Digital Image Restoration Using Regularization Filter

Computer Science Graduation Project

Submitted by:

Mohamed Ahmed Hemdan Abd El-Hakeem

Amr Mohamed Abdel Moneim Morsi

Yasmin Walid Fathy Ahmed

Ahmed Abdel Moneim Abdel Halim

Abd El-Zaher Waleed Abd El-Zaher Afify

Peter Nagy Selim Atallah

Mohamed El-Sayed Mohamed Abd El-Baki

Supervised by:

Prof. Dr. Alaa Abd El-Rahim

TA Reham Fares

TA. Nada Elsayed

Acknowledgement

Our greatest in deep and gratitude to the Modern University for Technology and Information (M.T.I) forthe great role it played in forming our character and personalities, providing us with all means necessary for our educational development and providing us as well with a wonderful environment to learn in throughout our educational years. This would not have been possible unless.

Prof. Dr. Olfat Kamel, President of the MTI University has established such a learning environment, providing the most effective facilities for our job to be done. It is a pleasure to thank those who maneuvered this Possible, **Prof. Dr. Nabeel El-Deeb**, and Dean of the faculty of Computers Information**Prof. Dr. Mohamed Taher El-Mayah** for his unlimited help, support, encouragement and guidance.

It is also our duty ton record our thankfulness to **Prof. Dr. Alaa Abd El-Raheem**, and assistant lecturer Mohammed Amin for his guidance and support. And who had to bear a heavy lead of responsibility and concern in bringing this project to a successful end.

Also, Special thanks to **Prof. Dr. Hafez Abdel-Wahab**, head of the Information System Department.

A Special thanks **to Prof. Dr. Hanafy Ismail**, head of the Computer Science Department. Special thanksto **Prof. Dr. Eman Taha**, head of Basic Science Department.

Special thanks to **Prof. Dr. Hesham El-Deeb** for his effort throughout our university studies. As well as the Department staff, and Prof. Dr. Sayed Bakar for their effort along our university studies, as well as **Prof.Dr. Abd El-Aziz El-Batta "may be in rest"** for their effort along our university studies. **Prof. Dr. Mahmoud El-Shishtawy** and for their effort along our university studies.

As well as the department staff, **Prof. Dr. Elsayed Bakkar** and for their effort teaching us the basics for our computer science degree. Special thanks **to Prof. Dr. Rasha Saeed, Dr. Rania Ahmed and Dr. Marwa Rafae** Special thanks to our families who always supported us during our academic life, and we couldn't do it without their help. We are grateful to all the teaching assistant without who helped us all through our studies with practical demonstrations that helped us to understand and to practice what we have learnt.

Abstract

Image restoration is one of the most fundamental techniques and commonly used applications in the image processing used to compensate for or undo defects which degrade an image.

- **Problem statement:**

Degradation is the process in which the quality of an image is destroyed or spoiled due to a variety of reasons like blurring on image that maybe caused by many factors: movement during the image capture process by the camera or out of focus and noise on image caused by many factors: poor lighting, high iso settings and long exposure times.

- **Objective:**

To restore a damaged or degraded image to its original quality and clarity, while preserving its authenticity and maintaining the integrity of the original content and enhance after that to make it more visually appealing.

- **Methodology:**

- Define the problem statement and the objective of the project. Identify the type of image restoration and enhancement required.
- Image Restoration Techniques: Choose appropriate image restoration techniques such as wiener, Regularization or Median to restore the degraded images.
- Image Enhancement Techniques: Choose appropriate image enhancement techniques such as contrast enhancement, color correction, sharpening, or smoothing to enhance the restored images.
- Performance Evaluation: Evaluate the performance of the implemented algorithms using metrics such as PSNR (Peak Signal-to-Noise Ratio), SSIM (Structural Similarity Index), or MSE (Mean Squared Error).

- **Conclusion:**

Image restoration and enhancement is an important field of image processing that aims to improve the quality of images by removing noise, blur, and other distortions. This project has demonstrated various techniques for restoring and enhancing images.

Table of Contents

List of Tables:

List of Figures:

Chapter 1

Introduction

1.1 Introduction

On a computer or other digital device, digital images are electronic representations of visual data that can be stored, processed, and displayed.

They are composed of pixels, which are minuscule dots that when put together create an image. Due to the widespread use of digital cameras and smartphones in recent years, digital images have become more and more popular.

They are simple to share online or print out for exhibition offline. Numerous software tools can be used to edit and manipulate digital images, opening countless creative possibilities.

However, Digital images can be easily changed or manipulated in ways that might not accurately depict reality, raising questions about the veracity and dependability of digital media.

Digital image restoration is a process that involves the use of various techniques and tools to improve the quality of digital images. The goal is to restore images that have been damaged or degraded due to various factors such as noise, blur, distortion, or other forms of degradation.

The need for digital image restoration arises from the fact that digital images are often subjected to various forms of degradation during their capture, storage, transmission, or processing. These degradations can result in images that are:

- Blurry
- Noisy
- Distorted
- have other imperfections that affect their quality and usefulness.

Digital Image Restoration techniques can be broadly classified into two categories:

- Spatial domain techniques operate directly on the pixel values of an image.
- Frequency domain techniques operate on the Fourier transform of an image.

In addition to these techniques, digital image restoration also involves various preprocessing steps such as contrast, sharpening, and resizing. Contrast involves adjusting the color balance of an image so that it appears more natural or accurate. Sharpening involves enhancing edges in an image so that it appears more detailed or crisp. Resizing involves changing the size of an image while preserving its aspect ratio.

Digital image restoration is a complex process that requires a combination of different techniques and tools to achieve high-quality results. While there are many challenges associated with this process such as dealing with complex noise models or handling large datasets, advances in technology continue to make this field more accessible and effective for a wide range of applications including medical imaging, surveillance systems, and artistic expression.

1.2 Problem Definition

Observed images generated by image acquisition devices are usually not the same as the true images but are instead degraded versions of their true images. Degradations can occur in the entire process of image acquisition, and there are many different sources of degradation. For instance, in aerial reconnaissance, astronomy, and remote sensing, images are often degraded by atmospheric turbulence, aberrations of the optical system, or relative motion between the camera and the object.

Image degradations can be classified into several categories, among several point degradations (or, noise) and spatial degradations (or, blurring) are most common in applications. Other types of degradations involve chromatic or temporal effects.

Having some techniques for image denoising when there is no blurring in the observed image, each technique has its maximum noise variance value, most of them can preserve edges to certain degree, but some important edge structures such an angles, corners, and places where edges have large curvature, would be blurred, or rounded by them.

Having some techniques for image deblurring which involve identifying the type and extent of blur present in an image and developing algorithms or techniques to remove it while preserving as much detail and clarity as possible. This can be a challenging task, as different types of blurs require different approaches for effective removal, and there is often a trade-off between removing blur and preserving image quality.

1.3 Project Objective

The objective of the project is to learn how the restoration process is done by building different techniques or filters as well as other techniques being used and how it works, to restore degraded images to help us improve the quality of the image. The basic approach of this project is to get the highest resulting technique compared among the rest of the techniques used to determine the best performance to fulfil the needs of the desktop application we’re creating.

1.4 Project Management

1.4.1 Team Members:

1. Mohamed Ahmed Hemdan
2. Amr Mohamed Abd Elmoneim Morsy
3. Yasmeen Waleed Fathy
4. Abdelzaher Waleed abdelzaher
5. Mohamed eldeep
6. Ahmed abdelmoneim
7. Peter Nagy

1.4.2 Task Description

Our task is to restore or enhance Degraded image that happened in many forms due to a lot of reasons like miss focus, atmospheric turbulence etc.

The primary objective of this group is to make a technique that helps us to solve the problem of the degraded image through denoising techniques or deburring techniques. The project aims to design and implement an image restoration with a desktop application interface.

1.4.3 Task Definition

1. The literature review of different methods and techniques with its quality measurement.
2. Studying familiar applications and defining the objective of proposed application.
3. Implementation of mentioned filters and techniques.
4. Integrating methods together building the whole system with a well desktop application.

1.4.4 Documentation Organization

This documentation is organized as follows:

- **Chapter Two** defines the project tasks, provides guidance on how to plan and manage the project effectively.
- **Chapter Three** talks about the literature review including the definition of image restoration, its different forms and how it can be handled.
- **Chapter Four** introduces the block diagram and an initial design for the project.
- **Chapter Five** introduces the design of the proposed system and talks about the platform used in this project.
- **Chapter Six** introduces the evaluation and testing of the system, where the experiments are done, and results will be stated.
- **Chapter Seven** introduces the conclusion of the work established based on the results of the experiments done, and the future work is to be determined and done in the same field later.

Chapter 2
Project Planning

Project Planning:

This chapter introduces Project Planning. Section 2.1 describes the project tasks and its description. Section 2.2 shows the project management. Section 2.3 describes the Software tool that's used to build up the planning which consists of 2.3.1 Project Subsystem, 2.3.2 Gantt chart, and 2.3.3 The Network diagram of the project.

2.1 Project Tasks

The project includes the following tasks:

1. Analysis and Data Collection
2. System Design
3. System Implementation
4. Testing
5. Documentation
6. Demo
7. Final Presentation

The description of the above tasks is written bellow:

1. Data Collection Task

Involves actions and methods performed on data that help describe facts, detect patterns, develop explanations, and test hypotheses. This includes data quality assurance, statistical data analysis, modeling, and interpretation of analysis results**.**

2. **Data collection:** Acquisition involves collecting or adding to the data holdings. There are several methods of acquiring data:

- collecting new data.
- using your own previously collected data.
- reusing someone other data
- purchasing data

3. **System Design Task**

System design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could be seen as the application of systems theory to product development.

4. **System Implementation Task**

System implementation is the process of developing, deploying, and integrating a new system into an existing environment. It involves turning a design or plan into a functional system by building, configuring and installing software component to make the system function as intended, is reliable and meets the need of the user.

5. **Documentation**

Documentation is a set of documents provided on paper used as a quick reference or a guidebook to help readers understand what is going on.

6. **System Testing Task**

It Is the type of testing to check the behavior of a complete and fully integrated software product based on the software requirements.

7. **Demo**

A lost and found system demo is a freely distributed piece of an upcoming or recently released app. Demos are typically released by the system publisher to help consumers get a feel of the system before deciding whether to by the full version.

8. **Presentation**

Representing the final work and everything that was done project by explaining what was learned and what was done by whom and how long it took to be done.

2.2 Project Management

The project includes the following tasks:

Tasks No	Project Planning Tasks
1	Digital Image Processing and image restoration algorithm understanding and learning python
2	Problem definition
3	Survey of image restoration algorithm
4	Image enhancement techniques and noise reduction
5	Select image restoration filter algorithm
6	Project design and prototype
7	Implementation GUI
8	Testing and comparative study of results
9	Documentation
10	Project presentation

Table 2.1 project sheet tasks

Task	Duration	Preceding	Responsability
1	3 weeks	---------	All students
2	1 week	1	All students
3	2 weeks	1,2	All students
4	5 weeks	2	Abdelzaher,Ahmed,Peter
5	5 weeks	2,3	Mohamed Hemdan,Yassmin,Amr, ,Mohamed Elsayed
6	4 weeks	4,5	All students
7	5 weeks	4,5,6	All students
8	3 weeks	7	All students
9	4 weeks	4,5,6,7,8	All students
10	4 weeks	9	All students

Table 2.2 sheet for handling project

Gantt chart showing project tasks, duration times for these tasks and predecessors:

A network diagram for the project showing estimated times for each activity and the earliest and latest expected completion time for each activity.

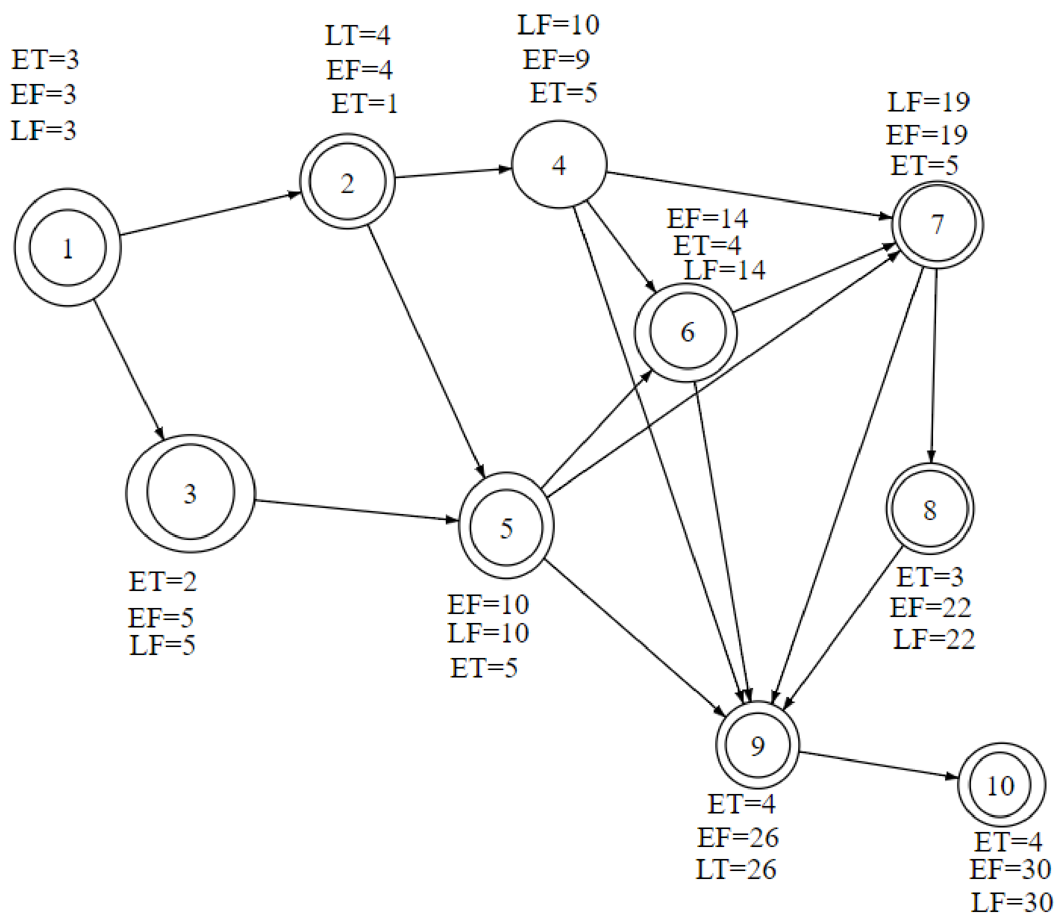


Figure 2.1 Network diagram

Task	Preceding Event	Expected duration (week)	EF	LF	Slack |LF – EF|	Critical path
1	**-**	**3**	**3**	**3**	**0**	✔
2	**1**	**1**	**4**	**4**	**0**	✔
3	**1,2**	**1**	**5**	**5**	**0**	✔
4	**2**	**1**	**9**	**10**	**1**	✘
5	**4**	**1**	**10**	**10**	**0**	✔
6	**3,5**	**1**	**14**	**14**	**0**	✔
7	**3,5**	**1**	**19**	**19**	**0**	✔
8	**3,5**	**1**	**22**	**22**	**0**	✔
9	**3,5**	**1**	**26**	**26**	**0**	✔
10	**5**	**2**	**30**	**30**	**0**	✔

Table 2.3 Project activity slack management

2.3 Software Tool

Task made by (**Microsoft project planning version 2010)**

Microsoft Project can

- Draw the action plan and represent it on both the Network Diagram and the Gantt chart.
- Allocate and organize resources for each activity.
- Follow-up progress of the project.
- Project budget management, and workload analysis.
- The possibility of programming work - as in all programs Microsoft Office package

Group of interconnected and interactive parts that perform an important job or task as a component of a larger system.
A subsystem, while a system, is also wholly contained within a larger system. The system would be subsystems, such as ERD, transactions, and GUI.

2.3.1 Project Subsystems

Group of interconnected and interactive parts that perform an important job or task as a component of a larger system.
A subsystem, while a system, is also wholly contained within a larger system. The system would be subsystems, such as GUI, Figure 2.2 shows that the subsystems.

	Task Mode	Task Name	Duration	Start	Finish	Predecessors	Resource Names
1		Digital Image Processing and Image Restoration Understanding and Learning python	3 wks	Mon 10/3/22	Sun 10/23/22		All Students
2		Problem Definition	1 wk	Mon 10/24/22	Sun 10/30/22	1	All Students
3		Survey of Image Restoration Algorithms	2 wks	Mon 10/31/22	Sun 11/13/22	2	All Students
4		Select of an appropriate image Enhancement Technique and Noise Reduction	5 wks	Sat 11/26/22	Fri 12/30/22	2	Abd El-Zaher Walid,Ahmed Abd elmoneim,Peter Nagy
5		Select of an appropriate Image Restoration Filters Algorithms	5 wks	Sat 11/26/22	Fri 12/30/22	2,3	Mohamed Ahmed Hemdan,Yasmin Walid,Mohamed El-Sayed,Amr Mohamed
6		Project Design and Prototype	3 wks	Mon 2/6/23	Sun 2/26/23	4,5	All Students
7		Implementation the GUI	5 wks	Mon 2/27/23	Sun 4/2/23	4,6,5	All Students
8		Testing and Comparative Study of Results	3 wks	Thu 4/13/23	Wed 5/3/23	7	All Students
9		Documentation	4 wks	Thu 5/4/23	Wed 5/31/23	8,4,6,7,5	All Students
10		Project Presentation	4 wks	Thu 6/1/23	Wed 6/28/23	9	All Students

Figure 2.2 project subsystem

2.3.2 Gantt chart

A Gantt chart, commonly used in project management, is one of the most popular and useful ways of showing activities (tasks or events) displayed against time. On the left of the chart is a list of the activities and along the top is a suitable time scale. Each activity is represented by a bar; the position and length of the bar reflects the start date, duration and end date of the activity, Figure 2.4 shows that the Gantt chart.

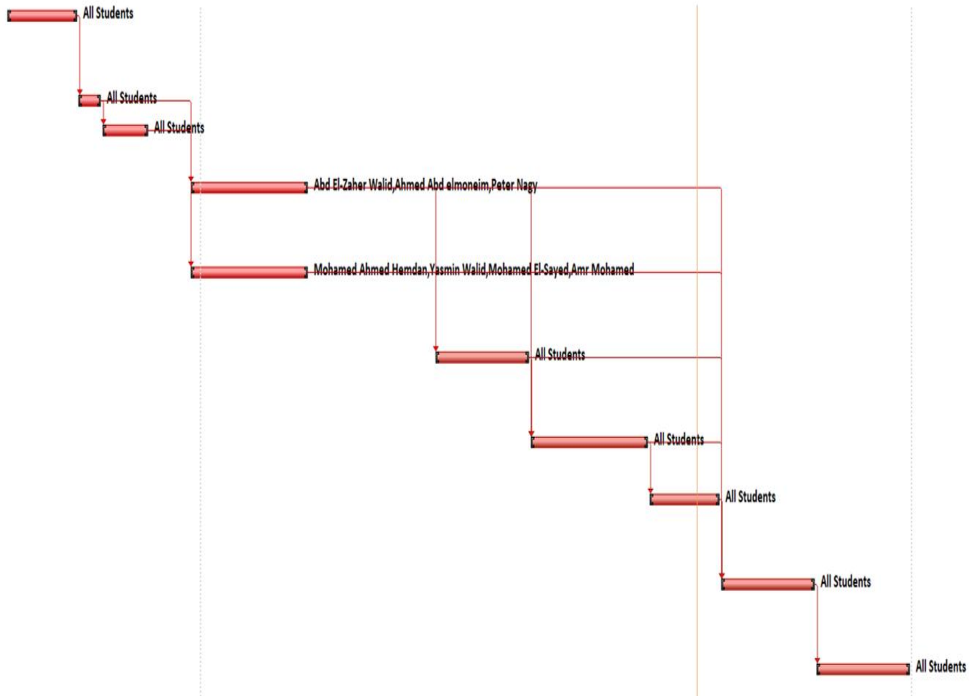


Table 2.4 Gantt chart of project

2.3.3 Network Diagrams

A Network Diagram is a visual representation of a project's schedule. Well known complements to network diagrams. A network diagram in project management is useful for planning and tracking the project from beginning to finish. It represents a project's critical path as well as the scope for the project, Figure 2.3 & 2.4 shows the Network Diagram.

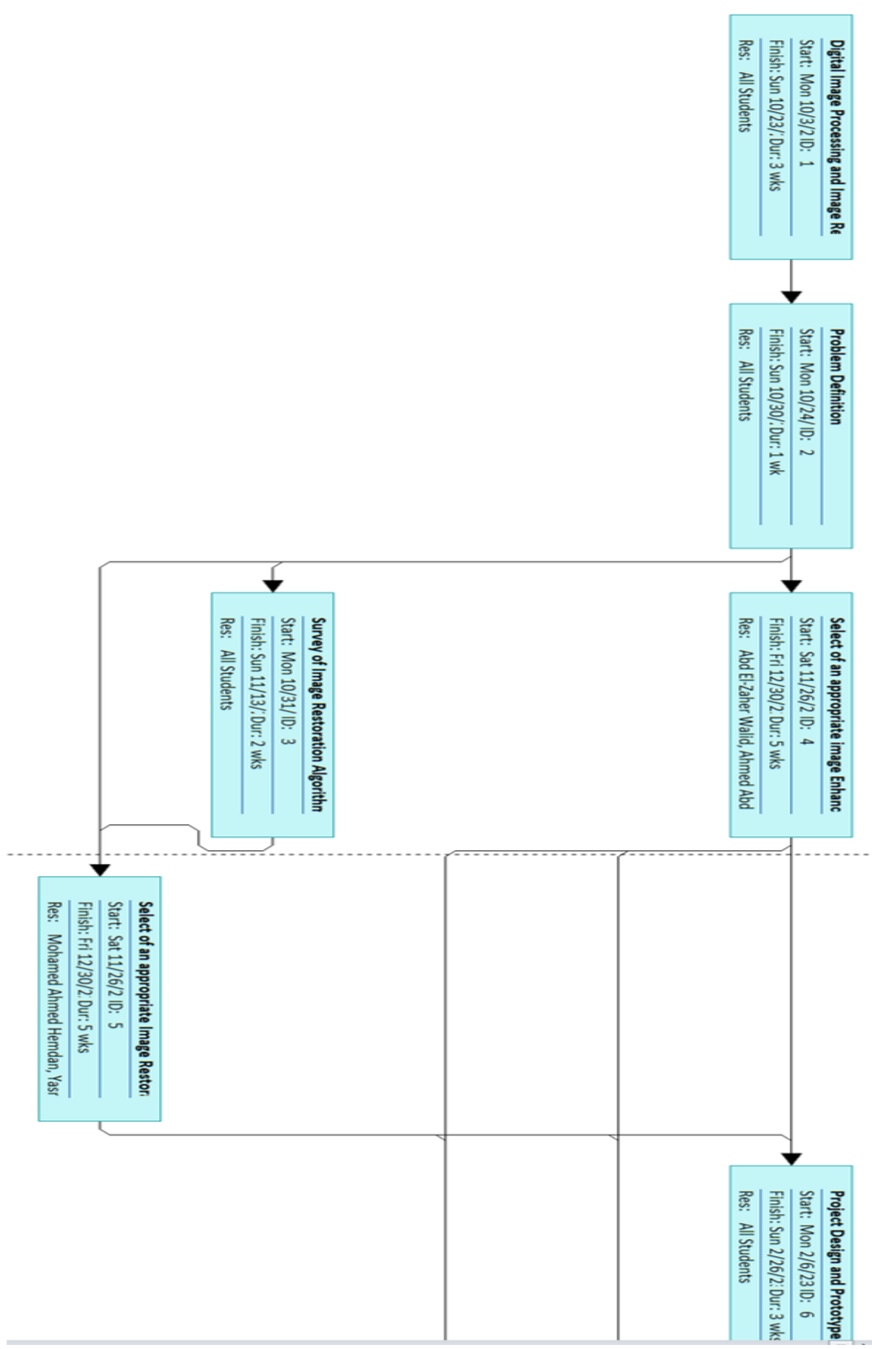


Figure 2.3 network diagram

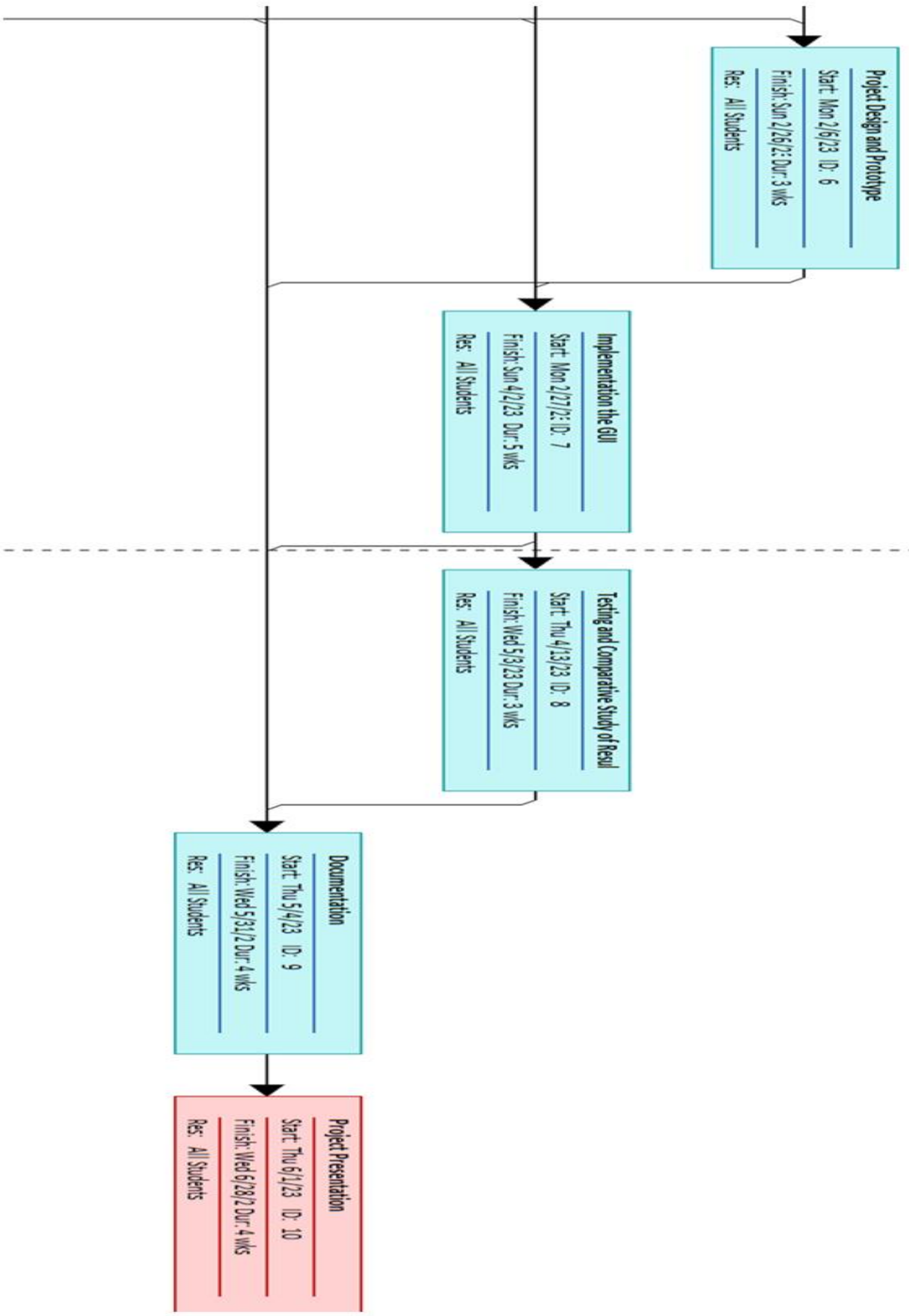


Figure 2.4 network diagram

Chapter 3
Literature Review

3.1 Digital Image Processing

3.1.1 Digital Image Processing Overview

An image may be defined as a two-dimensional function, f (x, y), where x and y are spatial (plane) coordinates called pixel which pixel is the term most widely used to denote the elements of a digital image, and the amplitude off at any pair of coordinates (x, y) is called the intensity or gray level of the image at that point. When x, y, and the amplitude values off are all finite, discrete quantities, we call the image a digital image. The field of digital image processing refers to processing digital images by means of computer.[1]

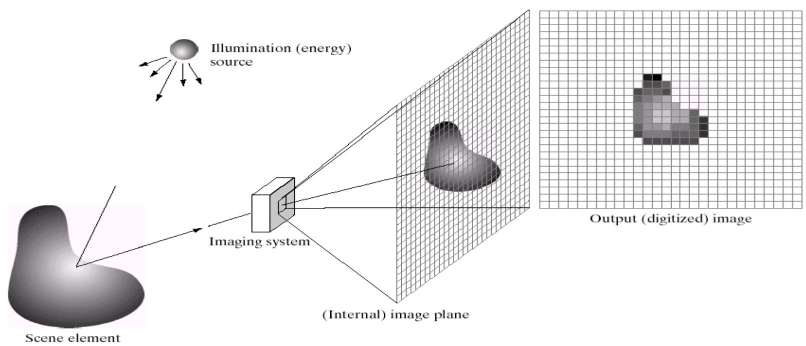


Figure3.1 image plane

Although the digital image processing field is built on a foundation of mathematical and probabilistic formulations, human intuition and analysis play a central role in the choice of one technique versus another, and this choice often is made based on subjective, visual judgments. Hence, developing a basic understanding of human visual perception is a first step in our journey.[1]

3.1.2 Levels of Digital Image Processing

The continuum from image processing to computer vision can be broken up into low-, mid- and high-level processes.

Low Level Process	Mid-Level Process	High Level Process
Input: Image **Output:** Image **Examples:** Noise removal, image sharpening	**Input:** Image **Output:** Attributes **Examples:** Object recognition, segmentation	**Input:** Attributes **Output:** Understanding **Examples:** Scene understanding, autonomous navigation

Figure 2.2 level of digital image

3.1.3 History of Digital Image Processing

- **Early 1920s**: One of the first applications of digital imaging was in the newspaper industry.
  - The Bartlane cable picture transmission service
  - Images were transferred by submarine cable between London and New York
  - Pictures were coded for cable transfer and reconstructed at the receiving end on a telegraph printer.
- **Mid to late 1920s**: Improvements to the Bartlane system resulted in higher quality images.
  - New reproduction processes based on photographic techniques.
  - Increased number of tones in reproduced images
- **1960s**: Improvements in computing technology and the onset of the space race led to a surge of work in digital image processing.
- **1964**: Computers used to improve the quality of images of the moon taken by the Ranger 7 probe
- Such techniques were used in other space missions including the Apollo landings.
- **1970s**: Digital image processing begins to be used in medical applications.
- **1979**: Sir Godfrey N. Hounsfield & Prof. Allan M. Cormack share the Nobel Prize in medicine for the invention of tomography, the technology behind Computerized Axial Tomography (CAT) scans

- **1980s - Today**: The use of digital image processing techniques has exploded, and they are now used for all kinds of tasks in all kinds of areas.
    - Image enhancement/restoration
    - Artistic effects
    - Medical visualization
    - Industrial inspection
    - Law enforcement
    - Human computer interfaces

3.1.4 Fundamental Steps of Digital Image Processing

DIM divided into two board categories:

- Methods whose input and output are images.
- Methods whose input may be images and whose output are attributed.

Methods extracted from those images the organization is summered in fig 3.3 The diagram does not imply that every process is applied to an image. Rather, the intention is to convey an idea of all the methodologies that can be applied to images for different purposes and possibly with different objectives.[1]

- **Image Acquisition**: is the first process shown in Fig 3.3 acquisition could be as simple as being given an image that is already in digital form. Generally, the image acquisition stage involves preprocessing, such as scaling.
- **Image Enhancement:** is among the simplest and most appealing areas of digital image processing. Basically, the idea behind enhancement techniques is to bring out detail that is obscured, or simply to highlight certain features of interest in an image.

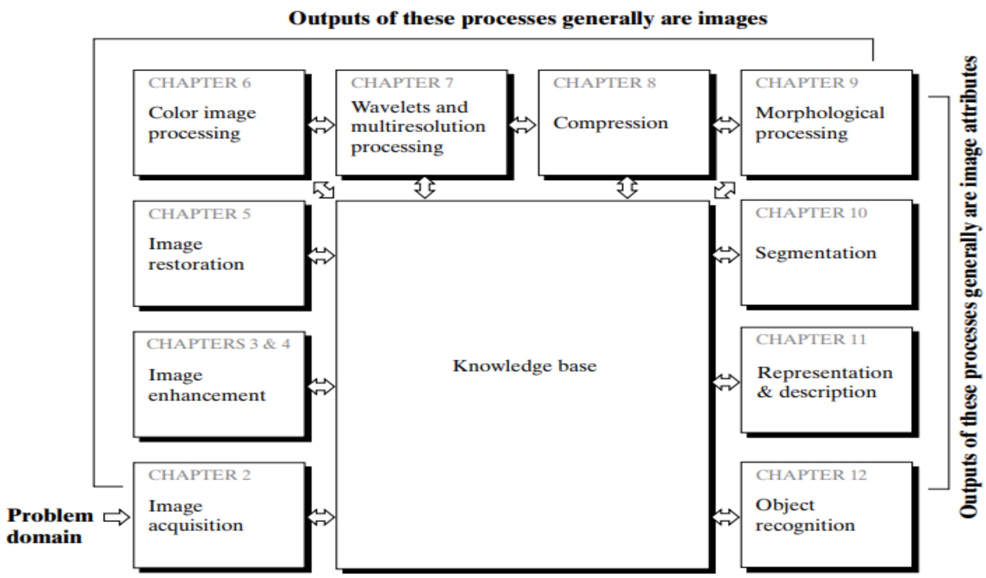


Figure 2.3 steps of digital image processing

- **Image Restoration:** is an area that also deals with improving the appearance of an image. However, unlike enhancement, which is subjective, image restoration is objective, in the sense that restoration techniques tend to be based on mathematical or probabilistic models of image degradation.
- **Color Image Processing:** is an area that has been gaining in importance because of the significant increase in the use of digital images over the Internet.
- **Wavelets:** are the foundation for representing images in various degrees of resolution. This material is used for image data compression and for pyramidal representation, in which images are subdivided successively into smaller regions.
- **Compression**: as the name implies, deals with techniques for reducing the storage required to save an image, or the bandwidth required to transmit it.
- **Morphological Processing:** deals with tools for extracting image components that are useful in the representation and description of shape.
- **Segmentation**: procedures partition an image into its constituent parts or objects. In general, autonomous segmentation is one of the most difficult tasks in digital image processing. A rugged segmentation

procedure brings the process a long way toward successful solution of imaging problems that require objects to be identified individually. the segmentation, the more likely recognition is to succeed.

- **Representation and Description:** almost always follow the output of a segmentation stage, which usually is raw pixel data, constituting either the boundary of a region (i.e., the set of pixels separating one image region from another) or all the points in the region itself.
- **Recognition:** is the process that assigns a label (e.g., “vehicle”) to an object based on its descriptors. We conclude our coverage of digital image processing with the development of methods for recognition of individual objects.

3.2 Image Enhancement

The principal objective of enhancement is to process an image so that the result is more suitable than the original image for a specific application and adjusting digital images so that the results are more suitable for display or further image analysis.

Image enhancement approaches fall into two broad categories: spatial domain methods and frequency domain methods. The term spatial domain refers to the image plane itself, and approaches in this category are based on direct manipulation of pixels in an image. Frequency domain processing techniques are based on modifying the Fourier transform of an image.[1]

3.2.1 Spatial Domain Methods

Processes will be denoted by the expression:

$$g(x, y) = T[f(x, y)]$$

where f (x, y) is the input image, g (x, y) is the processed image, and T is an operator on f, defined over some neighborhood of (x, y). In addition, T can operate on a set of input images, such as performing the pixel-by-pixel sum of K images for noise reduction.

The principal approach in defining a neighborhood about a point (x, y) is to use a square or rectangular sub image area centered at (x, y), as Fig. 3.4. shows.[1]

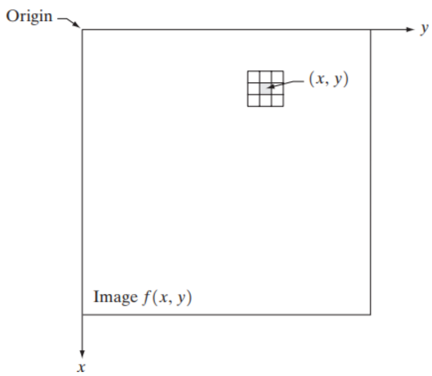


Figure 2.4 convolution process move.

The center of the sub image is moved from pixel to pixel starting, say, at the top left corner.

Spatial domain techniques:

1. Image Negatives:

The negative of an image with gray levels in the range [0, L-1] is obtained by using the negative transformation shown in Fig. 1.3, which is given by the expression. [1]

- Math equation:

$$S = L - 1 - r$$

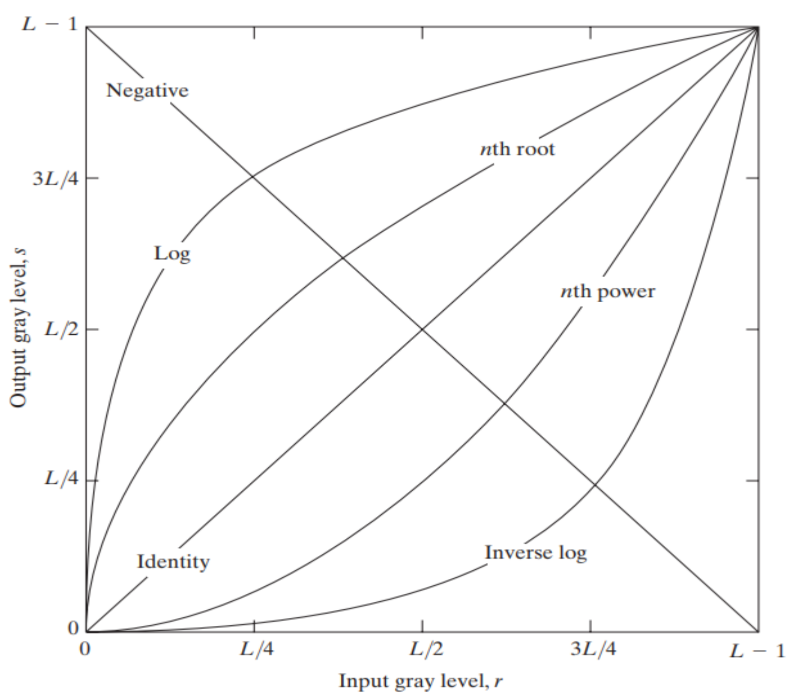


Figure 2.5 enhancement technique

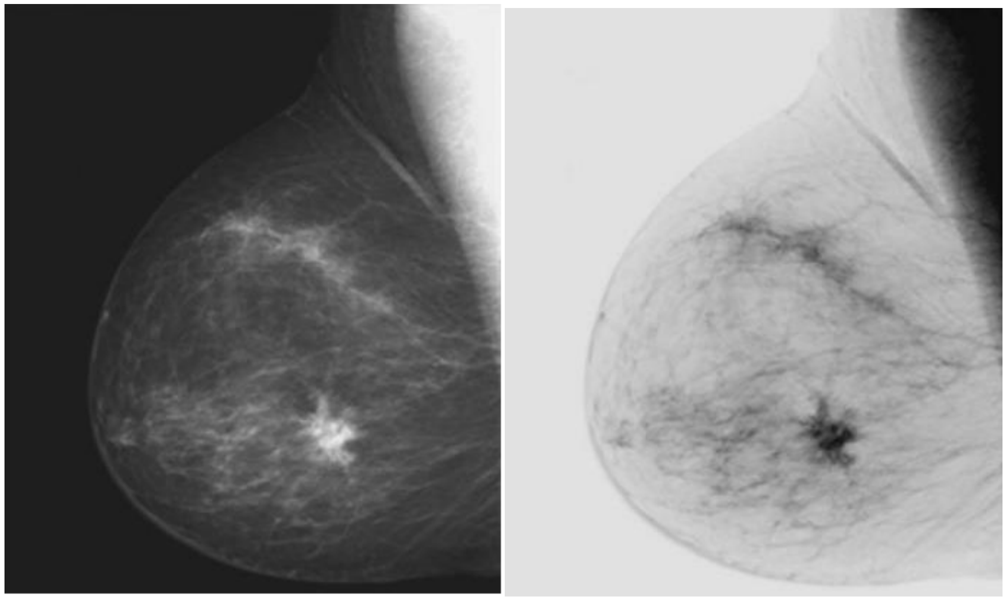


Figure 2.6 example of negative image

Reversing the intensity levels of an image in this manner produces the equivalent of a photographic negative. This type of processing is particularly suited for enhancing white or gray detail embedded in dark regions of an image, especially when the black areas are dominant in size. An example is shown in Fig. 3.6. The original image is a digital mammogram showing a small lesion. Although the visual content is the same in both images, note how much easier it is to analyze the breast tissue in the negative image in this case.

2. Log Transformations

- Math Equation

$$S = c\log 1 + r$$

Where c is a constant, and it is assumed that r = 0.[1]

Example of Log Transformation:

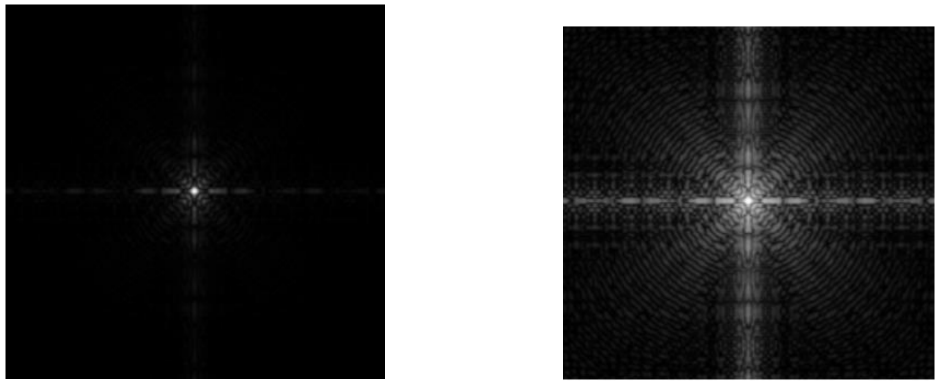


Figure 2.7 example of log

3. Power-Law Transformations:

- Math Equation

$$s = cr^{\gamma}$$

Power law curves with fractional values of γ map a narrow range of dark input values into a wider range of output values, with the opposite being true for higher value of input levels.[1]

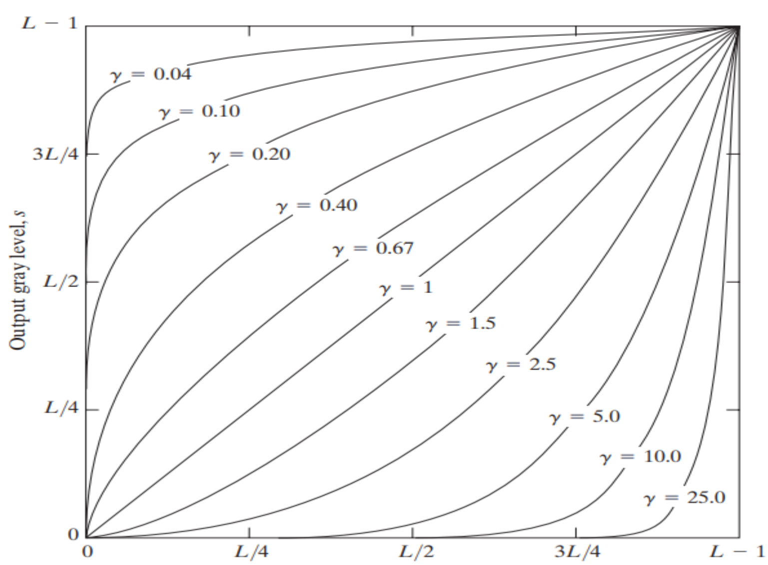


Figure 2.8 power law diagram

Example of Power Law Transformation:

Figure 2.9

- At gamma = 2.2 more than one the light in the image decreases and becomes dark. As shown in fig.3.9.
- At gamma = 0.4 less than one the dark in the image becomes decrease image becomes light. As shown in fig.3.9.

4. <u>Histogram equalization</u>

Histogram equalization is a technique used in image processing to adjust the contrast and brightness of an image by redistributing the pixel intensities.

Histogram equalization is to spread out the intensity's values of an image so that they cover the full available range of values.[1]

- Math Equation

  Let H(i) be the histogram of an input image I, where i

  Is the intensity value ranging from 0 to L-1 (L is the maximum intensity level)

  - First, we Calculate pdf.

$$\mathrm{PDF}(1) = \frac{\mathrm{H(I)}}{\mathrm{(MxN)}}$$

Where M and N are the height and width of the image

- Next, we calculate CDF.

$$CDF(I) = \sum PDF(j), j = 0 \; to \; 1$$

- Then we map the original pixel intensities to new values using the following formula:

$$I'(x,y) = round((L-1)x \; CDF\,(I\,(x,y)))$$

Where I (x, y) is the intensity value of the pixel at position (x, y) in the original image and I' (x, y) is the intensity value of the corresponding pixel in the equalized image.
The round function is used to round the result to the nearest integer value.
After this mapping, the resulting image will have a more evenly distributed histogram, which can improve the contrast and brightness of the image.

- **<u>Smoothing Spatial Filter</u>**

Smoothing filters are used for blurring and for noise reduction. Blurring is used in preprocessing steps, such as removal of small details from an image prior to (large) object extraction and bridging of small gaps in lines or curves. Noise reduction can be accomplished by blurring with a linear filter and by nonlinear filtering.[1]

- **Smoothing Linear Filters:**

The output (response) of a smoothing, linear spatial filter is simply the average of the pixels contained about the filter mask. These filters sometimes are called averaging filter. The idea behind smoothing filters is straightforward. By replacing the value of every pixel in an image by the average of the gray levels in the neighborhood defined by the filter mask, this process results in an image with reduced “sharp” transitions in gray levels. Because random noise typically consists of sharp transitions in gray levels, the most obvious application of smoothing is noise reduction. However, edges (which almost always are desirable features of an image) also are characterized by sharp transitions in gray levels, so averaging filters have the undesirable side effect that they blur edges.[1]

$$\frac{1}{9}\times \begin{array}{|c|c|c|}\hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array} \qquad \frac{1}{16}\times \begin{array}{|c|c|c|}\hline 1 & 2 & 1 \\ \hline 2 & 4 & 2 \\ \hline 1 & 2 & 1 \\ \hline \end{array}$$

*Figure 2.10 (Two 3*3 smoothing (averaging) filter masks. The constant multiple er in front of each mask is equal to the sum of the value (s of its coefficients, as is required to compute an average.)*

Use of the first filter yields the standard average of the pixels under the mask, the second mask shown in Fig. 3.10 is a little more interesting. This mask yields a so-called weighted average, terminology used to indicate that pixels are multiplied by different coefficients, thus giving more importance (weight) to some pixels at the expense of others, the size of the mask establishes the relative size of the objects that will be blended with the background. As an illustration[1]

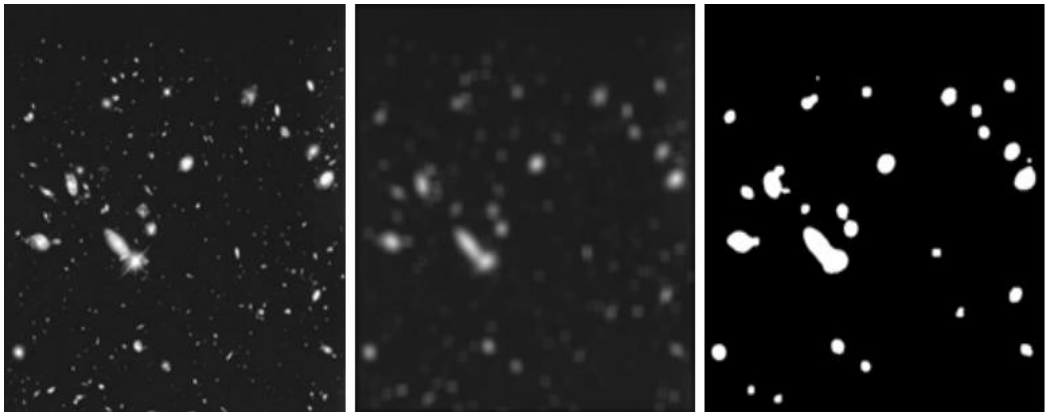


*Figure 2.11 (Two 3*3 smoothing (averaging) filter masks. The constant multiple er in front of each mask is equal to the sum of the value (s of its coefficients, as is required to compute an average.)*

15*15 averaging mask to this image. We see that a number of objects have either blended with the background or their intensity has diminished considerably. It is typical to follow an operation like this with thresholding to eliminate objects based on their intensity. The result of using the thresholding function of Fig. 3.11(b) with a threshold value equal to 25% of the highest intensity in the blurred image is shown in Fig. 3.11(c). Comparing this result with the original image, we see that it is a reasonable representation of what we would consider to be the largest, brightest objects in that image.[1]

3.2.2 Frequency Domain

Frequency domain is a mathematical representation of an image that describes the spatial frequencies present in the image. The frequency domain is used to analyze and manipulate images by transforming them from the spatial domain (where the image is represented as a grid of pixels) to the frequency domain (where the image is represented as a set of sinusoidal waves).

The transformation from spatial to frequency domain is achieved using a mathematical technique called Discrete Fourier Transform.[1]

- Math Equation

$$F(u,v)=\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}f(x,y)e^{-j2\pi(ux/M+vy/N)}$$

Figure 2.12 fft equation

Where:

- f (x, y) is the original image
- f (u, v) is the new image
- j is an imaginary number

The Fourier Transform decomposes an image into its constituent frequencies, which can be visualized as a spectrum. The spectrum shows how much of each frequency is present in the image.[1]

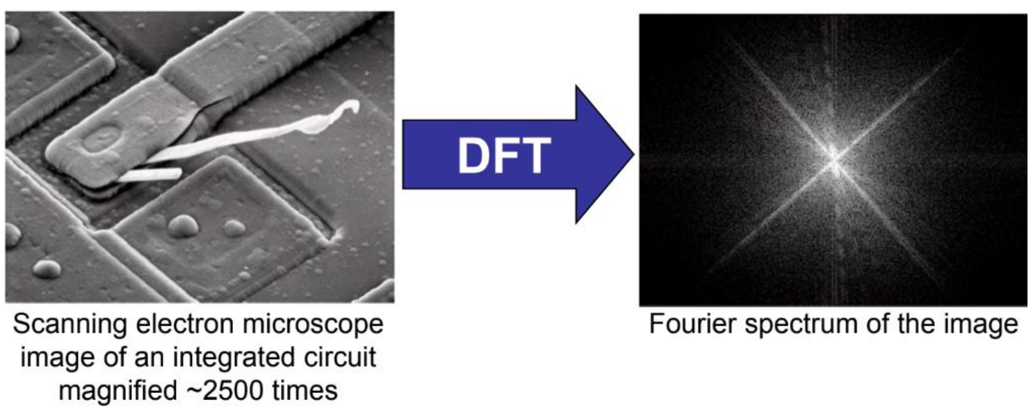


Figure 2.13 example of converting to fft

Then after applying filtering in the image then we return image to its spatial domain through inverse Discrete Fourier transform.

- Math Equation of Inverse DFT

$$f(x,y)=\frac{1}{MN}\sum_{u=0}^{M-1}\sum_{v=0}^{N-1}F(u,v)e^{j2\pi(ux/M+vy/N)}$$

Figure 2.14 equation of Idft

Where:

- f (x, y) is the original image
- f (u, v) is the new image
- j is an imaginary number

In the frequency domain we have low pass and high pass filter which operate on the Fourier transform of an image.

Which low pass filter is a filter that allows low frequency components to pass through while attenuating high frequency components. This means that the filter removes high frequency noise and preserves the low frequency details in an image. Low pass filters are commonly used for smoothing or blurring an image.

On the other hand, a high pass filter is a filter that allows high frequency components to pass through while attenuating low frequency components. This means that the filter removes low frequency details and preserves the high frequency edges and features in an image. High pass filters are commonly used for edge detection or sharpening an image.[1]

3.2.2.1 Low Pass Filter

1- <u>Butterworth Low Pass Filter</u>

The Butterworth low pass filter is a type of filter used in image processing to remove high-frequency noise from an image while preserving the low-frequency components.

The Butterworth filter works by attenuating the high-frequency components of an image while allowing the low-frequency components to pass through. The amount of attenuation depends on the order of the filter and the cutoff frequency. The cutoff frequency is the frequency at which the filter starts to attenuate the signal.[1]

- Math equation

$$H(u,v) = \frac{1}{1+[D(u,v)/D_0]^{2n}}$$

Figure 2.15 Butterworth equation

Where:

- H (u, v) is the transfer function of the filter.
- D (u, v) is the distance between a point (u, v) in the frequency domain. and the center of that domain.
- D0 is the cutoff frequency.
- n is the order of the filter.

The convolution operation involves sliding the kernel over each pixel in the image and computing a weighted sum of its neighboring pixels. The weights are determined by the values in the kernel matrix and are designed to emphasize low-frequency components while suppressing high-frequency components.

The resulting filtered image has reduced noise and smoother edges but may also appear slightly blurred due to the attenuation of high-frequency details. The degree of blurring depends on several factors, including the size of the kernel and the cutoff frequency of the filter.

3.3 Degradation in Image

What is image degradation?

Image degradation refers to the process by which an image loses quality or fidelity due to various factors such as noise and blur.

What are the reasons for the deterioration?

- Compression: When an image is compressed, such as when it is saved as a JPEG or other compressed file format, it can lose some of its quality and appear deteriorated.
- Resolution: If an image is displayed at a low resolution or is stretched too much, it can appear deteriorated.
- Physical damage: If an image has been physically damaged, such as through scratches or fading, it can appear deteriorated. This can happen over time due to exposure to light, heat, or humidity.
- Editing: If an image has been edited or manipulated in some way, it may appear deteriorated if the changes were not made carefully or if the image was saved in a lower quality format.

The degradation in image is noise or blur, Noise means the pixels in the image show different intensity values instead of true pixel values. Noise refers to unwanted variations or distortions in the image that can affect its quality and clarity, Noise is a random variation of image Intensity. blur refers to the loss of sharpness or detail in an image, resulting in a smoother and indistinct appearance. It occurs when the image is not in focus, or when there is motion blur caused by movement during the image capture process.[1]

3.3.1 Type of Noise

1- **Gaussian Noise:**

Gaussian Noise is a statistical noise having a probability density function equal to normal distribution, also known as Gaussian Distribution. Random Gaussian function is added to Image function to generate this noise.[1]

Math Equation

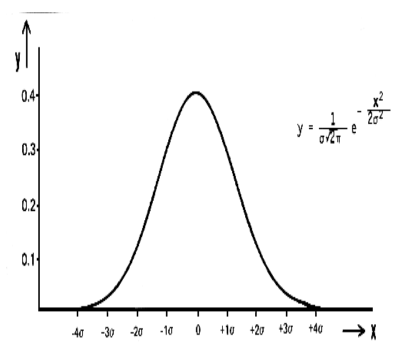
$$p_G(z) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(z-\mu)^2}{2\sigma^2}}$$

Figure 2.16 Gaussian equation

Where:

- Z is the noise value
- μ is the mean value of the noise
- σ is the standard deviation of the noise

Graph



- **Reasons of occurrence Gaussian Noise:**
  Gaussian noise is created because of (an) electronic circuit noise, sensor noise because of high temperature, sensor noise because of poor brightening.
- **Filters deal with Gaussian Noise**:
  - Mean filter.
  - Median filter.

2- **Salt & Pepper noise**

Salt & Pepper noise is called Impulse noise. It can likewise be named as spike or fat–tailed distributed noise. The salt and pepper noise is brought about by sudden and sharp disorder in the image signal. It is appearance as arbitrarily scattered dark or white (or together) pixels above the image. It is digitized as great qualities in the image. Impulse noise contained image has dark pixels in the bright region and splendid pixels in dull areas.[1]

- **Math Equation**

$$p(x) = \begin{cases} P_a, for\ x = a; \\ P_b, for\ x = b; \\ 0, otherwise \end{cases}$$

Figure 2.17 equation of salt and pepper

If b>a, gray level b will perform as a light dot in the image. On the contrary, level a will act like a dark dot. The impulse noise is called uni-polar when Pa or Pb is zero.

- **Reasons of occurrence salt & pepper noise:**

  This is caused generally due to errors in data transmission, the salt and pepper noise is generally caused by malfunctioning of pixel elements in the camera sensors, faulty memory locations, or timing errors in the digitization process.

.

- **Filters deal with salt & pepper noise**:
  - Median filter.

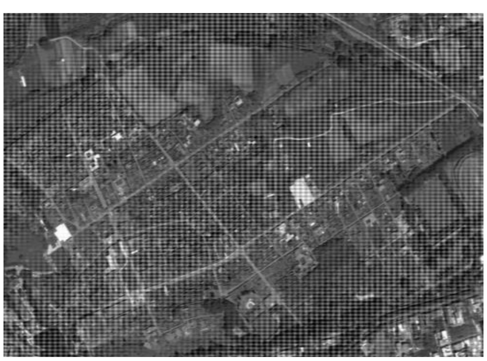
3- <u>Periodic Noise:</u>

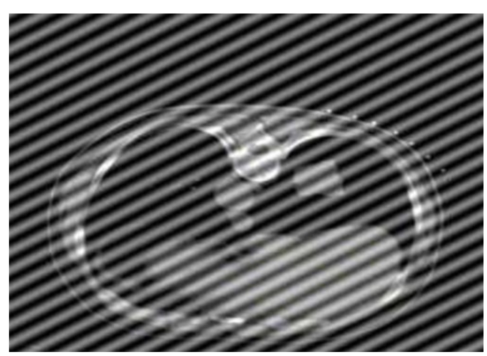
Refers to a type of noise that appears as regular patterns of repeating structures in an image.

Include electrical interference, vibrations during image acquisition, and interference from nearby electronic devices.[8]

Periodic noise can affect various types of image applications, including:

- Medical Imaging
- Astronomy
- Industrial Inspection





3.3.2 Types of Blurs

1- Out of Focus

When the camera is shooting a 3D scene on a 2D launch plane, some areas of the scene are centered whilst others are not. If the camera aperture is circular, a small disk is formed at any point source of the image, recognized as the circle of confusion (COC).[1]

Figure 2.18 out of focus example

2- Motion Blur

The effect of Motion Blur was a filter that made the image show like it moves through adding blur in a certain direction. Can be controlled the motion through the angle or direction such '(0 into 360 ° or from -90 into +90) and / or by the distance or intensity in pixels (0 to 999)', that depending on the software are used.[1]

Figure 2.19 motion blur example

3- Gaussian Blur

It was widely utilized in graphics programs, usually to minimize image blur. In computer vision algorithms, it is also used as a pre-processing step to enhance image structures at different levels.[1]

$$R = \sqrt{g}\,2 + f2$$

Figure 2.20 equation of Gaussian blur

Figure 2.21 example of Gaussian blur

4- Average Blur

That was one of numerous tools which might to be utilize into removed the blur and noise from an image. Can be used it when there was noise on the input image. That kind of camouflage might be distributing in the horizontal and vertical direction and may be distribute like a circular mean during the radius R. It was estimated by the following:

That g was the direction of the horizontal size blurring, and f was the direction of the vertical blurring volume.[1]

Figure 2.22 example of average blur

3.4 Restoration Algorithm

3.4.1 Band Reject Filter

Remove or attenuate a band of frequencies about the origin of the Fourier transform.[8]

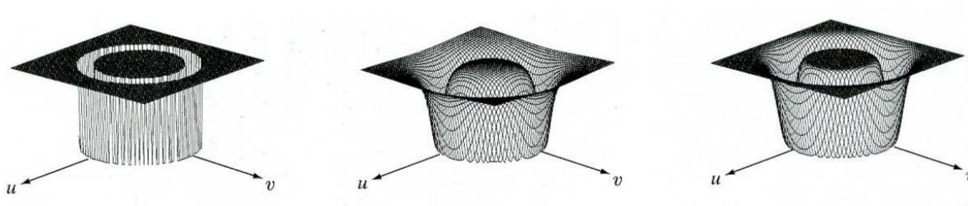


Figure 2.23 From left to right, perspective plots of ideal, Butterworth (of order 1), and Gaussian band reject filters.

- **Math equation**

$$H(u,v) = \frac{1}{1 + \left[\frac{D(u,v)W}{D^2(u,v) - D_0^2}\right]^{2n}}$$

Where:

- D (u, v): distance from the origin of the centered
- W: is width of band
- D0: is radial Center.

<u>Example of band reject filter:</u>

Use of band reject filtering for periodic noise removal.

Figure 2.25 Image corrupted by sinusoidal noise.

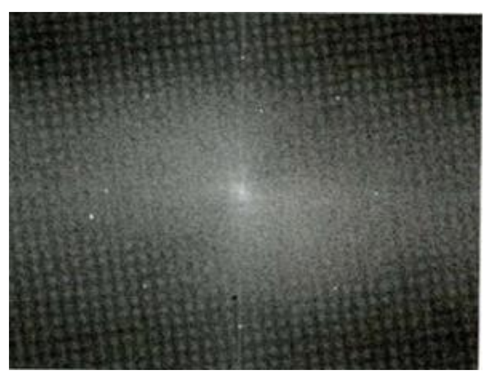


Figure 2.24 Spectrum of (3.18)

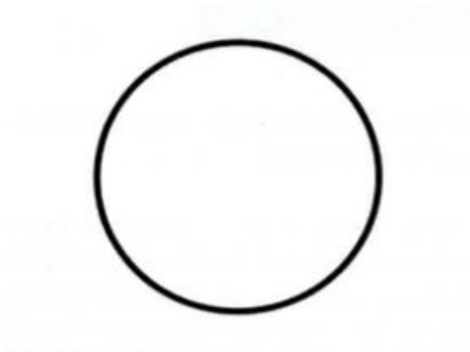


Figure 2.27 Butterworth band reject filter.

Figure 2.26 Result of filtering.

3.4.2 Inverse Filter

Inverse filtering is a technique used in picture processing to eliminate the effects of blurring from an image brought on by different elements like motion, defocus, or atmospheric turbulence. Inverse filtering's main objective is to estimate the original picture by reversing the blurring process's effects.

We must have a full knowledge of the blurring process and its characteristics to conduct inverse filtering. A linear system that converts the initial image into a blurred version can be used to model the blurring process. The point spread function (PSF), a mathematical function that describes how each point in the original image is spread out in the blurred image, can be used to illustrate this linear system.[5]

The equation for Inverse Filter:

$$G(u, v) = F(u, v) / H(u, v)$$

Where:

- $G(u, v)$ is the restored image.
- $F(u, v)$ is the degraded image
- $H(u, v)$ is the blur kernel in the frequency domain.

The inverse filter works by applying an inverse function to the blurred image that reverses the effect of the PSF. This inverse function is obtained by taking the Fourier transform of both the PSF and the blurred image, and then dividing one by the other in frequency domain. The resulting function is then transformed back into spatial domain to obtain an estimate of the original image.

However, inverse filtering can be very sensitive to noise and other artifacts present in real-world images. To overcome this problem, various regularization techniques are used that impose constraints on the estimated image to make it smoother and more realistic.

Overall, inverse filtering is a powerful technique for restoring images that have been degraded by blurring, but it requires careful consideration of various factors such as noise, regularization, and accuracy of PSF estimation.[4]

3.4.3 Wiener Filter

Wiener filtering is a technique that uses statistical information about both the image and noise to reduce noise while preserving edges and other important features. It works by minimizing the mean square error between the original image and a filtered version of it.

The Wiener filter is a mathematical algorithm used in image processing to remove noise from an image. It is a linear filter that works by minimizing the mean square error between the original image and the filtered image.[1]

The mean square error is calculated by the following equation:

$$e^2 = E((f - \hat{f})^2)$$

where E(X) is the expected value of X.

The minimum is calculated by the following equation:

$$\hat{F}(u,v) = (\frac{\overline{H}(u,v)}{|H(u,v)|^2 + K})$$

where H(u,v) is the degradation function.

$$\overline{H}(u,v)$$

It is the complex conjugate of the degradation function. In

$$K = \frac{S_\eta(u,v)}{S_f(u,v)}$$

The numerator is the noise power spectrum, and the denominator is the undegraded image power spectrum. This quotient can be interpreted as the inverse of the signal-to-noise ratio.

$$\hat{F}(u, v)$$

The Wiener filter works by estimating the power spectral density of both the noisy image and underlying signal. It then applies a frequency-dependent weighting function to attenuate frequencies with high noise levels while preserving frequencies with high signal-to-noise ratios. The constant K controls how much noise reduction is applied to each frequency component.[4]

The Wiener filter assumes that the noise in an image is additive and follows a Gaussian distribution. It estimates the power spectrum of the noise and uses it to adjust the frequency response of the filter. The filter amplifies high-frequency components of the image while attenuating low-frequency components, which are more likely to be affected by noise.

The Wiener filter can be applied to both grayscale and color images. It is particularly useful for removing noise from images that have been degraded by blurring or other types of distortion. However, it may not be effective for removing certain types of noise, such as salt-and-pepper noise.

Overall, the Wiener filter is a powerful tool for improving image quality and reducing noise in digital images.

Figure 2.28 example of wiener filter

3.4.4 Regularization Filter

Regularization filter is a type of image filter used in image processing to reduce noise and enhance the quality of an image. It is also known as a smoothing filter or a low-pass filter. The main purpose of regularization filters is to remove high-frequency noise from an image while preserving the edges and details.[1]

The equation for Regularization Filter is:

If the degraded image is modeled in the spatial domain by

$$g = Hf + \eta$$

the constrained least squares filter seeks to find the minimum of

$$\sum_{x}\sum_{u}(\nabla^2 f(x,y)^2)$$

subject to the constraint

$$\parallel g - H\hat{f} \parallel^2 = \parallel \eta \parallel^2$$

The solution to this problem is given by the following equation:

$$\hat{F}(u,v) = (\frac{\overline{H}(u,v)}{|H(u,v)|^2 + \gamma|P(u,v)|^2})G(u,v)$$

where *H (u, v)* is the degradation function and

$$\overline{H}(u,v)$$

It is the complex conjugate of the degradation function. *P (u, v)* is the Laplacian operator frequency domain, and γ is a parameter which is set to meet the previously mentioned constraint. *G (u, v)* is the frequency domain of the observed image, and

$$\hat{F}(u,v)$$

It is the frequency domain estimate of the undegraded image.[4]

The objective of the equation is to reduce noise and enhance the quality of an image by smoothing out the rough edges of the image. The regularization filter accomplishes this by applying a penalty to high-frequency components in an image, which effectively removes small-scale variations or noise from the image. The result is a

smoother image that is more visually pleasing and easier to analyze. Additionally, regularization filters can help prevent overfitting in machine learning and computer vision models by reducing the complexity of the input data while retaining important features.
The choice of λ is critical in determining the quality of the filtered image. A small value of λ will result in a filtered image that closely resembles the noisy input image but may still contain significant noise. A large value of λ will result in a smoother but less faithful representation of the input image. Therefore, finding an optimal value for λ requires balancing these competing objectives.
The regularization filter works by averaging the pixel values about each pixel. The size of the neighborhood is determined by the size of the kernel used in the filter. The kernel is a matrix that specifies how much weight should be given to each pixel in the neighborhood.
The most common types of regularization filters are Gaussian, Median, and Bilateral filters. The Gaussian filter uses a Gaussian function to weigh the pixels in the neighborhood, while the Median filter replaces each pixel with the median value of its neighborhood. The Bilateral filter combines both spatial and intensity information to preserve edges while reducing noise.
Regularization filters are widely used in various applications such as medical imaging, satellite imaging, and computer vision. They are particularly useful for denoising images that have been corrupted by random noise or other artifacts. However, they may also blur out important details if applied excessively or with inappropriate parameters.

In summary, regularization filters are essential tools for enhancing image quality by reducing noise and preserving important features. They work by smoothing out pixel values in a neighborhood using different weighting functions depending on their type.[1]

Figure 2.29 example of regularization filter

3.4.5 Radial Filter

Radial filtering is a modified form of inverse filtering in which high frequency values are truncated using a frequency domain low pass filter. This is done under the assumption that natural pictures have gradual changes in intensity and that noise contributes to very high frequency changes.[1]

The equation for Radial Filter is:

$$\hat{F}(u,v) = \frac{G(u,v)}{H(u,v)} L(u,v)$$

where $\hat{F}(u,v)$ is the output filtered image.

G (u, v) is the degraded image in the frequency domain.

H (u, v) is the Fourier transforms of the input image and a radial filter function.

L (u, v) is a frequency domain low pass filter.

In this equation, the radial filter function, H (u, v), is used to emphasize or suppress image details at different spatial frequencies, based on their radial frequency. This means that the filter will have different effects on features that are distant from the center of the frequency spectrum.

The filter function L (u, v) is applied to the Fourier transform of the radial filter to further modify its frequency response. This additional filtering can be used to boost or attenuate specific frequencies or frequency ranges.[4]

This equation represents a flexible way to apply frequency-based filtering to an image, potentially allowing for better control over the image's appearance and features.

A radial filter is a tool used in image editing software that allows the user to apply adjustments to a circular or elliptical area of an image. The filter is applied by selecting a center point and dragging it outwards to create a circular or elliptical shape. The adjustments made within the filter affect only the area inside the shape, leaving the rest of the image unchanged.

Radial filters are commonly used for selective adjustments such as adding vignettes, adjusting exposure, contrast, saturation, and sharpness. They can also be used for creative effects such as blurring or adding color gradients.

One of the advantages of using radial filters is that they allow for precise control over where adjustments are applied in an image. This can be especially useful when working with complex images where different areas require different levels of adjustment.

Overall, radial filters are a powerful tool in image editing software that can help photographers and designers achieve their desired results quickly and efficiently.[1]

3.4.5 Non-Local Mean Filter

Non-local mean filter is a popular image denoising technique that has gained significant attention in recent years. It is a type of filter that uses the similarity between different patches in an image to remove noise. The non-local mean filter is based on the idea that similar patches in an image have similar noise characteristics, and hence, averaging them can effectively reduce the noise.[5]

The non-local mean filter works by comparing each pixel in an image with its neighboring pixels and finding similar patches in the image. The similarity between two patches is measured using a distance metric, such as Euclidean distance or cosine similarity. Once similar patches are identified, their pixel values are averaged to obtain a denoised pixel value.

One of the key advantages of the non-local mean filter is its ability to preserve edges and textures in an image while removing noise. This is because the filter considers the spatial structure of an image and ensures that pixels with similar texture and color are treated similarly.

Another advantage of the non-local mean filter is its ability to handle different types of noise, including Gaussian, salt-and-pepper, and speckle noise. This makes it a versatile tool for denoising images in various applications, but it works better on Gaussian noise.

The mathematical equation for the non-local mean filter is as follows:

$$f(x, y) = 1/Z(x, y) \sum\sum w(I, j)\, g(I, j)$$

Where:

- $f(x, y)$ is the filtered image,
- $g(I, j)$ is the noisy input image,
- $w(I, j)$ is a weight function that measures the similarity between patches centered at (I, j) and (x, y)

$Z(x, y)$ is a normalization factor given by:

$$Z(x, y) = \sum\sum w(I, j)$$

The weight function $w(I, j)$ is defined as:

$$w(I, j) = \exp(-||P(I)-P(x, y)||\text{^}2/h\text{^}2)$$

where P(I) and P (x, y) are patches centered at (I, j) and (x,y), respectively, ||.|| denotes the Euclidean distance between two patches, and h is a parameter that controls the amount of smoothing

However, one limitation of the non-local mean filter is its computational complexity. The algorithm requires comparing each patch in an image with every other patch, which can be computationally expensive for large images. To address this issue, several variants of the non-local mean filter have been proposed that use approximations or reduce the search space for similar patches.

In conclusion, non-local mean filter is a powerful technique for denoising images that considers both spatial and texture information. While it has some limitations regarding computational complexity, it remains a popular choice for many applications due to its ability to preserve edges and textures while removing noise.[5]

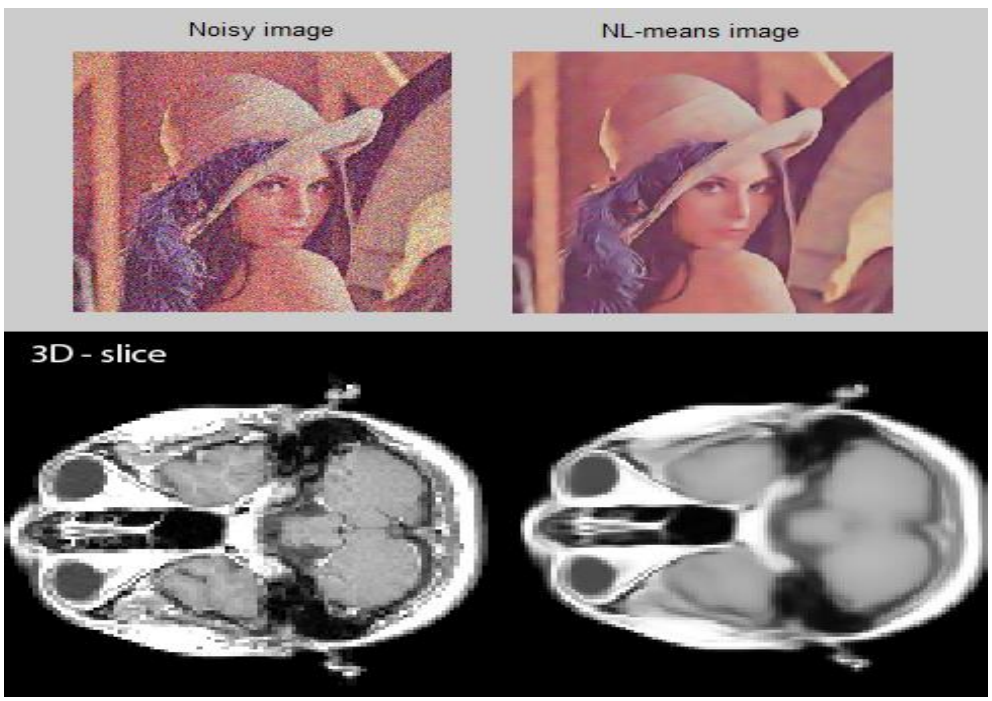


Figure 2.30 example of non-local mean filter

3.4.6 Average Filter

It is considered as a linear filter method of 'smoothing' images by reducing the amount of intensity variation between neighboring pixels. The average filter works by moving through the image pixel by pixel, replacing each value with the average value of neighboring pixels, including itself. which helps to smooth out any irregularities or inconsistencies in the image.[1]

Algorithm works by:

1- Centers the window to the pixel of interest, value of the nine gray levels, 3x3 window mask
2- Multiplies the weights with the pixel values.
3- Then, add the multiplication values.
4- Then, Replace the pixel (I, j) with the new average value (put the result in place of the old pixel value)
5- Repeat the process over all pixels in the image.

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

Figure 2.32 mask

$$\text{Average} = \frac{1}{9} * \begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix}$$

This can be particularly useful for improving the clarity and quality of images that have been degraded by factors such as low light conditions or poor camera settings. Additionally, average filters are relatively simple and computationally efficient, making them a popular choice for real-time applications such as video processing or surveillance systems.[9]

An average filter removes high-frequency noise from an image. This process smooths out the image and reduces the amount of noise present, can also result in a loss of detail and sharpness, but from its disadvantage If a pixel:

- has a very unrepresentative value, it can affect the average value of all the neighboring pixels.
- When edges are important or required in the output, it may be a problem using average filtering since it will blur the edges needed.

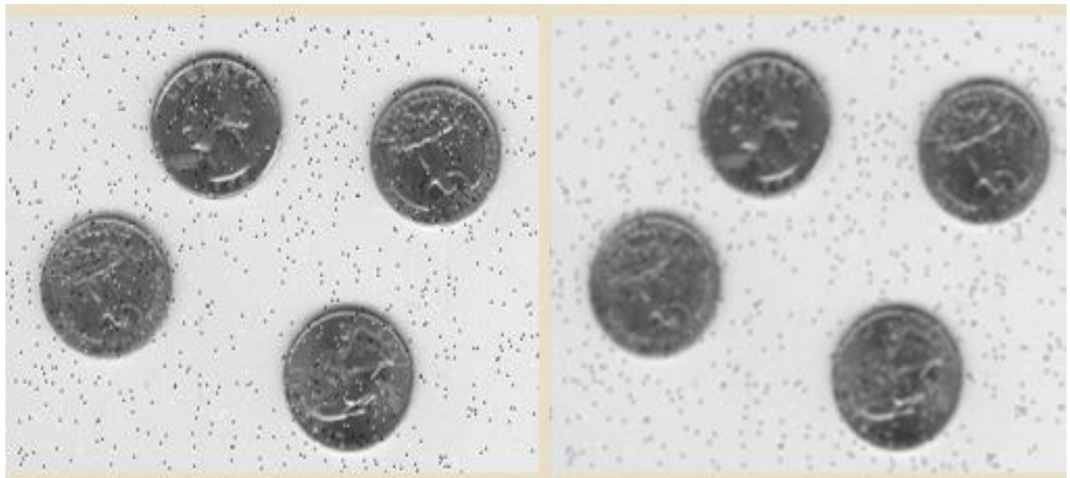


Figure 2.33 average filter example

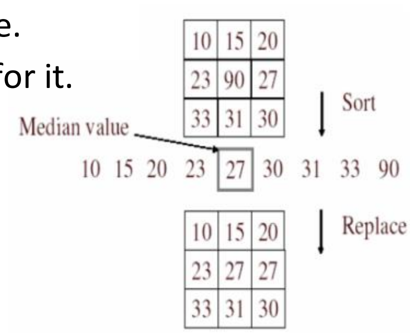
3.4.7 Median Filter

It Is a non-linear filter that is used to remove noise from an image. It works by replacing each pixel in the image with the median value of its neighboring pixels. The median value is the middle value in a sorted list of values.

The median filter removes noise from an image by smoothing out small variations in pixel values while preserving edges and other important features. It is particularly effective at removing salt-and-pepper noise, which appears as isolated white or black pixels in an image.

How algorithm works: by Supposing the window size to be 3x3 and suppose this window is centered at (i, j).

- sort the gray level values of the nine pixels in pix (I, j) that are in the window according to gray level.
- Assign pix (I, j) the middle gray level value appearing in the sort.
- Replace the pix (I, j) with the new median value.
- Repeat the process over all pixels in the image.
- The pixels that are at the boundaries we use for it. replicate or zero padding.



The advantage of using a median filter is that it can effectively remove noise without blurring or distorting important features in the image. However, it may not be as effective at removing other types of noise, such as Gaussian noise.[2]

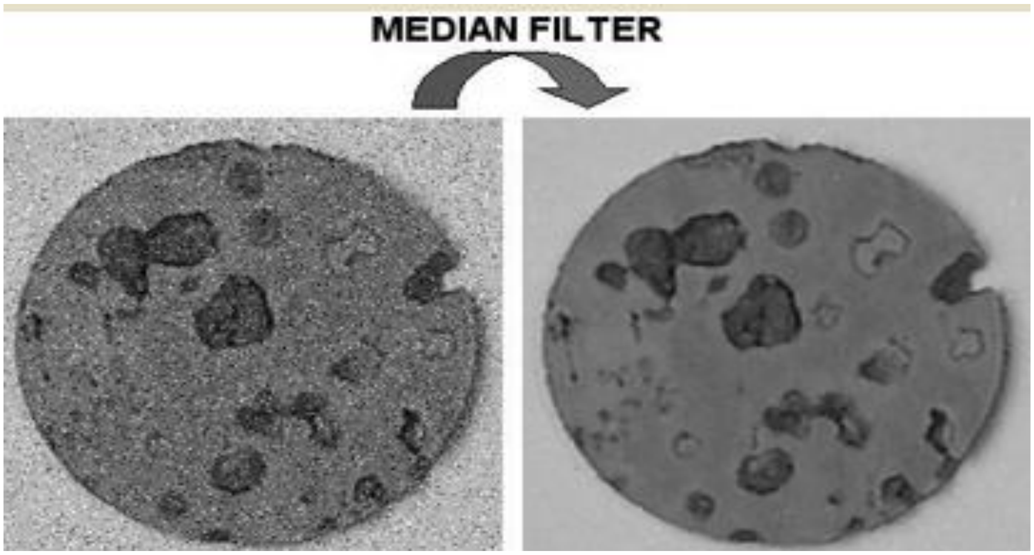


Figure 2.34 median filter example

3.5 Quality Measurement

It refers to the evaluation of the quality of an image based on various parameters such as sharpness, contrast, noise, resolution, color accuracy, and overall visual appearance. These measurements are used to assess the effectiveness of image processing algorithms and techniques in enhancing or restoring images.

Quality measurements can be **objective** or **subjective**: depending on whether they are based on mathematical calculations or human perception.

- **Objective measures:** include metrics such as peak signal-to-noise ratio (PSNR), structural similarity index (SSIM), and mean squared error (MSE)
- **subjective measures:** involve human observers rating the quality of an image based on their visual perception.

3.5.1 MSE (Mean Square Error)

It is a metric used to measure the difference between two images. It is commonly used in image processing to evaluate the quality of an image after it has been compressed or processed in some way.

MSE measures the average squared difference between the pixel values of two images. It calculates the difference between each pixel in the original image and its corresponding pixel in the processed image, squares that difference, and then takes the average of all those squared differences.

- Math equation

Mean Squared Error (MSE): $$MSE = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left|x_{ij} - y_{ij}\right|^2$$

Figure 2.35 MSE equation

Where:

- MN is the number of rows and columns of the matrix.
- Xij is the number of pixels in the original image.
- Yij is the number of pixels in the processed image.

The resulting value represents how much information was lost or changed during processing. A lower MSE value indicates that there is less difference between the original and processed images, meaning that less information was lost or changed. A higher MSE value indicates that more information was lost or changed during processing. It determines which one produces the best results with minimal loss of information.[1]

3.5.2 PSNR (Peak Signal to Noise Ratio)

It is a metric used in image processing to measure the quality of an image after it has been compressed or processed. PSNR measures the difference between the original image and the processed image, and it is expressed in decibels (dB).

- Math equation

Peak Signal-to-Noise Ratio (PSNR) → in decibel (dB):

$$PSNR = 10\log_{10}\left(\frac{L^2}{MSE}\right) = 10\log_{10}\left(\frac{L^2}{\frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left|x_{ij} - y_{ij}\right|^2}\right)$$

Figure 2.36 PSNR equation

Where:

- L is the dynamic range of pixel intensity
- B is the number of bits per pixel

$$L = 2^B - 1$$

PSNR measures how much noise or distortion has been introduced into an image during compression or processing. A higher PSNR value indicates that there is less distortion in the processed image, and therefore it is closer to the original image.[7]

3.5.3 SSIM (Structural Similarity Index)

It is a widely used metric in image processing that measures the similarity between two images. It is used to evaluate the quality of an image after it has been compressed or processed in some way.

SSIM measures the structural similarity between two images by comparing their luminance, contrast, and structure. It considers both local and global differences between the images.[7]

- Math equation

$$SSIM(x,y) = [l(x,y) * c(x,y) * s(x,y)]$$

Where:

- x and y are two images being compared
- l (x, y) is the luminance comparison
- c (x, y) is the contrast comparison
- s (x, y) is the structural comparison

The luminance comparison measures how similar the brightness of each pixel in the two images are. The contrast comparison measures how similar the contrast of each pixel in the two images are. The structural comparison measures how similar the patterns of pixels are in both images.

The output of SSIM ranges from 0 to 1, with 1 indicating perfect similarity between two images. A higher SSIM value indicates a higher degree of similarity between two images.[7]

3.6 Related work

3.6.1 ISNR measure on deburred blind images using Weiner filter.[3]

- **Objective:**

- evaluate the performance of the Wiener filter for image deblurring in the case of blind deconvolution.
- provide a quantitative evaluation of the Wiener filter performance in restoring blind-deconvolved images.

- **Restoration Algorithm:**

- Weiner Filter

- **Quality Measurements:**

- Increased Signal-to-Noise Ratio (ISNR)

- **Block Diagram:**

- Image degradation

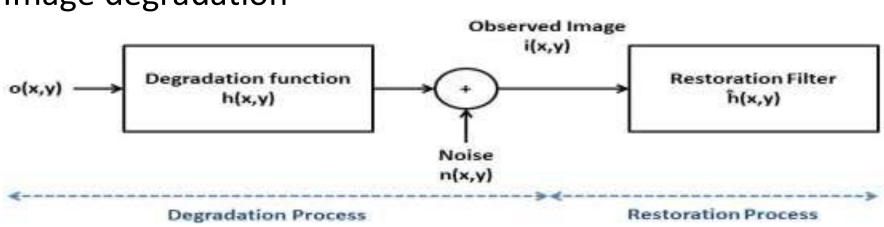


Figure 2.37 block diagram

- Weiner Filter

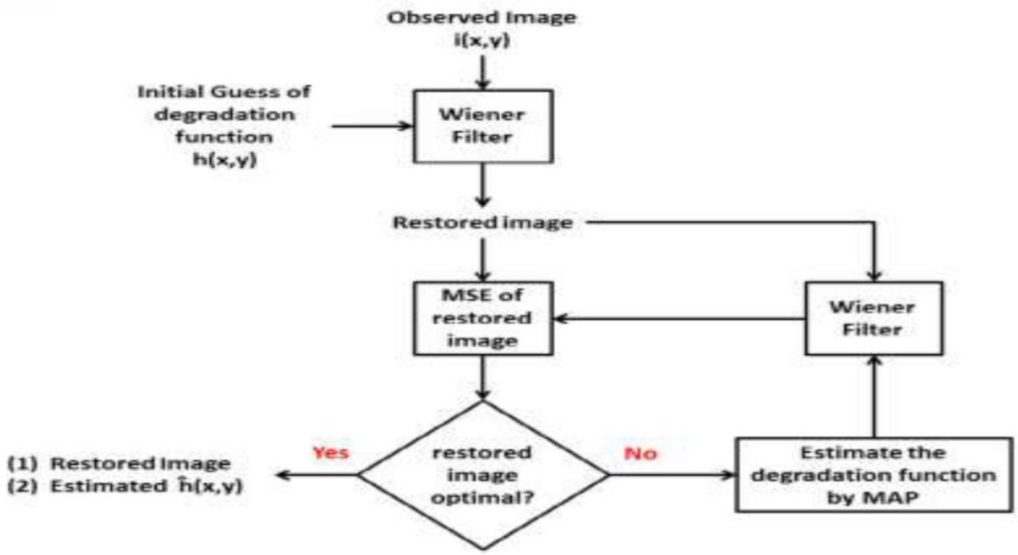
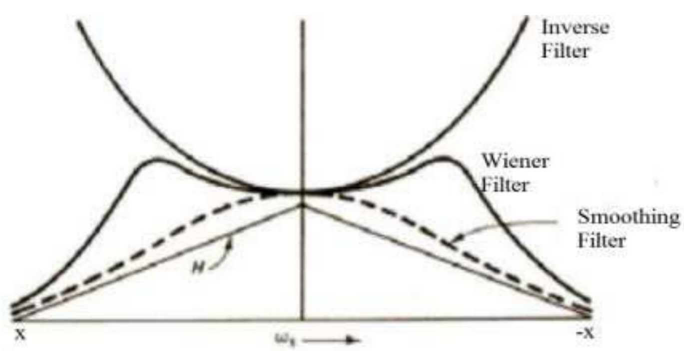


Figure 2.38 wiener BD in paper

- **Comparison between Weiner and inverse filtering**

The inverse filter is a method that seeks to directly invert the blurring process, whereas the Wiener filter is a linear filter that can be used to restore images that have been corrupted by additive noise and linear blur. The research demonstrates that, in terms of the Improved Signal-to-Noise Ratio (ISNR) measure, which is used to assess the



effectiveness of the restored images, the Wiener filter outperforms the inverse filter. Even in the case of blind deconvolution where the point spread function (PSF) is unknown or uncertain, the Wiener filter can efficiently remove noise while maintaining image structures and details. On the other hand, the inverse filter amplifies noise and is sensitive to inaccuracies in the estimated PSF.

- **System Architecture & Implementation**

  The image affected by Gaussian blur with noise is eliminated by 7 different steps in our proposed method, as shown below.

- **Input camera image:** The system's input image is a camera image, which can be structured or unstructured.
- **Simulate a motion blur:** Simulate a blurred image obtained from a camera gesture using a digital image that is blurred without noise. Create a PSF that conforms to the linear motion across 31 pixels at an angle of 11 degrees.
- **Restore the blurred image without noise:** This step removes Gaussian blur without noise; because the blurred image formed in the previous step has no noise, we'll use 0 for NSR.
- **Simulate blur and noise:** In this step, blur is combined with noise; the blur considered in this paper is Gaussian blur.
- **Restore the blurred and noisy image: Level-1:** There is no noise in this step (NSR = 0). The Weiner filter amplifies the noise to the point where only the faintest hint of the shape of the object is visible, therefore displaying the total noise in the image.
- **Restore the blurred and noisy image**: Level-2: The image is recovered from Gaussian blur with noise in this step.
- **Computing ISNR (increase in signal to noise ratio):** This step obtains the image quality ISNR for each of the preceding processes.

- **Advantage & Disadvantage of Paper**

Advantage	Disadvantage
1. **The paper proposes a new approach for using the Wiener filter to improve the quality of deblurred blind images. This approach leads to an improvement in image quality.** 2. The proposed method employs the Weiner filter, a well-known and widely used image deblurring technique. 3. **The paper provides quantitative measures for evaluating the performance of the proposed approach using the Improved Signal-to-Noise Ratio (ISNR) measure.**	1. **The proposed method only works with images that have been blurred using a specific type of blur kernel (i.e., Gaussian blur). It may not be effective for other types of blur kernels.** 2. The paper does not compare the proposed method with other existing methods for measuring image quality, which makes it difficult to assess its superiority over other approaches. 3. **The paper does not provide any insights into how the proposed method can be extended or improved to address some of its limitations.**

Table 2.1 adv. & disadv of the first paper

- **Results:**

1) input image

2) Create a blurred image like one you might get from a moving camera. Make a point-spread function (PSF) that represents a linear motion through 31 pixels (LEN=31) at an 11-degree angle (THETA=11). Convolute the filter with the image using imfilter to duplicate the blur.

Simulate a Motion Blur

3) Noise-free restoration of the original image after Gaussian blur. The most straightforward syntax for calling the Weiner filter is deconvwnr (A, PSF, NSR), where A denotes the blurred image, PSF denotes the point-spread function, and NSR denotes the noise-to-signal-power ratio. Since there is no noise in the blurred image, we will use 0 for NSR.

Restore the Blurred Image without noise.

4) Here, the image is subjected to additive noise added with a Gaussian blur, which is an assumption. Now let's try adding noise with some criteria's noise_mean = 0; noise_var = 0.0001

Simulate Blur and Noise.

5) We will inform (deconvwnr) in our level-1 restoration attempt that there is no noise (NSR = 0). The Wiener restoration filter is an ideal inverse filter when NSR = 0. The Weiner filter amplifies the noise to the point where only the faintest hint of the man's shape is visible, making the ideal inverse filter extremely sensitive to noise in the input image.

Restore the Blurred and Noisy Image:

6) We estimate the noise power to signal power ratio in our level-2 attempt.

Restore the Blurred and Noisy Image: Second Attempt

- **Conclusion**

  The goal of this paper's development is to deblur an image that is primarily affected by Gaussian blur with noise using a special type of filter called a Weiner filter. This filter offers better flexibility than other filters like the median filter and least mean square filter, among others, and it is very helpful in the fields of remote sensing, astronomy, biomedical imaging, etc.

3.6.2 Compare of Median Filter & Wiener Filter to Detect Concealed Weapons. [2]

- **Objective:**

- The objective of this paper is to compare the effectiveness of two different image filtering techniques, namely the median filter and Wiener filter, in detecting concealed weapons in images.
- The study aims to determine which technique is more accurate and efficient in detecting concealed weapons, which can be useful for security and law enforcement purposes.

- **Restoration Algorithm:**

- Weiner Filter
- Median Filter

- **Quality Measurements:**

- Peak signal to noise ratio (PSNR)
- Mean Square Error (MSE)

- **Advantage & Disadvantage**

Advantage	Disadvantage

1. **The paper compares two commonly used filters for detecting concealed weapons in security screening images.** 2. The study evaluates the performance of the filters using real-world data, which increases the validity and reliability of the results. 3. **The paper emphasizes the importance of considering various factors, such as noise level and object size, when choosing a filter for concealed weapon detection.**	**1. The study only compares two filters, limiting the scope of the research and potentially providing an incomplete understanding of all available options.** 2. The paper does not explain how to implement or optimize the filters for various scenarios, which may limit their practical applicability. **3. The paper uses a single dataset to evaluate the performance of the two filters. This could limit the generalizability of the findings since the results might differ with other datasets or images captured under different conditions.**

Table 2.2 adv. &disadv of the second paper

- **Results**

The experimental results ensured that median filter could decrease noise and the result image clearer than that of Wiener Filter. Wiener Filter produces a good output image with a little noise, which is also not clear enough.

(a) Original image (b) Gaussian noise (c) Median filter (d) Wiener Filter

Figure (2) (a) is the original image (b) is the Gaussian noise image (c) Median filter (d) Wiener filter.

- **Conclusion:**

  We use two filters that smooth the image to reduce noise to reduce it in some other way. First, we used median filtering to select the kernel size, list the pixel values covered by the kernel, and determine the median filter. The average of two median values is applied if the kernel passes by each number of pixels. applying median filtering. Second, the Weiner filter was used to reduce the Gaussian noise in the degraded image by estimating statistics for each neighbor per pixel. These two filters were effective at enhancing images of a person with a concealed weapon, but we discovered that the median filter is superior to the Wiener Filter in terms of reducing noise.

3.6.3 Image Restoration and de-blurring using Inverse and Wiener filtering.[4]

- **Objective:**

- It investigates and compares the efficacy of two commonly used image restoration and de-blurring techniques: inverse filtering and Wiener filtering.
- It provides a complete knowledge of these methods, as well as their limitations, and to show their application in real-world scenarios.

- **Restoration Algorithm:**

- Weiner Filter
- Inverse Filter

- **Quality Measurements:**

- Mean Square Error (MSE)

- **Advantage & Disadvantage**

Advantage	Disadvantage
1. **The research includes both simulation and real-world experiments, which provide a thorough assessment of the effectiveness of these techniques.** 2. The study highlights the significance of selecting appropriate parameters for each technique to achieve the best results. 3. **The paper provides light on the theoretical concepts of image restoration and de-blurring using inverse and Wiener filtering.**	**1. The research focuses primarily on the technical aspects of the techniques and does not go into detail about the broader implications and applications of image restoration and de-blurring.** 2. The paper does not have a detailed comparison of the computational complexity of the two techniques, which could be important in certain applications. **3. The study's experiments were limited to a small set of images, and it is unclear how the results would generalize to other types of images and data.**

Table 2.3 adv. and disadv of third paper

- **Results**

- The output images obtained while preprocessing the image for deblurring & restoration are shown below:

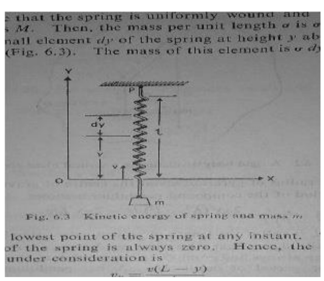


Fig1(a): Original Image

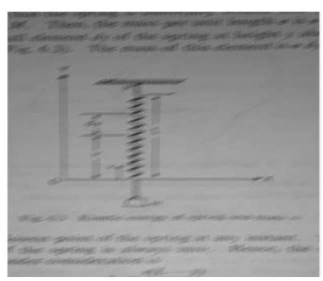


Fig1(b): Blurred Image (len =41, theta=31)

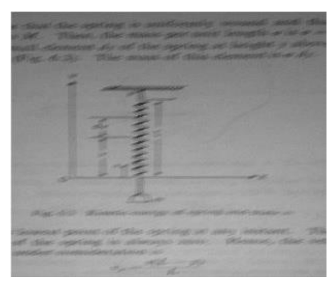


Fig1(c): Blurred Noisy Image (lnoise_var=0.01)

- For Inverse and Wiener Filter, after performing deconvolution the following results were obtained:

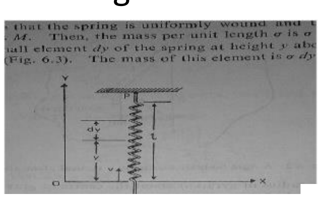


Fig. 2(a): Restored Image using Inverse filter with no additive nois

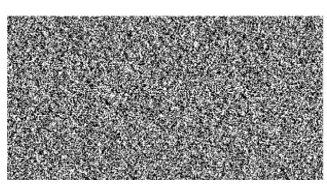


Fig. 2(b): Restored Image using Inverse filter with additive noise (noise_var= 0.01)

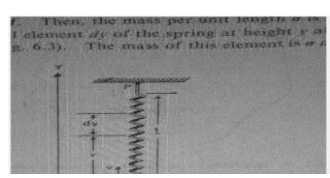


Fig. 2(c): Restored Image using Wiener filter with additive noise (noise_var =0.01)

- Image obtained after subtracting the Restored Image from the Original Image are as follows:

Fig. 3(a): Image obtained after subtraction of Original & Restored image {Fig. 2(a)}

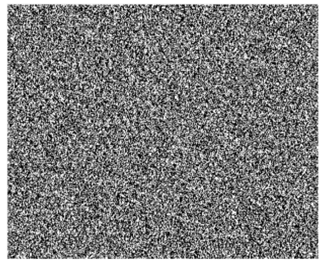


Fig. 3(b): Image obtained after subtraction of Original & Resto image {Fig. 2(b)}

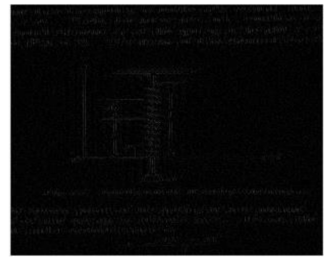


Fig. 3(c): Image obtained after subtraction of Original & Restored image {Fig. 2(c)}

The restoration process for the Inverse filter was only successful when there was no noise present. The Inverse filter is extremely noise sensitive. Furthermore, the subtraction produced a distorted and noisy image, showing that the restoration process fails when noise is present in the image.

The restoration of the Wiener filter was observed to be successful even in the presence of noise. However, when the Restored image was subtracted from the original image, it was discovered that some details on the image had not yet been restored after applying the Wiener filter.

- **Conclusion:**

A comparison of the filters used for deblurring and image restoration, including the Inverse and Wiener filters, was conducted. If we know what the blurring function is and there is no noise, we can use inverse filtering to restore an image quickly and accurately. Because an inverse filter is a high pass filter, it amplifies noise, as demonstrated by our results. As a result, even a small amount of noise can ruin the entire image and the restoration process. However, the Wiener filter is a modified and efficient method of deblurring that operates in the frequency domain. It works well by minimizing the MSE and accounting for noise that the inverse filter ignored.

3.6.4 Image Restoration Report [5]

- **Objective:**

- It was tested against a set of degraded images with varying levels of noise and known / unknown blur kernels.

- **Restoration Algorithm:**

- Inverse filtering.
- Radial filtering.
- Minimum mean square error (Weiner) filtering.
- Constrained least squares filtering.

- **Quality Measurements:**

- The Structural Similarity Index Measure (SSIM)
- The Peak Signal to Noise ratio (PSNR)

- **Advantage & Disadvantage**

Advantage	Disadvantage
1. The proposed method can handle degraded images with varying levels of noise and known/unknown blur kernels, making it useful in a wide range of applications.	1. The paper only evaluates the proposed method on a limited set of test images, so its effectiveness on other types of images or in different contexts is not well-established.

Table 2.4 adv. & disadv of fourth paper

- **Results**
- Images with known blur kernels and varying levels of additive white Gaussian noise.
  - The results of filters with low noise


Input Image


Inverse filter output

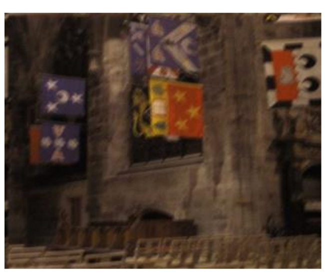

Radial Filter


Weiner filter

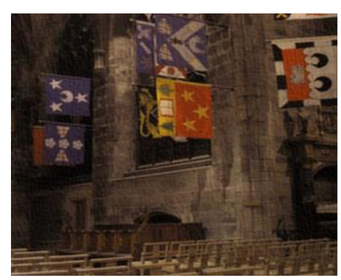

Constrained LS filter

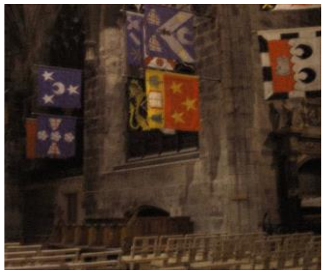
	Input image	Inverse filter	Radial filter (R=150)	Wiener filtering (K=0.02)	Constrained LS (G=0.04)
PSNR	24.86	23.78	32.89	29.48	**33.25**
SSIM	0.978	0.980	0.999	0.997	**0.999**

- The results of filters with moderate noise


Input Image


Inverse filter output


Radial Filter



Weiner filter

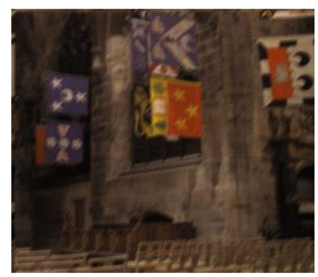
Constrained LS filter

	Input image	Inverse filter	Radial filter (R=100)	Wiener filtering (K=0.05)	Constrained LS (G= 0.2)
PSNR	24.86	23.18	29.15	25.52	**29.99**
SSIM	0.978	0.980	0.999	0.997	**0.999**

- The results of filters with high noise

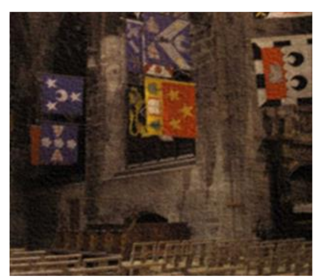
Input Image

Inverse filter output



Radial Filter

Weiner filter

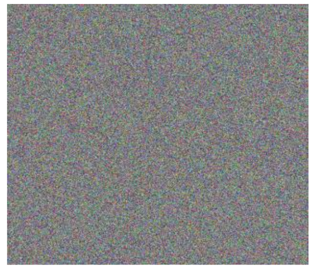


Constrained LS filter

	Input image	Inverse filter	Radial filter (R=75)	Wiener filtering (K=0.2)	Constrained LS (G= 0.9)
PSNR	24.14	23.15	26.32	24.38	**27.51**
SSIM	0.972	0.560	0.996	0.955	**0.997**

- The results obtained for different filtering techniques when the kernel is unknown.

Input Image



Inverse filter output

Radial Filter (R=160)

Weiner filter, K =0.07

Constrained LS filter(G=0.009)

The best visual result is obtained on using Weiner and constrained least square filtering. Due to error in kernel estimation, linear degradation model approximation and assumption of same blur kernel in all RGB channels.

- **Conclusion:**

  On images degraded by blurring and noise, the constrained least square filter produced the best image restoration results. If the image does not have very high frequency components, truncated inverse filtering produces comparable results. Inverse filtering produces acceptable results only when there is no or very little noise present. The performance of Wiener filtering by approximating noise to undegraded image power spectral density as a constant was found to be inferior in some cases, particularly when high noise components were present.

3.6.5 Adapted Non-Local Means Filter using Variable Window Size.[6]

- **Objective:**
- propose a new method for image denoising that can potentially improve the performance of existing techniques by adapting the Non-Local Means filter with a variable window size.
- The paper proposes an adapted non-local means (ANLM) filter that uses a variable window size to improve the quality of denoised images.

- **Restoration Algorithm:**
- Non-Local Means Filter.

- **Quality Measurements:**
- The Peak Signal to Noise ratio (PSNR)

- **Advantage & Disadvantage:**

Advantage	Disadvantage
1. The filter can effectively remove noise while preserving image details, which is important for many image processing applications. 2. It improved filtering method, which helps to calculate each pixel's weight in a better way, using non-Local means filter. **3. The proposed ANLM filter can be applied to a wide range of images and is not limited to specific types of noise or blurring.**	**1. It may not be applicable to all types of images and may require further testing and refinement.** 2. The filter may not work well for images with complex textures or patterns, as it relies on the assumption that similar pixels are clustered together. 3. **The implementation of the ANLM filter may be more complex than other denoising methods due to the variable window size and the need to tune the filter parameters**.

Table 2.5 adv. & disadv of fifth paper

- **Results**

The proposed adaptive similarity window sizes have been tested on an 8-bit standard grayscale test images corrupted by an additive white Gaussian noise.

Noise Image
(AWGN =20)

NL-Means

	Barbara	Hill	Peppers
NL-means with Adaptive similarity window size	30.757	29.666	30.213
Original NL-means	30.660	29.437	30.050

- **Conclusion:**

  The proposed framework yields an improved NL-means filter, varying the (patch) window size according to the patch variance will produce results better than the fixed widow size.

3.6.6 Performance Analysis of Average and Median Filters for Denoising of Digital Images.[9]

- **Objective:**
  The paper aims to provide insights into the strengths and weaknesses of these two filters for removing noise from digital images, which can help inform the selection of an appropriate Denoising technique for different application.

- **Restoration Algorithm:**

- Median Filter
- Average Filter

- **Advantage & Disadvantage:**

Advantage	Disadvantage
1. Provides a detailed comparison of two commonly used de-noising filters, making it simple to understand the advantages and disadvantages of each approach. **2.** Evaluates the performance of these filters in a variety of noise scenarios, providing a comprehensive analysis of their effectiveness. 3. **Provides insights into these filters' underlying mechanisms, allowing readers to better understand how they work and why they may be effective.**	**1. The study only looks at grayscale images and does not investigate how these filters perform on color images, which may behave differently under different types of noise.** 2. The analysis is restricted to a specific set of noise types and may not provide an accurate picture of how the filters perform in other types of noise or noise combinations. **3. The study does not consider each filter's computational complexity, which could be an important factor to consider when evaluating their effectiveness in real-world applications.**

Table 2.6 adv. & disadv of sixth paper

- **Results**

- 10% and 20% Salt and pepper noise is added to both the gray scale and the color images.

  - Results of Performing Average Filter on Grayscale Image

Noisy image

3x3 filtered image.

Noisy image

5x5 filtered image.

Noisy image

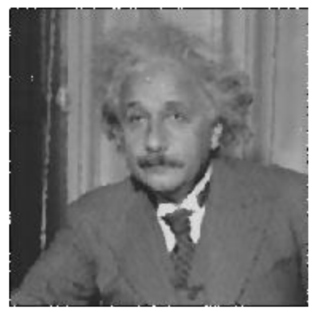
7x7 filtered image.

➢ Results of Performing Median Filter on Grayscale Image

Noisy image

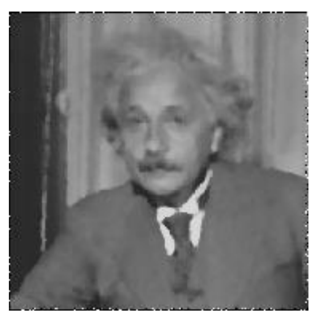
2x2 filtered image.

Noisy image



4x4 filtered image.

Noisy image



6x6 filtered image.

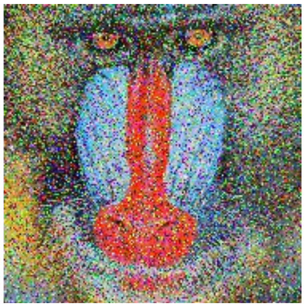
Noisy image

8x8 filtered image.

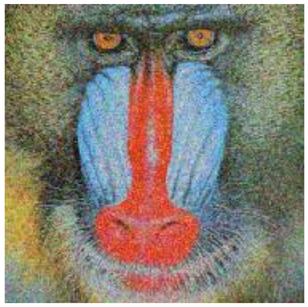
Noisy image

10x10 filtered image.

- Results of Performing Average Filter on Color Image

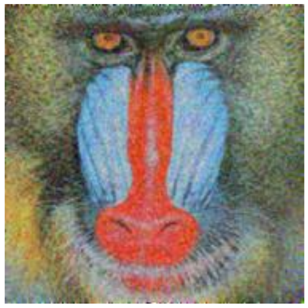


Noisy image



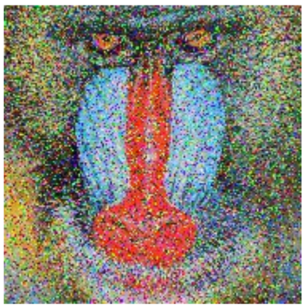
3x3 filtered image.

Noisy image

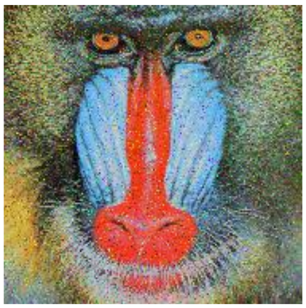


5x5 filtered image.

- Results of Performing Median Filter on Color Image



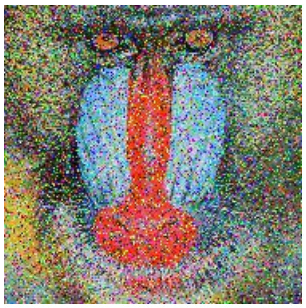
Noisy image



2x2 filtered image.

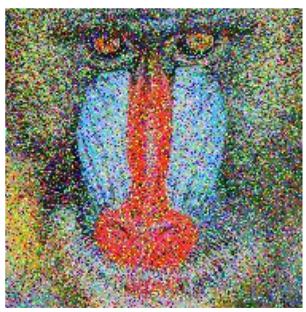
Noisy image

4x4 filtered image.



Noisy image

6x6 filtered image.



Noisy image

10x10 filtered image.

- **Conclusion:**

When the above results for 3x3, 5x5, and 7x7 Averaging filters are compared, it is discovered that the noise has been removed while the image has become blurred. As the size of the window increases, so does the ability to remove noise at the expense of image blurring.

The above results demonstrate the effect of increasing the size of the window in median filtering. It is discovered through comparison that noise is effectively removed as the size of the window increases. Furthermore, the ability to suppress noise increases only at the expense of edge blurring.

In this paper, the results of the Average and Median filters are compared, and it is discovered that the Median filter is more effective at removing salt and pepper noise. It retains image edge sharpness while removing noise.

3.6.7 Enhance Mobile Phone Digital Camera Images Using Band Reject Filter and Discrete Fourier Transform[8]

- **Objective:**
  The purpose of this paper is to suggest a method for improving the quality of digital images captured with mobile phone cameras. The proposed method is used to remove noise from a set of images by employing a Band Reject Filter, which employs a discrete Fourier transform to attenuate and replace a number of degradation bands.

- **Quality Measurements:**

- Mean Absolute Error (MAE)
- Computation Time (CT)
- The Peak Signal to Noise ratio (PSNR)

- **Advantage & Disadvantage:**

Advantage	Disadvantage
1. The proposed method effectively enhances the quality of mobile phone camera images, as demonstrated by the objective and subjective evaluations conducted in the paper. The method can remove noise and artifacts while preserving image details and improving overall image quality. 2. The proposed method is relatively simple to implement, using a combination of a band reject filter and discrete Fourier transform. The authors provide MATLAB code for implementing the method, which can be easily adapted and applied to a wide range of mobile phone camera images. **3. The proposed method is applicable to a wide range of mobile phone camera images, regardless of the specific camera hardware or software used. This makes it a useful tool for improving the quality of mobile photography in general.**	**1. the proposed method is specifically designed for enhancing the quality of digital camera images captured using a mobile phone. It may not be applicable or effective for other types of images or imaging systems.** 2. The proposed method involves several steps, including preprocessing, filtering, Fourier transform, and image enhancement, which can be computationally intensive. This may limit its practical applicability in situations where processing time is critical. **3. While the proposed method is effective at enhancing the quality of mobile phone camera images, it may not be able to fully compensate for low-quality images that are severely degraded by factors such as motion blur or low light conditions. In such cases, additional processing or hardware improvements may be necessary.**

Table 2.7 adv. & disadv of seventh paper

- **Results:**

  It shows how to remove periodic noise from images using BRF, and how to create a better image by increasing the PSNR value. The larger value of PSNR of each image is taken to form improved image. To create the final image, the enhanced image is transformed from frequency domain to spatial domain. The efficiency of this proposed method has been compared with Windowed Gaussian Notch Filter (WGNF) (Aizenberg & Butakoff 2008) and Adaptive threshold-based filter (ATBF) (Varghese 2016).

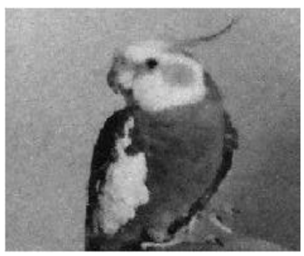
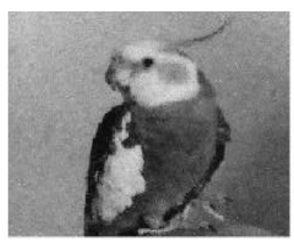


Image1 with 0.3 strength noise

Image2 with 0.5 strength noise

Image3 with 0.8 strength noise

Noise removal of image1



Noise removal of image2

Noise removal of image3

Improved image

Denoising method	Noise strength 0.3			Noise strength 0.5			Noise strength 0.8		
	PSNR	**CT**	**MAE**	**PSNR**	**CT**	**MAE**	**PSNR**	**CT**	**MAE**
WGNF	25.18	12.22	4.18	17.88	12.32	15.35	13.45	12.42	29.34
ATBF	32.23	4.2	2.34	22.8	4.1	9.23	17.43	4.3	17.56
Proposed method	**37.43**	**1.71**	**1.23**	**34.23**	**2.68**	**5.20**	**26.5**	**1.78**	**7.46**

Image1 with 0.3 strength noise

Image2 with 0.5 strength noise

Image3 with 0.8 strength noise

Noise removal of image1

Noise removal of image2

Noise removal of image3

Improved image

Denoising method	Noise strength 0.3			Noise strength 0.5			Noise strength 0.8		
	PSNR	**CT**	**MAE**	**PSNR**	**CT**	**MAE**	**PSNR**	**CT**	**MAE**
WGNF	23.16	11.22	3.17	15.82	12.26	13.33	11.35	11.24	27.24
ATBF	30.21	4.1	2.36	22.8	4.1	7.25	17.43	4.3	16.36
Proposed method	**38.33**	**1.74**	**1.21**	**33.21**	**1.66**	**4.23**	**24.7**	**1.72**	**7.43**

- **Conclusion**

A Band Reject Filter was proposed in this proposal to remove periodic noise from mobile phone images at the frequency domain. The spatial domain techniques are good for removing random noise, whereas the frequency domain techniques are good for removing periodic noise. The blending process with the Band Reject Filter produces a clearer image with higher quality than the original images. The improved image is constructed based on the Peak signal to noise ratio. The larger value for the measure is used to create a merged image, which is then transformed to spatial domain to create an improved image.

Chapter 4
Analysis & Design

4.1 Prototype

Prototype: is the Initial design of the application

- Start Page

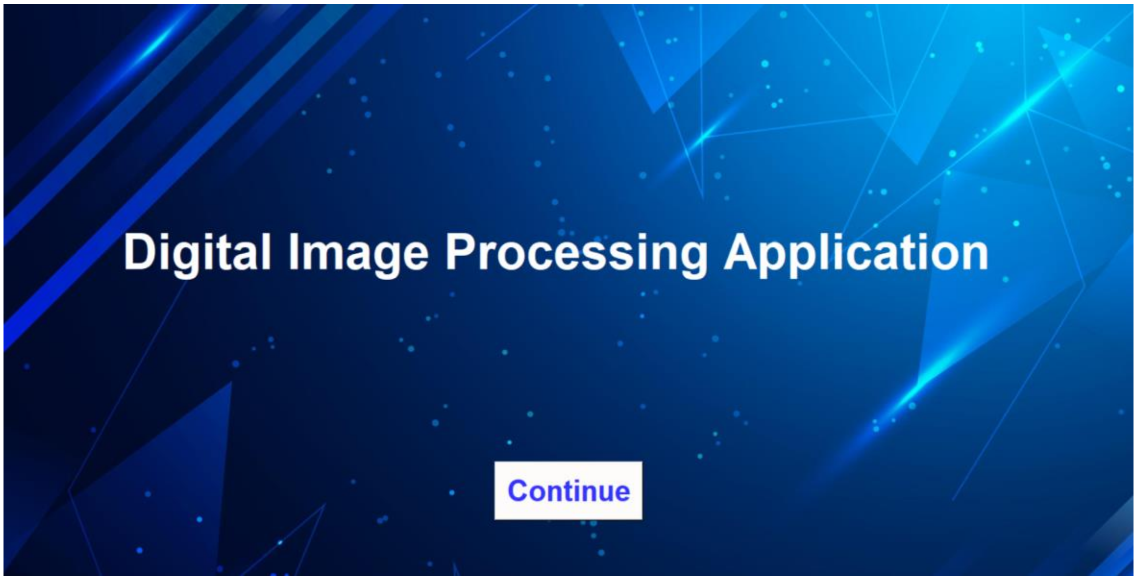


Figure 4.1 first page PT

- **File Menu**

- Home Page

Figure 4.2 home page PT

- **Image Processing**

- Crop Page

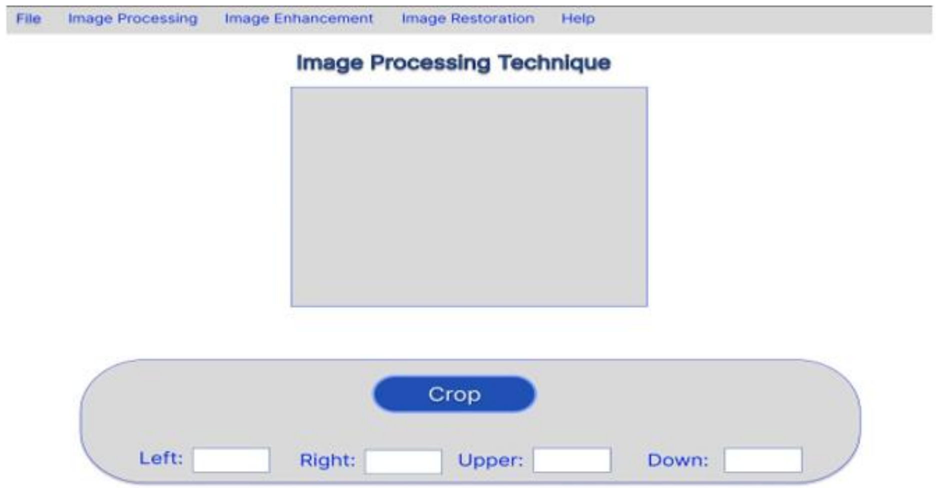


Figure 4.3 crop page PT

- Mirror Page

Figure 4.4 Mirror page PT

- Rotate Page

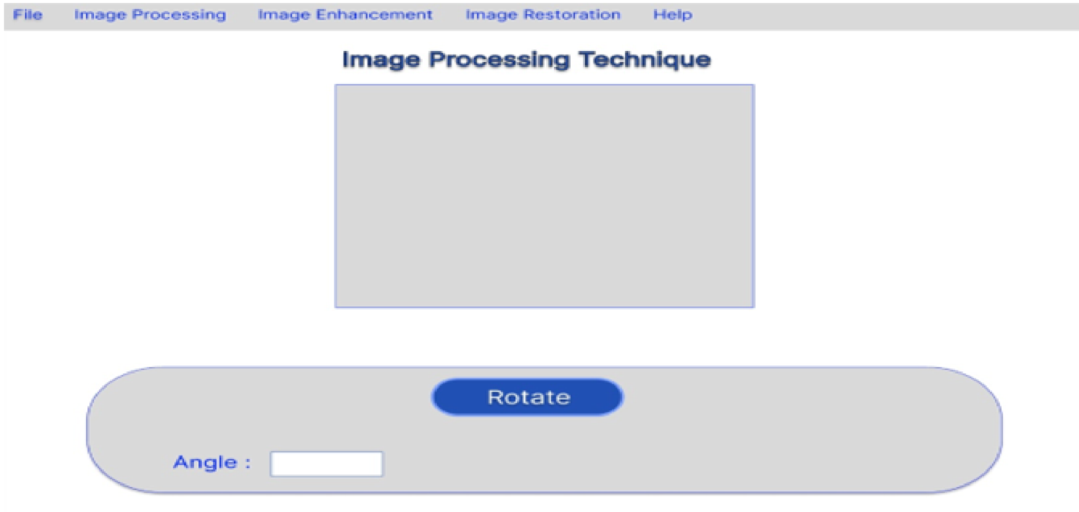


Figure 4.5 Rotate page PT

- Zoom Page

Figure 4.6 Zoom page PT

- Pixelate Page

Figure 4.7 Pixelate page PT

- Gray Scale Page

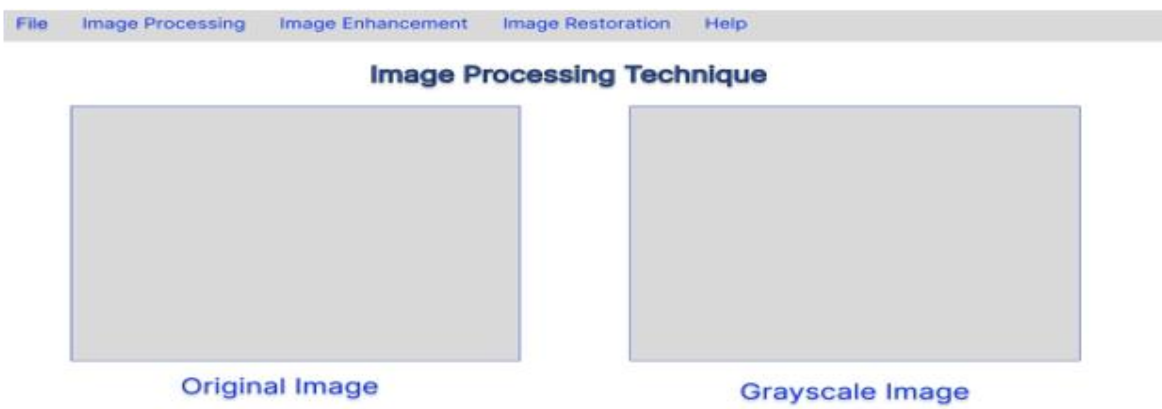


Figure 4.8 Grayscale page PT

- Salt & Pepper Page

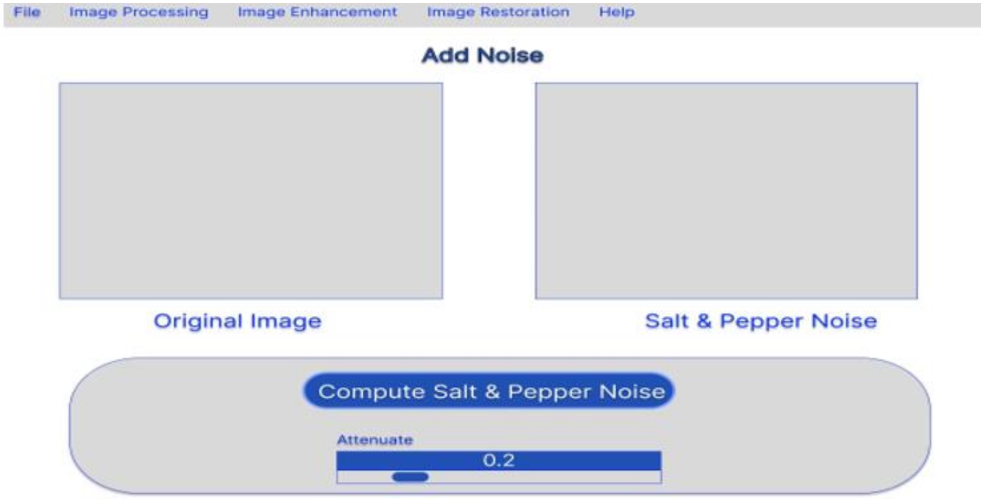


Figure 4.9 Salt and pepper page PT

- Gaussian Noise Page

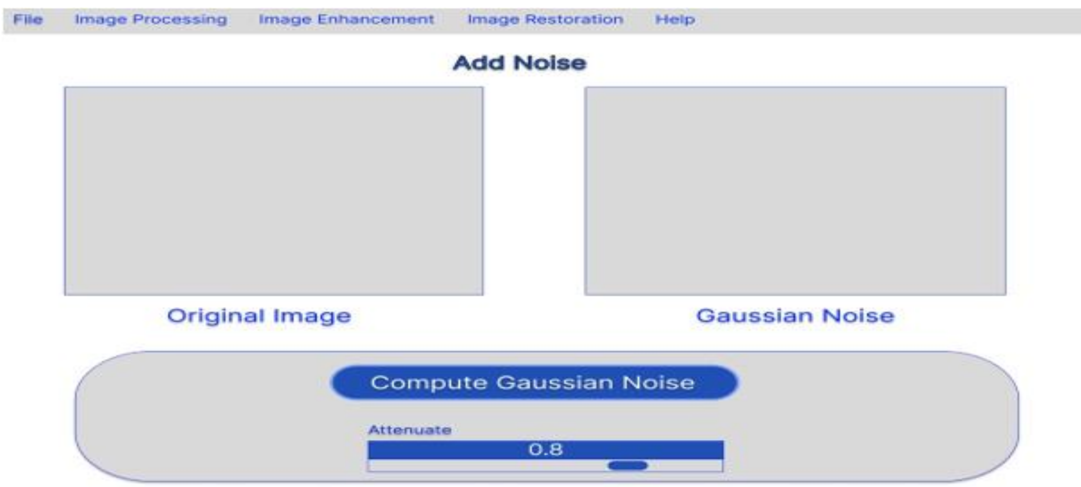


Figure 4.10 Gaussian noise page PT

- **Image Enhancement**
  - Negative Page

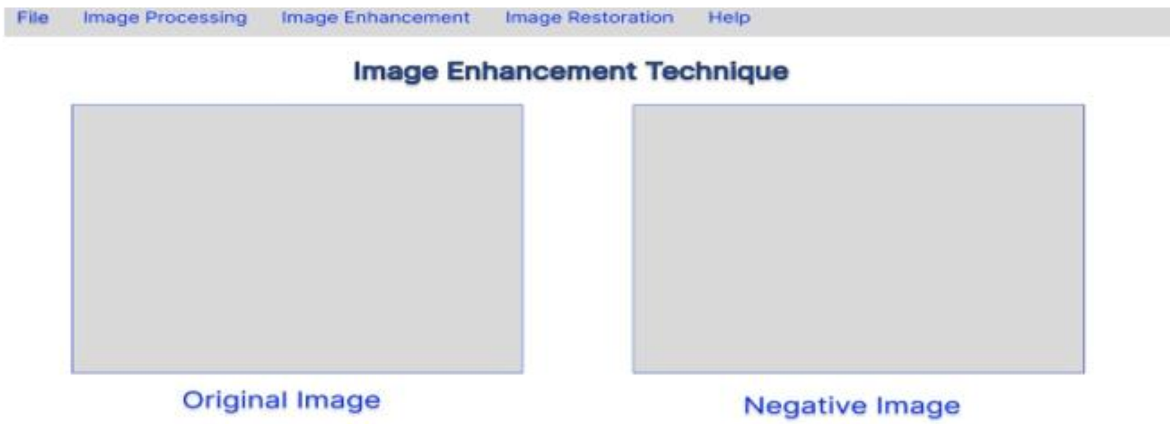


Figure 4.11 Negative page PT

  - Log Page

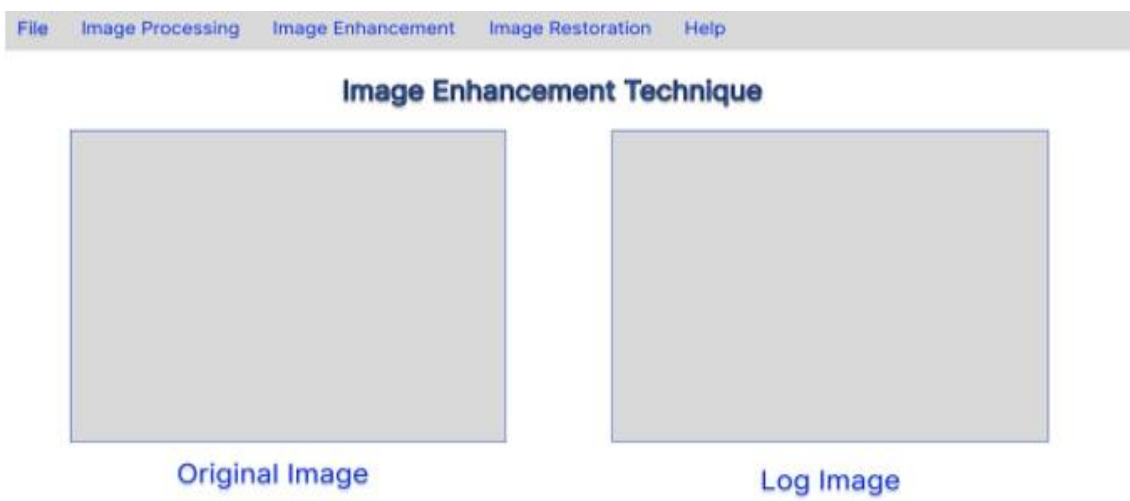


Figure 4.12 log page PT

- Inverse log Page

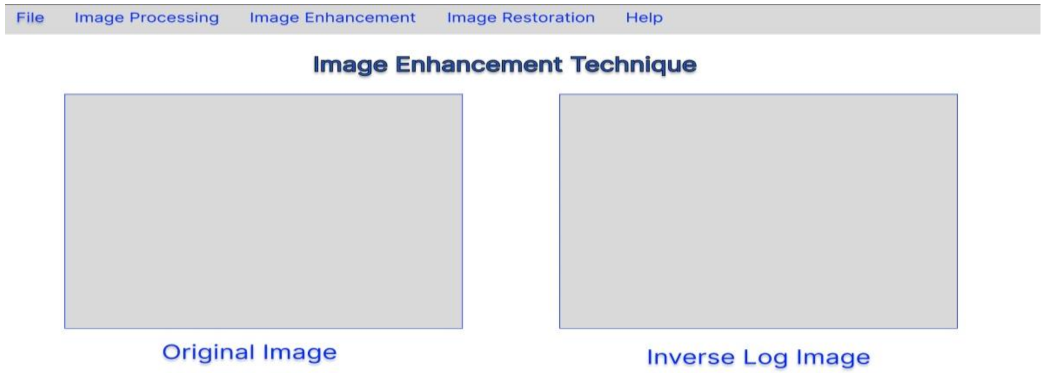


Figure 4.13 Inverse log page PT

- Histogram Equalization Page

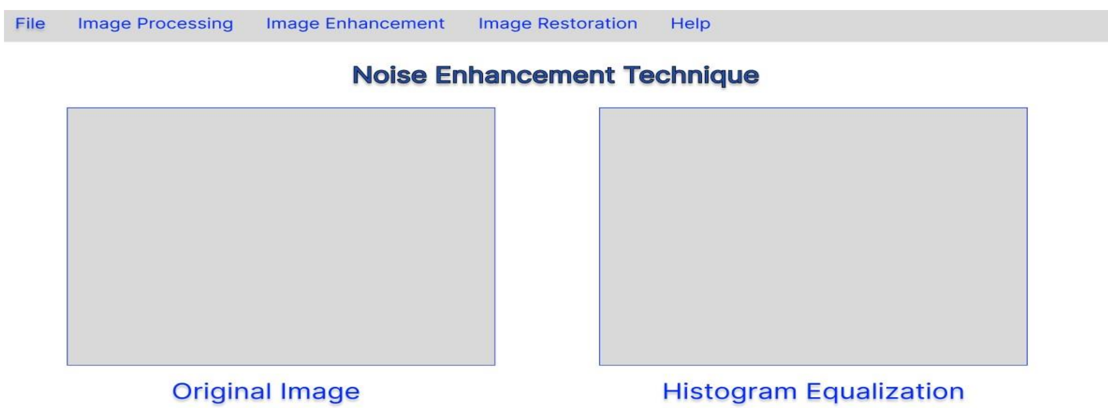


Figure 4.14 histogram page PT

- Contrast Page

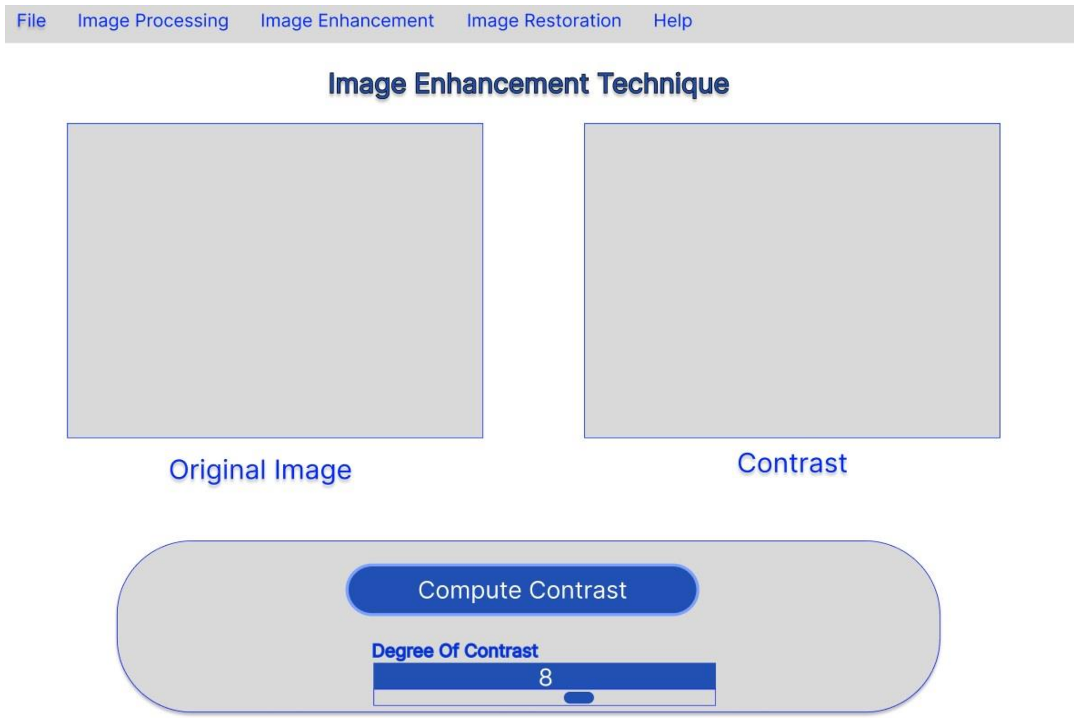


Figure 4.15 contrast page PT

- Sharpness

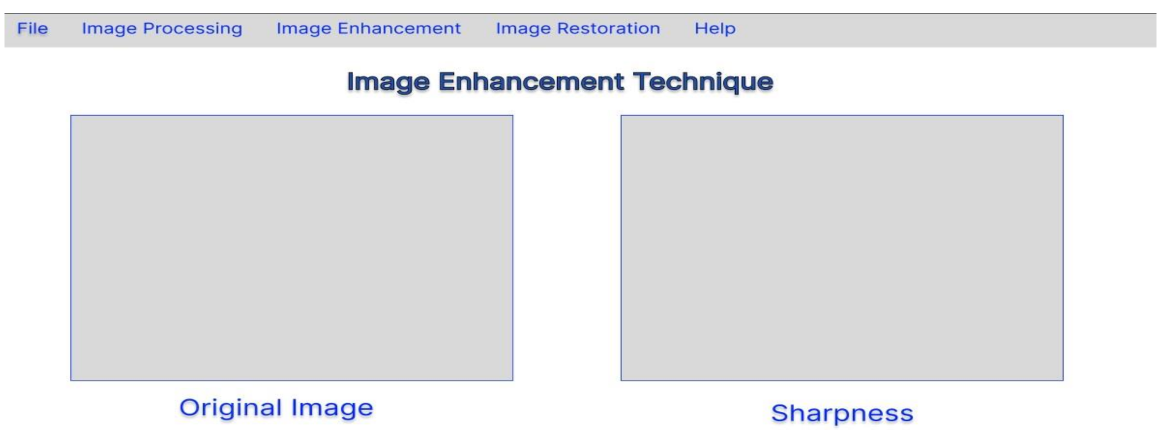


Figure 4.16 sharpness page PT

- Edge Detection Page

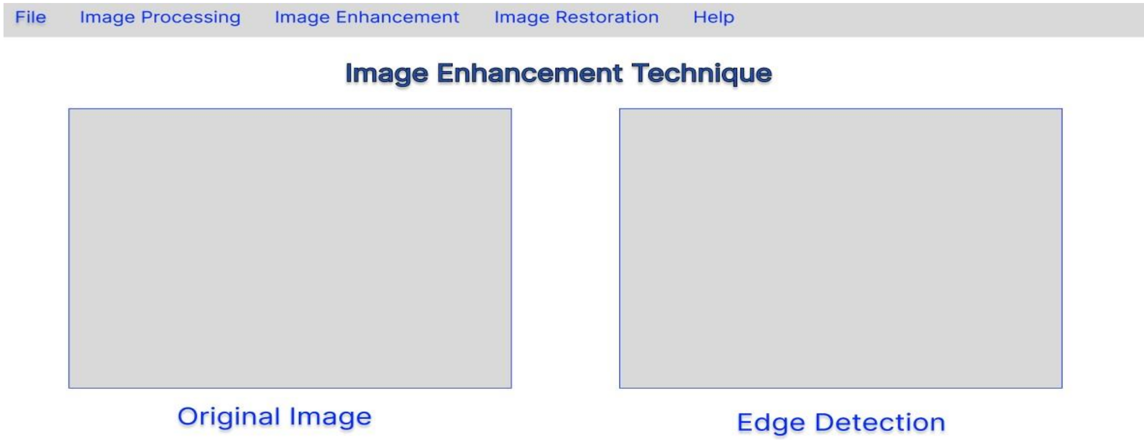


Figure 4.17 Edge Detection page PT

- **Image Restoration Page**
  - Median Filter Page

Figure 4.18 Median filter page PT

- Average Filter

Figure 4.19 Average filter page PT

- Non-Local-Mean Filter

Figure 4.20 Non-Local mean filter page PT

- Inverse Filter

Figure 4.21 Inverse page PT

- Radial Filter

Figure 4.22 Radial filter page PT

- Wiener Filter

Figure 4.23 wiener filter page PT

- Regularization Filter

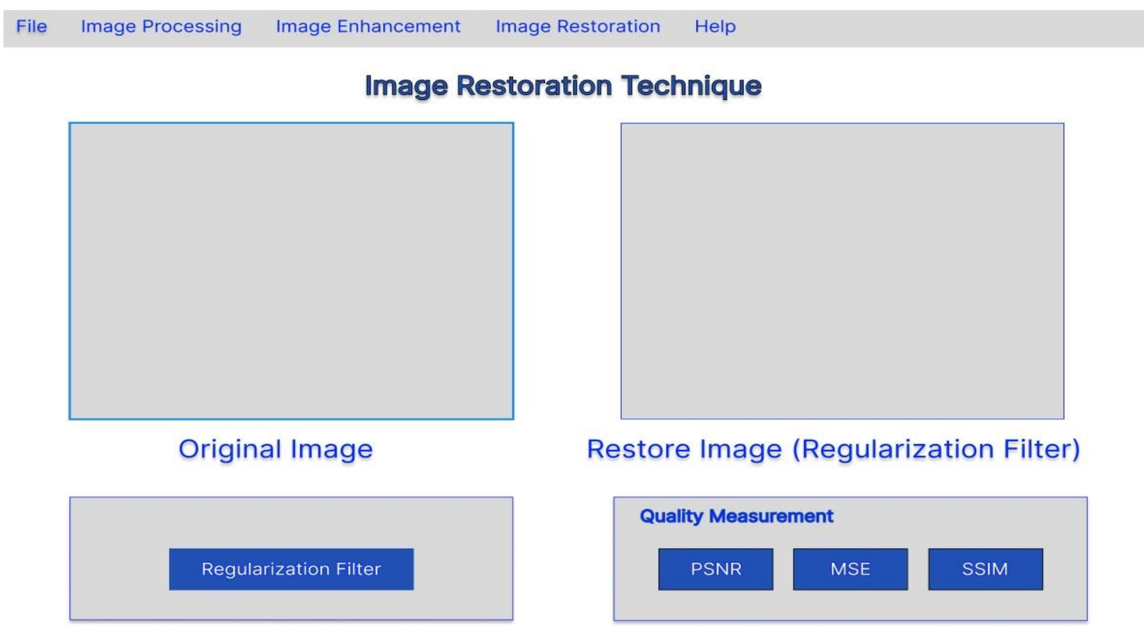


Figure 4.24 Regularization filter page PT

- Band Reject Filter

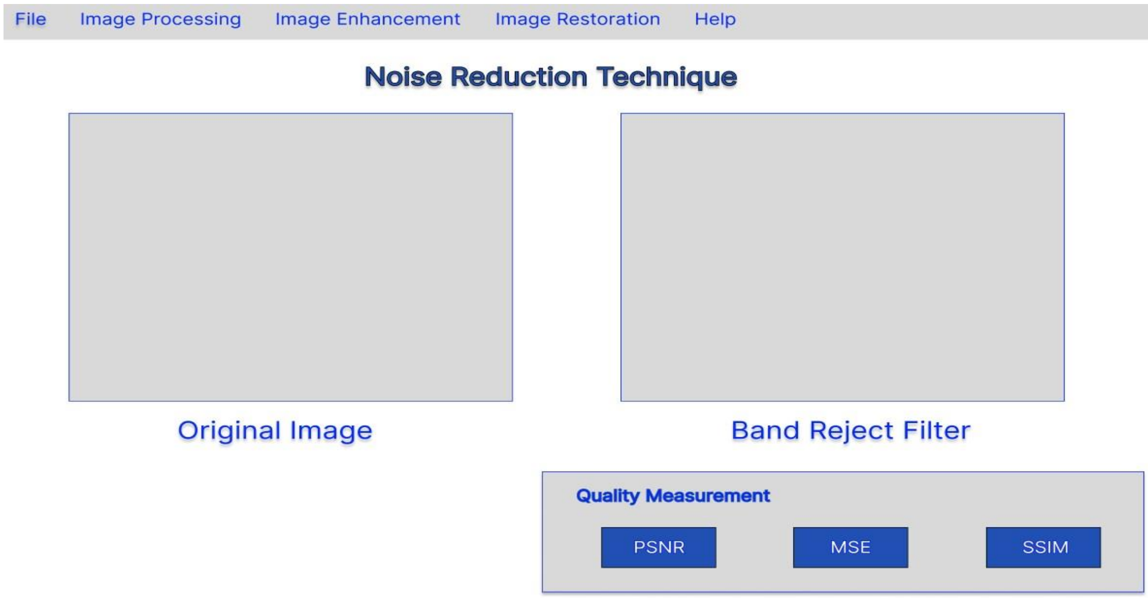


Figure 4.25 Band Reject page PT

- ## Help Page

Figure 4.26 Help page PT

4.2 Block Diagrams

4.2.1 Main Block Diagram of the System

A block diagram can represent a package, a block, or a constraint block, as indicated by the model element *type* in square brackets. The model element name is the name of the package, block, or constraint block, and the diagram name is user defined and is often used to describe the purpose of the system.

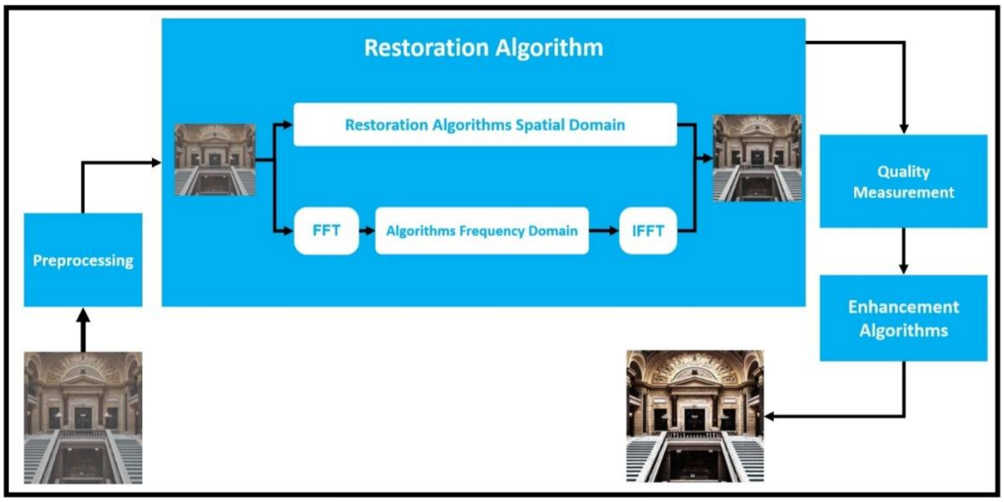


Figure 4.27 main block diagram of the system

The **above figure, Fig 4.27** provides the block diagram of the proposed system. Firstly, input the degraded image into the system, then it goes through some Image preprocessing is done to contrast enhancement, Rotate, Zoom, Crop and image filtering.

Afterwards, apply restoration algorithm It can be one of several types, including deblurring, denoising, the algorithm takes the input image and applies a mathematical transformation to restore the image, Then, Apply Quality Measurement such as (PSNR, MSE, SSIM) that evaluating the quality of an image.

Afterwards, Once the restoration algorithm has been applied, post-processing is done to further enhance the image quality. This can include sharpening, color correction, and other adjustments.

4.3 Flowchart

1- <u>Average Filter</u>

Flow Chart of a Average Filter provides a clear and concise explanation of the steps involved in implementing the algorithm.

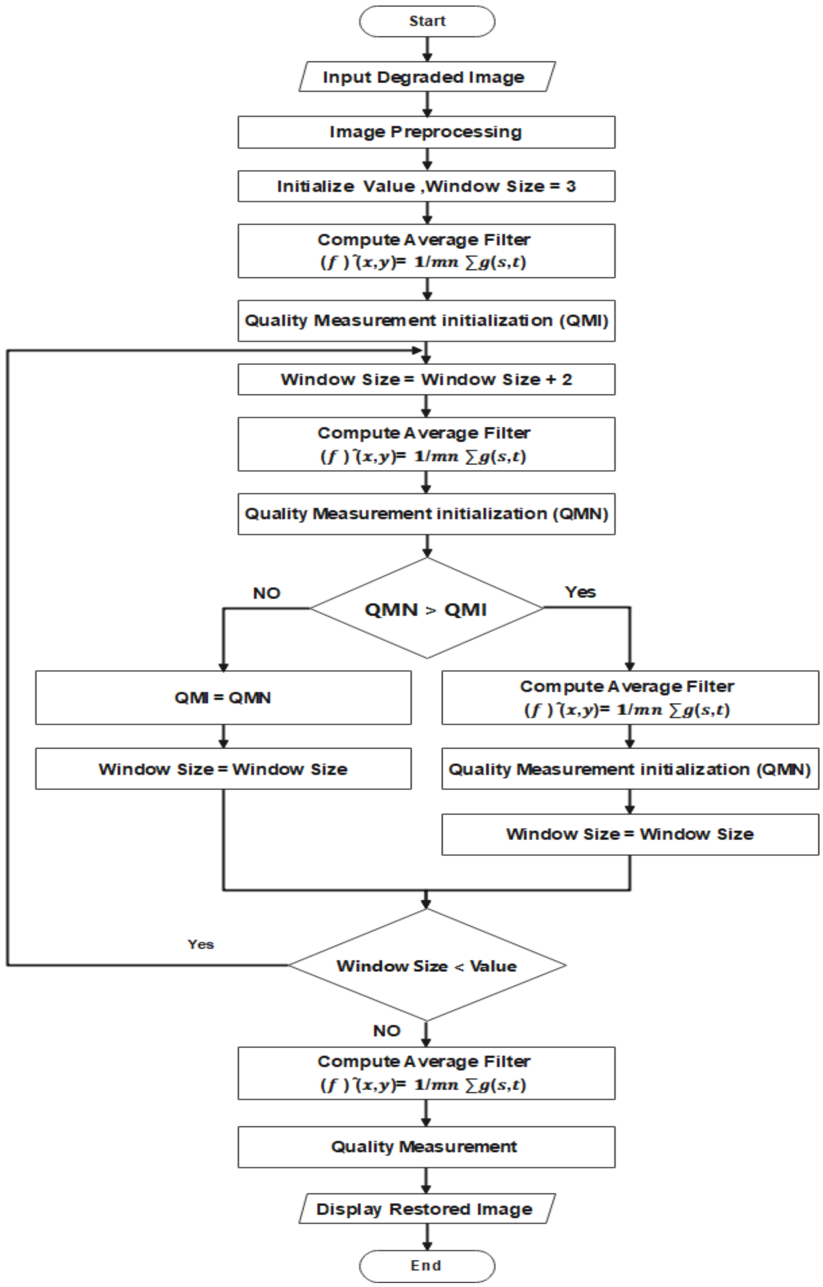


Figure 4.28 Average filter flowchart

2- Median Filter

Flow Chart of a Median Filter provides a clear and concise explanation of the steps involved in implementing the algorithm.

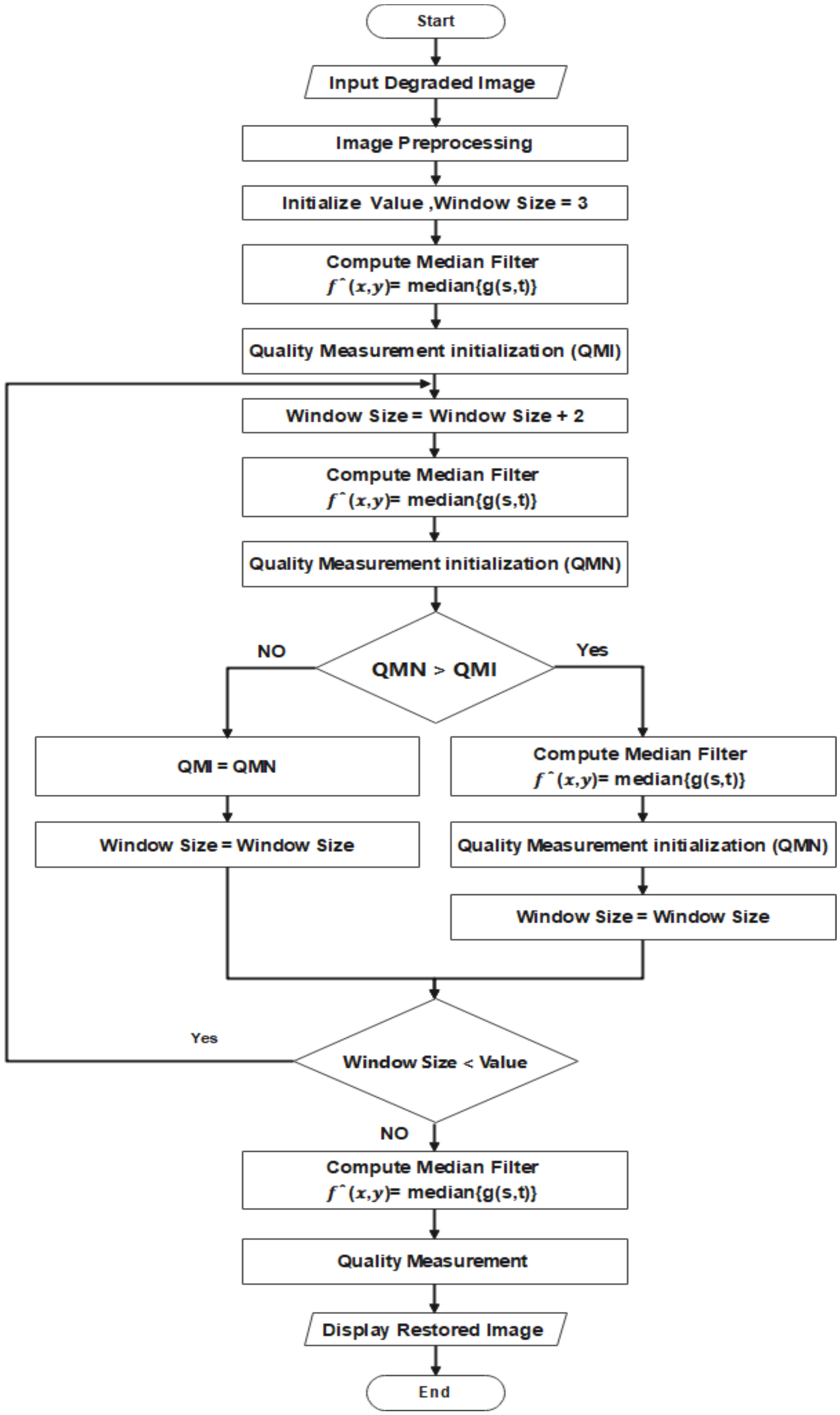


Figure 4.29 Median filter flowchart

3- Non-Local Mean

Flow Chart of a Non local Mean Filter provides a clear and concise explanation of the steps involved in implementing the algorithm.

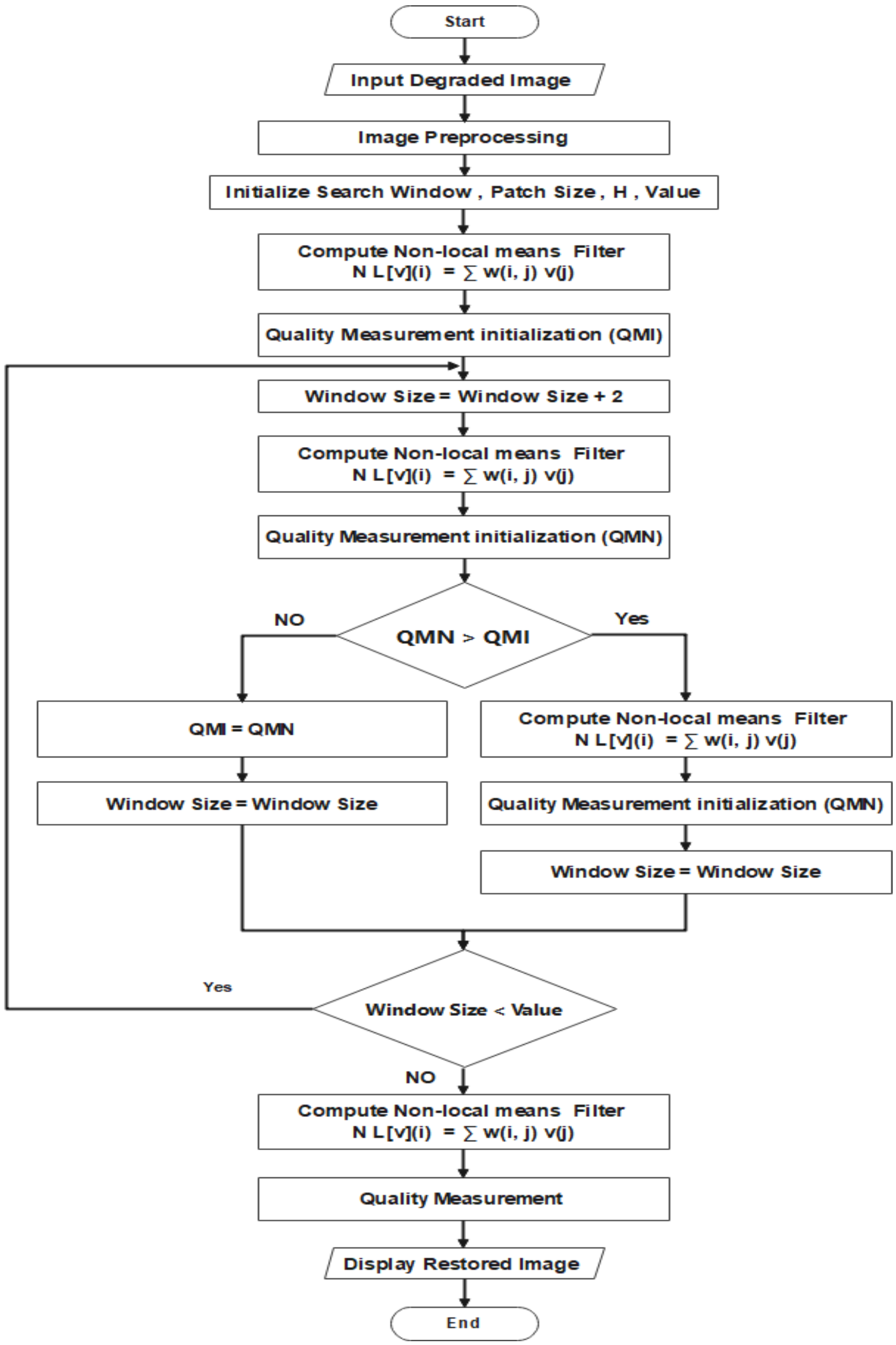


Figure 4.30 Non-Local Mean flowchart

4- <u>Inverse Filter</u>

Flow Chart of a Inverse Filter provides a clear and concise explanation of the steps involved in implementing the algorithm.

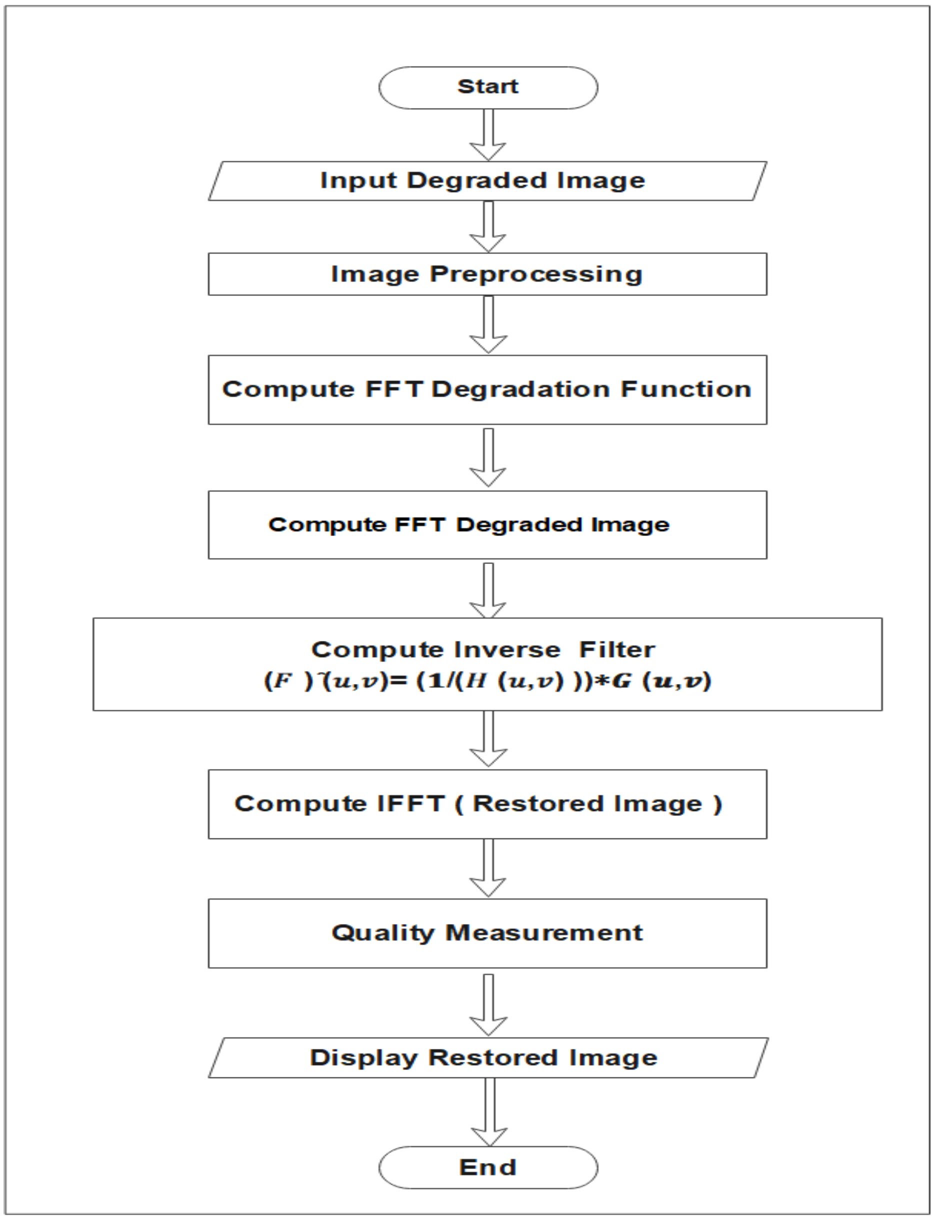


Figure 4.31 Inverse Filter flowchart

5- Radial Inverse Filter

Flow Chart of a Radial Filter provides a clear and concise explanation of the steps involved in implementing the algorithm.

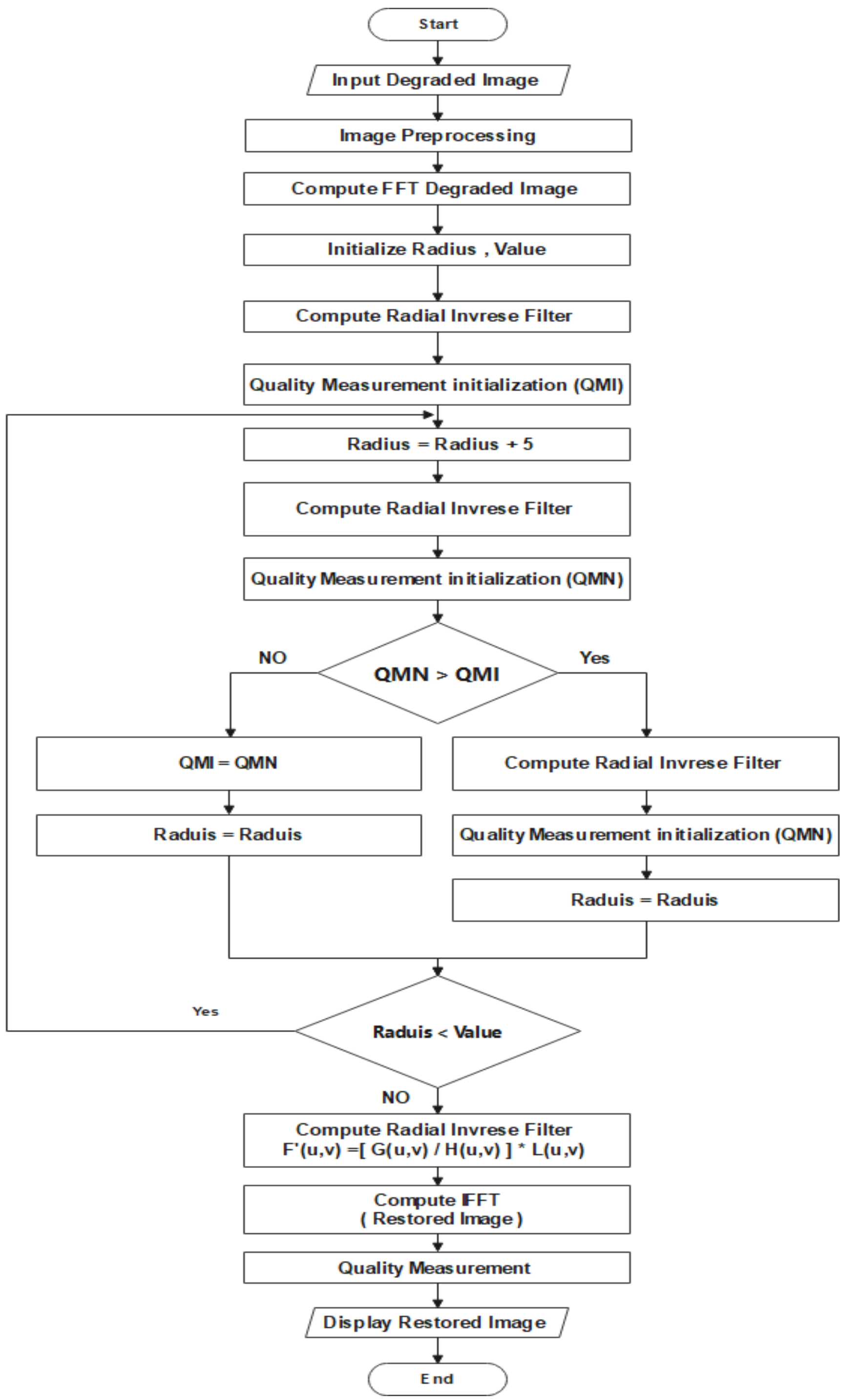


Figure 4.32 Radial Inverse filter flowchart

6- Wiener Filter

Flow Chart of a Wiener filter provides a clear and concise explanation of the steps involved in implementing the algorithm.

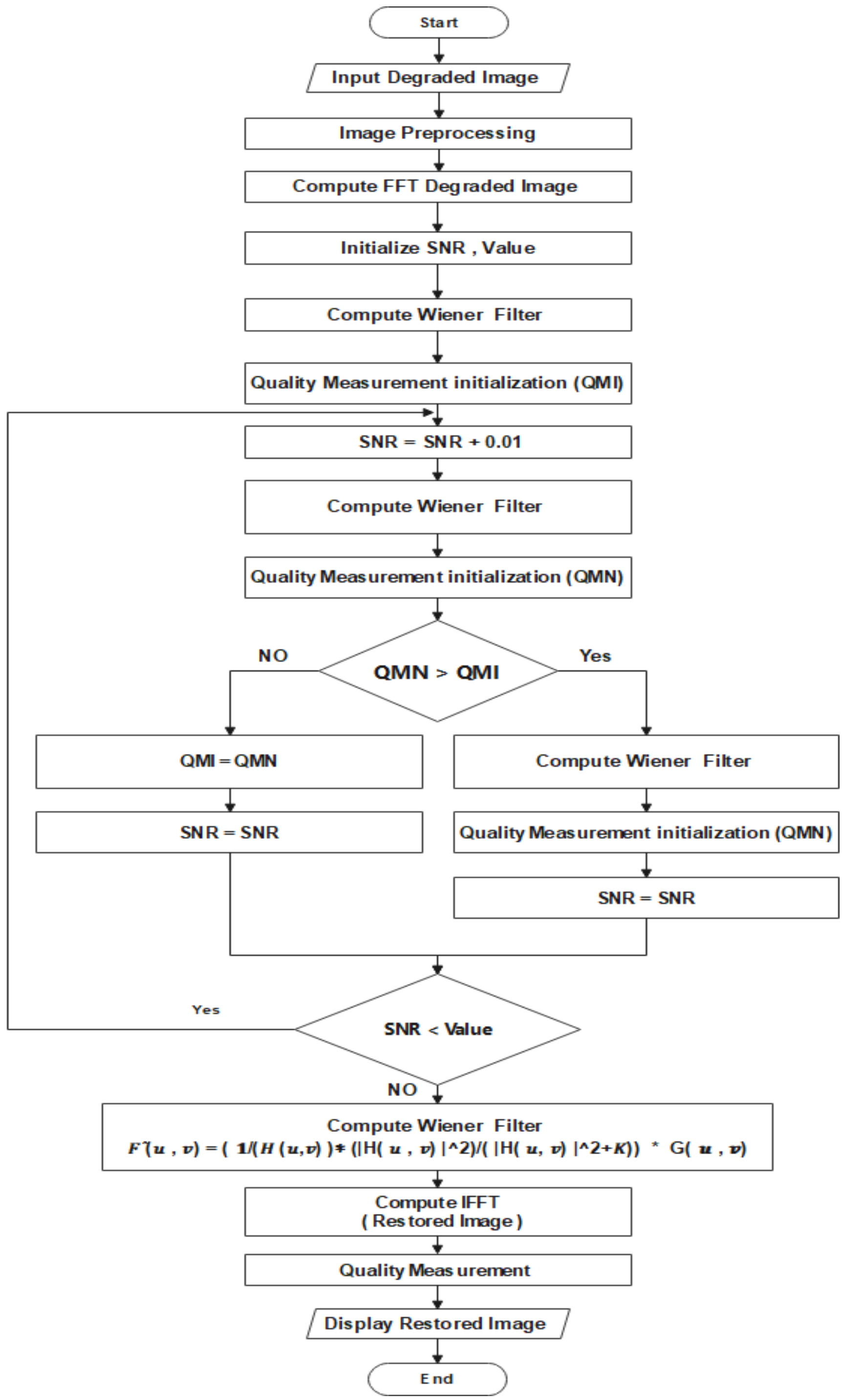


Figure 4.33 Wiener filter flowchart

7- <u>Regularization Filter</u>

Flow Chart of a Regularization Filter provides a clear and concise explanation of the steps involved in implementing the algorithm.

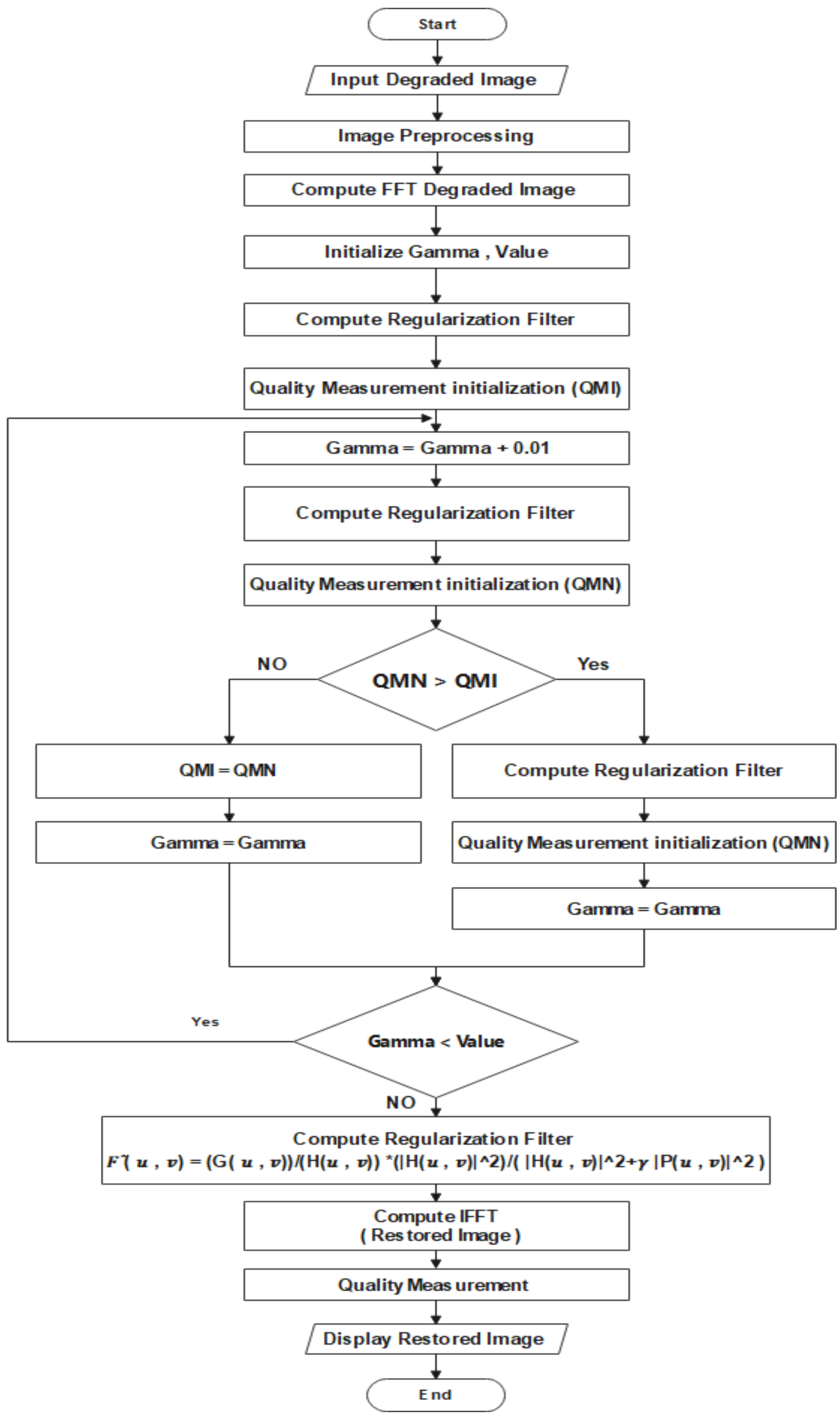


Figure 4.34 Regularization filter flowchart

8- Band Reject Filter

Flow Chart of a Band Reject Filter provides a clear and concise explanation of the steps involved in implementing the algorithm.

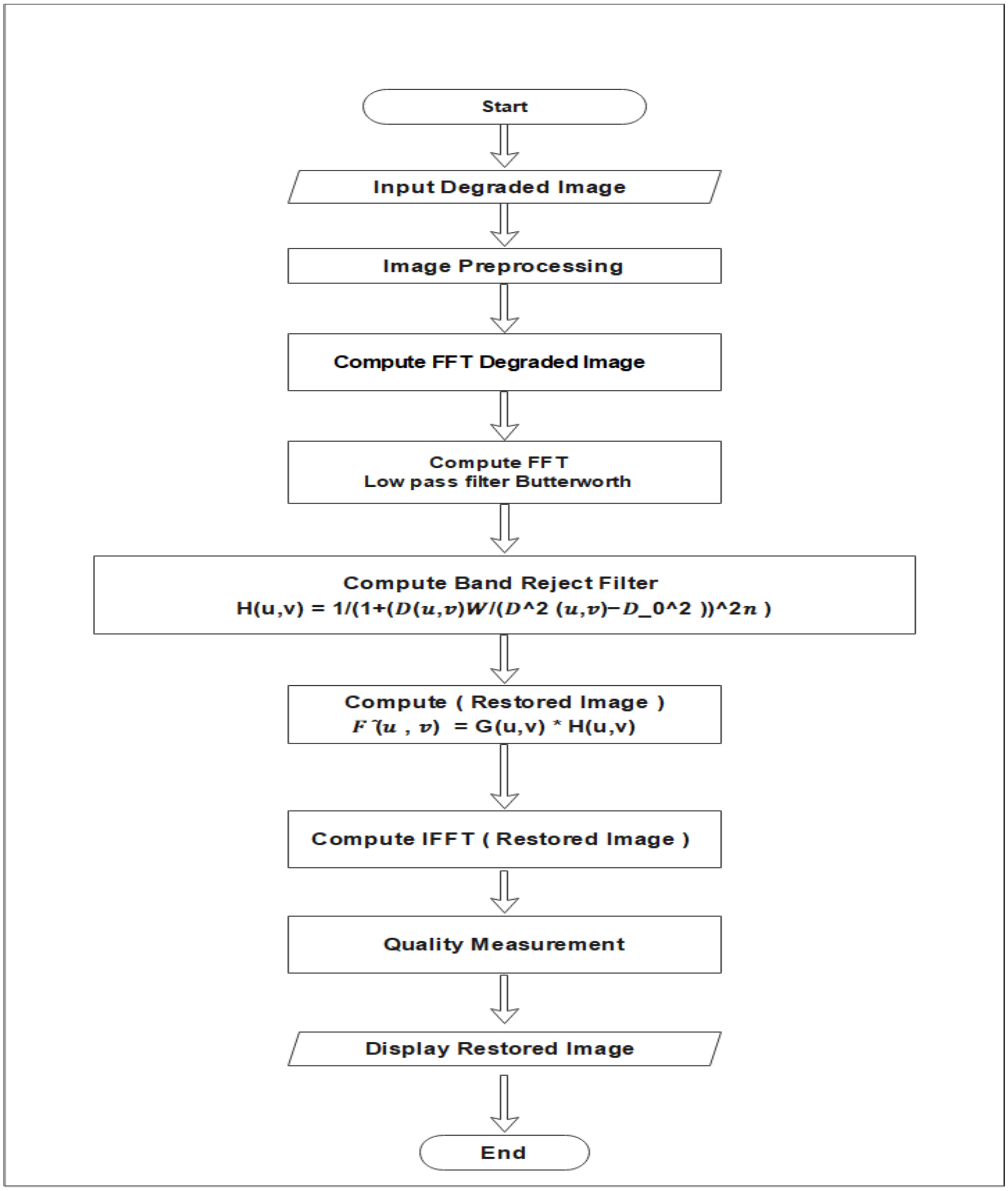


Figure 4.35 Band Reject Filter flowchart

Chapter 5

System Implementation

5.1 System Description:

Digital image restoration using regularization filter.

Digital image restoration is the process of recovering a degraded or damaged image to its original quality or near to it. One of the common techniques used for digital image restoration is the regularization filter.

Regularization is a technique used to reduce the noise and artifacts in an image by incorporating prior knowledge about the image into the restoration process. In image restoration regularization can be used to reduce the impact of noise, blur, and other distortions in the image.

The regularization filter involves applying a filter to the degraded image to reduce the noise, followed by a regularization operator to enhance the sharpness of the image. The regularization operator is designed to restore the edges and details of the image while minimizing the effect of noise and blur.

The regularization filter can be implemented using various algorithms such as the total variation method, wavelet transform, and bilateral filtering. These algorithms have different strengths and weaknesses, and the choice of algorithms have different strengths and weaknesses, and the choice of algorithm depends on the specific requirements of the restoration task.

The regularization filter is a useful technique for digital image restoration, as it can improve the quality of degraded images and enhance their details and sharpness while reducing the impact of noise and blur.

5.2 Platform Used:

- **Python:**

  Python is a high-level, interpreted programming language that is widely used in various domains such as web development, data science, machine learning, and more. It was created by Guido van Rossum and was first released in 1991.

  One of the key features of Python is its simplicity and ease of use, which makes it an excellent choice for beginners. Its syntax is easy to learn and read, and it supports a wide range of programming paradigms, including object-oriented, functional, and procedural programming.

Python is a popular programming language for image processing. There are several libraries available for image processing in Python, some of the most widely used ones are:

1. Pillow: It is a fork of the Python Imaging Library (PIL) and provides a set of functions for opening, manipulating, and saving various image file formats.
2. OpenCV: It is an open-source computer vision library that has many image processing functions such as image filtering, feature detection, object detection, and segmentation.
3. scikit-image: It is a collection of algorithms for image processing, including filters, feature detection, image restoration, and segmentation.
4. NumPy: It is a library for scientific computing in Python and has many functions that can be used for image processing such as array manipulation, linear algebra, and image filtering.
5. matplotlib: It is a library for data visualization and has functions that can be used for plotting images and histograms.

These libraries can be used to perform various tasks in image processing such as image enhancement, noise reduction, edge detection, object detection, and segmentation.

- **PyCharm and jupyter notebook:**

PyCharm is an Integrated Development Environment (IDE) for Python programming language. It is developed by JetBrains and is available in two editions - Community (free and open source) and Professional (paid).

PyCharm provides a wide range of features and tools that make it easier for developers to write, test, and debug their Python code.

1. Code completion and suggestions.
2. Debugging.
3. Testing.
4. Version Control.
5. Code Refactoring.

PyCharm also supports various web frameworks like Django, Flask, and Pyramid, making it an ideal choice for web development projects.

Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations, and narrative text. It supports various programming languages, including Python, R, Julia, and others.

Jupyter Notebook provides an interactive computational environment where you can write and execute code in small chunks called cells. Each cell can contain code, text, or equations, and can be executed independently. This allows you to easily test and debug your code and visualize your results in real-time.

Jupyter Notebook is its ability to integrate code, text, and visualizations in a single document. This makes it an ideal tool for data exploration, scientific computing, and data analysis. Jupyter Notebook also allows you to export your notebook as HTML, PDF, or other formats, making it easy to share your work with others.

Jupyter Notebook also supports a wide range of extensions and plugins that can enhance its capabilities. For example, you can use extensions to add support for additional languages, integrate with version control systems, or create interactive visualizations.

- **Pillow Image:**

Pillow is a Python Imaging Library that provides a set of functions to open, manipulate, and save various image file formats. Pillow is a fork of the Python Imaging Library (PIL) and is a widely used image processing library in Python.

Pillow supports a wide range of image file formats, including JPEG, PNG, BMP, GIF, and TIFF. It provides various functions to perform image manipulation operations such as cropping, resizing, rotating, and filtering.

- **Open CV:**

OpenCV (Open Source Computer Vision) is a popular open-source library for computer vision and image processing tasks. It was originally developed by Intel and is now maintained by the OpenCV community. OpenCV provides a wide range of functions and algorithms for image and video processing, including:

1. Image and video input/output.
2. Basic image processing operations such as blurring, resizing, and thresholding.
3. Advanced image processing operations such as edge detection, feature detection, and image segmentation.
4. Object detection and tracking.
5. Face recognition.
6. 3D reconstruction.

OpenCV supports multiple programming languages such as C++, Python, Java, and MATLAB, making it widely accessible to developers and researchers. The library is cross-platform and can be used on Windows, Linux, macOS, Android, and iOS.

OpenCV also provides support for deep learning frameworks such as Tensor Flow and PyTorch, allowing users to easily integrate deep learning models into their computer vision applications.

OpenCV has a large community of developers and researchers who contribute to its development and provide support through forums, documentation, and tutorials. It is widely used in various applications such as robotics, augmented reality, autonomous vehicles, and medical imaging.

- **Tkinter:**

Tkinter is a Python library that provides a way to create graphical user interfaces (GUIs) using the Tk GUI toolkit. Tkinter comes bundled with the standard Python distribution, so it is a very popular choice for building desktop applications using Python.
Tkinter provides a set of widgets (such as buttons, labels, text boxes, and menus) that can be placed on a GUI window, as well as methods for handling events (such as mouse clicks and keyboard input) and managing layouts. Tkinter is a cross-platform library, which means that code written using Tkinter can run on any operating system that has a Python interpreter and a Tk installation.
Tkinter is relatively easy to learn and use, making it a good choice for beginners who want to create simple GUI applications. However, it

also has more advanced features for building complex and customizable GUIs.
Tkinter is a built-in GUI (Graphical User Interface) toolkit in Python used for creating desktop applications. It provides a set of tools and widgets that allow developers to create user interfaces quickly and easily.
Tkinter can be used for image processing by utilizing the Pillow library, which provides support for opening, manipulating, and saving many different images file formats.

5.3 Front End:

- **Crop image:** Cropping an image in image processing means removing a portion of the image by selecting a rectangular region.

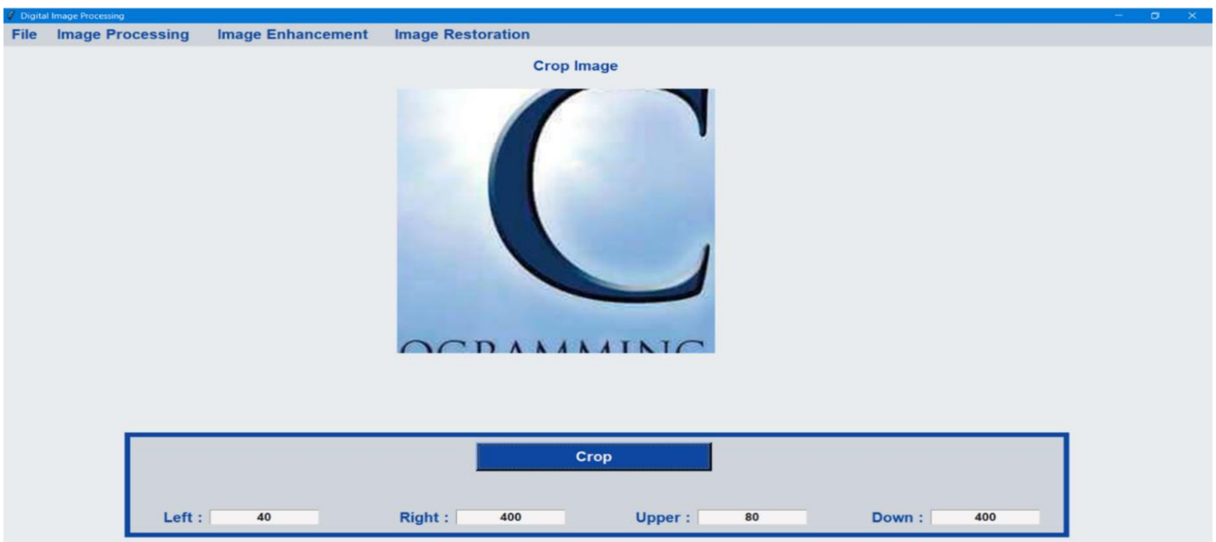


Figure 6.1 crop

Here we achieve this by initializing the left, right, upper, and down value. Which helps user to enter the value he wants to make the crop.

- **Mirror:** reversed or mirrored horizontally or vertically.

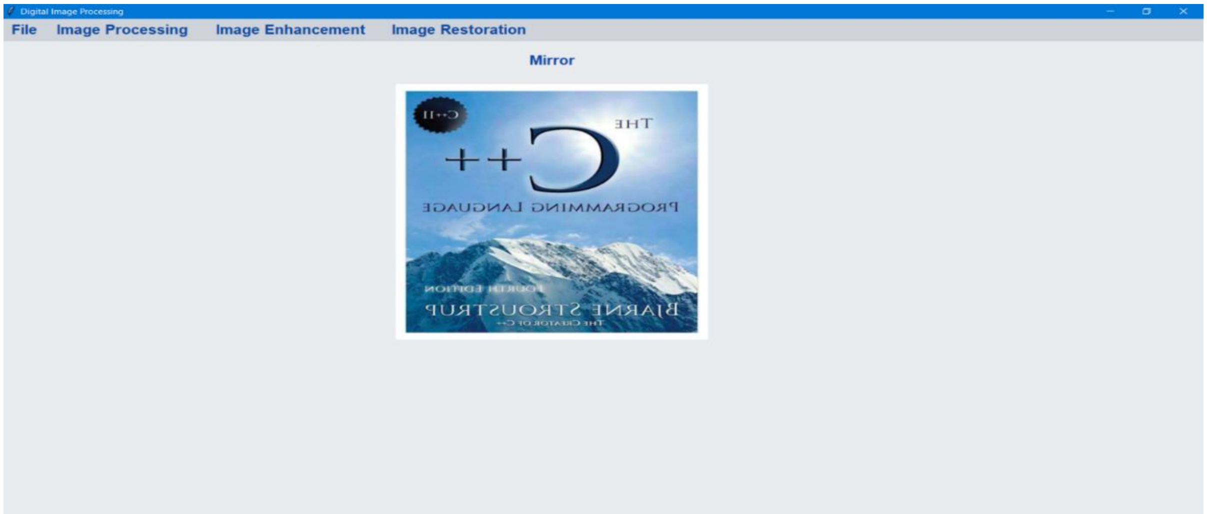


Figure 6.2 Mirror page

- **Rotation:** rotate image by defining the degree of an angle(preprocessing).

Figure 6.3 Rotate page at 180 Deg.

- Degree of angle =180

Figure 6.4 Rotate at 45 Deg.

- Degree of angle = 45

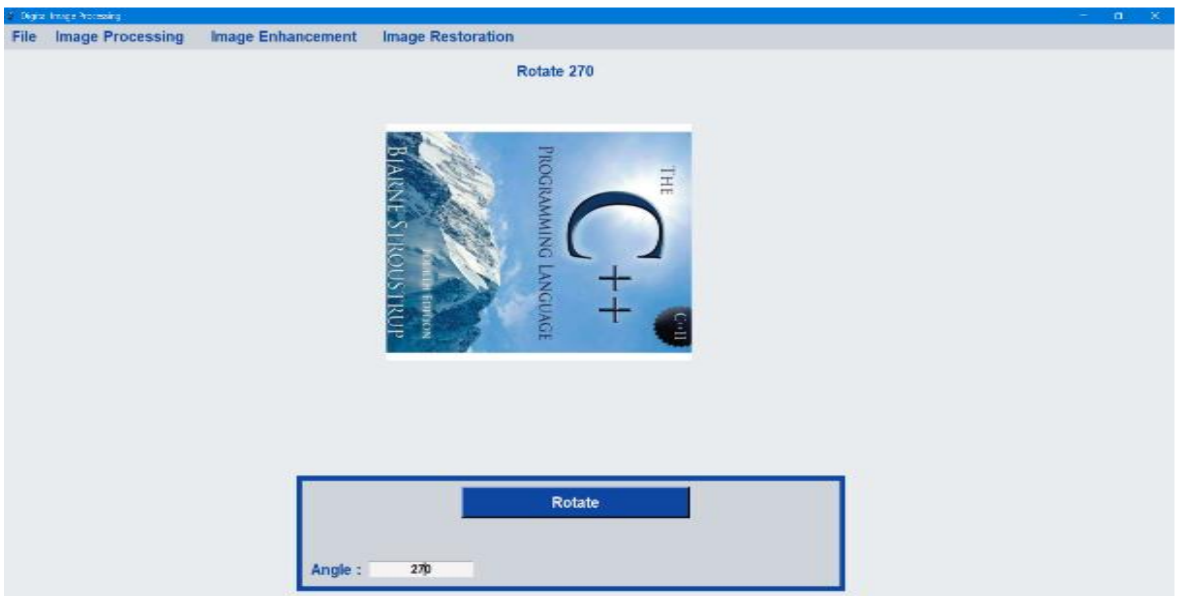


Figure 6.5 Rotate at 270 Deg.

- Degree of angle =270

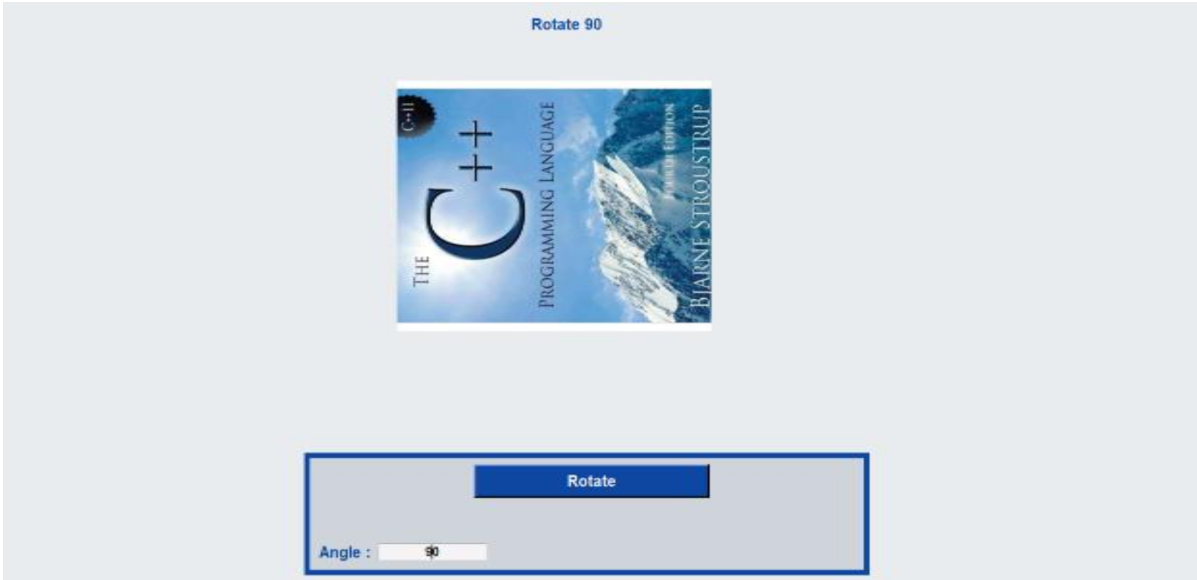


Figure 6.6 Rotate at 90 Deg.

- Degree of angle = 90

- **Zoom:**

Zoom in image processing refers to the process of enlarging or magnifying a portion of an image while maintaining its quality and clarity (preprocessing).

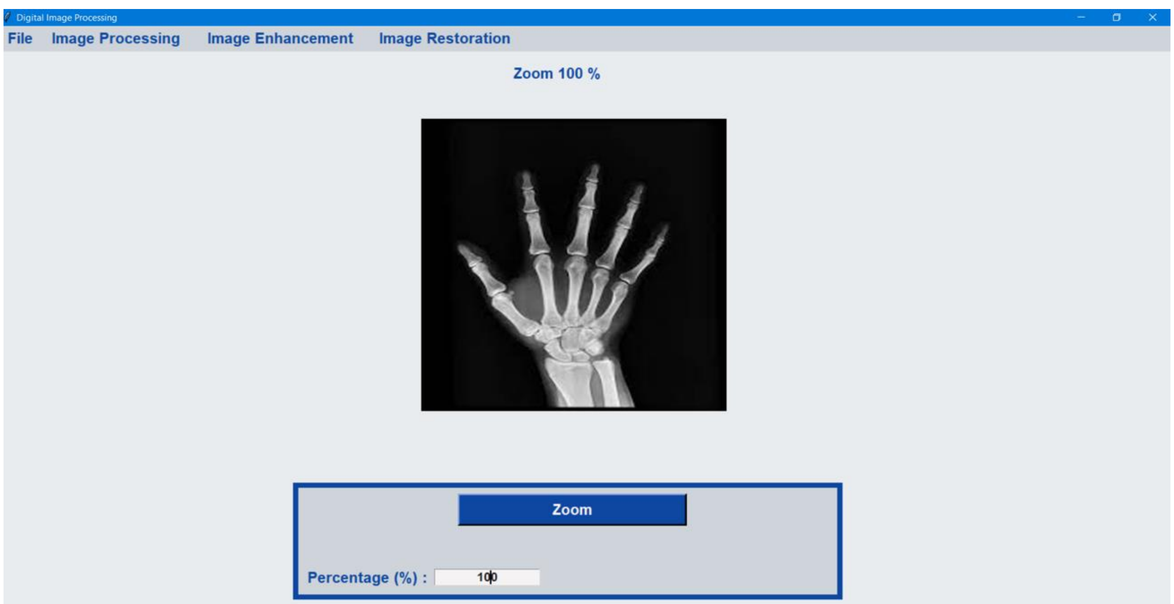


Figure 6.7 Zoom page at 100%

- The percentage of zoom is 100%

Figure 6.8 Zoom page at 160%

- The percentage of zoom is 160%

Figure 6.9 Zoom page at 60%

- The percentage of zoom is 60%

Preprocessing:

- **Pixelate**

  Pixilation is the process of intentionally blurring or distorting an image by reducing its resolution or enlarging it beyond its original size. This is done by breaking down the image into small squares called pixels, which are then enlarged or reduced in size to create a blocky, pixelated effect.

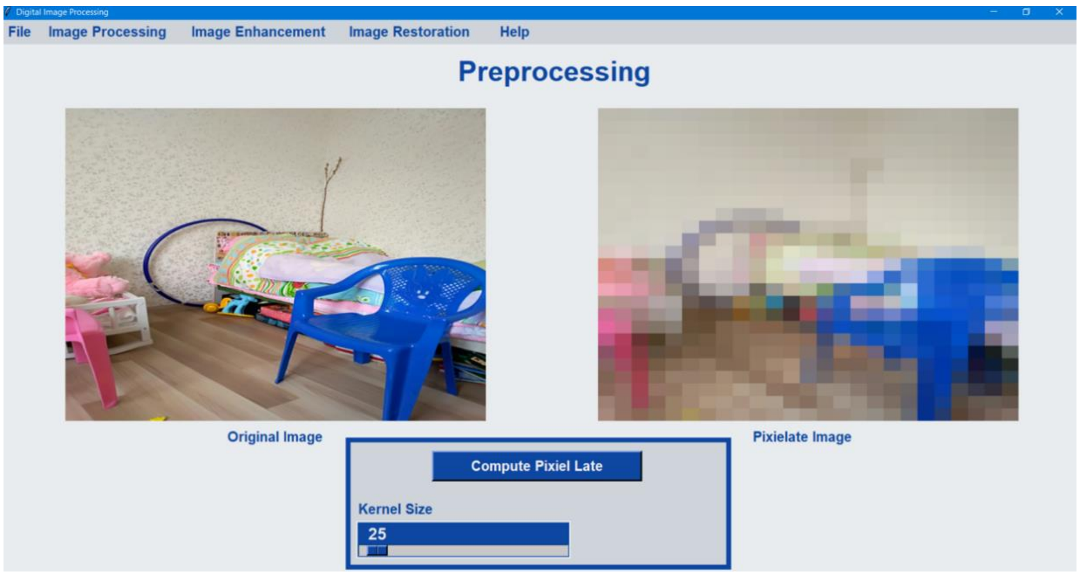


Figure 6.10 Pixelate

Figure 6.11 pixelate at 255

- **Gray scale**
  For each pixel, take the average of the red, green, and blue pixel values to get the grayscale value.

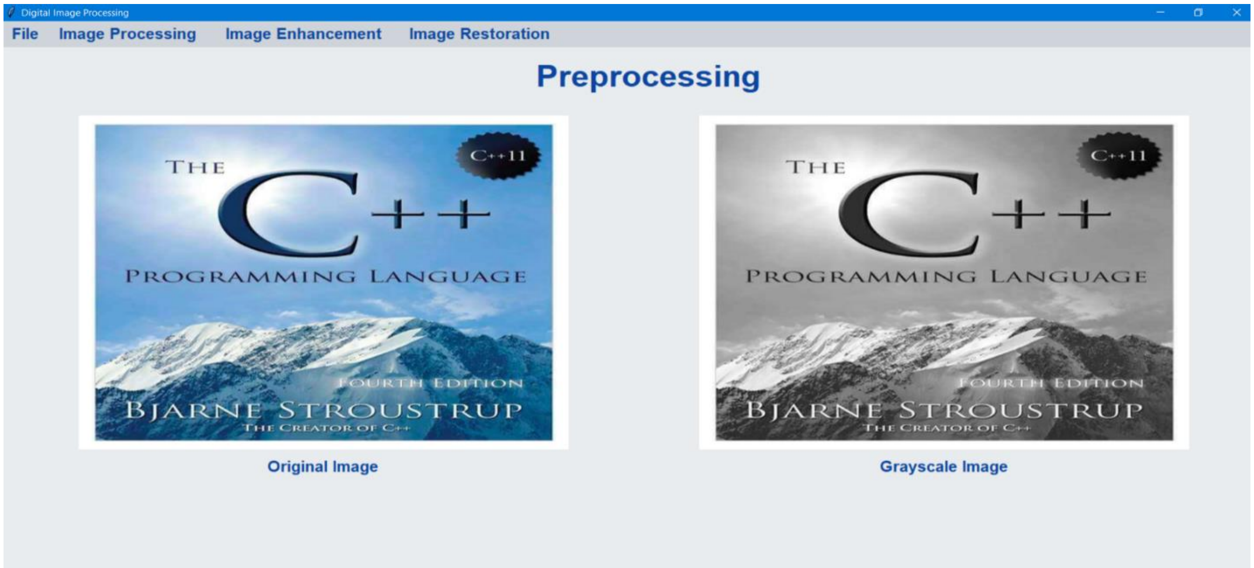


Figure 6.12 Grayscale page

Add Noise:

1- Salt and Pepper Noise: the black and white.

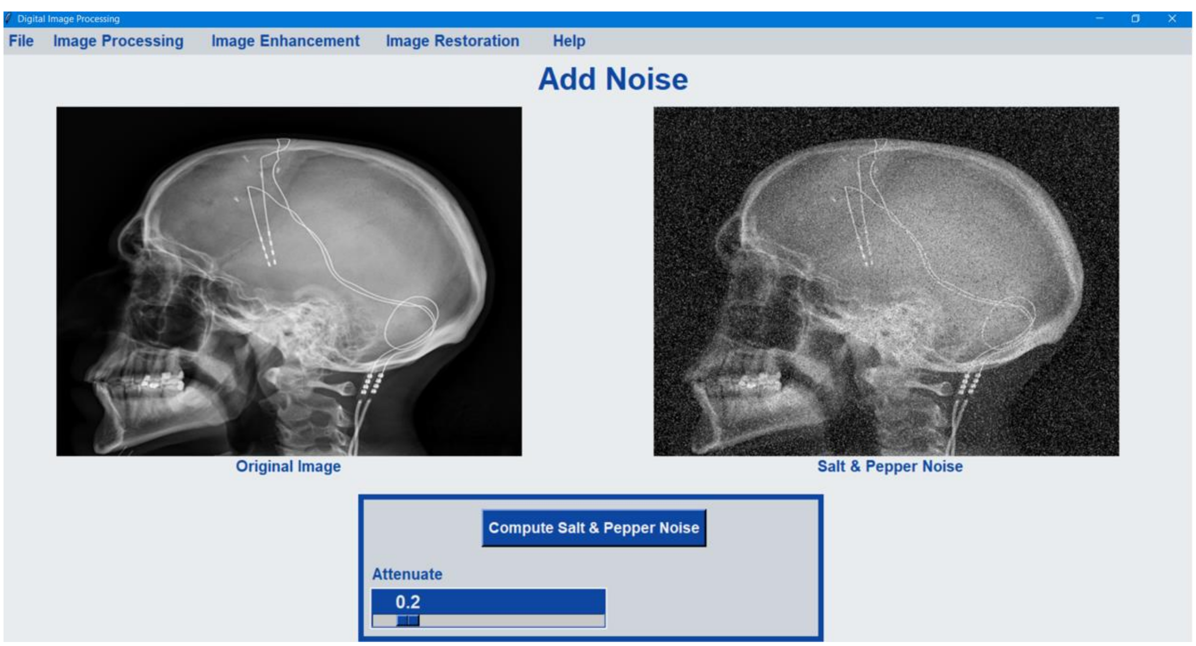


Figure 6.13 Salt & Pepper noise at variance 20%

- Attenuate (noise variance) = 0.2 (20%)

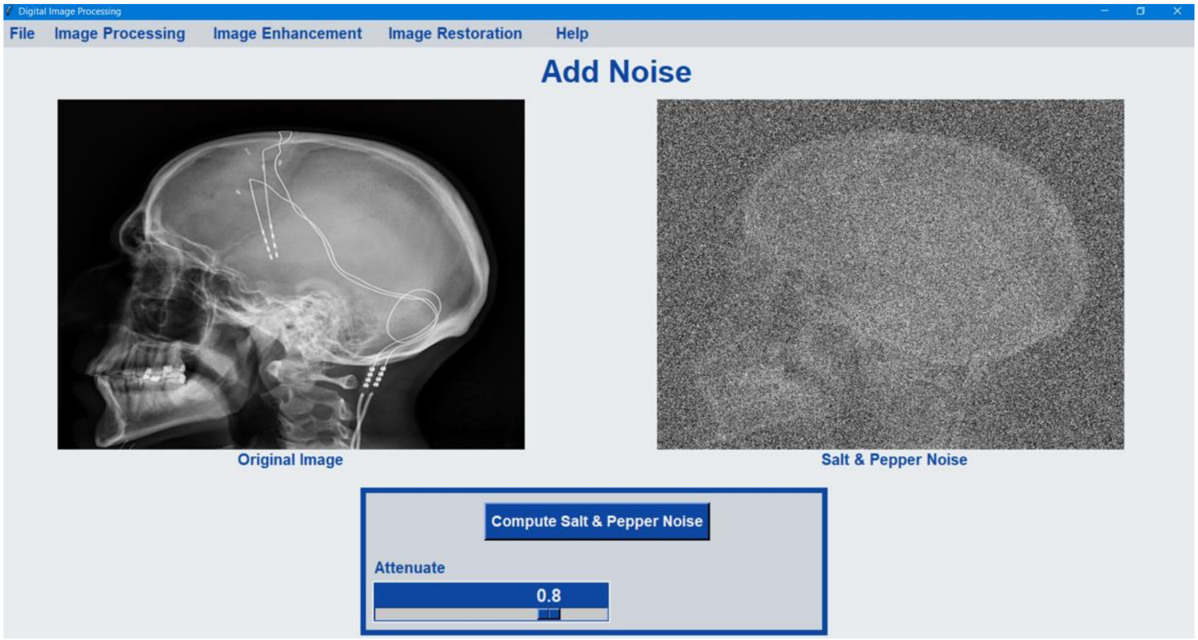


Figure 6.14 Salt & Pepper noise at variance 80%

- Attenuate (noise variance) = 0.8 (80%)

- Increasing the value of attenuate lead to increasing the value of noise in the image
- We make this function to test algorithms.

2- Gaussian Noise:

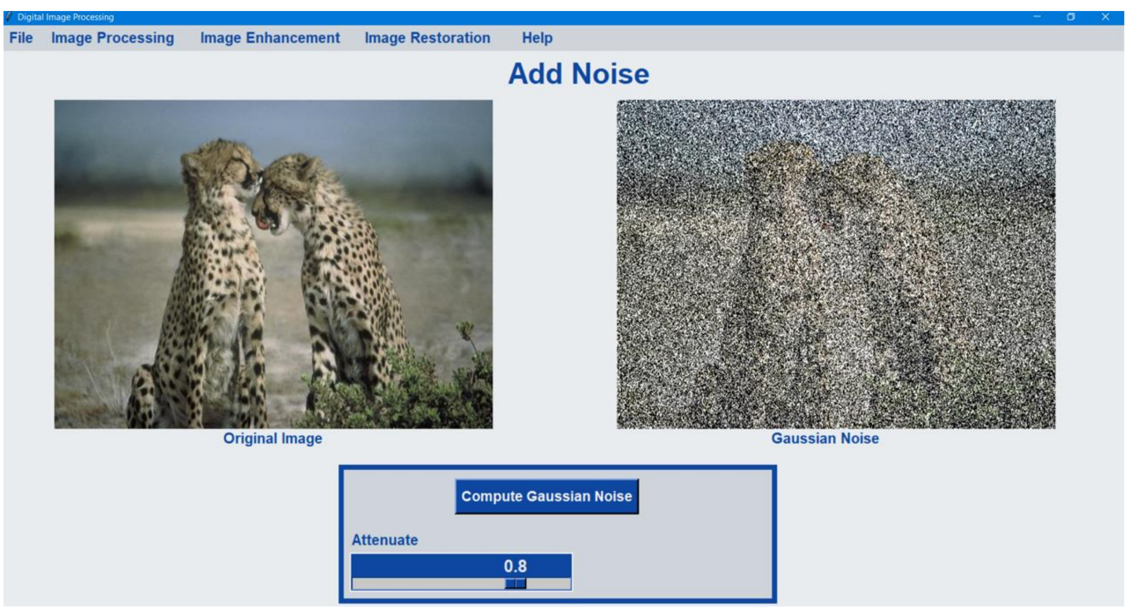


Figure 6.15 Gaussian noise at variance 30%

- value of attenuate (noise variance) =0.8 (80%)

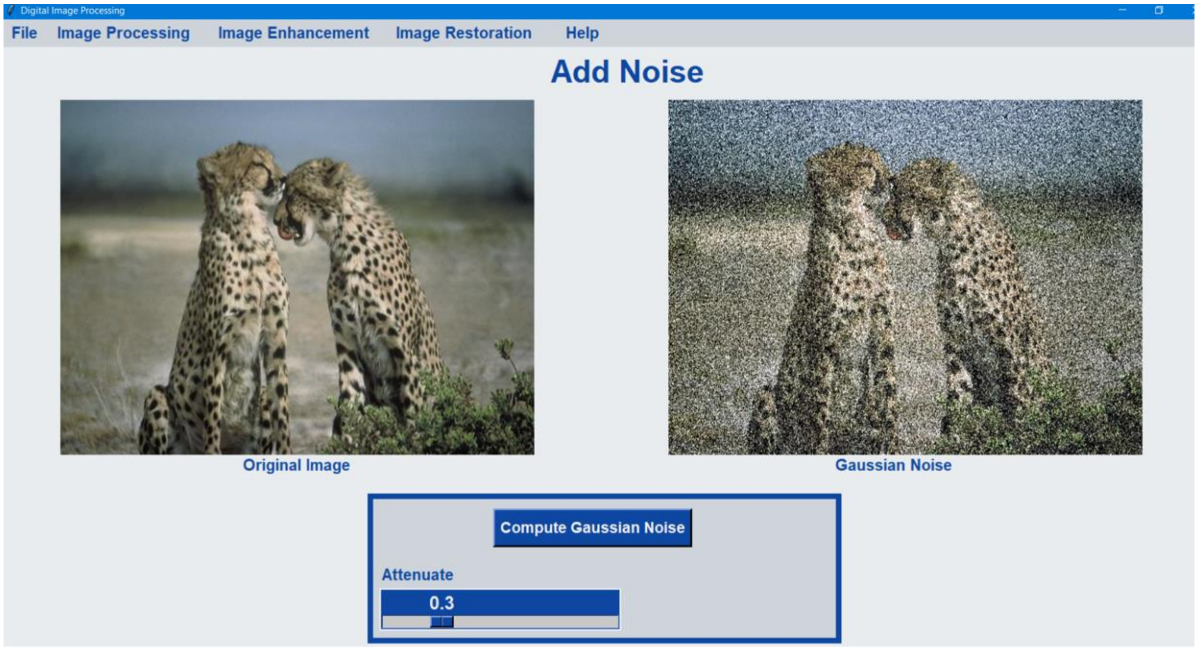


Figure 6.16 Gaussian noise at variance 80%

- value of attenuate (noise variance) =0.3 (30%)
- Increasing the value of attenuate lead to increasing the value of noise in the image

Image Enhancement Technique:

1- Histogram Equalization

2- Negative Image:

- Image negative is produced by subtracting each pixel from the maximum intensity value.

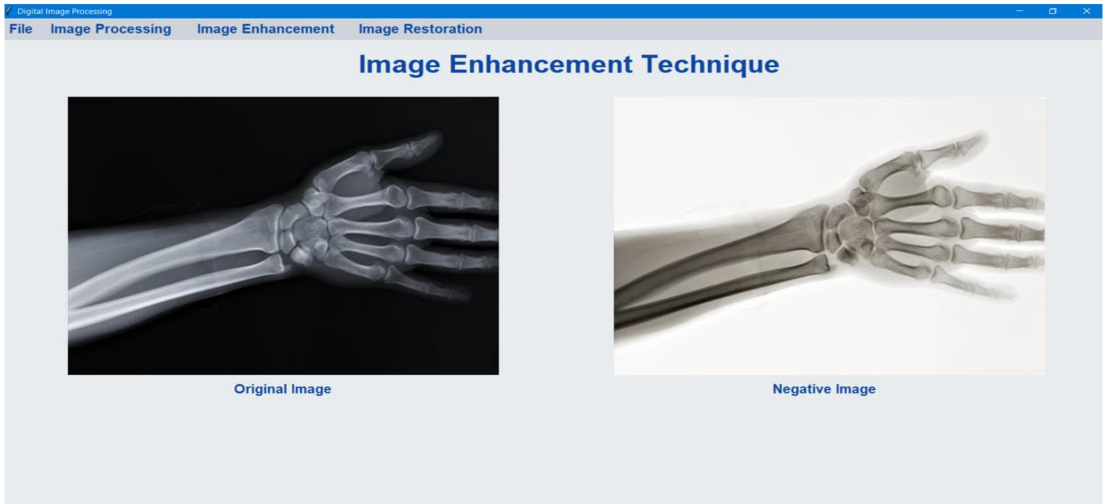


Figure 6.17 Negative image page

3- Log Image:

- Log transformation of an image means replacing all pixel values present in the image, with its logarithmic values.

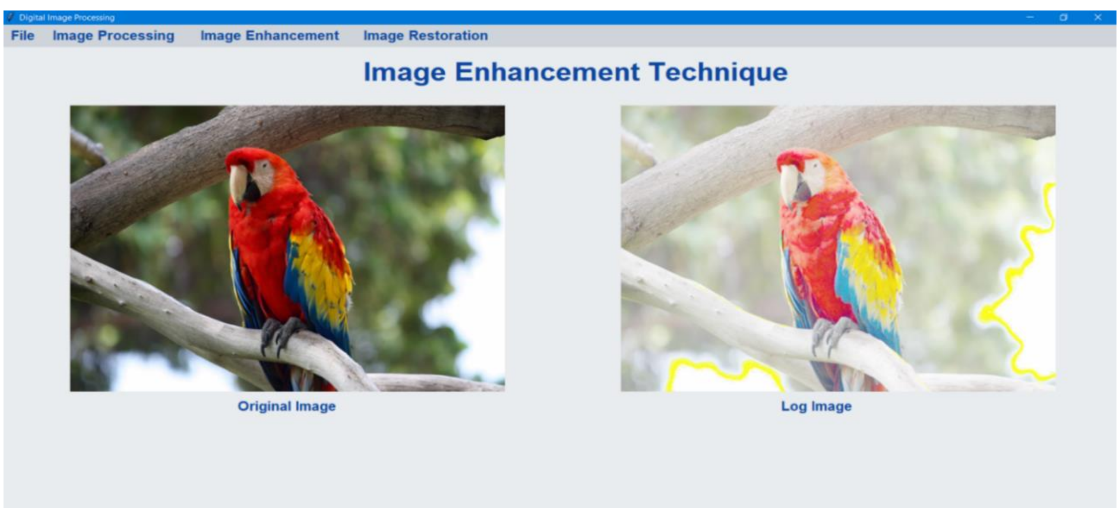


Figure 6.18 Log image page

4- Inverse Log Image:

- It is a mathematical operation used in image processing to convert the logarithmic representation of an image back to its original linear representation.

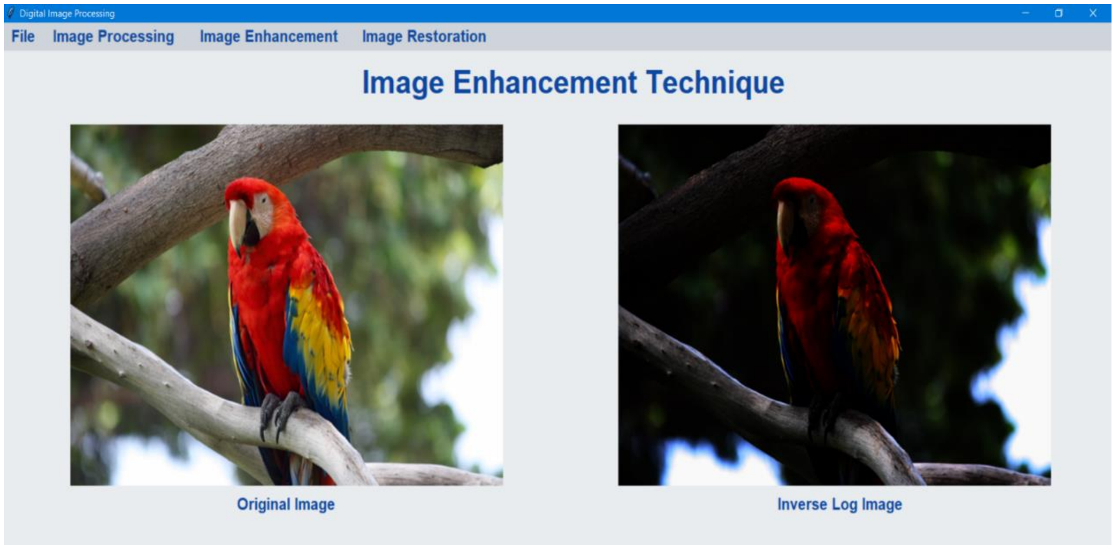


Figure 6.19 Inverse Log image page

5- Sharpness:

- refers to the clarity and detail of an image. It is a measure of how well-defined the edges and boundaries of objects in an image appear.

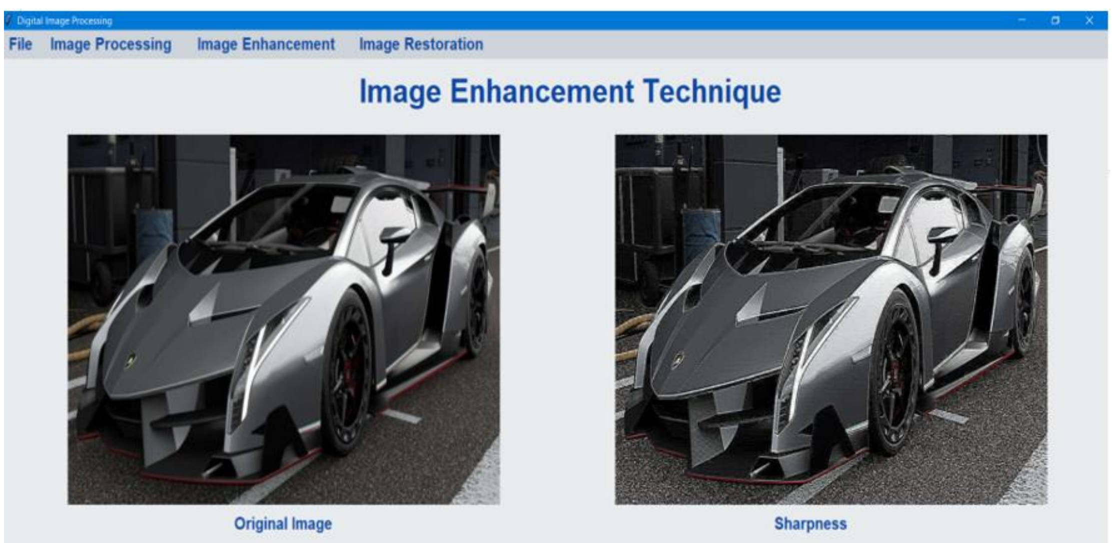


Figure 6.20 Sharpness page

6- Edge Detection:

- Involves identifying the boundaries between different regions in an image. The edges in an image represent the areas of rapid intensity changes, which can be used.

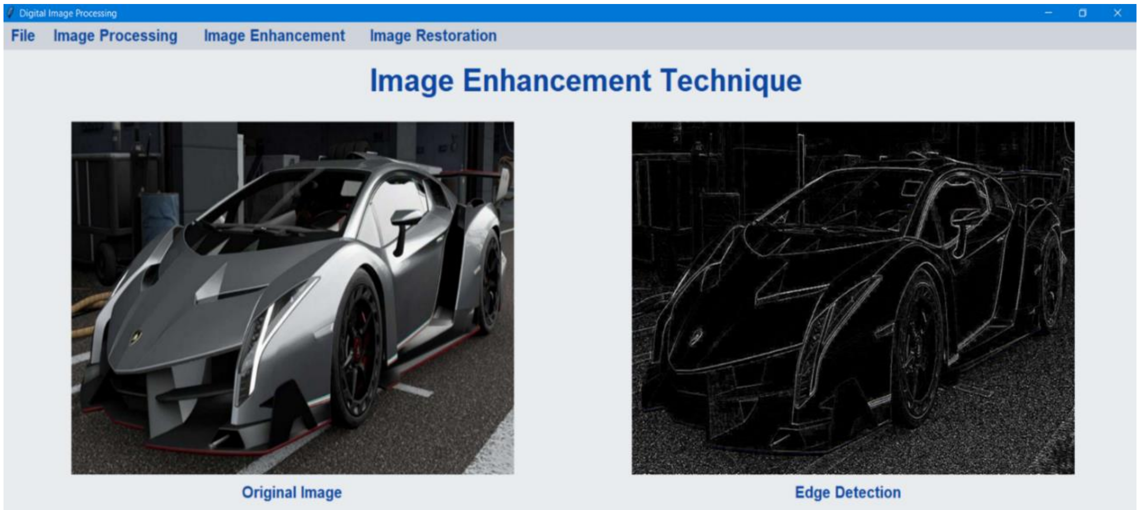


Figure 6.21 Edge Detection page

7- Image Contrast:

Image contrast is to enhance the difference between the light and dark areas of an image, making it easier to distinguish details and improve overall visual clarity.

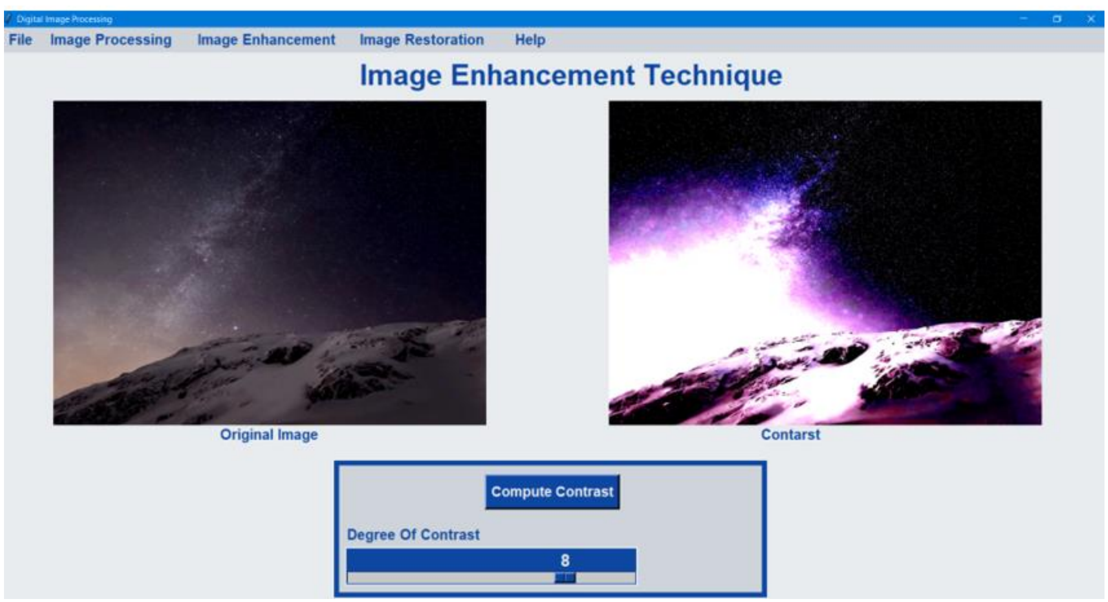
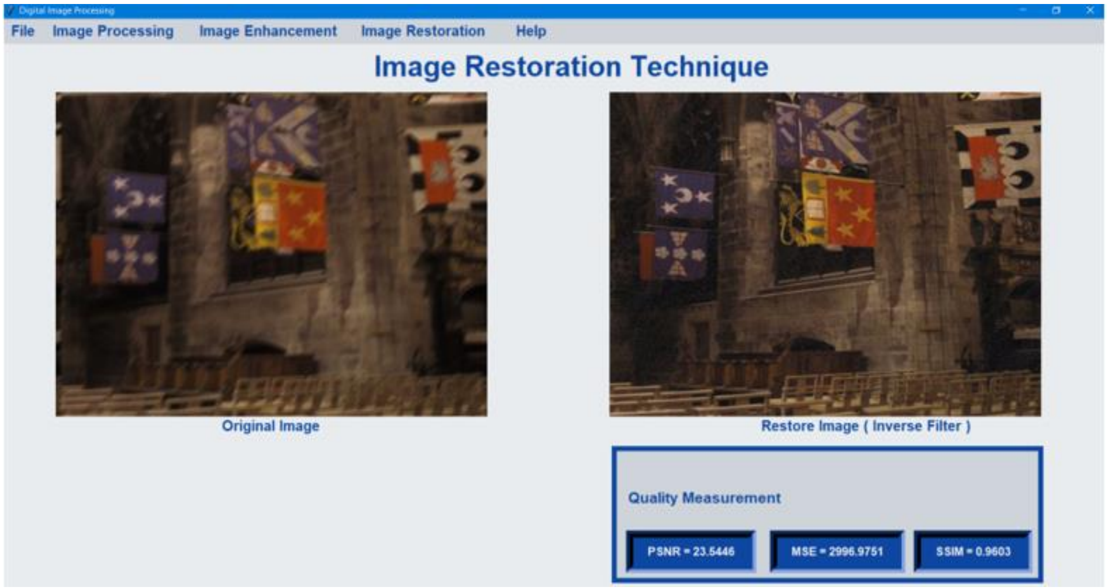


Figure 6.22 Contrast page at 8

Image Restoration Technique:

1- Inverse Filter



- Inverse filters remove blur from this image well by value we get it from iteration depending on quality measurement.

2- Averaging Filter:

- The average filter is to smooth out an image by replacing each pixel value with the average value of its neighboring pixels, this is done by sliding a window of a certain size.

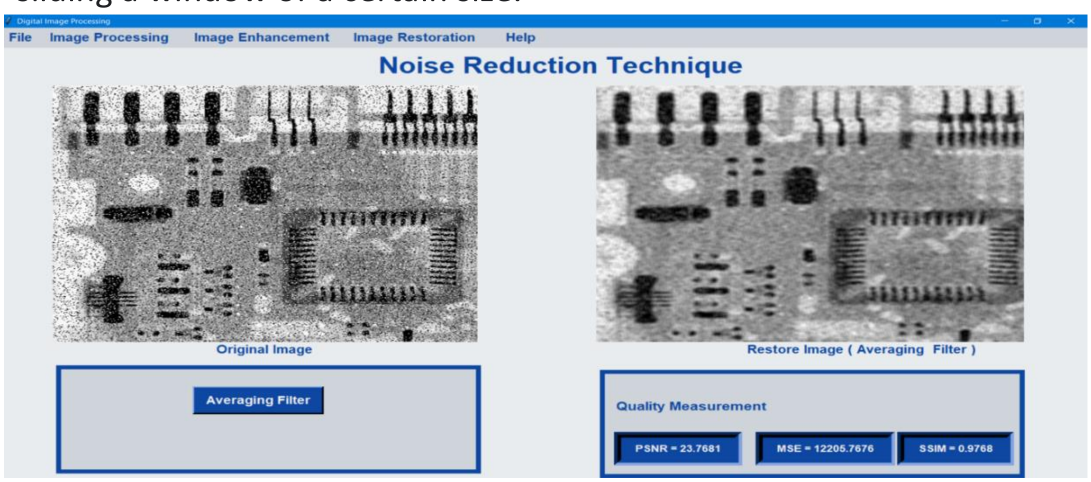


Figure 6.23 Average filter

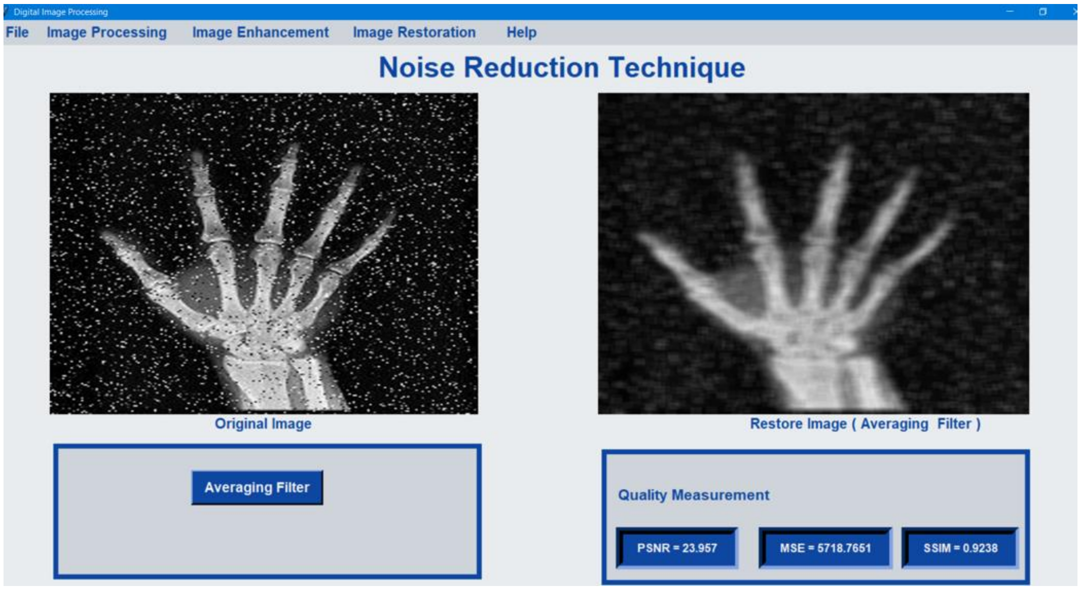


Figure 6.24 Average filter

- Kernel size Specifies the percentage of noise that will be removed from the image, We choose kernel size by iteration depend on quality measurement such as MSE (mean square error)
- Average filter is not better in images that you need to preserve its details and edges.

3- Median Filter:

- It works by replacing each pixel in the image with the median value of its neighboring pixels.

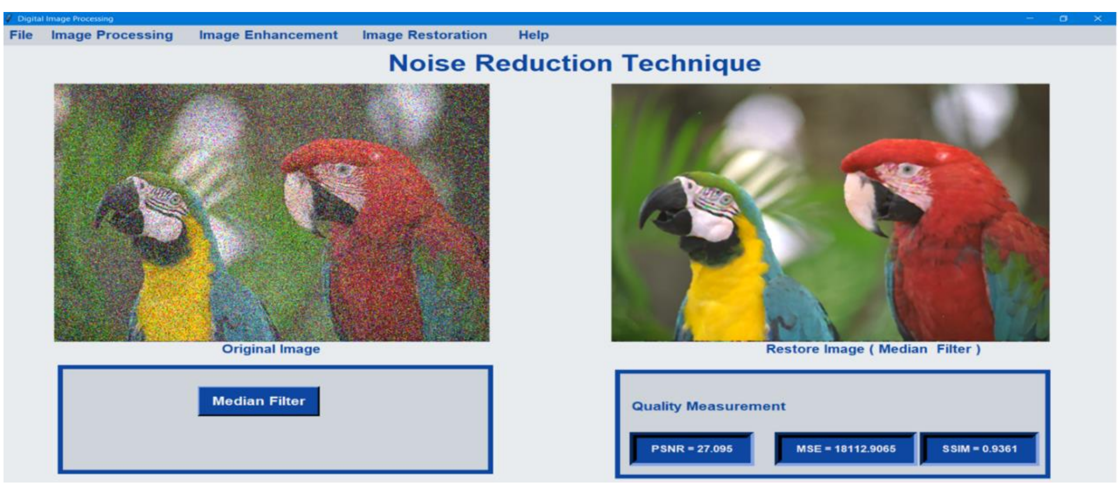


Figure 6.25 Median filter

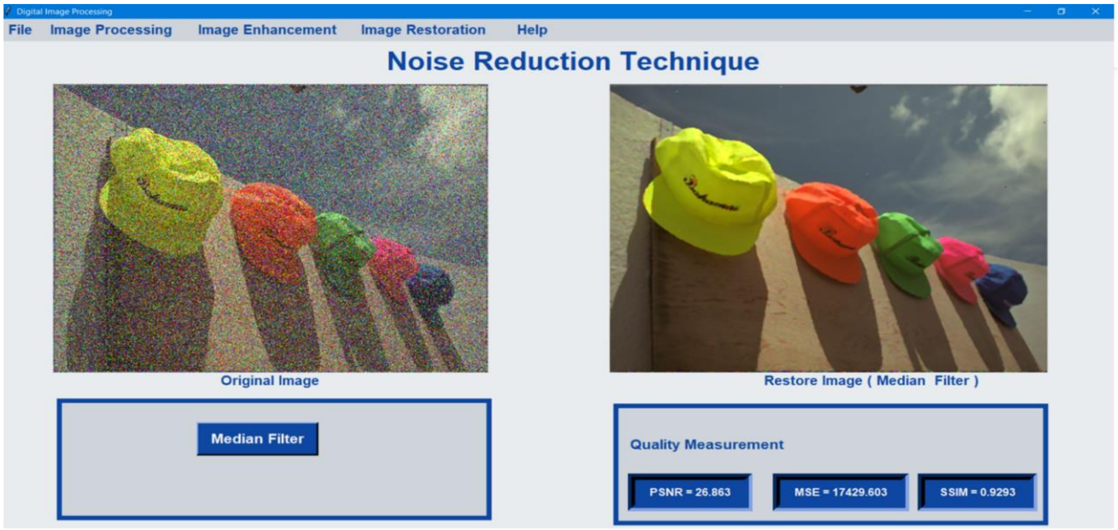


Figure 6.26 Median filter

- Kernel size Specifies the percentage of noise that will be removed from the image, We choose kernel size by iteration(default) depend on quality measurement such as MSE (mean square error)
- It's better at removing noise and preserving in detail compared to the average filter.

4- Non-Local Means Filter:

- remove noise from an image by averaging the pixels in a larger area around each pixel, rather than just the neighboring pixel.

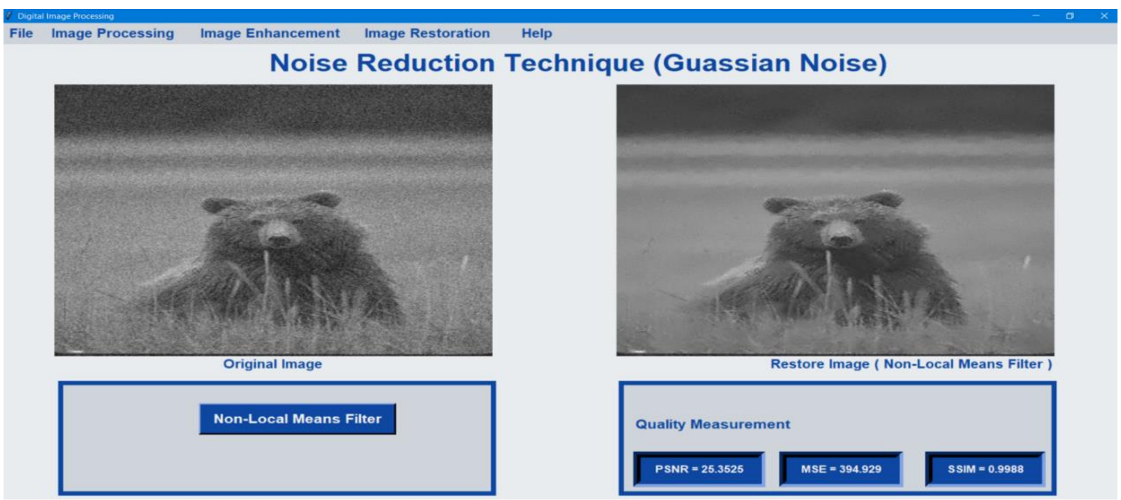


Figure 6.27 non-Local mean filter

Figure 6.28 non-Local mean filter

- weighted average Specifies the percentage of noise that will be removed from the image, we choose weighted average by iteration(default) depend on quality measurement such as (MSE & PSNR)
- In this image nonlocal mean filter remove Gaussian noise but make image smoother and some details is removed

5- Weiner Filter

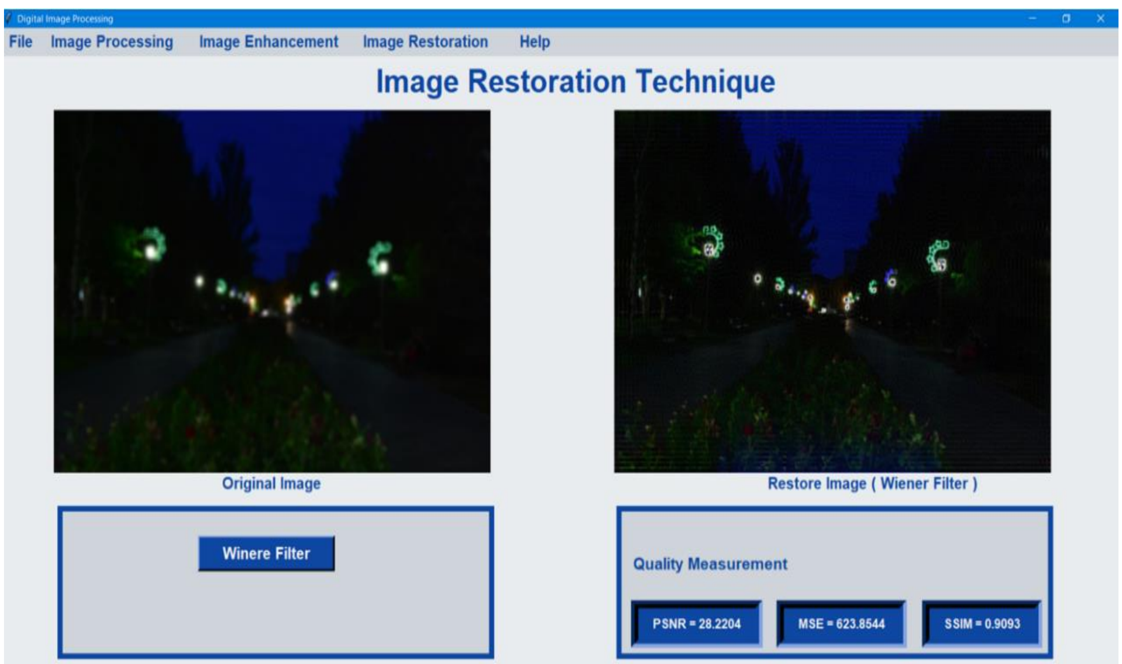


Figure 6.29 Wiener filter

Figure 6.30 Wiener filter

- The value of SNR, user should enter the value of it but we make iteration depend on the value of quality measurement to choose the best value of this image.
- In this image blur removed from the image and make it seen but without preserving edges.

6- Regularization Filter

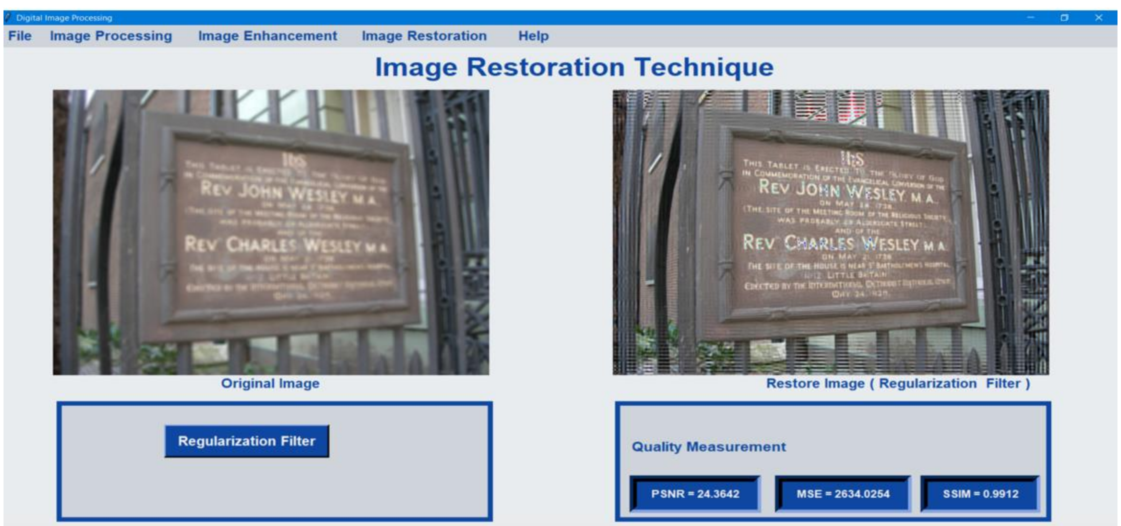


Figure 6.31 Regularization filter

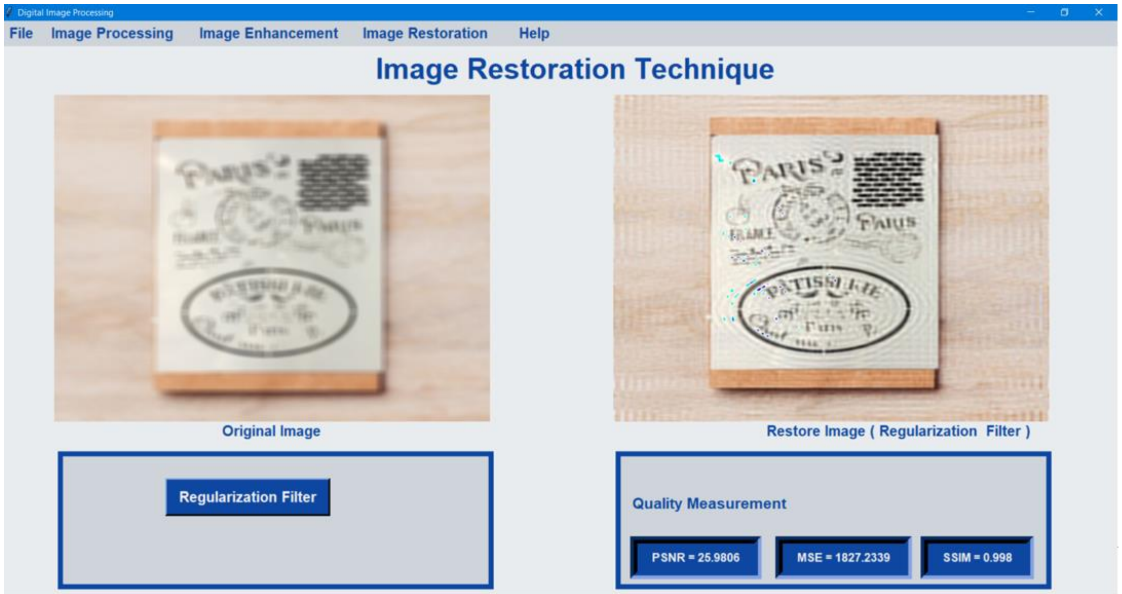


Figure 6.32 Regularization filter

Figure 6.33 Regularization filter

- The value of gamma, user should enter the value of it but we make iteration depend on the value of quality measurement such as (SSIM (structure similarity index measurement)) to choose the best value of this image.
- In this image blur removed from the image and make it seen but without preserving edges.

7- Radial Filter

Figure 6.34 Radial filter

Figure 6.35 Radial filter

- The value of radius, user should enter the value of it, but we make iteration depend on the value of quality measurement to choose the best value of this image.
- In this image blur is removed from the image and make it seen but without preserving edges and some details in the image.

8- Band Reject Filter

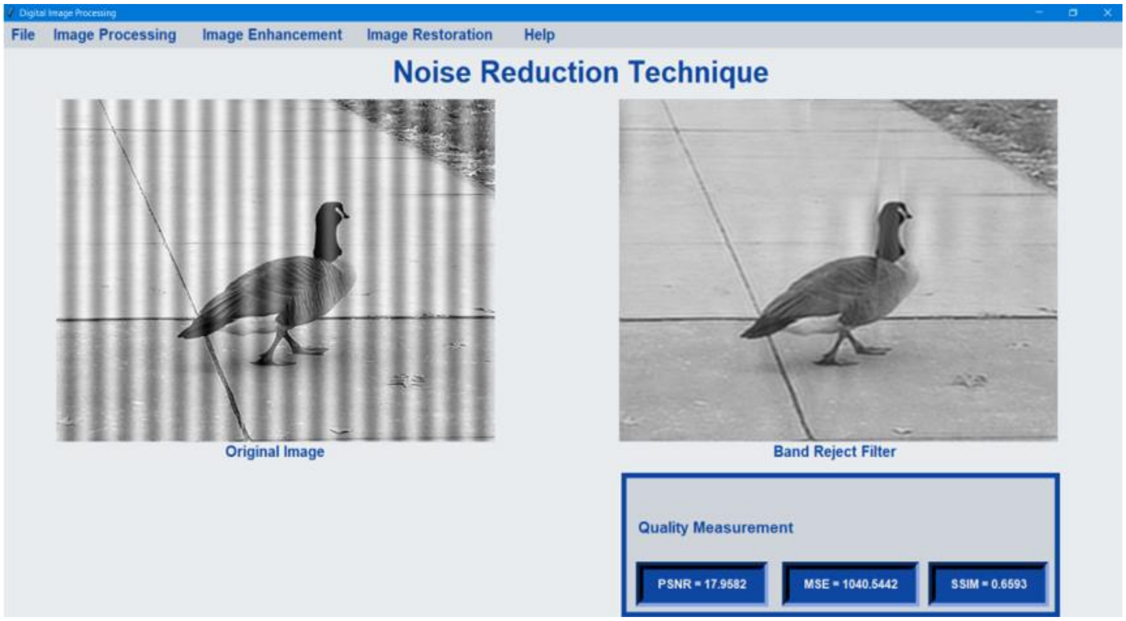


Figure 6.36 band reject

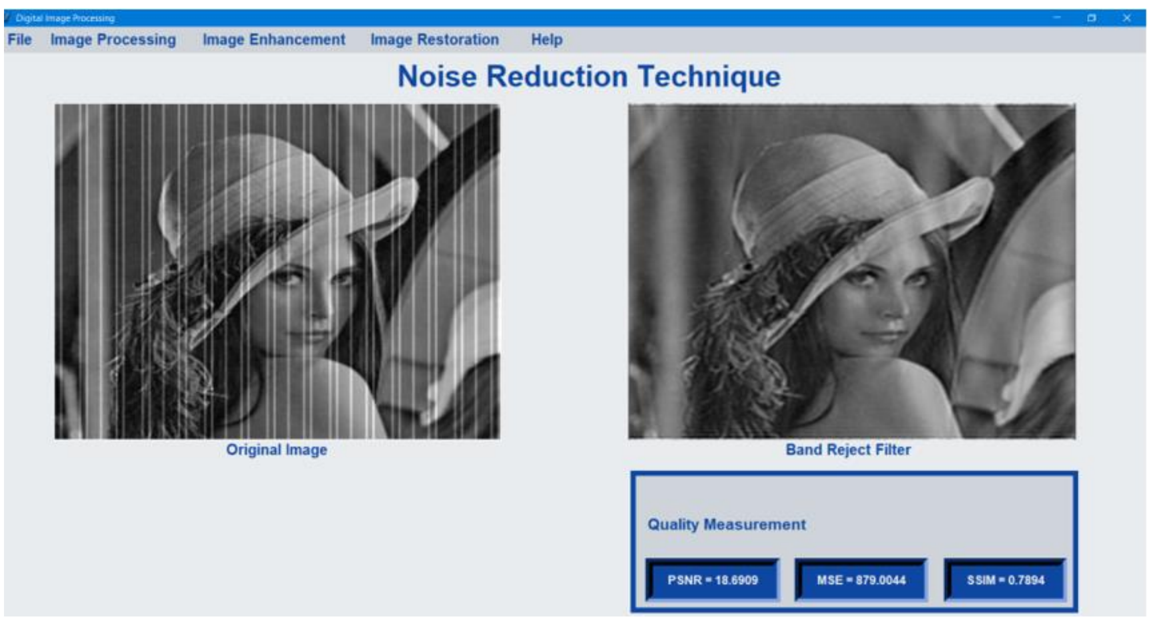


Figure 6.37 band reject

- The value of radial, user should enter the value of it, but we make iteration depend on the value of quality measurement to choose the best value of this image.
- In this image band reject remove periodic noise well from the image and make it clear.

Chapter 6
System Testing

System Testing:

Digital image processing projects involve the development of algorithms and techniques to enhance or analyze images.

In this chapter, we will discuss the testing of algorithms of all parts for proposed system are represented. Showing the experiment result by all training options.

6.1 Test Case

1. User tries to select image

Test case	Scenario	Test step	Expected result	Actual outcome
TC-001	User tries to select image.	Users open the program & selects image	When user select image, it should take him to choose whether, operation or enhancement or restoration.	The selected option is opened successfully

2. User choose Image Processing

Test case	Scenario	Test step	Expected result	Actual outcome
TC-002	User tries to edit the image.	Users select image operation	When user select operation, it should take him to choose whether, crop or rotate or mirror or zoom or pixelate or grayscale or add noise	the selected image is successfully edited.

3. User choose Image Enhancement

Test case	Scenario	Test step	Expected result	Actual outcome
TC-003	User tries to choose right algorithm to enhance the image.	Users select image enhancement algorithm	When user select image, it should take him to choose whether, negative or log or inverse log or sharpness or edge detection or contrast.	the selected image is successfully enhanced.

4. User choose Image Restoration

Test case	Scenario	Test step	Expected result	Actual outcome
TC-004	User tries to choose right algorithm to restore the image.	User selects image Restoration algorithm.	When user select image, it should take him to choose whether, average or median or non-local or wiener or regularization or radial.	the selected image is successfully restored.

5. User save Image

Test case	Scenario	Test step	Expected result	Actual outcome
TC-005	User tries to save the edited image	Users selects File	After the selected image had been edited, it should take him to save it.	the selected image is saved successfully.

6.2 Datasets:

6.2.1 Noisy Image Dataset

Name	Blur
Content	This dataset contains a total of 14,001 images. The images are in JPEG format & have a resolution of 256 X 256 pixels.
Description	The dataset is a collection of 9 classes of natural images that have been degraded by adding different types of noise and distortions. The images in the dataset were created by adding different types of noise, such as: 1- erlang noise. 2- exponential noise. 3- Gaussian noise. 4- lognormal noise. 5- Poisson noise. 6- Rayleigh noise. 7- salt and pepper noise. 8- speckle noise. 9- uniform noise.

Table 6.1 noise dataset

6.2.2 Salt and Pepper Noise Dataset

Name	Salt and Pepper Noise
Content	This dataset is a collection of 500 images in JPEG format, each of size 256 x 256 pixels
Description	The images have been artificially corrupted with salt and pepper noise. The dataset includes images with different levels of noise, ranging from 10% to 90%. The images in the dataset are diverse and cover a range of subjects and scenes, including people, animals, landscapes, and objects.

Table 6.2 salt & pepper dataset

6.3 Algorithms Test and Comparison

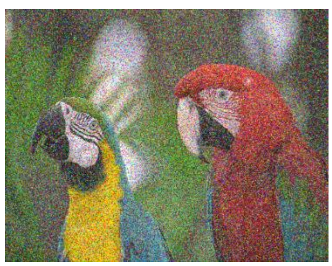
6.3.2 Median Filter:

- Salt & Pepper Noise

Noisy image

image 1

Quality measurement	Value
PSNR	26.8
MSE	17429.603
SSIM	0.929



Noisy image

image 2

Quality measurement	Value
PSNR	27.09
MSE	18112.90
SSIM	0.93

- ❑ Median Filter Is High effect In Noise Reduction(**Salt & pepper Noise**)

6.3.3 Average Filter:

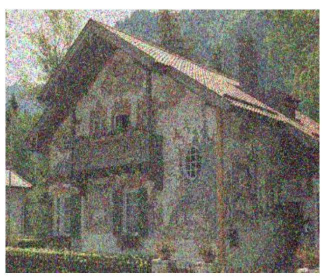
- **Grayscale Image with Salt and Pepper Noise**

Noisy image

image 1

Quality measurement	Value
PSNR	25.5
MSE	2245.89
SSIM	0.968

➢ **RGB Image with Salt and Pepper Noise**



Noisy image

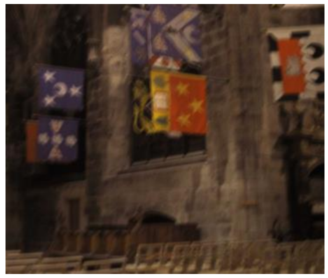
image 2

Quality measurement	Value
PSNR	23.47
MSE	16589.91
SSIM	0.94

❑ Average Filter is Moderate Effect in Noise Reduction (Salt & pepper Noise)

6.3.4 Inverse Filter:

- RGB Image with Motion Blur Noise

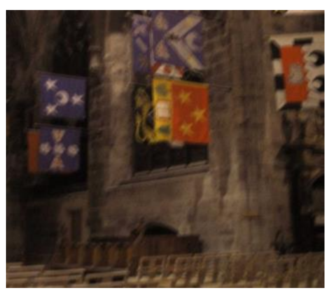


Blurry image

image 1

Quality measurement	Value
PSNR	23.54
MSE	2996.97
SSIM	0.96

- RGB Image with Motion & Gaussian Noise



Noisy image

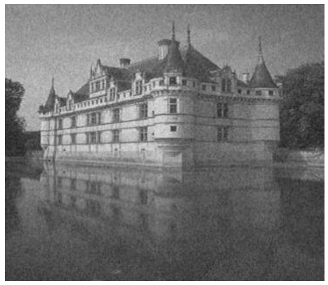
Image 2

Quality measurement	Value
PSNR	23.17
MSE	17240.10
SSIM	0.709

- ❑ Inverse Filter Is Effect Blurring image when extract the Degradation Function is true.
- ❑ Inverse Filter Is Not Effective with noise image.

6.3.5 Non-Local-mean Filter:

- **Greyscale Image with Gaussian Noise**

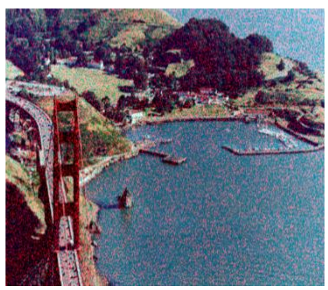


Noisy image

Image1

Quality measurement	Value
PSNR	25.76
MSE	355.604
SSIM	0.998

- **RGB Image with Gaussian Noise**



Noisy image

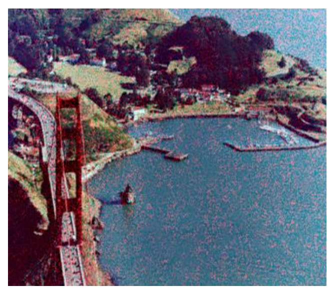


image 1

Quality measurement	Value
PSNR	24.94
MSE	3984.27
SSIM	0.985

- Non-Local Means Filter Is High Effect in Noise Reduction (Gaussian Noise)

6.3.6 Regularization Filter:

- **RGB Image**

Blurry image

image 1

Quality measurement	Value
PSNR	24.05
MSE	3099.76
SSIM	0.98

Blurry image

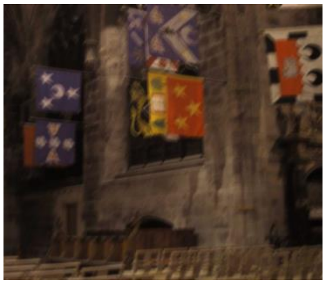
image 2

Quality measurement	Value
PSNR	23.8
MSE	4011.17
SSIM	0.98

- Regularization Filter Effect with Blurring & noise image when extract the Degradation Function is true.

6.3.7 Radial Inverse Filter:

- **RGB Image with Motion Blur Noise**

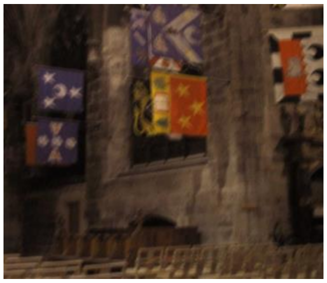


Blurry image

image 1

Quality measurement	Value
PSNR	24.86
MSE	1532.28
SSIM	0.979

- **RGB Image with Motion & Gaussian Noise**



Blurry image

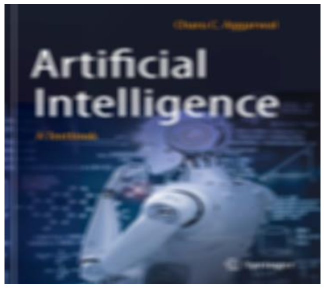
image 2

Quality measurement	Value
PSNR	24.54
MSE	1623.97
SSIM	0.977

- Radial Inverse Filter Is Effect with Blurring & noise image when extract the Degradation Function is true.

6.3.8 Wiener Filter

➢ **Degraded RGB Image Blurring**



Blurry image

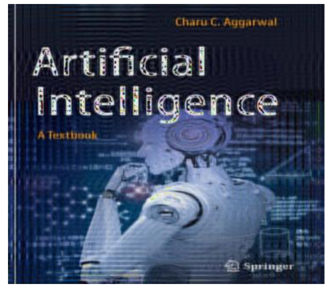
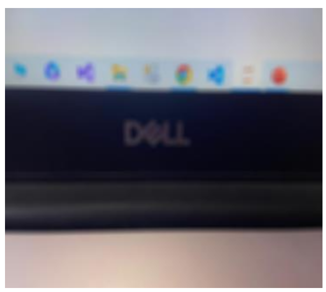


image 1

Quality measurement	**Value**
PSNR	24.47
MSE	5319.14
SSIM	0.96



Blurry image

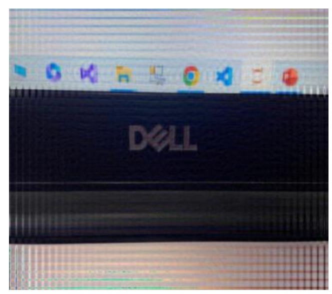


image 2

Quality measurement	**Value**
PSNR	25.33
MSE	2090.52
SSIM	0.99

❑ Wiener Filter Effect with Blurring & noise image when extract the Degradation Function is true.

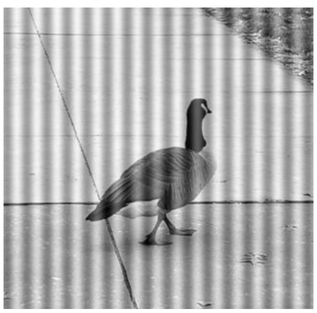
6.3.9 Band Reject Filter

- **Grayscale Image**

Noisy image

image 1

Quality measurement	Value
PSNR	18.6909
MSE	879.0044
SSIM	0.7894



Noisy image

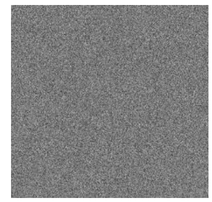
image 2

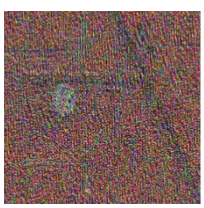
Quality measurement	Value
PSNR	17.9582
MSE	1040.5442
SSIM	0.6593

- Band Reject Filter Effect with periodic noise

6.4 Algorithms Comparison

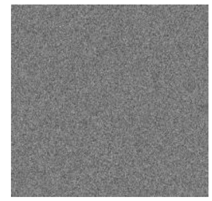
6.4.2 Experiment 1 (Noise Reduction [Salt & Pepper])

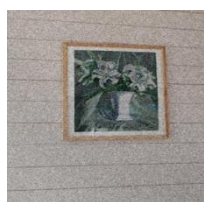
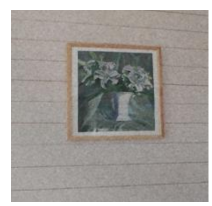
Degraded Image	Wiener	Regularization	Inverse
			
PSNR	23.13	**23.22**	23.12
MSE	25993.88	19908.30	37630.21
SSIM	0.59	0.91	0.782

Radial	Median	Average	Non-Local Means
			
23.16	**26.15**	23.32	23.34
320823.23	19484.82	17419.09	15032.64
0.86	0.877	0.92	0.93

- Median Filter is best Result in PSNR [26.15]

6.4.3 Experiment 2 (Noise Reduction Gaussian Noise)

Degraded Image	**Wiener**	**Regularization**	**Inverse**
			
PSNR	23.06	**23.08**	23.01
MSE	15867.55	14895.91	33357.02
SSIM	0.947	0.97	0.91

Radial	**Median**	**Average**	**Non-Local Means**
			
23.03	23.11	23.04	**23.24**
14571.97	13645.75	13470.09	3804.13
0.972	0.974	0.974	0.992

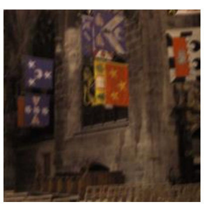
- Non-Local Means Filter is best Result in PSNR [23.24]

6.4.4 Experiment 3 (Blurring)

Degraded Image	**Wiener**	**Regularization**	**Inverse**	**Radial**
				
PSNR	27.6	**28.1**	25.42	26.2
MSE	7529.9	7081.88	4912.19	6943.4
SSIM	0.93	0.94	0.96	0.94

- Regularization Filter is best Result in PSNR [28.1]

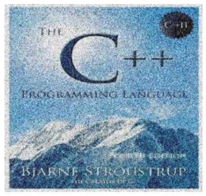
6.4.5 Experiment 4 (Blurring + Gaussian Noise)

Degraded Image	**Wiener**	**Regularization**	**Inverse**	**Radial**
				
PSNR	23.3	**23.7**	23.17	23.5
MSE	4441.57	2369.06	17240.10	2680
SSIM	0.939	0.968	0.709	0.964

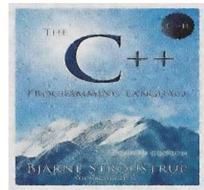
- Regularization Filter is best Result in PSNR [23.7]

6.5 Enhancement and Restoration

6.5.2 Experiment 1 (Gaussian Noise)


Degraded Image


Non-Local Means


Sharpness

- Apply Non-Local Means Filter (Restoration) to reduce Gaussian Noise Then
- Apply Sharpness (Enhancement) to improve quality of image.

6.5.3 Experiment 2 (Blurring)

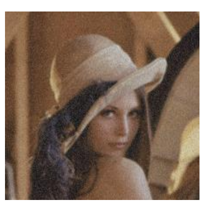

Degraded Image

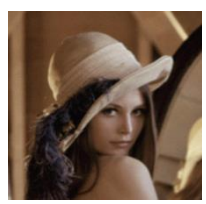

Regularization


Contrast

- Apply Regularization Filter (Restoration) to reduce Blurring Then Apply Contrast (Enhancement) to improve quality of image.

6.5.4 Experiment 3 (Salt & Pepper Noise + Blurring)


Degraded Image


Median


Regularization

Algorithm	Regularization	Regularization
PSNR	23.89	23.3
MSE	2256.57	10029.55

6.6 System Usability

6.6.1 Program Testing

The purpose of testing is to find errors in programmers' systems so that they can be fixed. Development in any system takes only 60-65% of the time, leaving the rest for testing.

System testing involves operating the system with derived test cases. When testing a program, it is essential to try and simulate everything that the user may try to do with the program. This should include both legitimate actions and actions that aren't necessarily in accordance with the program's procedures.

6.6.2 User Testing

Following user testing, it is critical to understand the users' perceptions of the system considering the project's context and goals.

A questionnaire is created for user testing to determine the user's opinion of the program. It is briefed on the project's history and goals. and asked ten users to fill out a table rating the various aspects of the project out of five. The table and the questions I asked them to consider are included below.

Score	Bad	Average	Good	Very Good	Excellent
User Interface					
Design					
Functionality					
Usefulness					
Repeat Usage					

- **User Interface:** How simple the interface is for the user to use and navigate.
- **Design:** The user's perception of the GUI's aesthetics and lag.
- **Functionality:** How well the program works and whether it meets the requirements.
- **Usefulness:** Does the user believe this program will be useful to them?
- **Repeat Usage:** Would the user use this program again if they had the choice?

- **User 1**

Score	Bad	Average	Good	Very Good	Excellent
User Interface			√		
Design			√		
Functionality				√	
Usefulness				√	
Repeat Usage				√	

- **User 2**

Score	Bad	Average	Good	Very Good	Excellent
User Interface			√		
Design				√	
Functionality				√	
Usefulness			√		
Repeat Usage			√		

- **User 3**

Score	Bad	Average	Good	Very Good	Excellent
User Interface				√	
Design			√		
Functionality				√	
Usefulness			√		
Repeat Usage				√	

- **User 4**

Score	Bad	Average	Good	Very Good	Excellent
User Interface		✓			
Design				✓	
Functionality					✓
Usefulness		✓			
Repeat Usage					✓

- **User 5**

Score	Bad	Average	Good	Very Good	Excellent
User Interface				✓	
Design					✓
Functionality					✓
Usefulness		✓			
Repeat Usage					✓

- **User 6**

Score	Bad	Average	Good	Very Good	Excellent
User Interface				✓	
Design					✓
Functionality			✓		
Usefulness			✓		
Repeat Usage				✓	

- **User 7**

Score	Bad	Average	Good	Very Good	Excellent
User Interface		√			
Design					√
Functionality					√
Usefulness				√	
Repeat Usage					√

- **User 8**

Score	Bad	Average	Good	Very Good	Excellent
User Interface			√		
Design				√	
Functionality				√	
Usefulness				√	
Repeat Usage		√			

6.6.3 User Evaluation

The evaluation is performed to determine whether or not the system is qualified, as well as how users interact with it.

User testing offers an intriguing glimpse into what people think of the program.

The functionality obviously did not receive high marks, as there is little visible evidence of the program producing any results. This received the lowest rating because the program did not accomplish as much as people expected.

The user interface appeared to be straightforward and simple to use. and everyone seemed to know which button did what.

Averages of rating the program is as follows:

Score	Bad	Average	Good	Very Good	Excellent
User Interface			4.2		
Design				4.1	
Functionality				4.2	
Usefulness			3		
Repeat Usage				4	

Comment on the previous results' correction and enhancement procedures:

- **User interface**
  From the system's user interface characteristics, it is clear, and the user can navigate easily without problems, and the user interface can be improved, but it is already very good, which is nice.
- **Design**
  The design is also very good, but it could be improved by changing the colors to make it more appealing to most users.
- **Usefulness**
  The utility is excellent, as it can significantly improve the quality of degraded or damaged images.
- **Functionality**
  The program's functionality is excellent, and it meets the requirements for a successful system.
- **Repeat Usage**
- Repeat usage is average because the user has already seen and dealt with the system, so the next time the user will not be as interested in using the system.

Chapter Seven
Conclusion & Future work

7.1 Conclusion

In real life, you don't know what the psf is. You guess the psf based on camera parameter or other system characteristics. It can be found even through experiments. However, the psf is usually a model, it's very complex to determine an exact psf.

Noise in image is commonly modeled as Gaussian. The noise to image power ratio is a good measure of how much noise is in an image. This information, along with the noise power, can be used in the inverse filtering process to realize a much less degraded image than expected. The more information you add to the inverse filtering process, the better the output.

Autocorrelation matrices are often used in inverse filtering. They come from noise and blur information about the image. This information expressed as a correlation is a convenient way to implement into an inverse filter.

With no noise information. The wiener and regularization filters do a poor job at realizing the original, non-degraded, image. However, the

With noisy information. The wiener and regularization filters do a great job at restoring the image. However, the wiener filter is much better at blur than the regularized filter.

7.2 Future work

- There are many types of image restoration that can be used.
- Also, this project works on optical camera, image restoration can also work with astronomy and radar.

References

[1] Rafael C. Gonzalez and Richard E. Woods “Digital Image Processing”

[2] Moumena Salah YASSEN1 “COMPARE OF MEDIAN FILTER AND WIENER FILTER TO DETECT CONCEALED WEAPONS”, June 2022.

[3] Rev. Téc. Ing. Univ. Zulia. Vol. 39 “ISNR measure on deblurred blind images using Weiner filter”, 2016.

[4] Kaushick Parui “Image Restoration and de-blurring using Inverse and Wiener filtering”, Jan 2018.

[5] Shyama P“Image restoration ”, Jan, 2018.

[6] Ammar M. Alhosainy and Ehab F. Badran “Adapted Non-Local Means Filter using Variable Window Size”,2010.

[7] Hore, Alain, and Djemel Ziou. "Image quality metrics: PSNR vs. SSIM." Pattern recognition (icpr), 2010 20th international conference on. IEEE, 2010.

[8] Rusul H. Al_taie1 and Hawraa Abd Al_kadum Hassan. “Enhance Mobile Phone Digital Camera Images Using Band Reject Filter and Discrete Fourier Transform”,2023.

[9] Alamuru Susmitha, Ishani Mishra and Dr. Sanjay Jain3 “Performance Analysis of Average and Median Filters for De noising Of Digital Images.”, Nov 2016