Experiment Number: 5

NAME: **Kartik Banshi Katkar** ROLLNO: 36

CLASS: TY IT A BATCH: 1

DATE OF PERFORMANCE: 15/08/2023

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Question:  
  
Write Matlab code to design Mean Filter, Weighted Mean Filter, Median Filter, High Pass Filter, High Boost Filter.**

**Code:**

%low pass filter

clc;

clear all;

close all;

img=imread('grayscale.jpg');

a=imnoise(img,'salt & pepper',0.30);

subplot(1,2,1);

imshow(a);

title('Noisy image');

% masking the image

[n,m,l]=size(img);

mask=(1/9)\*[[1,1,1];[1,1,1];[1,1,1]];

img=padarray(img,[1,1]);

for i=2:(n+1)

for j=2:(m+1)

val=0;

for k=1:3

for l=1:3

val=val+mask(k,1)\*img(i-2+k,j-2+l);

end

end

img\_f(i-1,j-1)=val;

end

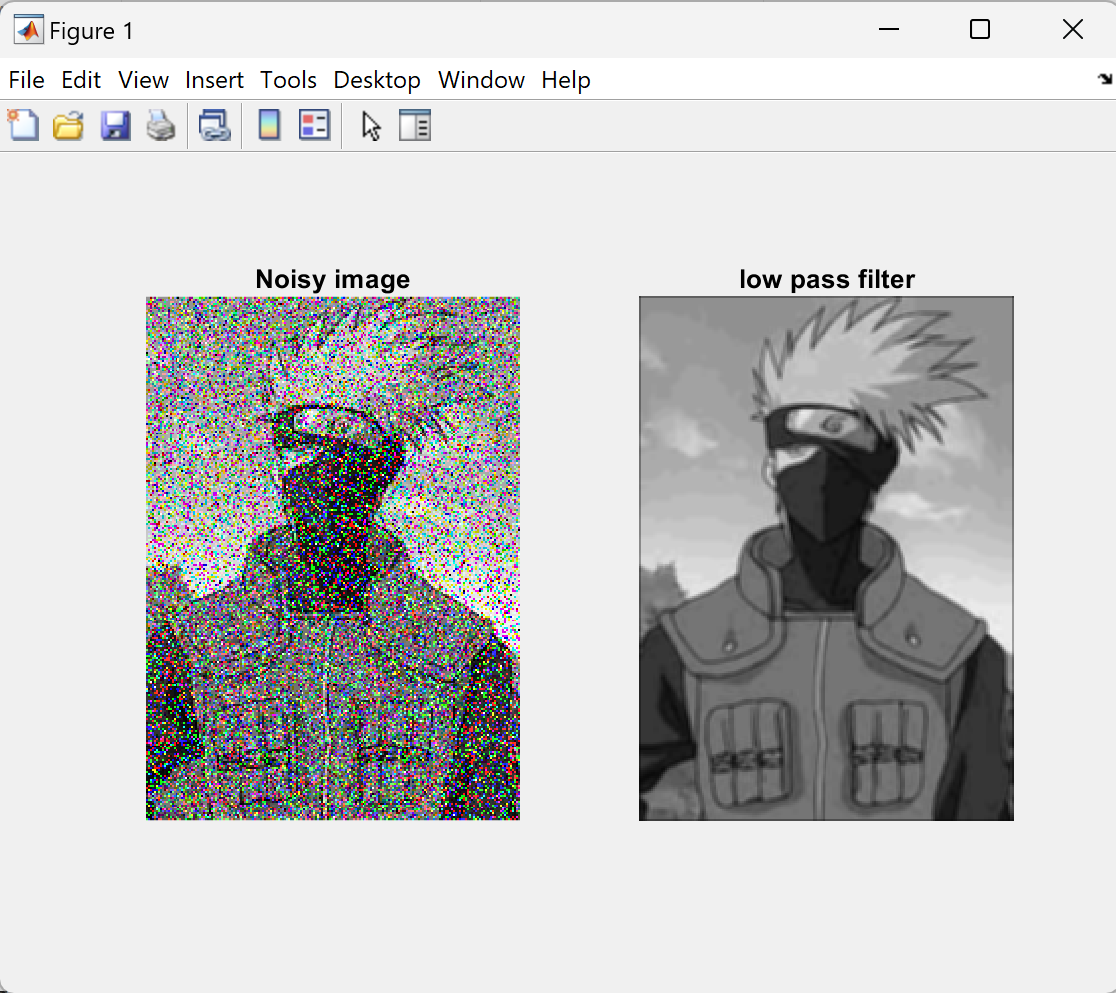
end

subplot(1,2,2);

imshow(img\_f);

title('low pass filter');

**Input:**

**Output:**

**Code:**%weighted low pass filter

clc;

clear all;

close all;

img=imread('grayscale.jpg');

a=imnoise(img,'salt & pepper',0.04);

subplot(1,2,1);

imshow(a);

title('Noisy image');

% weighted masking of the image

[n,m,l]=size(img);

mask=(1/16)\*[[1,2,1];[2,4,2];[1,2,1]];

img=padarray(img,[1,1]);

for i=2:(n+1)

for j=2:(m+1)

val=0;

for k=1:3

for l=1:3

val=val+mask(k,1)\*img(i-2+k,j-2+l);

end

end

img\_f(i-1,j-1)=val;

end

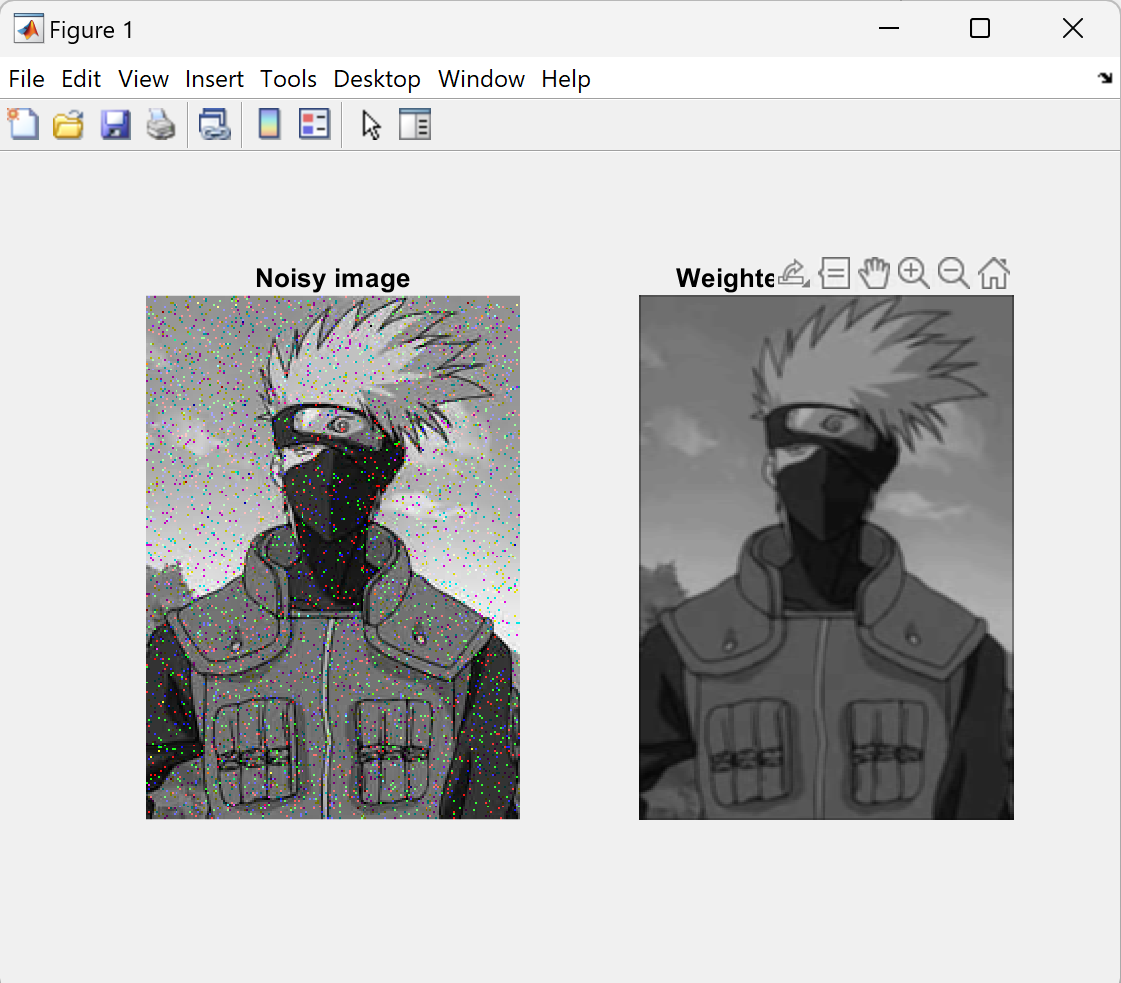
end

subplot(1,2,2);

imshow(img\_f);

title('Weighted low pass filter');

**Input:**

**Output:**

**Code:**% Median filterclc;

clear all;

close all;

img = imread('grayscale.jpg');

a = imnoise(img, 'salt & pepper', 0.04);

[n, m] = size(img);

window\_size = 3; % Size of the median filter window

img\_f = zeros(n, m); % Initialize filtered image

for i = 2 : n - 1

for j = 2 : m - 1

% Extract the window

window = a(i-1:i+1, j-1:j+1);

% Convert the window into a 1D array and sort it

window\_values = reshape(window, 1, []);

sorted\_values = sort(window\_values);

% Find the median value

median\_index = (window\_size^2 + 1) / 2;

median\_value = sorted\_values(median\_index);

% Assign the median value to the filtered image

img\_f(i, j) = median\_value;

end

end

figure;

subplot(1, 2, 1);

imshow(a);

title('Noisy image');

subplot(1, 2, 2);

imshow(uint8(img\_f));

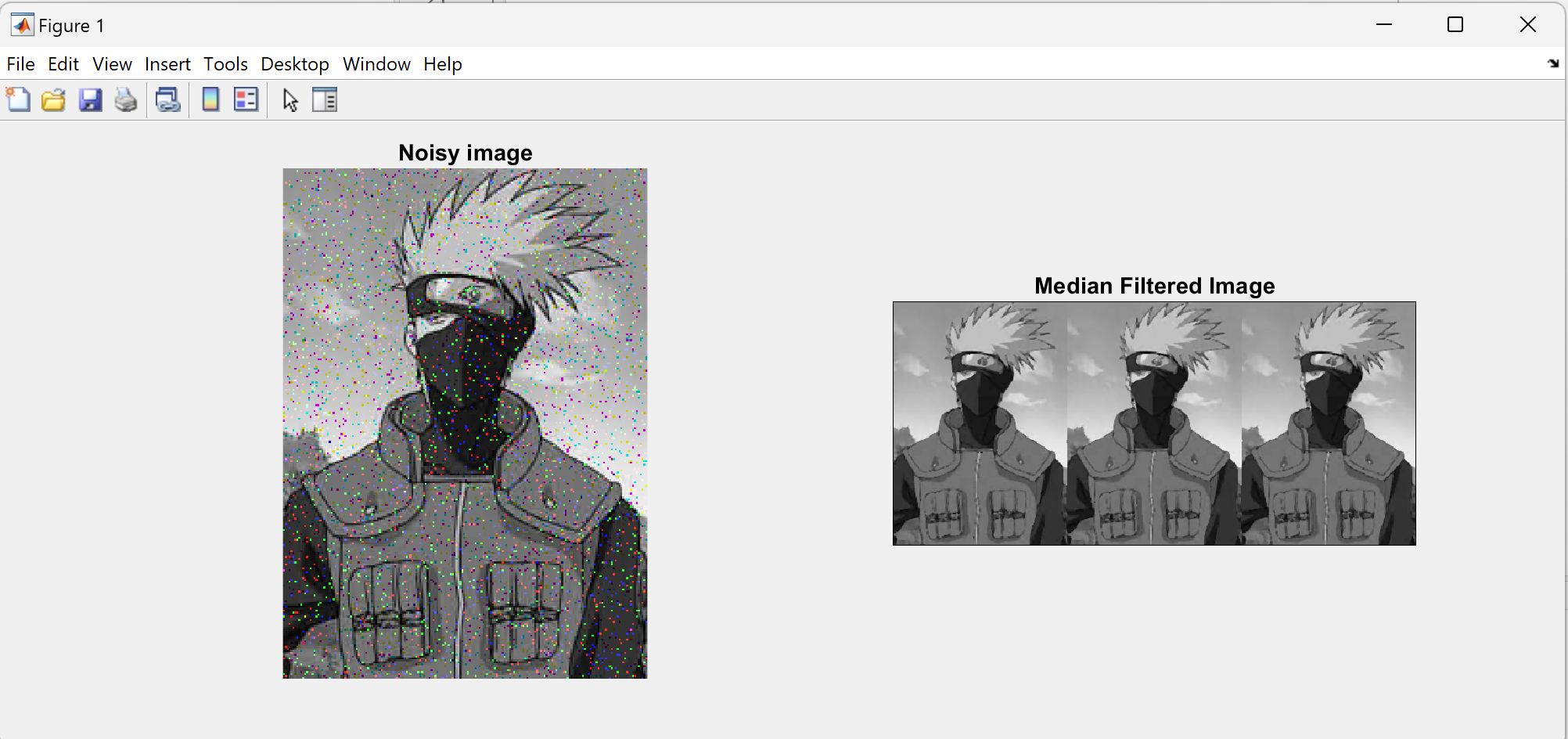
title('Median Filtered Image');

% Show only one figure with both images

set(gcf, 'Position', [100, 100, 1000, 400]); % Adjust figure size

**Input:**

**Output:**



**Code:**% High pass filterclc;

clear all;

close all;

img = imread('grayscale.jpg');

a = imnoise(img, 'salt & pepper', 0.04);

subplot(1, 2, 1);

imshow(a);

title('Noisy image');

[n, m] = size(img);

laplacian\_kernel = [0 -1 0; -1 4 -1; 0 -1 0];

img = double(img); % Convert to double for processing

img\_f = zeros(n, m); % Initialize filtered image

for i = 2 : n - 1

for j = 2 : m - 1

% Extract the window

window = img(i-1:i+1, j-1:j+1);

% Apply the Laplacian kernel to the window

filtered\_value = sum(sum(window .\* laplacian\_kernel));

% Assign the filtered value to the image

img\_f(i, j) = filtered\_value;

end

end

% Normalize the filtered image to 0-255 range for display

img\_f = img\_f - min(img\_f(:));

img\_f = img\_f / max(img\_f(:)) \* 255;

% Display high-pass filtered image in a larger size

figure;

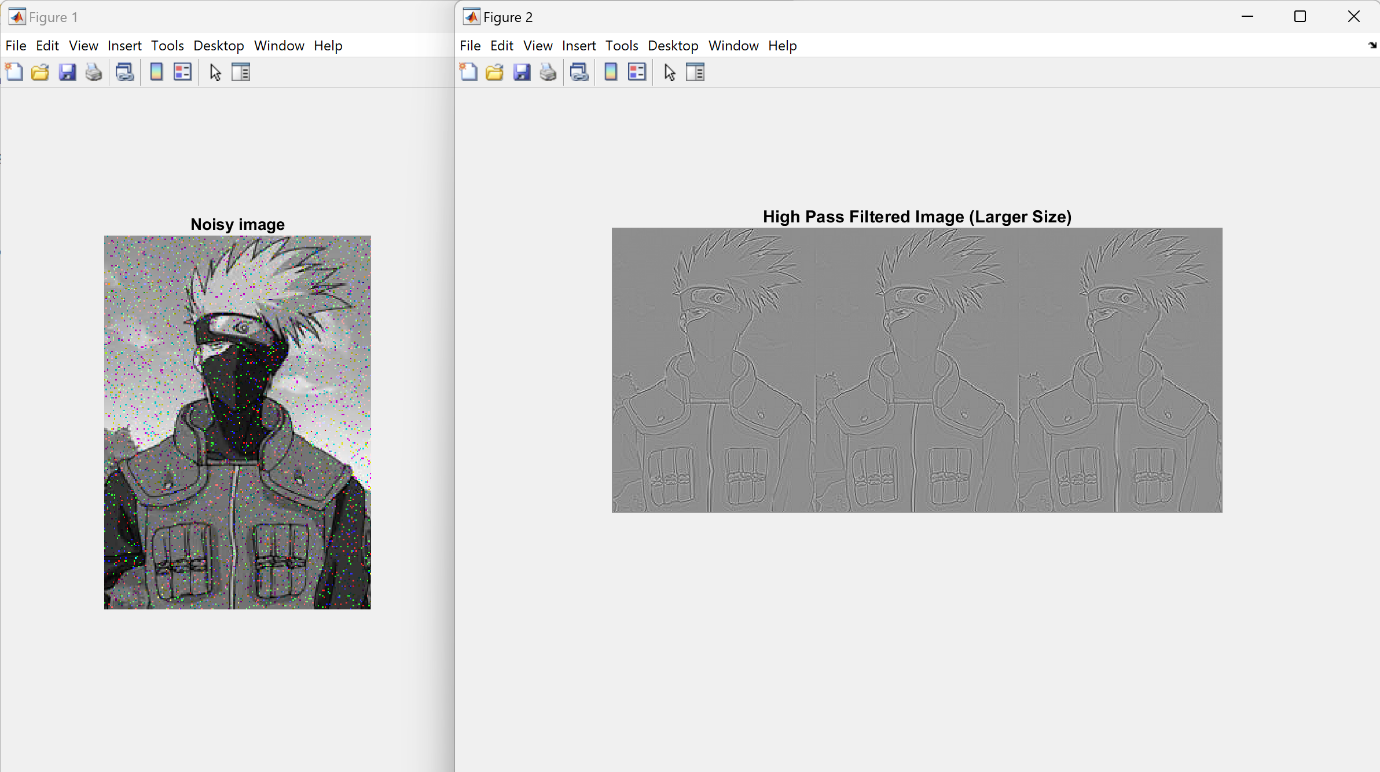
imshow(uint8(img\_f));

title('High Pass Filtered Image (Larger Size)');

set(gcf, 'Position', [100, 100, 800, 600]); % Adjust figure size

**Input:**

**Output:**

 **Code:**% High Boost filterclc;

clear all;

close all;

img = imread('grayscale.jpg');

a = imnoise(img, 'salt & pepper', 0.04);

subplot(1, 2, 1);

imshow(a);

title('Noisy image');

[n, m] = size(img);

laplacian\_kernel = [0 -1 0; -1 4 -1; 0 -1 0];

boost\_factor = 2; % Adjust this factor to control enhancement

img = double(img); % Convert to double for processing

img\_f = zeros(n, m); % Initialize filtered image

for i = 2 : n - 1

for j = 2 : m - 1

% Extract the window

window = img(i-1:i+1, j-1:j+1);

% Apply the Laplacian kernel to the window

filtered\_value = sum(sum(window .\* laplacian\_kernel));

% Calculate the enhanced pixel value

enhanced\_pixel = img(i, j) + boost\_factor \* filtered\_value;

% Ensure the enhanced pixel is within [0, 255] range

img\_f(i, j) = min(max(enhanced\_pixel, 0), 255);

end

end

subplot(1, 2, 2);

imshow(uint8(img\_f));

title('High-Boost Filtered Image');

**input :** **output:**