Question_5

April 22, 2018

0.1 Q.5 : read an image and call it, I_garden. Find the Fourier Transform of I_garden; call it F_garden. Do the following 2 steps :

0.1.1 (a) Create a new spectrum, F_50, where you choose to retain only 50% of the spectrum co-efficients in F_garden, centered around the origin. Zero out the remaining. Use F_50 to reconstruct image I_50.

0.1.2 (b) Create a new spectrum, F_25, where you choose to retain only 25% of the spectrum co-efficients in F_garden, centered around the origin. Zero out the remaining. Use F_25 to reconstruct image I_25.

0.1.3 Compare images I_50 and I_25, with respect to I_garden. What do you observe ?

```
In [1]: # importing necessary packages

        import numpy as np
        import matplotlib.pyplot as plt
        import cv2
        %matplotlib inline
```

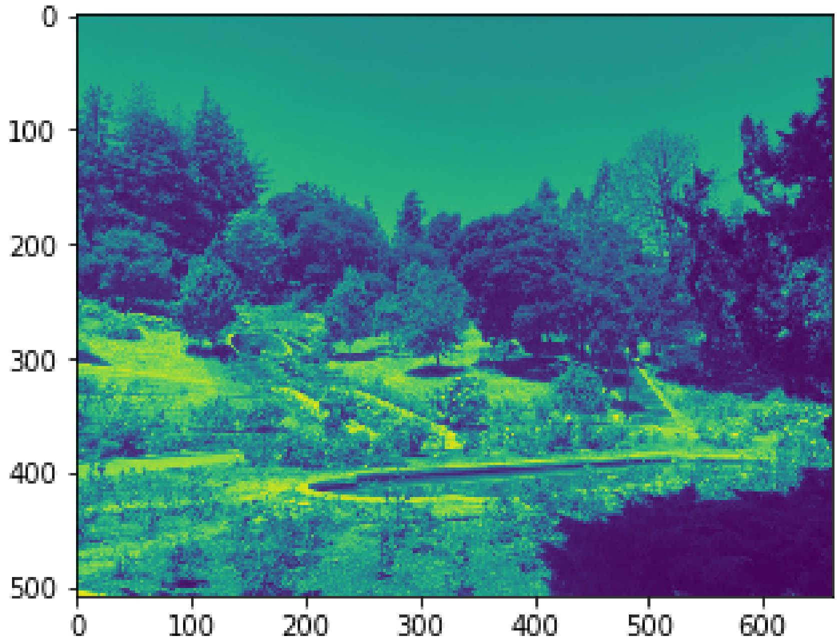
0.1.4 Defining a function for fourier transform of an image

```
In [47]: def fft(img):
             fft = np.fft.fft2(img)
             fft_shift = np.fft.fftshift(fft)

             return fft_shift

         def mag_ph(fft_img):
             fft_shift = fft_img
             magnitude_spectrum = 20*np.log(0.00001+np.abs(fft_shift))
             phase_spectrum = np.angle(fft_shift)

             return magnitude_spectrum, phase_spectrum
```

0.1.5 1. Reading an image into I_garden

```
In [48]: I_garden = cv2.imread('./images/5.png',0)
         plt.imshow(I_garden)
         print(I_garden.shape)
```

```
(508, 660)
```



0.1.6 2. Taking Fourier transform of an image

```
In [49]: F_garden = fft(I_garden)
         F_garden.shape
```

```
Out[49]: (508, 660)
```

0.1.7 Flow of steps :

First I will create a mask / low pass filter for an image , which will allow low pass frequncies to pass out and zero out the higher frequency. In fftshift of an image low frequencies are around DC component of an image : which is at the middle.So we create a filter accordingly

0.1.8 3. creating a mask / low pass filter : L_50 : which will allow 50% of spec coeff to pass around an origin

```
In [50]: # taking size of an image to create a mask
         ncols = int(660)
         nrows = int(508)

         #print(type(nrows))

         # 50% mask
```

```
L_50 = np.zeros((nrows,ncols))
L_50[int(nrows/2-nrows/4) : int(nrows/2+nrows/4) , int(ncols/2-ncols/4) : int(ncols/2-
```

0.1.9 4. Applying low pass filter / 50% Mask / L_50 to fourier transformed image F_garden

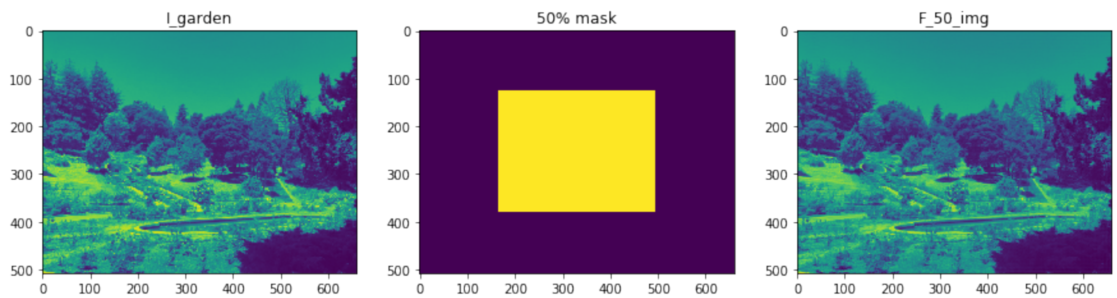
```
In [51]: F_50 = F_garden * L_50
```

0.1.10 5. recovering image frm fft by tkng inv fftshift first then tkng inv fft

```
In [52]: x = np.fft.ifftshift(F_50)
         y = np.fft.ifft2(x)
         F_50_img = np.array(np.abs(y))          # np.abs removes imaginary parts
```

0.1.11 6. Plotting an original image + Mask representation + Image recovered after applying low pass filter

```
In [53]: f, axarr = plt.subplots(1,3,figsize=(15,15))
         axarr[0].imshow(I_garden); axarr[0].set_title('I_garden')
         axarr[1].imshow(L_50) ; axarr[1].set_title('50% mask')
         axarr[2].imshow(F_50_img) ; axarr[2].set_title('F_50_img')

Out[53]: Text(0.5,1,'F_50_img')
```

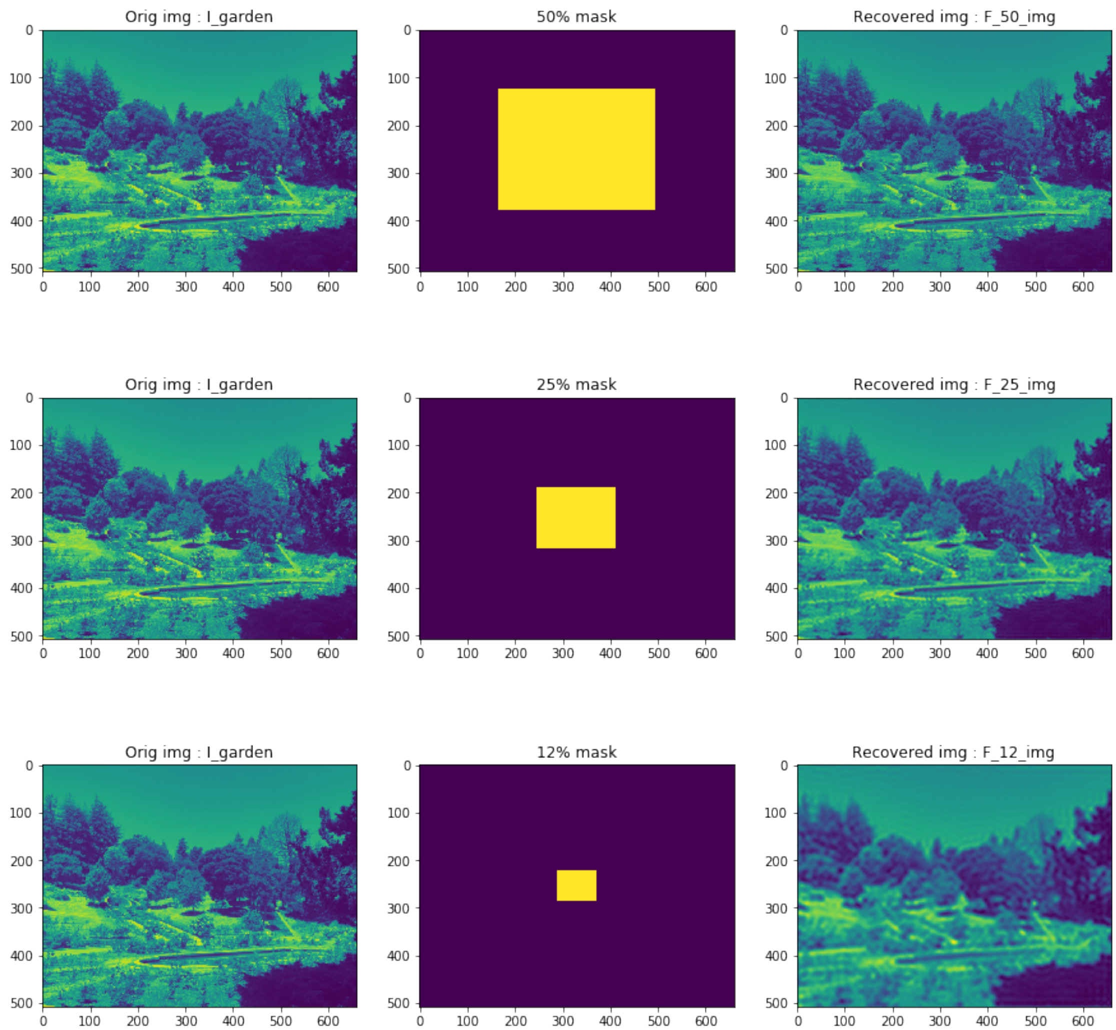


0.1.12 7. Likewise , follwing similar steps for two more masks / low pass filters : 25% and 12%

```
In [54]: # 25% mask
         L_25 = np.zeros((nrows,ncols))
         L_25[int(nrows/2-nrows/8) : int(nrows/2+nrows/8) , int(ncols/2-ncols/8) : int(ncols/2-

         # 12.5% mask
         L_12 = np.zeros((nrows,ncols))
         L_12[int(nrows/2-nrows/16) : int(nrows/2+nrows/16) , int(ncols/2-ncols/16) : int(ncols


         # applying low pass filter to fourier transformed img
         F_25 = F_garden * L_25
```

```python
F_12 = F_garden * L_12


# recovering image frm fft by tkng inv fftshift first then tkng inv fft

x = np.fft.ifftshift(F_25)
y = np.fft.ifft2(x)
F_25_img = np.array(np.abs(y))

x = np.fft.ifftshift(F_12)
y = np.fft.ifft2(x)
F_12_img = np.array(np.abs(y))


# plotting all the results
f, axarr = plt.subplots(3,3,figsize=(15,15))
axarr[0,0].imshow(I_garden); axarr[0,0].set_title('Orig img : I_garden')
axarr[0,1].imshow(L_50) ; axarr[0,1].set_title('50% mask')
axarr[0,2].imshow(F_50_img) ; axarr[0,2].set_title('Recovered img : F_50_img')
axarr[1,0].imshow(I_garden) ; axarr[1,0].set_title('Orig img : I_garden')
axarr[1,1].imshow(L_25) ; axarr[1,1].set_title('25% mask')
axarr[1,2].imshow(F_25_img) ; axarr[1,2].set_title('Recovered img : F_25_img')
axarr[2,0].imshow(I_garden) ; axarr[2,0].set_title('Orig img : I_garden')
axarr[2,1].imshow(L_12) ; axarr[2,1].set_title('12% mask')
axarr[2,2].imshow(F_12_img) ; axarr[2,2].set_title('Recovered img : F_12_img')
```

Out[54]: Text(0.5,1,'Recovered img : F_12_img')



0.2 Observation :

- As we can see after applying a mask / low pass filter , it allows only low frequency components or spectrum - -coefficeints around an origin (DC component) : This will in turn reduce an information or details of an image: Which results into a blur image with low amount of details
- As we are reducing the size of the mask / low pass filter : it is cutting off more frequencies that are high than given threshold or distant from center in this case : so it results into more blur image and less info