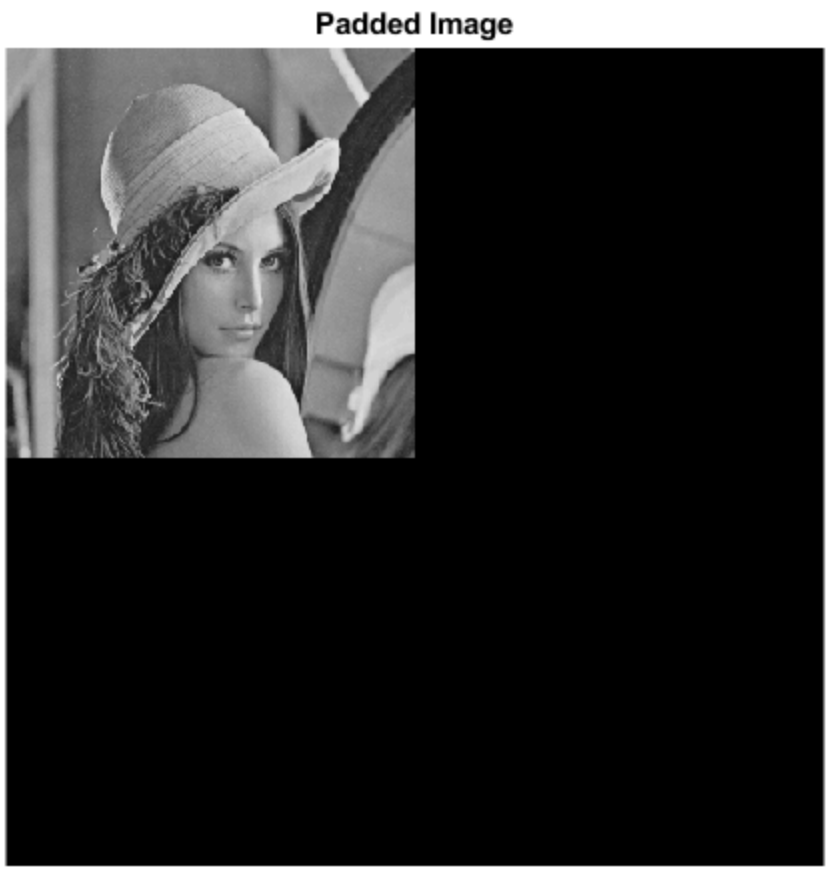
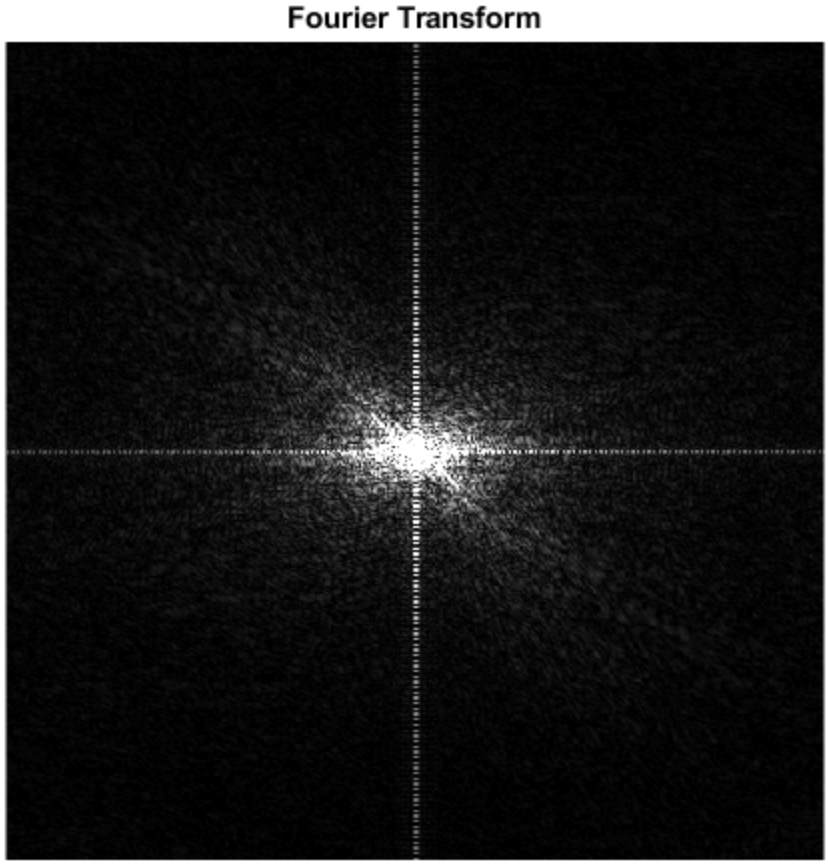
```
%NAME: Krishna Kodali
%INST: IIT Bhubaneswar
%DATE: 21/10/2020
%CATEGORY: BTech
%BRANCH: Computer Science
%Roll Number: 17CS01008

% Assignment-03
% Filtering in Frequency Domain & Morphological Operators
%Removing previous Buffer
clc;clear;close all;
```

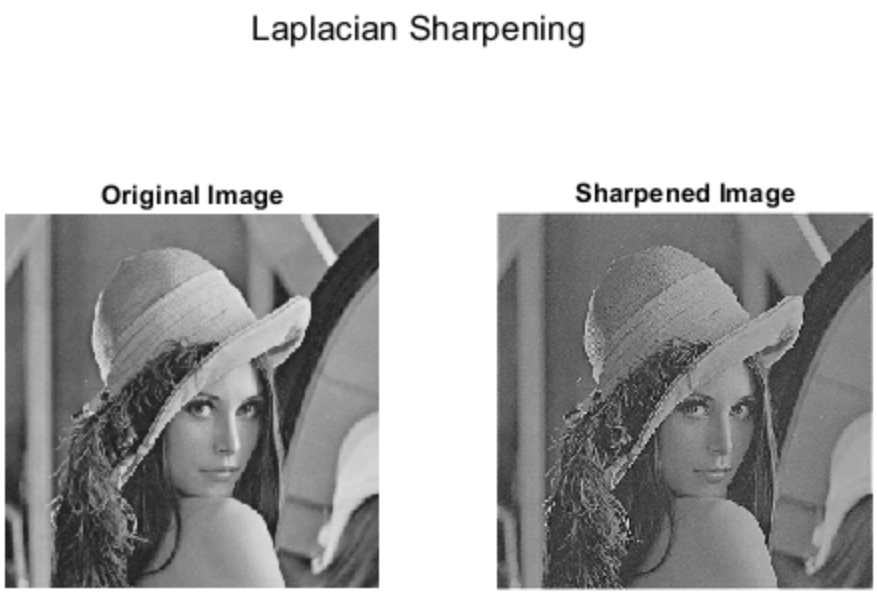
Question-1: Perform Laplacian Based Sharpening in Frequency Domain

```
%Input Image
Image = imread("lena_gray_256.tif");
Image = double(Image);

figure, imshow(uint8(Image)),
title("Input Image");
[r,c] = size(Image);
p = 2*r; q = 2*c;

%Padding the image
pad_Image = zeros(p, q);
pad_Image(1:r, 1:c) = Image;
figure, imshow(uint8(pad_Image)),
title('Padded Image')

%Shifting the image w.r.t the origin
center = zeros(p, q);
for i = 1 : p
    for j = 1 : q
        center(i, j) = (-1) ^ (i + j);
    end
end
pad_Image = pad_Image .* (center);

%Finding Fourier transform
F_I = fft2(pad_Image);
A_FI = abs(F_I);
figure, imshow(uint8(A_FI./255)),
title('Fourier Transform');

%Computing Laplacian Operator
Laplacian = zeros(p, q);
for i = 1 : p
    for j = 1 : q
        Laplacian(i, j) = (-4 * (pi .^ 2)) * ((i - r).^2 + (j -c).^2);
    end
end
```

```
%Multiplying in Frequency Domain and unpadding the image.
LF_I = F_I .* Laplacian;
LF_I = real(ifft2(LF_I)) .* (center);
LF_I = LF_I(1 : r, 1 : c);

%Normalizing in order to scale the values.
maxVal = max(max(LF_I));
minVal = min(min(LF_I));
LF_I = (LF_I - minVal) ./ (maxVal - minVal);
LF_I = LF_I * 255;
SI = (Image) - LF_I;

%Plotting the output
figure
subplot(1, 2, 1), imshow(uint8(Image)), title('Original Image')
subplot(1, 2, 2), imshow(mat2gray(SI)), title('Sharpened Image')
sgtitle('Laplacian Sharpening')
```


Padded Image


Fourier Transform



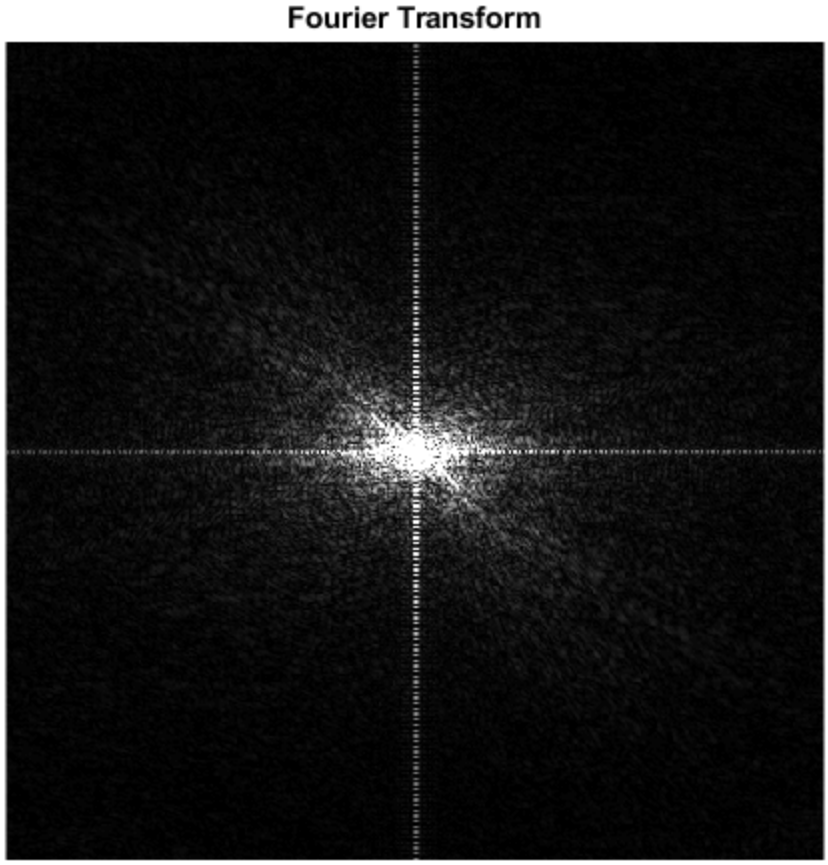
Question 2: Perform Gaussian based lowpass and highpass filtering on an image in the frequency domain

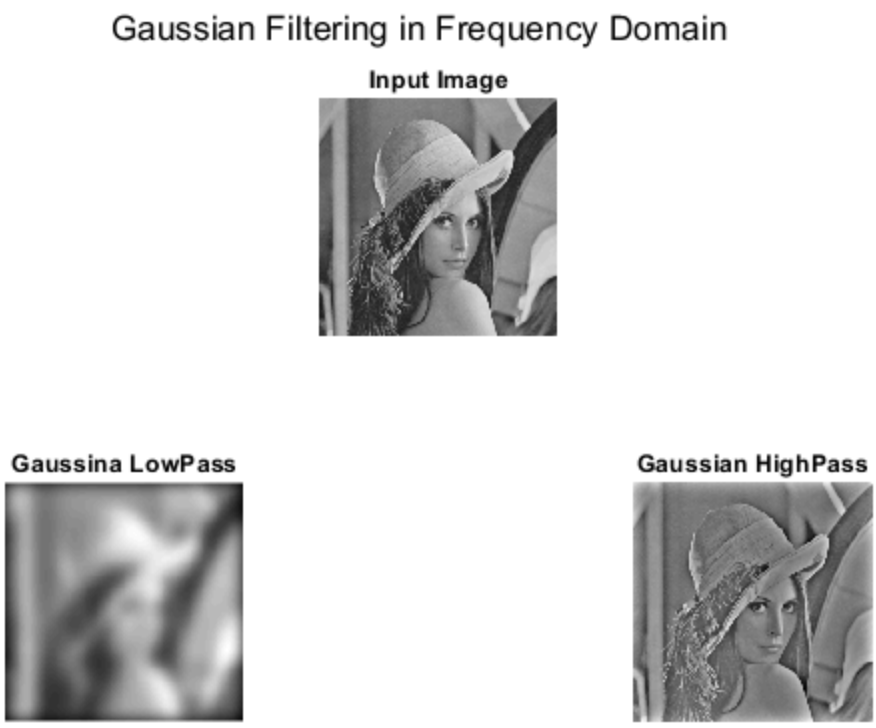
```
Image = imread("lena_gray_256.tif");
Image = double(Image);

figure, imshow(uint8(Image)),
title('Input Image')

[r,c] = size(Image);
p = 2*r; q = 2*c;

%Padding the image
pad_Image = zeros(p, q);
pad_Image(1:r, 1:c) = Image;
figure, imshow(uint8(pad_Image)),
title('Padded Image')

%Shifting the image w.r.t the center
center = zeros(p, q);
for i = 1 : p
    for j = 1 : q
        center(i, j) = (-1) ^ (i + j);
    end
end
pad_Image = pad_Image .* (center);
```

```
%Finding Fourier transform
F_I = fft2(pad_Image);
A_FI = abs(F_I);
figure, imshow(uint8(A_FI./255)),
title('Fourier Transform');

%Finding Gaussian Lowpass and Highpass Filters
LP_Gaussian = zeros(p, q);
HP_Gaussian = zeros(p, q);
Threshold = 10;
for i = 1 : p
    for j = 1 : q
        D = (i - r) .^ 2 + (j - c) .^ 2;
        LP_Gaussian(i, j) = exp(-1 * D / (2 * (Threshold .^2)));
        HP_Gaussian(i, j) = 1 - LP_Gaussian(i, j);
    end
end

% Multiplying in Frequency Domain and Unpadding the Image
LP_I = F_I .* LP_Gaussian;
LP_I = real(ifft2(LP_I)) .* (center);
LP_I = LP_I(1 : r, 1 : c);

HP_I = F_I.* HP_Gaussian;
HP_I = real(ifft2(HP_I)) .* (center);
HP_I = HP_I(1 : r, 1 : c);

figure
subplot(2, 3, 2), imshow(uint8(Image)),
title('Input Image');

%Normalizing the Image and Plotting them
maxVal = max(max(LP_I));
minVal = min(min(LP_I));
LP_I = (LP_I - minVal) ./ (maxVal - minVal);
LP_I = LP_I * 255;
SI = (Image) - LP_I;
subplot(2, 3, 4), imshow(mat2gray(LP_I)),
title('Gaussina LowPass');

maxVal = max(max(HP_I));
minVal = min(min(HP_I));
HP_I = (HP_I - minVal) ./ (maxVal - minVal);
HP_I = HP_I * 255;
SI2 = (Image) - HP_I;
subplot(2, 3, 6), imshow(mat2gray(HP_I)),
title('Gaussian HighPass');
sgtitle('Gaussian Filtering in Frequency Domain');
```

Input Image

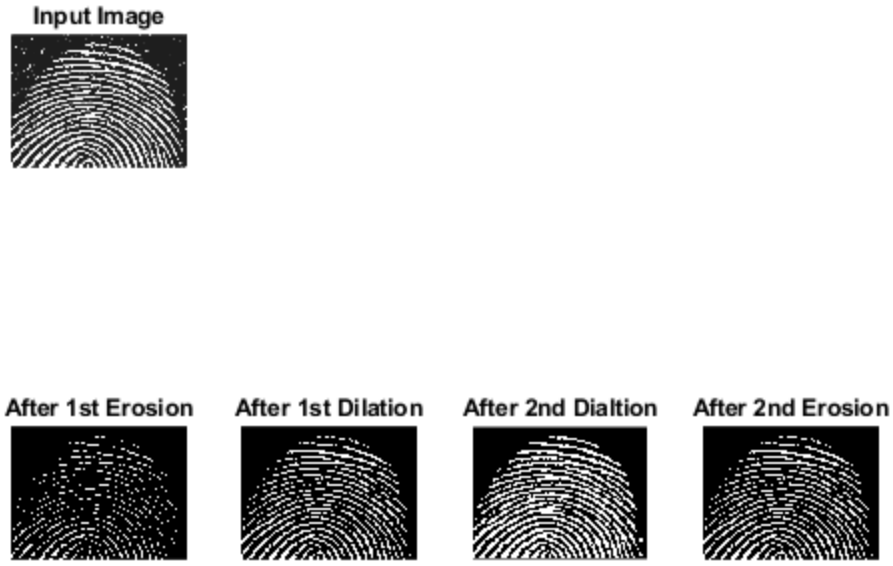
Padded Image


Fourier Transform




```
%Question-3: Take a noisy fingerprint image and find the processed
 images using different morphological operators.
Image = imread('Fingerprint.png');
Image = rgb2gray(Image);
Image = double(Image);

figure
subplot(2, 4, 1), imshow(uint8(Image)),
title('Input Image');

[p, q] = size(Image);

maxVal = max(max(Image));
minVal = min(min(Image));
Image = (Image - minVal) ./ (maxVal - minVal);

%Structuring Element
SE = ones(3, 3);

%For a Image Performing, Erosion, Dilation again followed by Dilation
 and then Erosion
%Erosion
E1_Image = zeros(p, q);
for i = 1 : p - 2
    for j = 1 : q - 2
        SE_Image = Image(i : i + 2, j : j + 2);
```

```
        V = SE_Image .* SE;
        if sum(sum(V)) == 9
            E1_Image(i + 1, j + 1) = 1;
        end
    end
end
subplot(2, 4, 5), imshow(uint8(E1_Image*255)),
title('After 1st Erosion');

%Dilation
D1_Image = zeros(p, q);
for i = 1 : p - 2
    for j = 1 : q - 2
        SE_Image = E1_Image(i : i + 2, j : j + 2);
        V = SE_Image .* SE;
        if sum(sum(V)) > 0
            D1_Image(i + 1, j + 1) = 1;
        end
    end
end
subplot(2, 4, 6), imshow(uint8(D1_Image*255)),
title('After 1st Dilation');

%Dilation
D2_Image = zeros(p, q);
for i = 1 : p - 2
    for j = 1 : q - 2
        SE_Image = D1_Image(i : i + 2, j : j + 2);
        V = SE_Image .* SE;
        if sum(sum(V)) > 0
            D2_Image(i + 1, j + 1) = 1;
        end
    end
end
subplot(2, 4, 7), imshow(uint8(D2_Image*255)),
title('After 2nd Dialtion');

%Erosion
E2_Image = zeros(p, q);
for i = 1 : p - 2
    for j = 1 : q - 2
        SE_Image = D2_Image(i : i + 2, j : j + 2);
        V = SE_Image .* SE;
        if sum(sum(V)) == 9
            E2_Image(i + 1, j + 1) = 1;
        end
    end
end
subplot(2, 4, 8), imshow(uint8(E2_Image*255)),
title('After 2nd Erosion');
sgtitle("Morphological Operations on Noisy Image");
```



Published with MATLAB® R2020a