**Practical – 4**

**Aim:** Frequency Domain Filters

1. **Notch Filter:**

**Code:**

clc

clear all

% read input image using imread() function

inImg = imread('Images\e14.tif');

[m,n] = size(inImg);

% 2d fourier transformation

f = fftshift(fft2(inImg));

% Defining the filter

H = ones(m,n);

H(uint8(m/2),uint8(n/2)) = 0;

% Appling notch filter

nf = H.\*f;

% inverse 2d fourier transformation

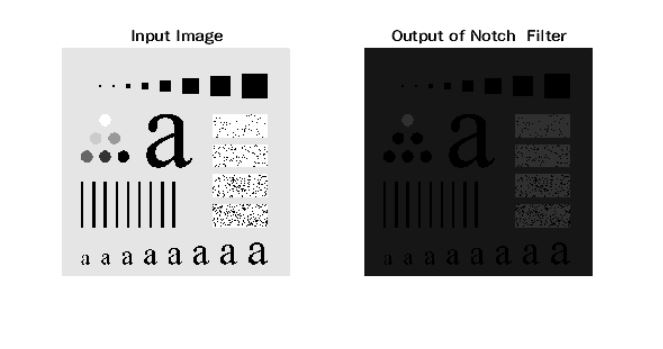
outImg = abs(ifft2(nf));

%Display input and output images in plot

subplot(1,2,1);imshow(inImg);title('Input Image');

subplot(1,2,2);imshow(uint8(outImg));title('Output of notch filter');

**Output:**



**Conclusion:**

By Performing this practical we get to know that in notch filter we force average value of an image to zero.

1. **Ideal low pass Filter:**

**Code:**

clc

clear all

% read input image using imread() function

inImg = imread('Images\e14.tif');

[m,n] = size(inImg);

% set cut-off frequency

cf = 200;

% 2d fourier transformation

f = fftshift(fft2(inImg));

p = round(m/2);

q = round(n/2);

% Defining the filter

H = ones(m,n);

for i = 1:m

for j = 1:n

d = (i-p).^2 + (j-q).^2;

if d <= cf

H(i,j) = 0;

end

end

end

% Appling the ideal low pass filter

ilpf = f.\*H;

% inverse 2d fourier transformation

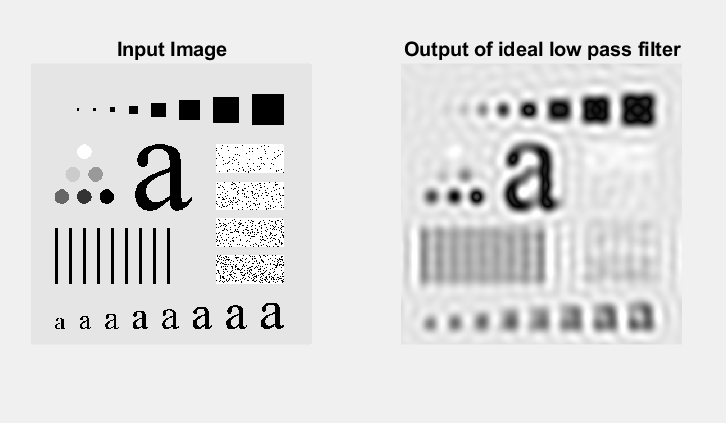
outImg = abs(ifft2(ilpf));

%Display input and output images in plot

subplot(1,2,1);imshow(inImg);title('Input Image');

subplot(1,2,2);imshow(uint8(outImg));title('Output of ideal low pass filter');

**Output:**



**Conclusion:**

By Performing this practical we get to know that in ideal low pass filter we will attenuate all the high frequencies and pass the lower ones.

1. **Butterworth lowpass Filter:**

**Code:**

clc

clear all

% read input image using imread() function

inImg = imread('Images\e14.tif');

[m,n] = size(inImg);

% set cut-off frequency

cf = 50;

% 2d fourier transformation

f = fftshift(fft2(inImg));

N = 1;

p = round(m/2);

q = round(n/2);

% Defining the filter

H = zeros(m,n);

for i = 1:m

for j = 1:n

d = (i-p).^2 + (j-q).^2;

H(i,j) = 1/(1+((d/cf/cf).^(2\*N)));

end

end

% Appling the Butterworth low pass filter

blp = f.\*H;

% inverse 2d fourier transformation

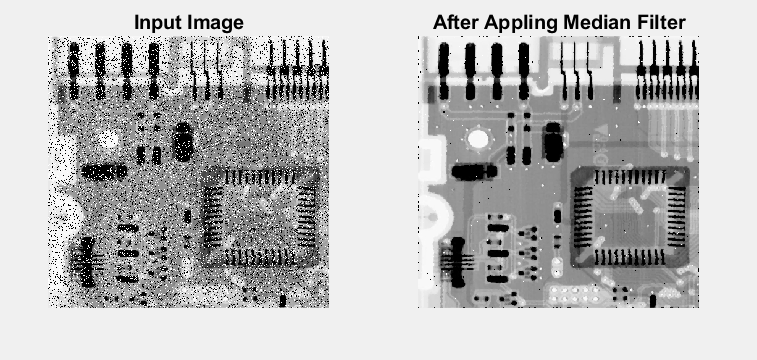
outImg = abs(ifft2(blp));

%Display input and output images in plot

subplot(1,2,1);imshow(inImg);title('Input Image');

subplot(1,2,2);imshow(uint8(outImg));title('Output of Butterworth low pass filter');

**Output:**



**Conclusion:**

By Performing this practical we get to know that in Butterworth low pass filter we will attenuate all the high frequencies and pass the lower ones to get sharpen image.

1. **Gaussian low pass Filter:**

**Code:**

clc

clear all

% read input image using imread() function

inImg = imread('Images\e14.tif');

[m,n] = size(inImg);

% set cut-off frequency

cf = 90;

% 2d fourier transformation

f = fftshift(fft2(inImg));

p = round(m/2);

q = round(n/2);

% Defining the filter

H = zeros(m,n);

for i = 1:m

for j = 1:n

d = (i-p).^2 + (j-q).^2;

H(i,j) = exp(-d/2/cf/cf);

end

end

% Appling the Gaussian low pass filter

glp = f.\*H;

% inverse 2d fourier transformation

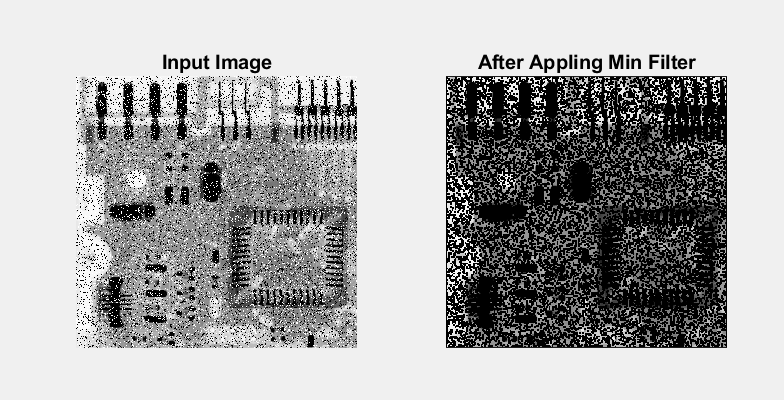
outImg = abs(ifft2(glp));

%Display input and output images in plot

subplot(1,2,1);imshow(inImg);title('Input Image');

subplot(1,2,2);imshow(uint8(outImg));title('Output of Gaussian low pass filter');

**Output:**



**Conclusion:**

By Performing this practical we get to know that in gaussian low pass filter we will attenuate all the high frequencies and pass the lower ones to get sharpen split image.

1. **Ideal high pass Filter:**

**Code:**

clc

clear all

% read input image using imread() function

inImg = imread('Images\e14.tif');

[m,n] = size(inImg);

% set cut-off frequency

cf = 200;

% 2d fourier transformation

f = fftshift(fft2(inImg));

p = round(m/2);

q = round(n/2);

% Defining the filter

H = ones(m,n);

for i = 1:m

for j = 1:n

d = (i-p).^2 + (j-q).^2;

if d <= cf

H(i,j) = 0;

end

end

end

% Appling the ideal high pass filter

ihp = f.\*H;

% inverse 2d fourier transformation

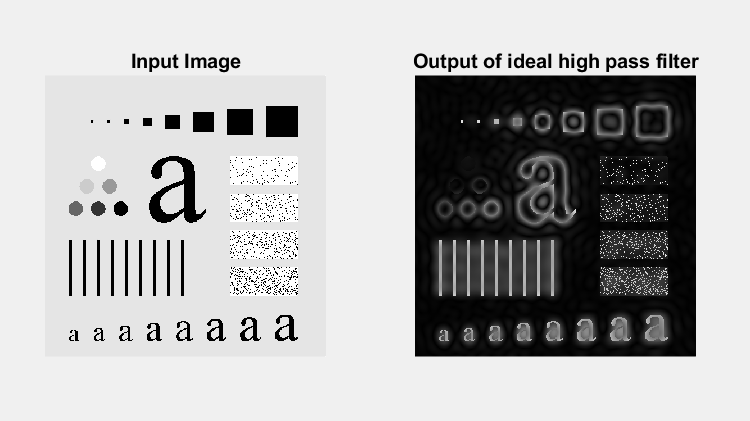
outImg = abs(ifft2(ihp));

%Display input and output images in plot

subplot(1,2,1);imshow(inImg);title('Input Image');

subplot(1,2,2);imshow(uint8(outImg));title('Output of ideal high pass filter');

**Output:**



**Conclusion:**

By Performing this practical we get to know that in ideal high pass filter we will attenuate all the low frequencies and pass the higher ones to get sharpen image.

1. **Butterworth high pass Filter:**

**Code:**

clc

clear all

% read input image using imread() function

inImg = imread('Images\e14.tif');

[m,n] = size(inImg);

% set cut-off frequency

cf = 90;

% 2d fourier transformation

f = fftshift(fft2(inImg));

N = 1;

p = round(m/2);

q = round(n/2);

% Defining the filter

H = zeros(m,n);

for i = 1:m

for j = 1:n

d = (i-p).^2 + (j-q).^2;

if d~=0

H(i,j) = 1/(1+((cf\*cf/d).^(2\*N)));

end

end

end

% Appling the Butterworth high pass filter

bhp = f.\*H;

% inverse 2d fourier transformation

outImg = abs(ifft2(bhp));

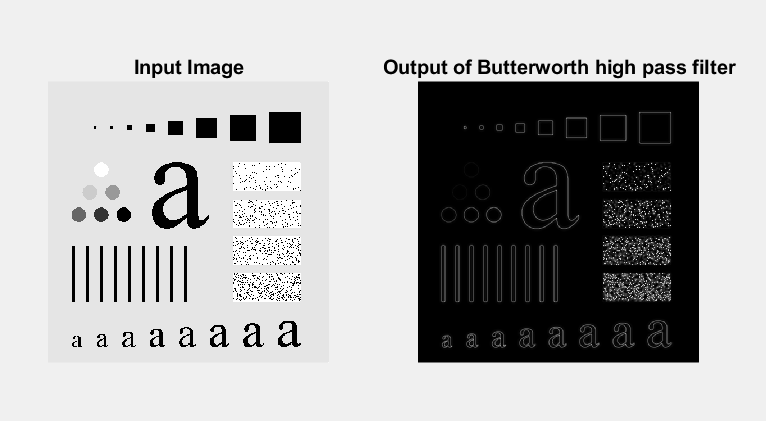
%Display input and output images in plot

%Display input and output images in plot

subplot(1,2,1);imshow(inImg);title('Input Image');

subplot(1,2,2);imshow(uint8(outImg));title('Output of Butterworth high pass filter');

**Output:**



**Conclusion:**

By Performing this practical we get to know that in Butterworth high pass filter we will attenuate all the low frequencies and pass the higher ones to get sharpen image.

1. **Gaussian high pass Filter:**

**Code:**

clc

clear all

% read input image using imread() function

inImg = imread('Images\e14.tif');

[m,n] = size(inImg);

% set cut-off frequency

cf = 50;

% 2d fourier transformation

f = fftshift(fft2(inImg));

p = round(m/2);

q = round(n/2);

% Defining the filter

H = zeros(m,n);

for i = 1:m

for j = 1:n

d = (i-p).^2 + (j-q).^2;

H(i,j) = exp(-d/2/cf/cf);

end

end

H = 1-H;

% Appling the Gaussian high pass filter

ghp = f.\*H;

% inverse 2d fourier transformation

outImg = abs(ifft2(ghp));

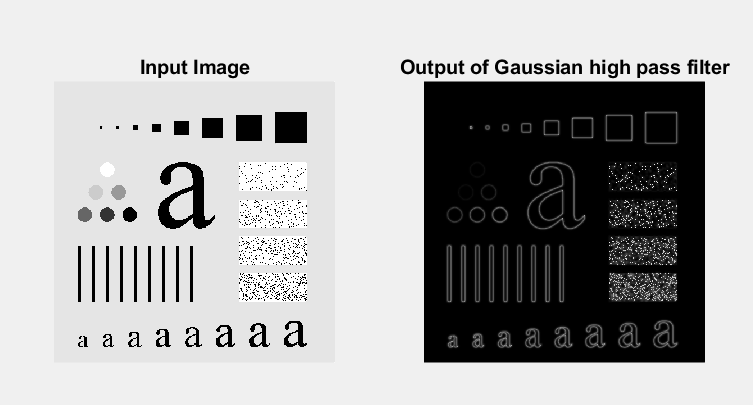
%Display input and output images in plot

%Display input and output images in plot

subplot(1,2,1);imshow(inImg);title('Input Image');

subplot(1,2,2);imshow(uint8(outImg));title('Output of Gaussian high pass filter');

**Output:**



**Conclusion:**

By Performing this practical we get to know that in gaussian high pass filter we will attenuate all the low frequencies and pass the higher ones.