Assignment 05 - Report

Student Name	*Student ID (DAIICT)*	*Student ID (IIT Jammu)*
Ambuj Mishra	202116003	2021PCS1017
Arpita Nema	202116004	2021PCS1018

Objective

Problem 1. Find out and show the Magnitude and Phase information of the cameraman image (Fig.1). Perform log transformation on the magnitude spectrum and show the result. Discuss the reason for the difference between the observations (with log and without log transformation). Observe the significance of DFT coefficients of the image by reconstructing the image with higher frequency coefficients and lower frequency coefficients, separately without changing the phase information. [Tips:Use FFT for finding out the DFT coefficients.]

Problem 2. Demonstrate the significance of magnitude and phase information by reconstructing an image with the magnitude of Fig.2(a) and phase of Fig. 2(b). Repeat the experiment with the mag- nitude of Fig.2 (b) and phase of Fig.2 (a). [Tips: First of all resize the images to create same dimensional images]
s
Problem 3. Implement low pass and high pass filters in frequency domain for Ideal, Butterworth (order 2), and Gaussian kernels and apply those on Fig.1. Discuss about the results.

Problem 4. Consider the image of Fig.1, and implement unsharp masking and high boost filtering in the frequency domain. Compare the results with the implementation of spatial domain operation. Tune the parameters (cut-off frequency for frequency domain, kernel size for spatial domain, etc.) to achieve the same results in both domains. Report the results.

Methods and Experiments

Problem 1.

We calculated the magnitude and phase of given image using following functions:

- Calculating 2D discrete fourier transformation-
  f = np.fft.fft2(img)

- Bringing high frequency component to center-
  fshift = np.fft.fftshift(f)

- Calculating magnitude spectrum and it's log transformation-
  magnitude_spectrum = np.abs(fshift)
  magnitude_spectrum_log = np.log(np.abs(fshift)+1)

- Calculating phase spectrum-
  phase_spectrum=np.angle(fshift)

Applying a low pass filter in the frequency domain means zeroing all frequency components above a cut-off frequency.

Applying a high pass filter frequency domain is the opposite to the low pass filter, that is, all the frequencies below some cut-off radius are removed.

Problem 2.

Using the same method used in Question-1, we have calculated the magnitude and phase spectrum of both the images and reconstructed the new images using magnitude of first and phase of second image and vice-versa.
To reconstruct the image, following function has been used-

```
reconstructed_image = np.multiply(magnitude_spectrum_image1,
phase_spectrum_image2)
```

Problem 3.

We implemented all the butterworth low-pass, butterworth low-pass, ideal low-pass, ideal high-pass, gaussian low-pass and gaussian high-pass filters. We have scratch coded below functions to implement these -

- ```
  def ideal_low_filter(lr, cr, cc, img):
   tmp = np.zeros((img.shape[0], img.shape[1]))
   for i in range(img.shape[0]):
     for j in range(img.shape[1]):
       tmp[i, j] = (1 if np.sqrt((i - cr) ** 2 + (j - cc) ** 2) <= lr else 0)
   return tmp
  ```

- ```
  def ideal_high_filter(lr, cr, cc, img):
   tmp = np.zeros((img.shape[0], img.shape[1]))
   for i in range(img.shape[0]):
     for j in range(img.shape[1]):
       tmp[i, j] = (0 if np.sqrt((i - cr) ** 2 + (j - cc) ** 2) <= lr else 1)
   return tmp
  ```

- ```
  def butterworth_low_pass(lr, cr, cc, n, img):
   tmp = np.zeros((img.shape[0], img.shape[1]))
   for i in range(img.shape[0]):
     for j in range(img.shape[1]):
       tmp[i, j] = 1 / (1 + np.sqrt((i - cr) ** 2 + (j - cc) ** 2) / lr) ** (2 * n)
  return tmp
  ```

- ```
  def butterworth_high_pass(lr, cr, cc, n, img):
   tmp = np.zeros((img.shape[0], img.shape[1]))
   for i in range(img.shape[0]):
     for j in range(img.shape[1]):
       tmp[i, j] = 1 / (1 + lr / np.sqrt((i - cr) ** 2 + (j - cc) ** 2)) ** (2 * n)
   return tmp
  ```

Problem 4.

Unsharp masking and high-boost filtering are the processes that are widely used to sharpen images in older times.
First, we created performed blurring on the original image using average filtering and subtracted this blurred image from the original image. This is done in order to create mask.
Then, we add k*mask created in previous step to original image.

- When value of K=1, then it is called unsharp masking.
- When K > 1, then it is termed as high-boost filtering.

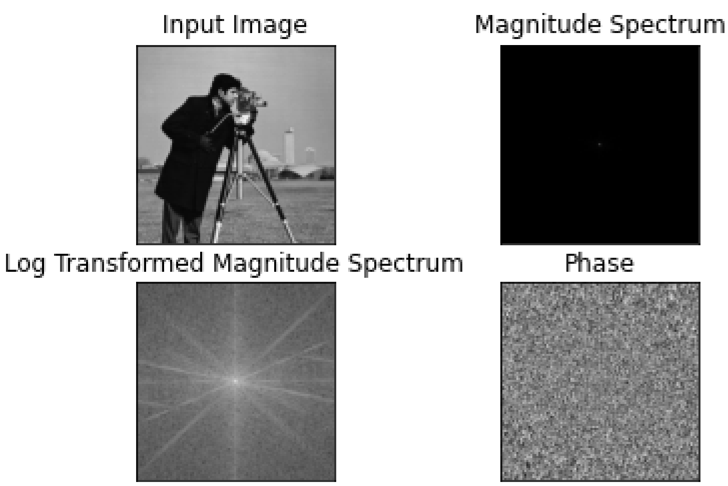
```
M(x, y) =  I (x, y) - I_blurred (x, y)
I_sharpend = I(x,y) + K * M(x,y)
```

where I(x,y) is original image;
I_blurred is image after performing blurring using averaging filtering. We used 9X9 averaging filter;
M(x,y) is generated mask
I_sharpend (x,y) corresponds to a sharpened image.

Observations

Problem 1.

Original Image



In magnitude spectrum some intensities will be orders of magnitude higher than others. Applying a log scale allows us to see the smaller intensity values as well. Otherwise, it will become difficult to analyze image.

===

Image reconstruction with entire magnitude spectrum and original phase information

==

Image reconstruction with with low frequency components   and original phase information

==

Image reconstruction with with high frequency components and original phase information

Problem 2.

Original Images

--

Lena Face

Einstein

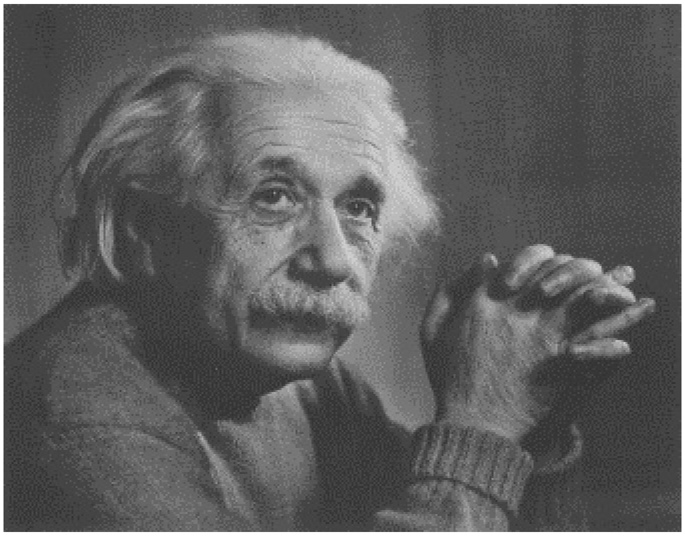


Image regenerated using magnitude of einstein's image and phase information from Lena's image

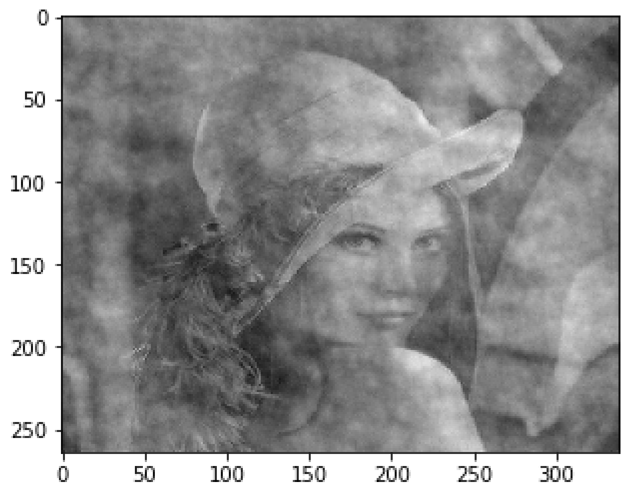
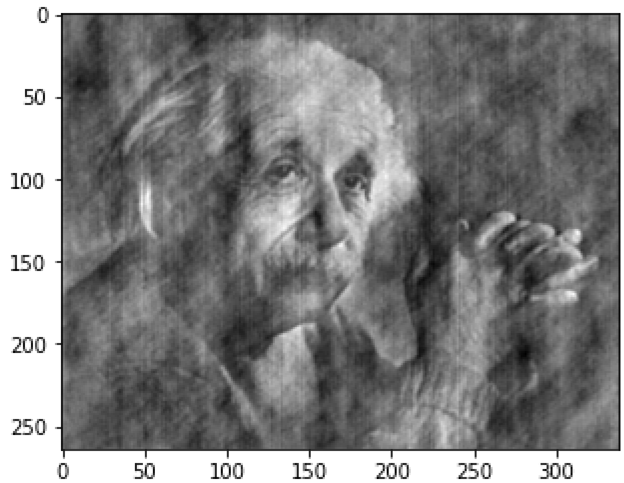


Image regenerated using magnitude of Lena's image and phase information from Einstein's image
<matplotlib.image.AxesImage at 0x7f208168ec50>



Problem 3.

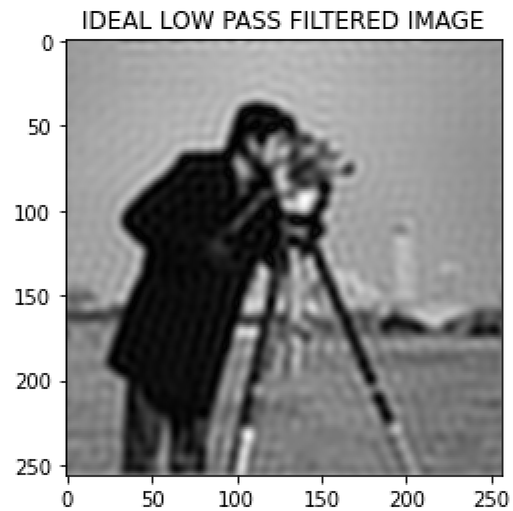
ORIGINAL

==

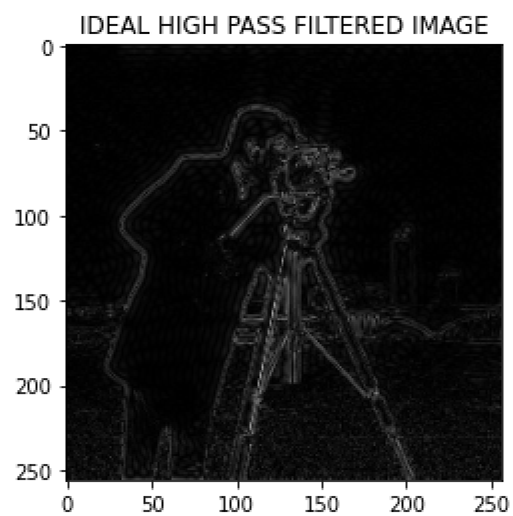
GAUSSIAN LOW PASS FILTERED IMAGE

===

GAUSSSIAN HIGH PASS FILTERED IMAGE

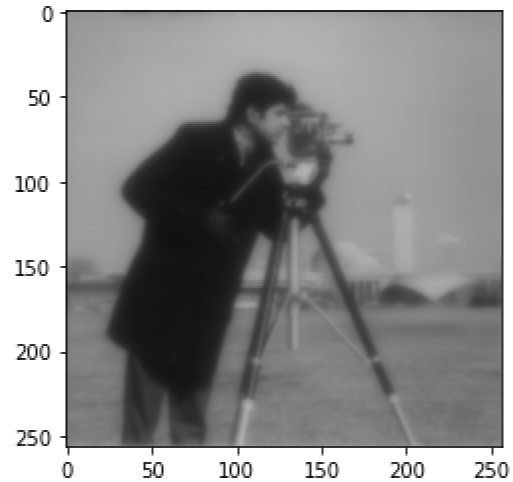

IDEAL LOW PASS FILTERED IMAGE
0
50
100
150
200
250
0
50
100
150
200
250

==


IDEAL HIGH PASS FILTERED IMAGE
0
50
100
150
200
250
0
50
100
150
200
250

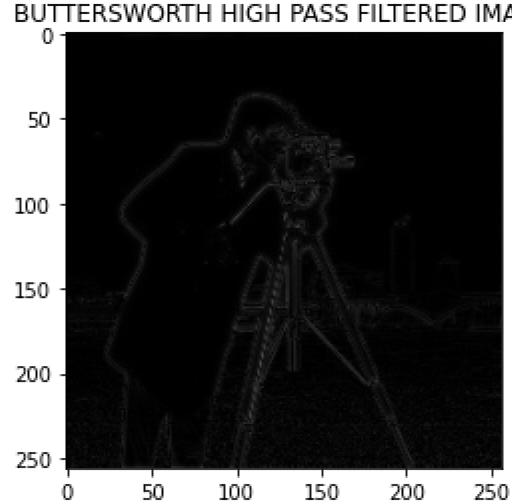
==

BUTTERSWORTH LOW PASS FILTERED IMAGE



==

BUTTERSWORTH HIGH PASS FILTERED IMAGE



Problem 4.

Original Image

==

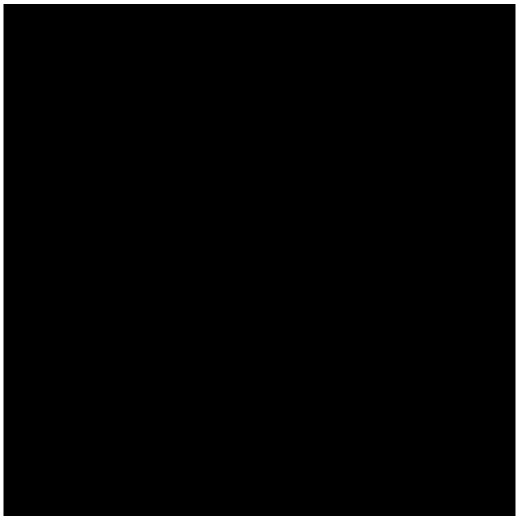
Blurred Image

==

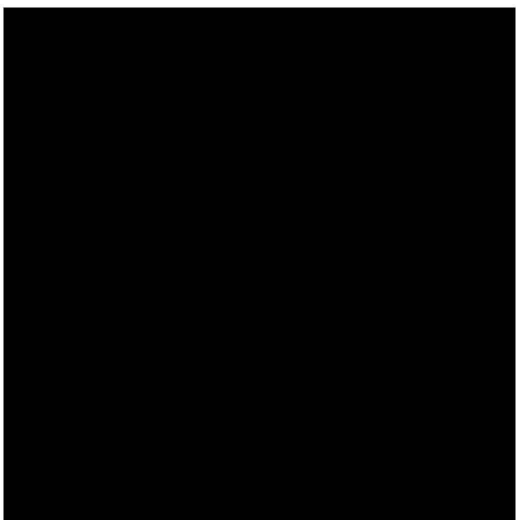
High-boost filtered image in spatial domain

ORIGINAL

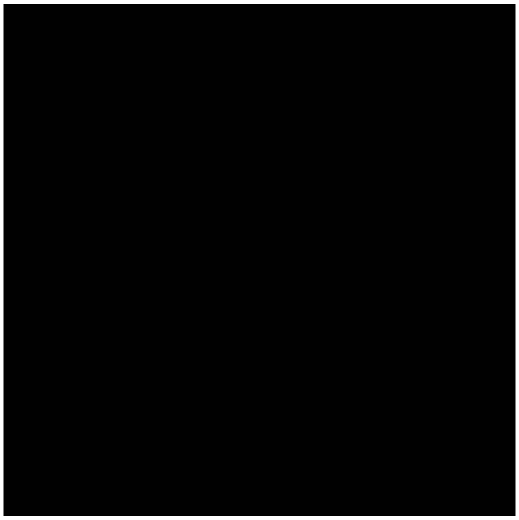
SPECTRUM



MASK



MASK2



ORIGINAL DFT/IFT

Image after Unsharp Masking in frequency domain

Image after high-boost filtering in frequency domain

==

Conclusion

Problem 1.

- In the magnitude spectrum some intensities will be orders of magnitude higher than others.
- Applying a log scale allows us to see the smaller intensity values as well. Otherwise, it will become difficult to analyze the image.
- Applying a low pass filter in the frequency domain means zeroing all frequency components above a cut-off frequency.

- Applying a high pass filter frequency domain is the opposite to the low pass filter, that is, all the frequencies below some cut-off radius are removed.

Problem 2.
If an image is reconstructed using the magnitude spectrum of one and phase spectrum of the other, the phase spectrum will decide the shape features and the magnitude spectrum will decide the intensities.

Problem 3.
Ideal, butterworth and gaussian low-pass filters are used for adding smoothening effects in an image while Ideal, butterworth and gaussian high-pass filters are used for image sharpening that is highlighting high frequency components like edges.

Problem 4.
Unsharp masking and high-boost filtering are used to perform sharpening on any image. They amplify the high-frequency components of a signal.
The high boost filter, which is a sharpening filter, is just 1 + fraction * high pass filter.To apply high-boost filter in frequency domain, a high pass filter is created in the range 0 to 1 rather than 0 to 255 for ease of use.
Observation - If we decrease the value of fraction, then the high-boost filtered image in the frequency domain becomes closer to the high-boost filtered image in the spatial domain.