Image processing
Exercise 4

Marc Bisgaard Petersen
Marpet18@student.aau.dk

Questions:

1. **Match the DFT and DCT content of the Lena image to a and b**
   Image b is the DCT content and image a is the DFT content

2. **Characterize the following 3 filters (a, b and c) as low pass and high pass. Which one you will use if you want to blur you image and which one if you want to find the edges?**
   Image a and c low pass filter while image b is a high pass filter. The low pass filter are used to blur images and the high pass filter are used for find edges.
   A and C = blurring image
   B = Edge finding

3. **Fourier transform is named after French mathematician:**
   The Fourier transform is named after Joseph Fourier

4. **Fourier transform domain is:**
   Frequency domain

5. **The $f(0,0)$ coefficient of a DCT of an image how is it called?**
   It is called ac component.

6. **Low pass filters are used for image**
   Blurring

7. **High pass filters are used for image**
   sharpening

Programming:

Illustrate the FFT of the three different grayscale images of your choice. One with really a lot of details and edges (lot of texture), one with a normal texture (eg a face) and the last one an almost uniform image (eg a image with a gradient gray). You can use Python and OpenCV (eg submit a JupyterLab notebook). Discuss the differences in the three different FFTs

Images:

Figure 1:lot of textures

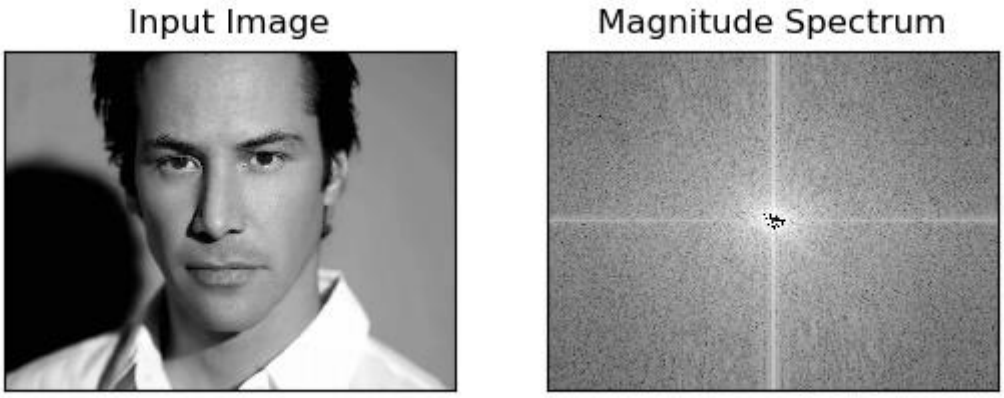

Figure 2: Normal texture

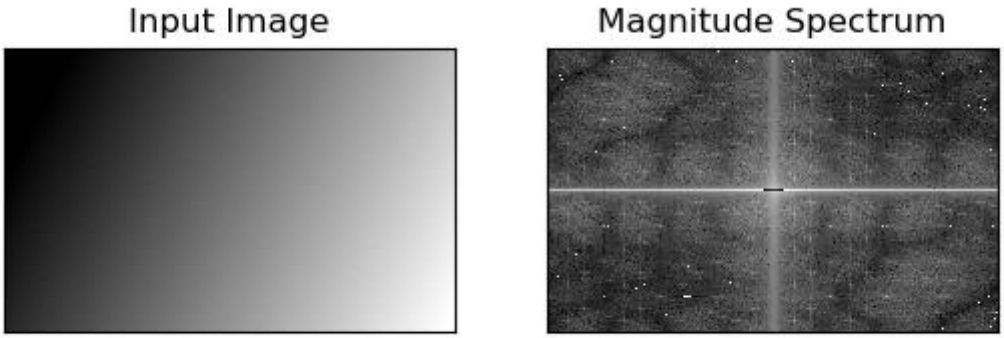

Figure 3: gradient grayscale

Code:

```
import cv2
import numpy as np
from matplotlib import pyplot as plt



routes =["images/details.jpg","images/face.jpg","images/uniform.jpg"]

imgWithDetails = cv2.imread(routes[0],cv2.IMREAD_GRAYSCALE)
imgWithNormalTexture = cv2.imread(routes[1],cv2.IMREAD_GRAYSCALE)
imgWithUniform = cv2.imread(routes[2],cv2.IMREAD_GRAYSCALE)

fd = np.fft.fft2(imgWithDetails)
fdshift = np.fft.fftshift(fd)
magnitude_spectrum1 = 20*np.log(np.abs(fdshift))
magnitude_spectrum1 = np.asarray(magnitude_spectrum1, dtype=np.uint8)

ft = np.fft.fft2(imgWithNormalTexture)
ftshift = np.fft.fftshift(ft)
magnitude_spectrum2 = 20*np.log(np.abs(ftshift))
magnitude_spectrum2 = np.asarray(magnitude_spectrum2, dtype=np.uint8)

fu = np.fft.fft2(imgWithUniform)
fushift = np.fft.fftshift(fu)
magnitude_spectrum3 = 20*np.log(np.abs(fushift))
magnitude_spectrum3 = np.asarray(magnitude_spectrum3, dtype=np.uint8)

plt.subplot(121),plt.imshow(imgWithDetails, cmap = 'gray')
plt.title('Input Image'), plt.xticks([]), plt.yticks([])
plt.subplot(122),plt.imshow(magnitude_spectrum1, cmap = 'gray')
plt.title('Magnitude Spectrum'), plt.xticks([]), plt.yticks([])
plt.show()

plt.subplot(121),plt.imshow(imgWithNormalTexture, cmap = 'gray')
plt.title('Input Image'), plt.xticks([]), plt.yticks([])
plt.subplot(122),plt.imshow(magnitude_spectrum2, cmap = 'gray')
plt.title('Magnitude Spectrum'), plt.xticks([]), plt.yticks([])
plt.show()

plt.subplot(121),plt.imshow(imgWithUniform, cmap = 'gray')
plt.title('Input Image'), plt.xticks([]), plt.yticks([])
plt.subplot(122),plt.imshow(magnitude_spectrum3, cmap = 'gray')
plt.title('Magnitude Spectrum'), plt.xticks([]), plt.yticks([])
plt.show()
```