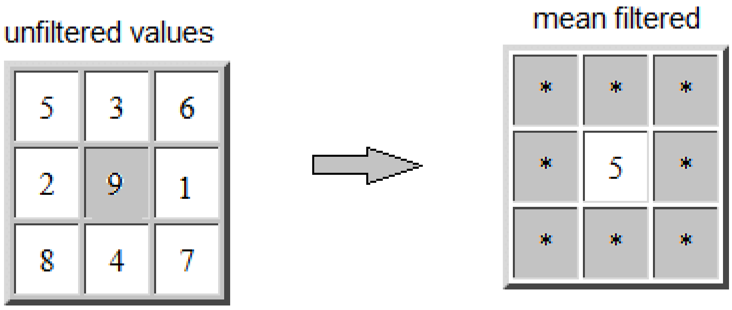
5. **Adjusting Filter Parameters**: Some filters allow users to adjust parameters, such as the size of the neighborhood or the weight assigned to each pixel in the calculation. These adjustments can impact the strength and style of the smoothing effect.

4.2.1 Mean filters

The mean filter, also known as the average filter, is a simple and widely used type of local smoothing filter in image processing.

In a 3 x 3 Mean filter, If f(x,y) is the pixel value at coordinate (x,y) and g(x,y) is the output pixel value after performing the mean filter, then

$$
\begin{aligned}
g(x,y) &= \sum f(x+i, y+j) \\
&= f(x,y) + f(x-1,y-1) + f(x-1,y) + f(x-1,y+1) + f(x,y+1) + f(x+1,y+1) \\
&\quad + f(x+1,y) + f(x+1,y-1) + f(x,y-1)
\end{aligned}
$$



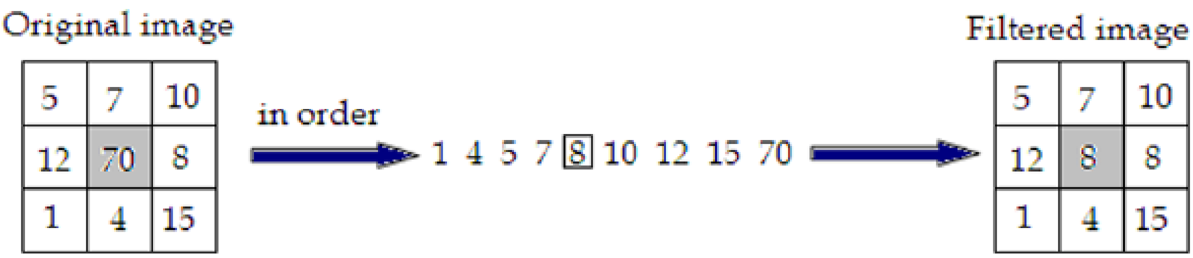
$5+3+6+2+9+1+8+4+7=45$
$45/9=5$

Code to Mean filter

```
import cv2
denoised_img = cv2.blur(img,(5, 5))
# here (5,5) is the kernel size
```

4.2.2 Median filters

Median Filter is another filtering technique which uses statistical measure to compute the denoised value. For each pixel the median filter looks for the median value in the kernel including its own value. It does so by arranging the values in an order and finding the median value as shown in the figure below. Finally, it replaces the median value with the original value.



Code to Median filter

```
import cv2
median_img = cv2.medianBlur(img,3)
# 3 is the kernel size
```

4.2.3 Gaussian Filter

Named after the Gaussian distribution, it applies a convolution operation with a Gaussian kernel to achieve its smoothing effect. Here's a detailed explanation of how the Gaussian filter works:

1. **Gaussian Kernel**: The core of the Gaussian filter is a two-dimensional Gaussian function, which is a bell-shaped curve. The function is defined by a mathematical formula that assigns weights to pixels based on their distance from the center. Pixels closer to the center have higher weights.
2. **Defining a Neighborhood**: Like other local filters, the Gaussian filter considers a local neighborhood around each pixel in the image. The size of this neighborhood is determined by the dimensions of the Gaussian kernel.
3. **Convolution Operation**: The Gaussian kernel is applied to the pixel values in the neighborhood using a convolution operation. This involves multiplying each pixel value by the corresponding weight in the Gaussian kernel, summing up the results, and assigning the weighted sum as the new value for the central pixel.

$$\text{New Pixel Value} = \sum_{i,j} (\text{Pixel Values}_{i,j} \times \text{Gaussian Weight}_{i,j})$$

4. **Adjustable Parameters**: The Gaussian filter has two main adjustable parameters: the standard deviation (σ) of the Gaussian distribution and the size of the filter kernel. A larger standard deviation results in a wider and smoother Gaussian curve.
5. **Smoothing Effect**: The Gaussian filter effectively blurs the image, reducing high-frequency noise and small-scale details. The degree of blurring is controlled by the standard deviation and the size of the kernel. Smoothing occurs more strongly for larger standard deviations and kernel sizes.

Code to Gaussian Noise

```
import cv2
gaussian_img =  cv2.GaussianBlur(img, (3, 3), 1)
# (3,3) is the kernel size and 1 is the standard deviation
```

4.2.4 Drawbacks.

In case of Local smoothing filters, Pixels are assumed to be similar in measurements just because they are spatially close inside the image. However, this assumption will be violated along the edges of each object, or everywhere a light or color gradient exists.

Although the traditional local smoothing filters like Gaussian filter, median filter, mean filter manage to remove the noise from an image, they achieve the same only by smoothing the entire image as a whole. It yields a lot of undesirable blur that destroys both the noise and the texture of the image. As a result of this process, the images lose the details around the edges that were originally present, degrading the overall quality. This might make the images unusable for certain scenarios.

4.3 Non-local Means Algorithm

4.3.1 Overview

Unlike local filtering methods that only consider nearby pixels, NLM takes into account similarities across the entire image. Here's a step-by-step explanation of how the Non-Local Means algorithm works:

1. **Patch Similarity**: For each pixel in the image, the NLM algorithm considers a small patch (a group of pixels) centered around that pixel. The algorithm then compares this patch to patches around all other pixels in the image.
2. **Similarity Measurement**: The similarity between two patches is measured using a similarity metric, often based on the Euclidean distance between the pixel intensities within the patches. Similar patches have lower distances, indicating higher similarity.
3. **Weighted Averaging**: To denoise the pixel at the center of the patch, the algorithm calculates weighted averages of the pixel values from similar patches. The weights are determined by the similarities between the reference patch and the patches around other pixels.

$$\text{New Pixel Value} = \sum_{i,j} w(i,j).\text{Pixel Value}_{i,j} \ / \sum_{i,j} w(i,j)$$

where w(i,j) represents the weight assigned to the pixel at position (i, j) based on the similarity between patches.

4. **Adjustable Parameters**: The NLM algorithm typically has parameters that control the amount of denoising, such as a filtering parameter (h) that

determines the threshold for considering patches as similar. Larger values of h result in more aggressive denoising.

4.3.2 Algorithm of Non-Local Denoising:

Here is a step-by-step algorithm for Non-Local Means (NLM) denoising:-
Input: Grayscale or color image I with Gaussian noise.
Parameters: Patch size p×p, similarity parameter h, and any other optional parameters.
Output: Denoised image $\hat{I}$

<u>Steps:</u>

1). **Initialize:** Set the denoised image $\hat{I}$ to be a copy of the noisy image I

2). **Iterate Over Pixels:** For each pixel P(x) centered at pixel x.

3). **Compute Similarities:** For each pixel x, compute the patch similarity S(x,y) with all other pixels y within the patch_distance using the similarity measure:

$$S(x,y)=e^{-\frac{||P(x)-P(y)||^2}{h^2}}$$

4). **Weighted Averaging:** For each pixel x, compute the denoised value $\hat{I}(x)$ using weighted averaging:

$$\hat{I}(x)=\frac{\sum_y S(x,y).I(y)}{\sum_y S(x,y)}$$

Where y iterates over all pixels in the image.

5). **Update Denoised Image:** Update the denoised image $\hat{I}$ with the computed values for each pixel.

6). **Repeat:** Repeat Steps 2-5 for all pixels in the image.

7). **Parameter Tuning:** Fine-tune parameters such as the patch size, smoothing parameter h, and any adaptive filtering parameters for optimal denoising results.

Code to Non-local Means algorithm

```
sigma = 0.05
nlm_img = denoise_nl_means(img, sigma = sigma, h=0.8 * sigma ,
patch_size=3, patch_distance=21 )
```

Chapter 5

Experiments

<u>Information related to tests performed</u>

Dataset

We performed the tests in 6 images whose results are shown below. This images were converted into grayscale to be considered as the ground truth images.

To denoise these images, we produced three types of noise and added them to the images to prepare for the input images. We have used the **gaussian noise** images, **salt and pepper noise** and **speckle noise** images.

Technologies

This project was done entirely using python version 3.12.0.

<u>Libraries Used:</u>

- opencv(cv2)
- numpy
- matplotlib
- skimage

Noise type added: Gaussian(for image 1,2,3), Speckle noise(for image 4)

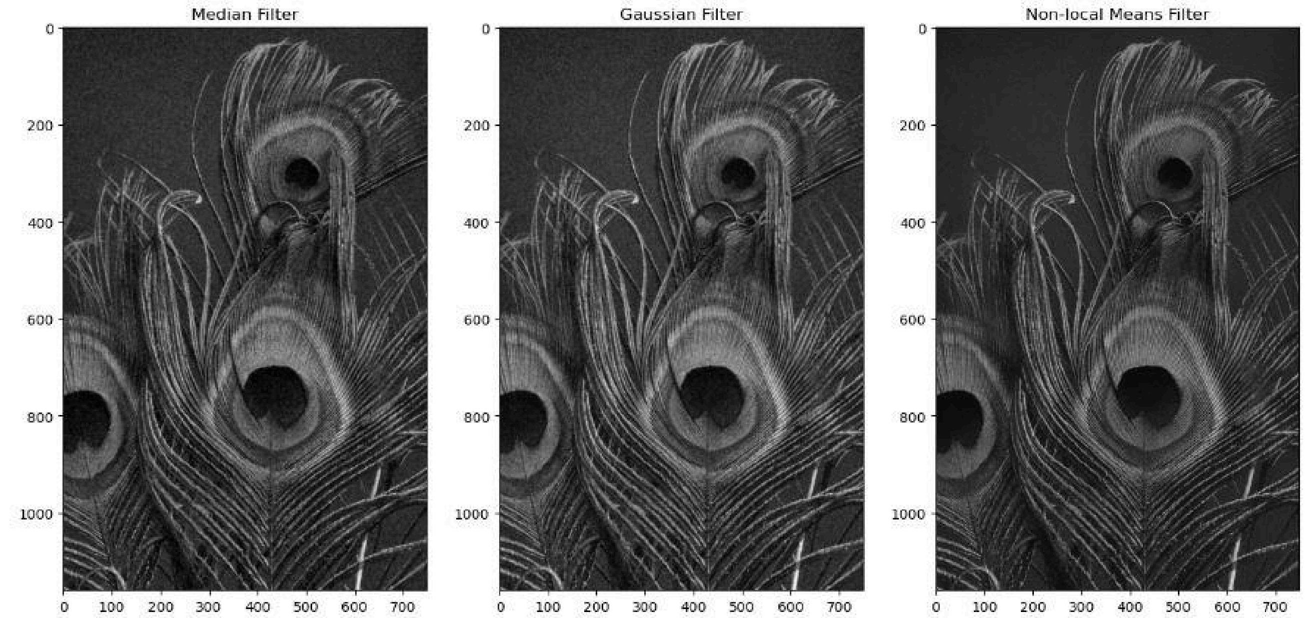
Denoising methods used:

a) Median Filter
b) Gaussian Filter
c) Non - local mean Filter

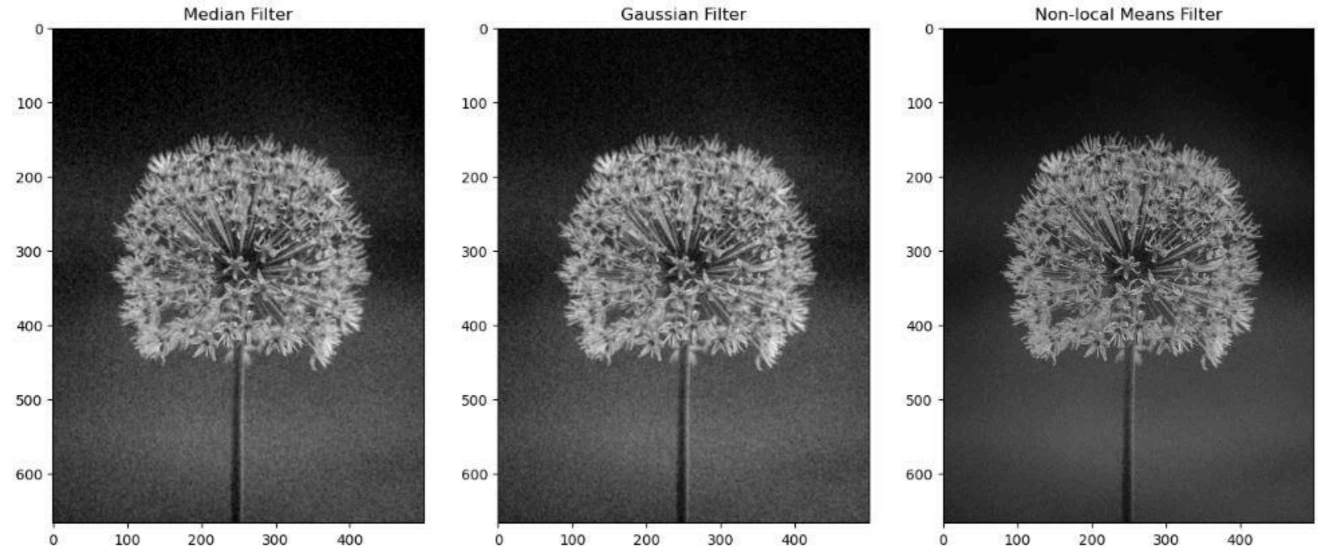
Parameters for Filtering:

Median Filter:
    kernel size: 3x3
Gaussian Filter:
    kernel size: 3x3
    Standard deviation(σ) = 1
Non-local Mean Filter:
    kernel size: 3x3
    patch_size: 3x3
    patch_distance: 21x21
    Sigma(standard deviation): 0.05

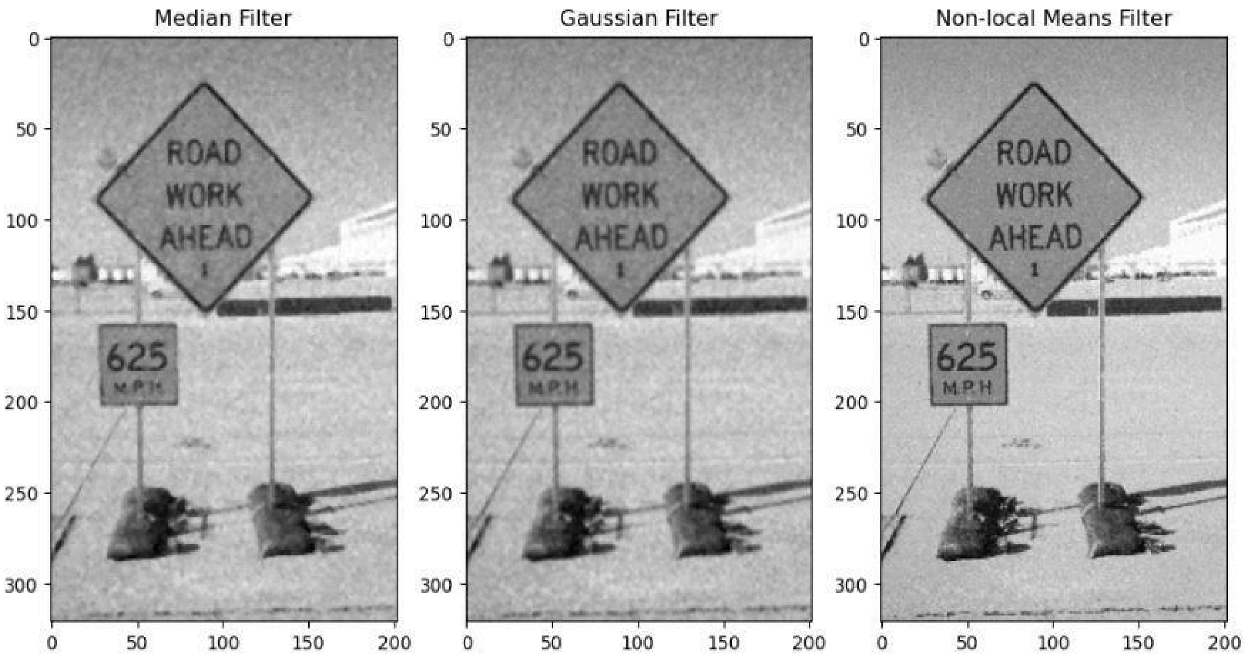
```
Median Filtering Complete. Time:0.0
Gaussian Filtering Complete. Time:0.0
NLMeans Filtering Complete. Time:23.475372076034546
```



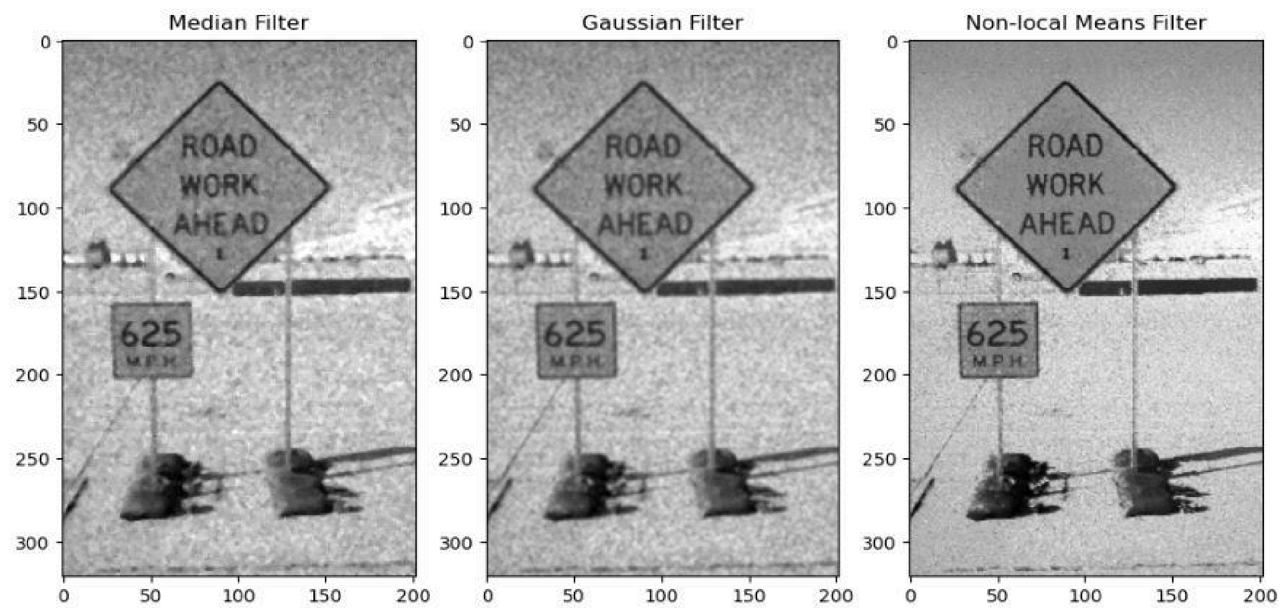

```
Median Filtering Complete. Time:0.0
Gaussian Filtering Complete. Time:0.0
NLMeans Filtering Complete. Time:9.038102626800537
```



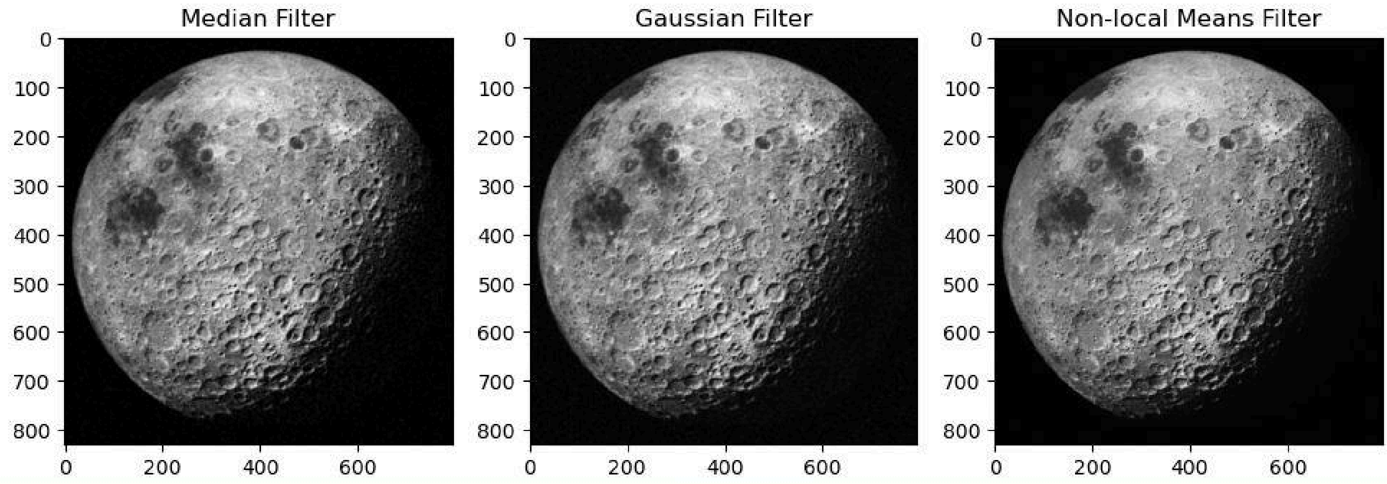
```
Median Filtering Complete. Time:0.0
Gaussian Filtering Complete. Time:0.0009996891021728516
NLMeans Filtering Complete. Time:2.015721082687378
```



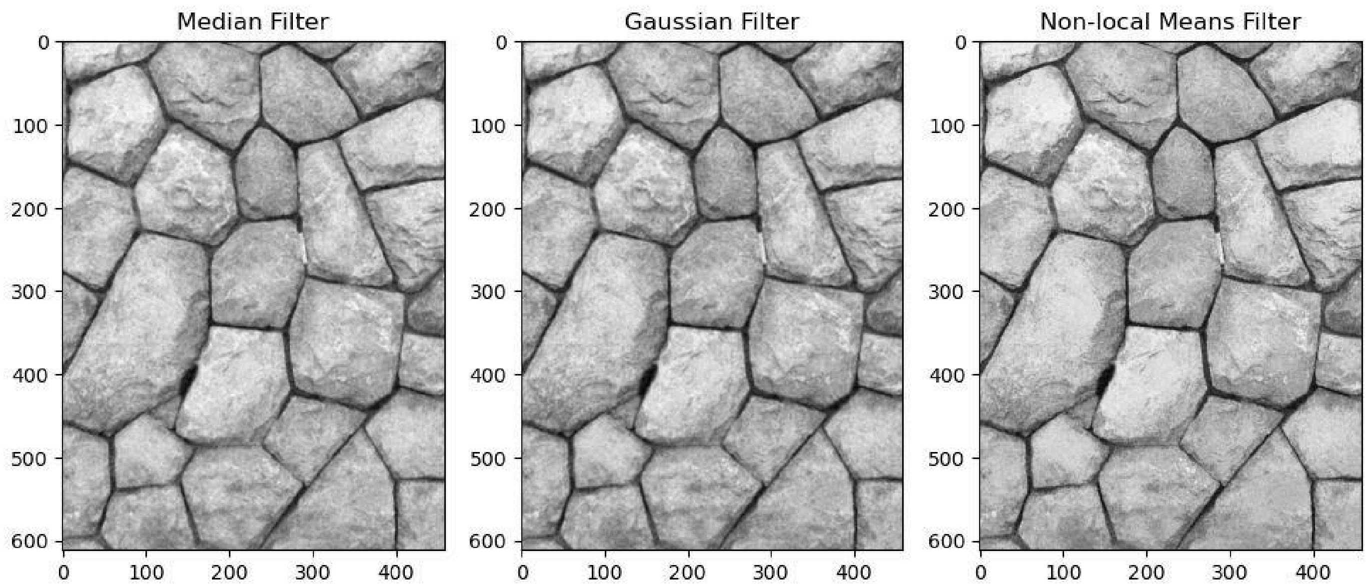

```
Median Filtering Complete. Time:0.0009944438934326172
Gaussian Filtering Complete. Time:0.0017027854919433594
NLMeans Filtering Complete. Time:2.1057353019714355
```



```
Median Filtering Complete. Time:0.0
Gaussian Filtering Complete. Time:0.015603780746459961
NLMeans Filtering Complete. Time:18.00157403945923
```




```
Median Filtering Complete. Time:0.0009961128234863281
Gaussian Filtering Complete. Time:0.0009801387786865234
NLMeans Filtering Complete. Time:7.85629415512085
```



Chapter 6

Results & Observations

Time taken:
It has been observed that median filter and Gaussian filter takes very less time as compared to the non local means which was expected since non local means algorithm has to perform some extra computations. Also the size of the image is a factor to look for. Here is the chart for detailed information.

Sl no.	Image size (in kb)	Noise type	Time taken (in secs)		
			Median filter	Gaussian filter	Non-local means
1	343	Gaussian	< 0.1	< 0.1	23.5
2	62	Gaussian	< 0.1	< 0.1	9
3	17.9	Gaussian	< 0.1	< 0.1	2
4	17.9	Speckle	< 0.1	< 0.1	2.1
5	164	Salt & pepper	< 0	< 0.01	18
6	55.8	Salt & pepper	<0.001	<0.001	7.89

Output Image Quality:

All the four output images are filtered with a 3 x 3 filter. However, the results are drastically different. Median and gaussian filters smoothen the image overall while the result from the non-local mean filter preserves the textures a lot better while also removing the noise to an ample amount.
However, the time required to perform the Non-local means algorithm is more as compared to other methods.

Note: All the judgements are made through observing the output images and comparing to the original image visually.

Chapter 6
Conclusion

From the above experiments we have come to the conclusion that the Non-local means algorithm is effective in reducing various types of noise, including Gaussian noise. It not only reduces noise from the image but also manages to preserve details and textures well especially in the edges.

It distinguishes between noise and true image features, allowing for denoising while retaining important details.

Further, it can also be said that although Non-local means can give better results at time, one of its disadvantages is that it is Computationally expensive comparatively
This is because it requires a lot of time to find similar patches around the image repetitively for each and every pixel and assign them weights based on their similarity with the main patch. This further is a time demanding task.

Also, in case of salt and pepper noise, the median filter may perform better as compared to any other algorithms discussed here. It is also observed that choosing the right parameters can be a bit tricky which needs to be changed depending on the image.

In summary, the Non-Local Means algorithm is a powerful denoising technique that leverages the similarities between patches across the entire image. It is particularly useful when preserving fine details is crucial, and it has found applications in medical imaging, photography, and computer vision.

References

1. A. Buades, B. Coll and J. -. Morel, "A non-local algorithm for image denoising," 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 2005, pp. 60-65 vol. 2, doi: 10.1109/CVPR.2005.38.

2. Antoni Buades, Bartomeu Coll, Jean-Michel Morel. A review of image denoising algorithms, with a new one. Multiscale Modeling and Simulation: A SIAM Interdisciplinary Journal, 2005, 4 (2), pp.490-530. ff10.1137/040616024ff. ffhal-00271141f

3. Angelo, Emmanuel. "NL-Means: Some Math | Computers Don't See (Yet)". Computers Don't See (Yet), 2020

4. https://kanchiuniv.ac.in/coursematerials/Digital%20image%20processing%20-Vijaya%20Raghavan.pdf

5. https://www.computersdontsee.net/post/2013-04-13-nl-means-some-math/

6. https://www.cs.auckland.ac.nz/courss/compsci373s1c/PatricesLectures/Image%20Filtering.pdf

7. https://scikit-image.org/docs/stable/api/skimage.restoration.html#skimage.restoration.denoise_nl_means

8. https://pages.stat.wisc.edu/~mchung/teaching/MIA/reading/diffusion.gaussian.kernel.pdf.pdf

9. UniHeidelberg (2013) *Non-Local Means for Image Denoising | Image Analysis Class 2013* [Online video] Available at: https://www.youtube.com/watch?v=k8hS6uTz-Uc [Accessed at: 7/12/2023]

10. DigitalSreeni (2020) *What is Non-local means (NLM) denoising filter?* Available at: https://www.youtube.com/watch?v=Va4Rwoy1v88 [Accessed at: 15/12/2023]