Assignment No. 2

Problem Statement: Read an image, plot its histogram then do histogram equalization. Comment about the result. Implement various spatial domain and frequency domain filters.

Aim: To read a grayscale image, plot its histogram, perform histogram equalization for contrast enhancement, and implement various spatial domain and frequency domain filters for image enhancement and analysis.

Objective:

1. To study the concept of image histogram and its importance in image processing.
2. To apply histogram equalization for contrast improvement.
3. To implement spatial domain filters (smoothing and sharpening) to enhance image details.
4. To implement frequency domain filters (low-pass and high-pass) using Fourier Transform.
5. To analyze and compare the results of spatial and frequency domain filtering.

Outcomes:

1. Understand histogram equalization and its effect on image contrast.
2. Gain practical experience with spatial domain filtering techniques.
3. Implement frequency domain filters using Fourier Transform.
4. Analyze and compare the impact of different image enhancement techniques.

Theory

1. Histogram & Histogram Equalization

- Image Histogram: A graph representing the distribution of pixel intensities in an image.
- Histogram Equalization: A technique to spread out pixel intensities more evenly, improving image contrast.

  Formula:

$$s_k = round\left((L-1) \sum_{j=0}^{k} p_r(r_j) \right)$$

where,

s_k = output intensity level

L = total number of intensity levels

$P_r(r_j)$ = probability of intensity level r_j

2. Spatial Domain Filtering

- Operates directly on pixels.
- A filter (mask/kernel) is convolved with the image.

Formula:

$$g(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) \, . \, f(x-s, y-t)$$

where,

f(x,y) = input image

g(x,y) = output image

w(s,t) = filter mask / kernel

Examples:

- Gaussian Blur (Smoothing) → reduces noise.
- Sharpening (Laplacian/High-boost) → enhances edges.

3. Frequency Domain Filtering

- Based on Fourier Transform, where the image is represented in terms of frequency components.
- High frequencies → edges, noise, fine details.
- Low frequencies → smooth background, general shape.

Discrete Fourier Transform Formula:

$$X(k) = \sum_{n=0}^{N-1} x(n)\, e^{-j2\Pi kn/N}$$

Inverse Discrete Fourier Transform:

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k)\, e^{j2\Pi nk/N}$$

- **Low-pass filter** → retains low frequencies, removes high frequencies → smoothing.
- **High-pass filter** → retains high frequencies, removes low frequencies → edge enhancement.

Pseudocode:

Algorithm ImageEnhancement

Input: Grayscale image f(x,y)
Output: Enhanced images using histogram equalization, spatial filters, and frequency domain filters

Step 1: Read the grayscale image
  f ← ReadImage("input_image.jpg")

Step 2: Plot original histogram
  hist_f ← ComputeHistogram(f)
  Plot(hist_f, title="Original Histogram")

Step 3: Perform histogram equalization
  f_eq ← HistogramEqualization(f)
  hist_eq ← ComputeHistogram(f_eq)
  Plot(hist_eq, title="Equalized Histogram")

Step 4: Comment on results
Print("Histogram equalization redistributes pixel intensities to enhance contrast")

Step 5: Implement Spatial Domain Filtering
// Smoothing filters
f_mean ← ApplyFilter(f, MeanFilter(3x3)) // Average filter
f_weighted ← ApplyFilter(f, WeightedAverageFilter()) // e.g., Gaussian weights
f_median ← ApplyFilter(f, MedianFilter(3x3)) // Non-linear, removes salt & pepper noise
f_gaussian ← ApplyFilter(f, GaussianFilter(σ=1.0)) // Smooths while preserving edges better than mean

// Display results
Display(f_mean, "Mean Filtered Image")
Display(f_weighted, "Weighted Average Filtered Image")
Display(f_median, "Median Filtered Image")
Display(f_gaussian, "Gaussian Filtered Image")

Step 6: Implement Frequency Domain Filtering
// Compute Fourier Transform
F ← DFT(f)

// Apply frequency domain filters
F_lowpass ← ApplyLowPassFilter(F, cutoff_radius)
F_highpass ← ApplyHighPassFilter(F, cutoff_radius)

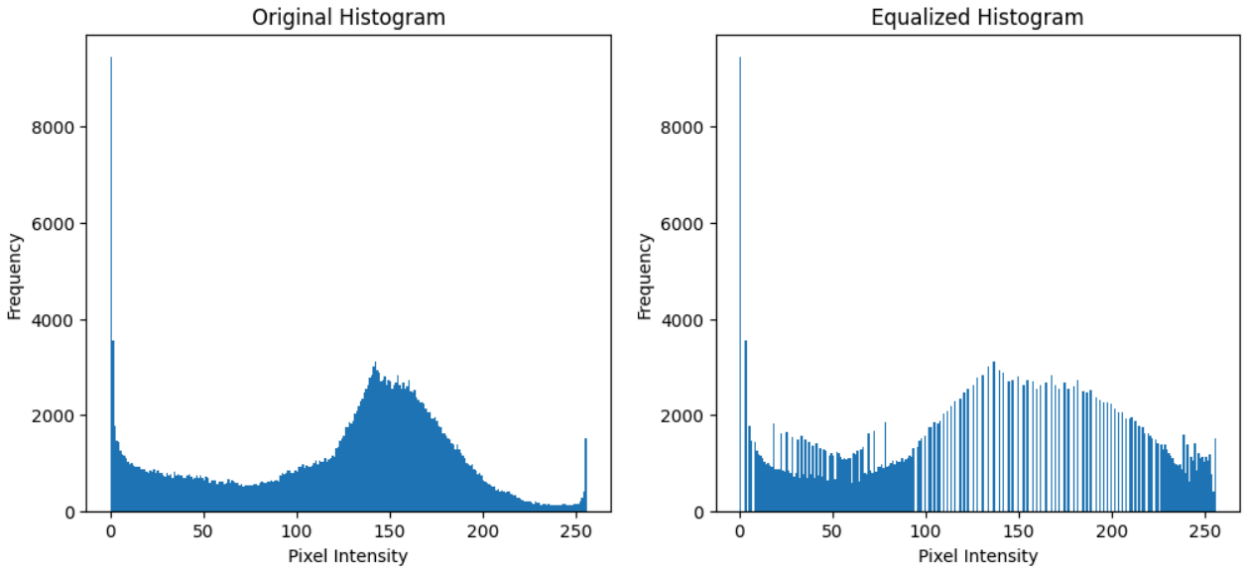
// Inverse transform to get spatial images
f_lowpass ← IDFT(F_lowpass)
f_highpass ← IDFT(F_highpass)

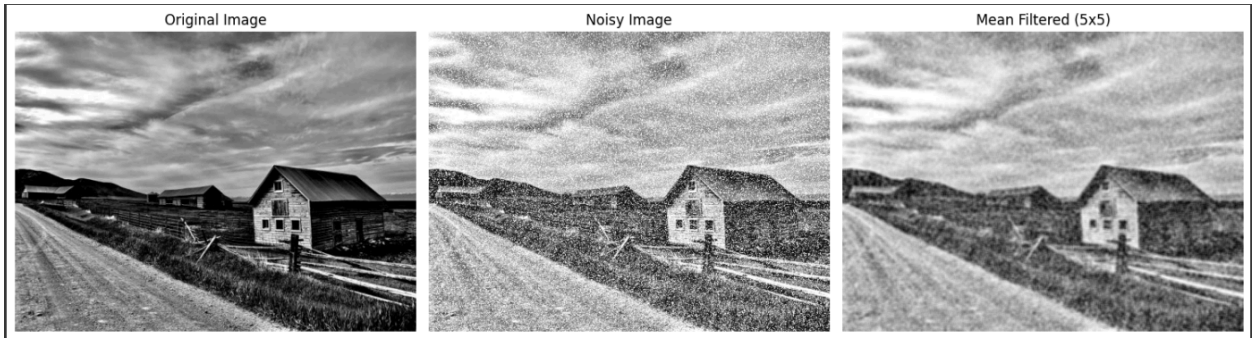
Display(f_lowpass, "Low-Pass Filtered Image")
Display(f_highpass, "High-Pass Filtered Image")

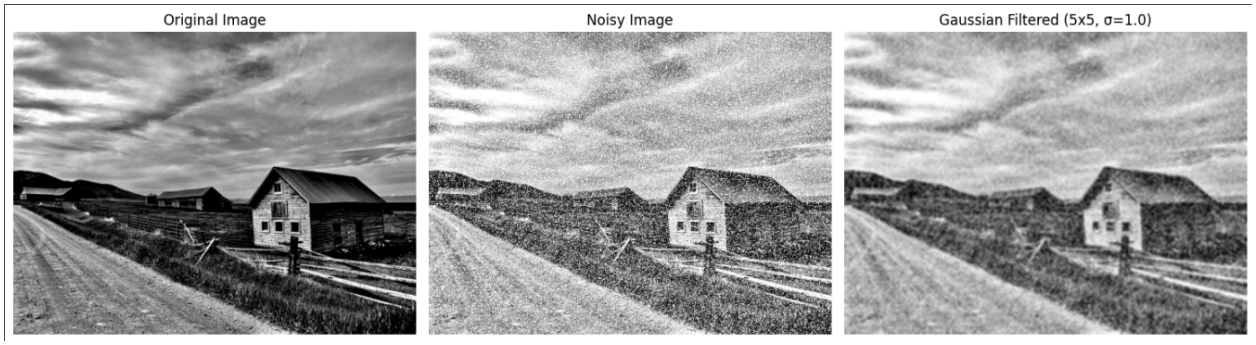
Step 7: Comment on overall results
Print("Spatial filters modify local pixel neighborhoods, useful for smoothing, sharpening, and edge detection.")
Print("Frequency domain filters selectively enhance or suppress certain frequency components.")
Print("Histogram equalization enhances overall contrast by redistributing intensity values.")

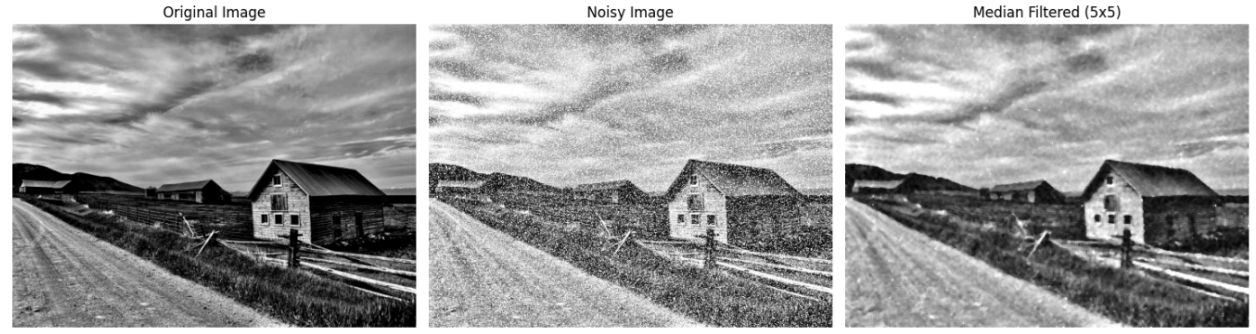
End Algorithm

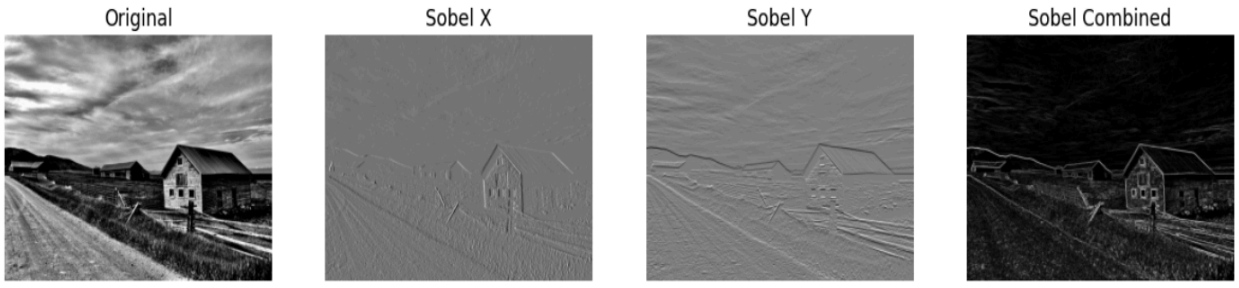
Result:




Original Image
Noisy Image
Mean Filtered (5x5)


Original Image
Noisy Image
Gaussian Filtered (5x5, σ=1.0)


Original Image
Noisy Image
Median Filtered (5x5)


Original
Sobel X
Sobel Y
Sobel Combined

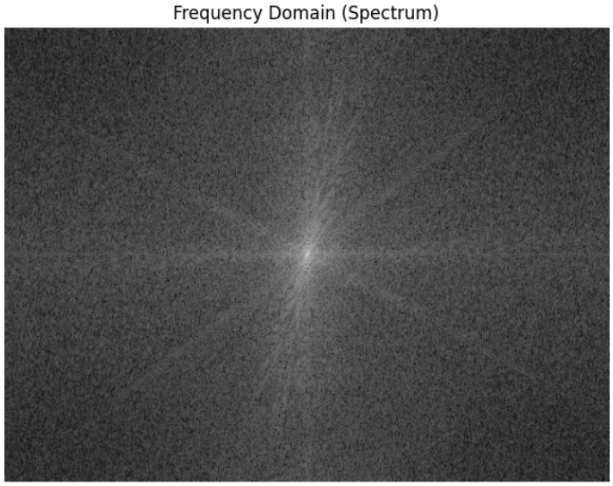
Original

Laplacian (Edges)

Sharpened Image

Original Image

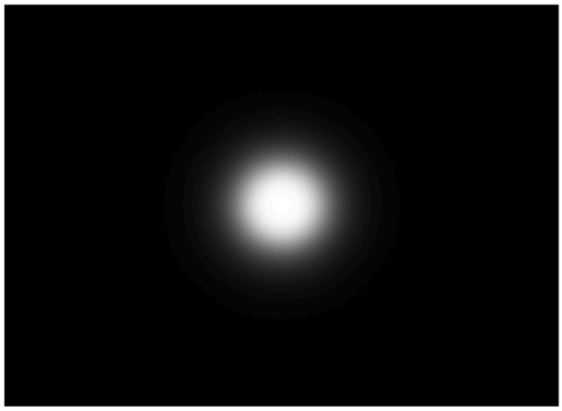
Frequency Domain (Spectrum)



Ideal Low-pass Mask

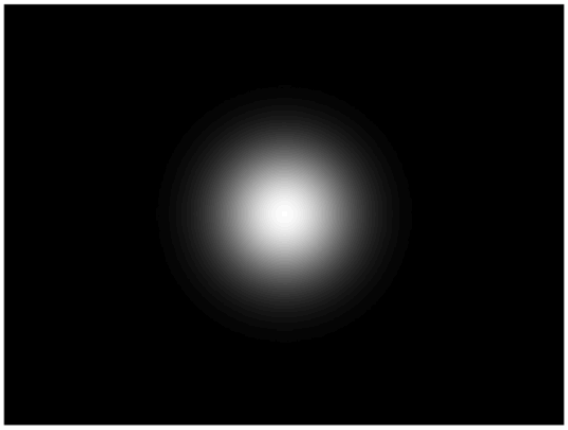
Ideal Low-pass Result

Butterworth LP Mask



Butterworth LP Result

Gaussian LP Mask



Gaussian LP Result

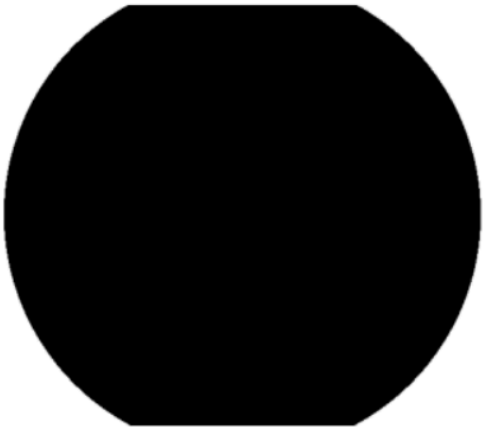
Ideal High-pass Mask

Ideal High-pass Result

Butterworth HP Mask

Butterworth HP Result

Gaussian HP Mask



Gaussian HP Result

Band-Pass Mask

Band-Pass Result

Conclusion:

Histogram equalization improves image contrast by redistributing pixel intensities, making hidden details clearer. Spatial domain filters help in noise removal and edge enhancement, while

frequency domain filters are effective for global transformations and periodic noise reduction. Each method has its own importance, and together they provide powerful tools for image enhancement and preprocessing in various applications like medical imaging and computer vision.

Github Link:

https://github.com/rachanadixit/FDIP/blob/main/123B5F138_FDIP_Assignment2.ipynb