LAB ASSIGNMENT 5 (SUBJECT: DIP LAB)

SUBMISSION DATE: 23/03/2022

NAME: Atharva Bhosale

BRANCH: IT

ID: 191080006

AIM : Write Programme for Filtering in frequency domain.

THEORY :

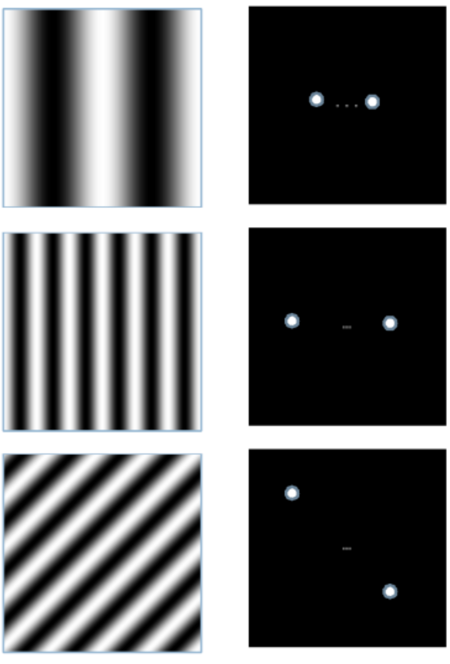
Filtering in Frequency Domain

Frequency Domain Filters are used for smoothing and sharpening of image by removal of high or low frequency components. Sometimes it is possible of removal of very high and very low frequency. Frequency domain filters are different from spatial domain filters as it basically focuses on the frequency of the images. It is basically done for two basic operation i.e., Smoothing and Sharpening.

Discrete Fourier Transform (DFT)

Fourier transform is a decomposition of a signal into some basis functions. Here basis functions are weighed sum of sin and cos functions Given a discrete image I(x,y) the fourier transform of it is :

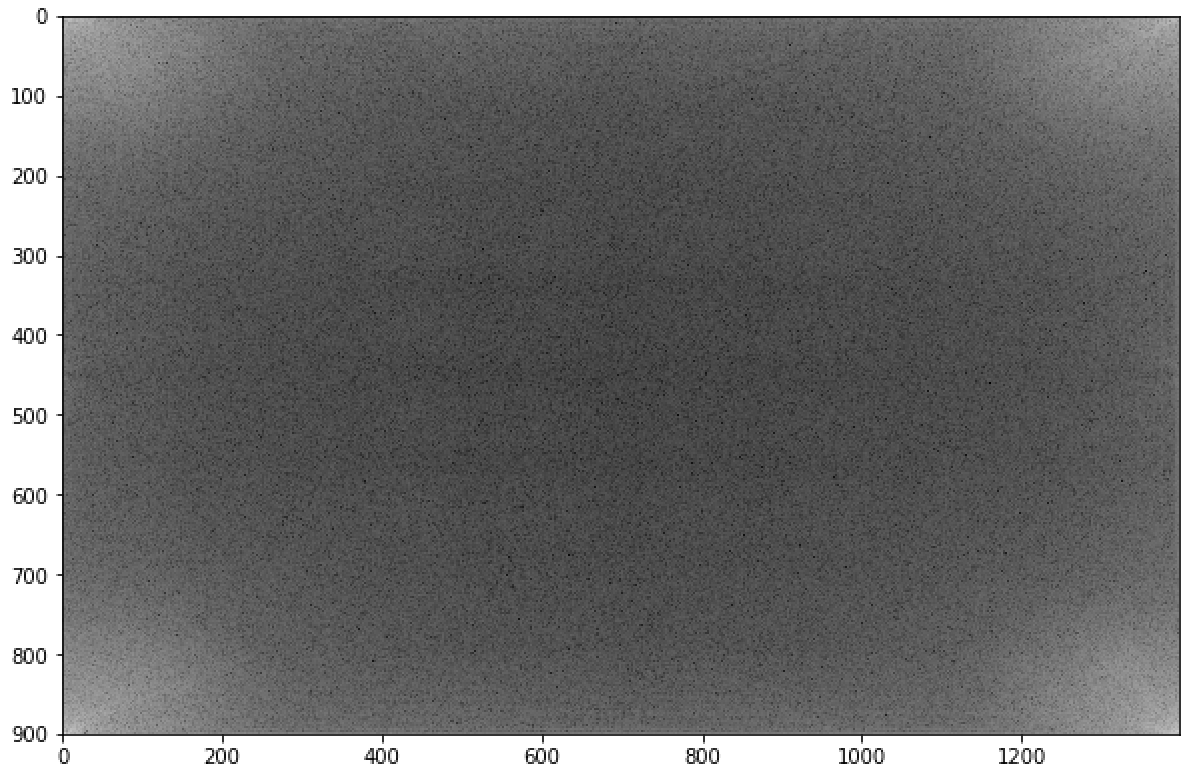
$$I(u,v) = \sum_{x=0}^{N_{cols}-1} \sum_{y=0}^{N_{rows}-1} I(x,y) e^{-i2\pi\left(\frac{xu}{N_{cols}} + \frac{yv}{N_{rows}}\right)}$$



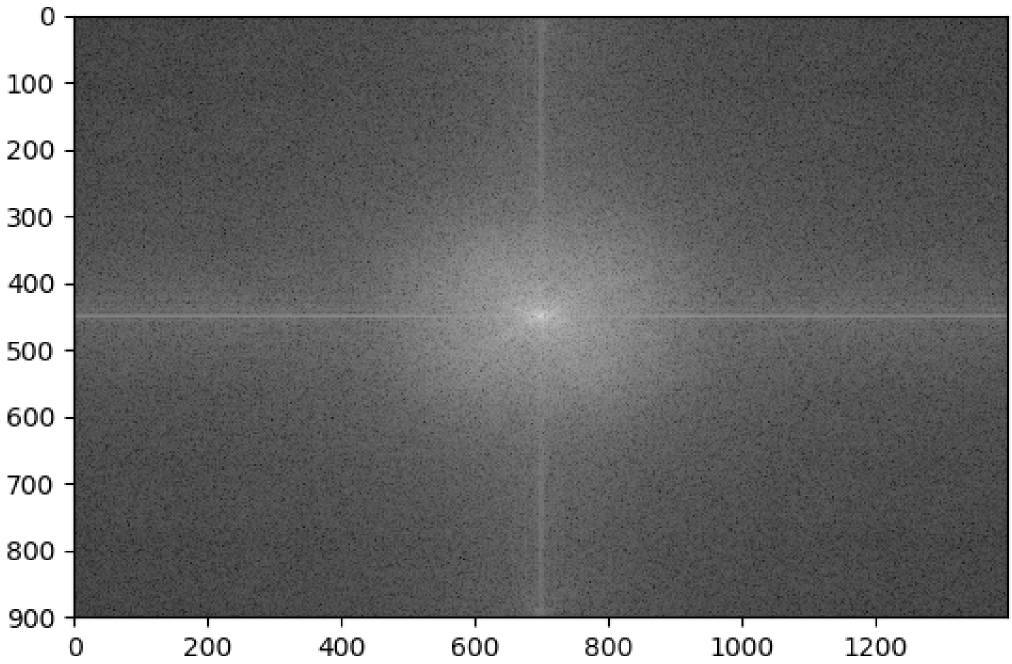
Low frequency components are found at the central regions while high frequency components are in peripherals.

The Original Image

Fourier Transform of the image Without shifting



Fourier Transform of the image after shifting. Shifting is done to move zero frequency component to the center of the image.



1. **Low pass filter:**
   Low pass filter removes the high frequency components that means it keeps low frequency components. It is used for smoothing the image. It is used to smoothen the image by attenuating high frequency components and preserving low frequency components.

   Mechanism of low pass filtering in frequency domain is given by:

   G(u, v) = H(u, v) . F(u, v)
   where F(u, v) is the Fourier Transform of original image
   and H(u, v) is the Fourier Transform of filtering mask

2. **High pass filter:**

   High pass filter removes the low frequency components that means it keeps high frequency components. It is used for sharpening the image. It is used to sharpen the image by attenuating low frequency components and preserving high frequency components.

   Mechanism of high pass filtering in frequency domain is given by:

   H(u, v) = 1 - H'(u, v)
   where H(u, v) is the Fourier Transform of high pass filtering
   and H'(u, v) is the Fourier Transform of low pass filtering

IMPLEMENTATION STEPS:

The assignment is done using Python Programming language and its libraries for displaying the image. The code for filtering in frequency domain is written from scratch without using any libraries that directly give the output.

The algorithm for generating low pass filter (LHF) and high pass filter (HPF) is written.

EXECUTION STEPS:

Just run all the cells of notebook.

CODE:

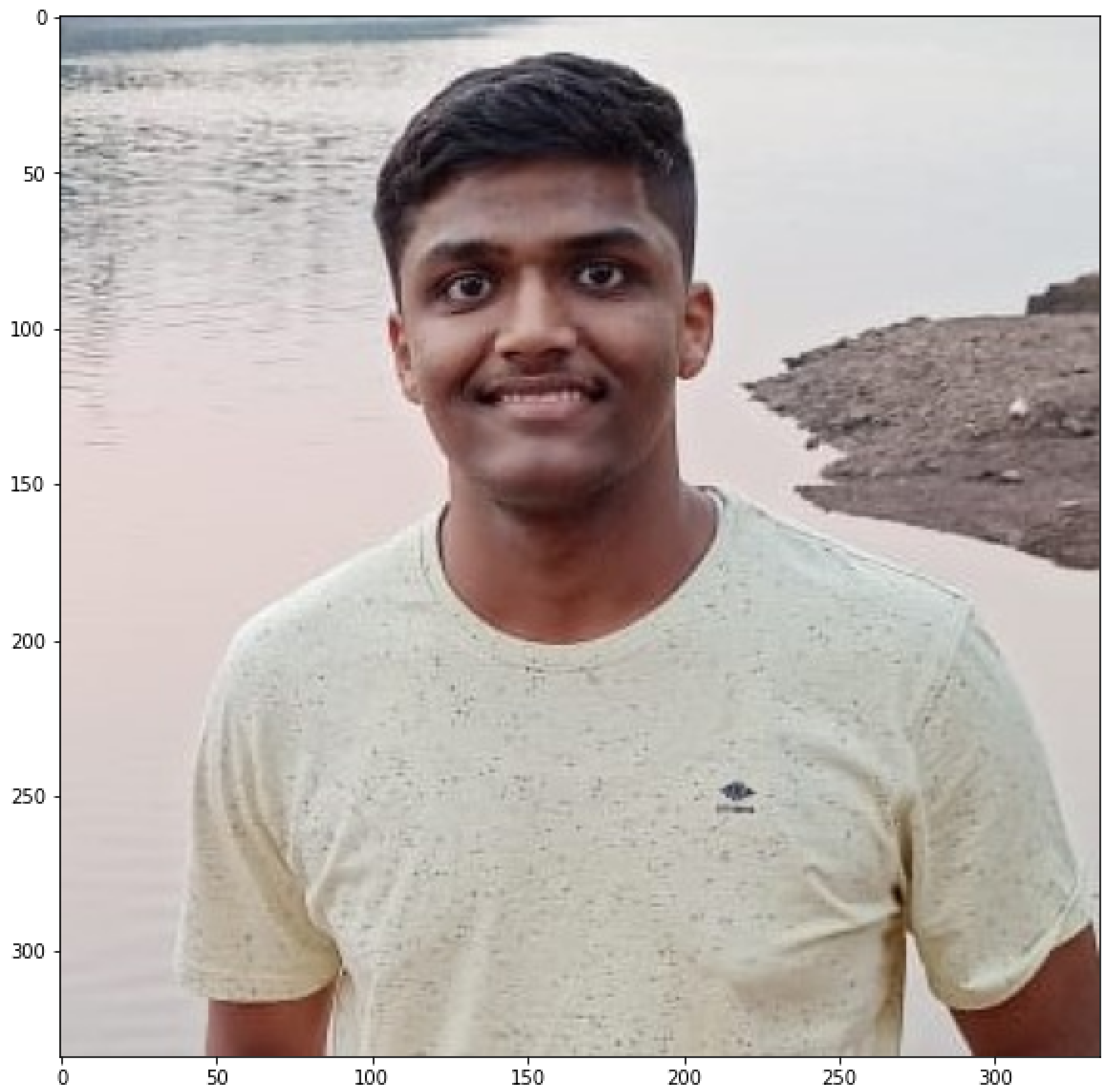
```
mport numpy as np
import matplotlib.pyplot as plt
import matplotlib.colors as col
import matplotlib.image as mpimg
from scipy import fftpack

def extractValueChannel(image):
    try:
        # Check if it has three channels or not
        np.size(image, 2)
    except:
        return image
    hsvImage = col.rgb_to_hsv(image)
    return hsvImage[..., 2]

def generateFilter(image,w,h, filtType):
    if w > 0.5 or h > 0.5:
        print("w and h must be < 0.5")
        exit()
    m = np.size(image,0)
    n = np.size(image,1)
    LPF = np.zeros((m,n))
    HPF = np.ones((m,n))
    xi = np.round((0.5 - w/2) * m)
    xf = np.round((0.5 + w/2) * m)
    yi = np.round((0.5 - h/2) * n)
    yf = np.round((0.5 + h/2) * n)
    LPF[int(xi):int(xf),int(yi):int(yf)] = 1
    HPF[int(xi):int(xf),int(yi):int(yf)] = 0
    if filtType == "LPF":
        return LPF
    elif filtType == "HPF":
        return HPF
    else:
```

```
        print("Only Ideal LPF and HPF are supported")
        exit()

print("he;;o")
image = mpimg.imread("atharva.bmp")
plt.figure("Original Image", figsize=(10,10))
plt.imshow(image)


valueChannel = extractValueChannel(image)
plt.figure("Value Channel of the Image", figsize=(10,10))
plt.imshow(valueChannel)

valueChannel = extractValueChannel(image)
FT = fftpack.fft2(valueChannel)
plt.figure("Fourier Transform of the Image", figsize=(10,10))
plt.imshow(np.log(1+np.abs(FT)))
plt.set_cmap("gray")

ShiftedFT = fftpack.fftshift(FT)
plt.figure("Fourier Transform of the Image", figsize=(10,10))
plt.imshow(np.log(1+np.abs(ShiftedFT)))
plt.set_cmap("gray")

ShiftedFT = fftpack.fftshift(FT)
plt.figure("Fourier Transform of the Image", figsize=(20,20))
plt.imshow(np.abs(ShiftedFT))
plt.set_cmap("gray")

LPF = generateFilter(ShiftedFT,0.05, 0.05, "LPF")
plt.figure("Ideal Low Pass Filter in frequency domain",
figsize=(10,10))
plt.imshow(LPF)

filteredVChannel = np.abs(fftpack.ifft2(LPF * ShiftedFT))
plt.figure("Filtered Value Channel of the Image", figsize=(10,10))
plt.imshow(filteredVChannel)

# Covert Image to hsv
hsvImage = col.rgb_to_hsv(image)
filteredVChannel = filteredVChannel/np.max(filteredVChannel)
# Add filtered value channel to hsv image
hsvImage[...,2] = filteredVChannel
```

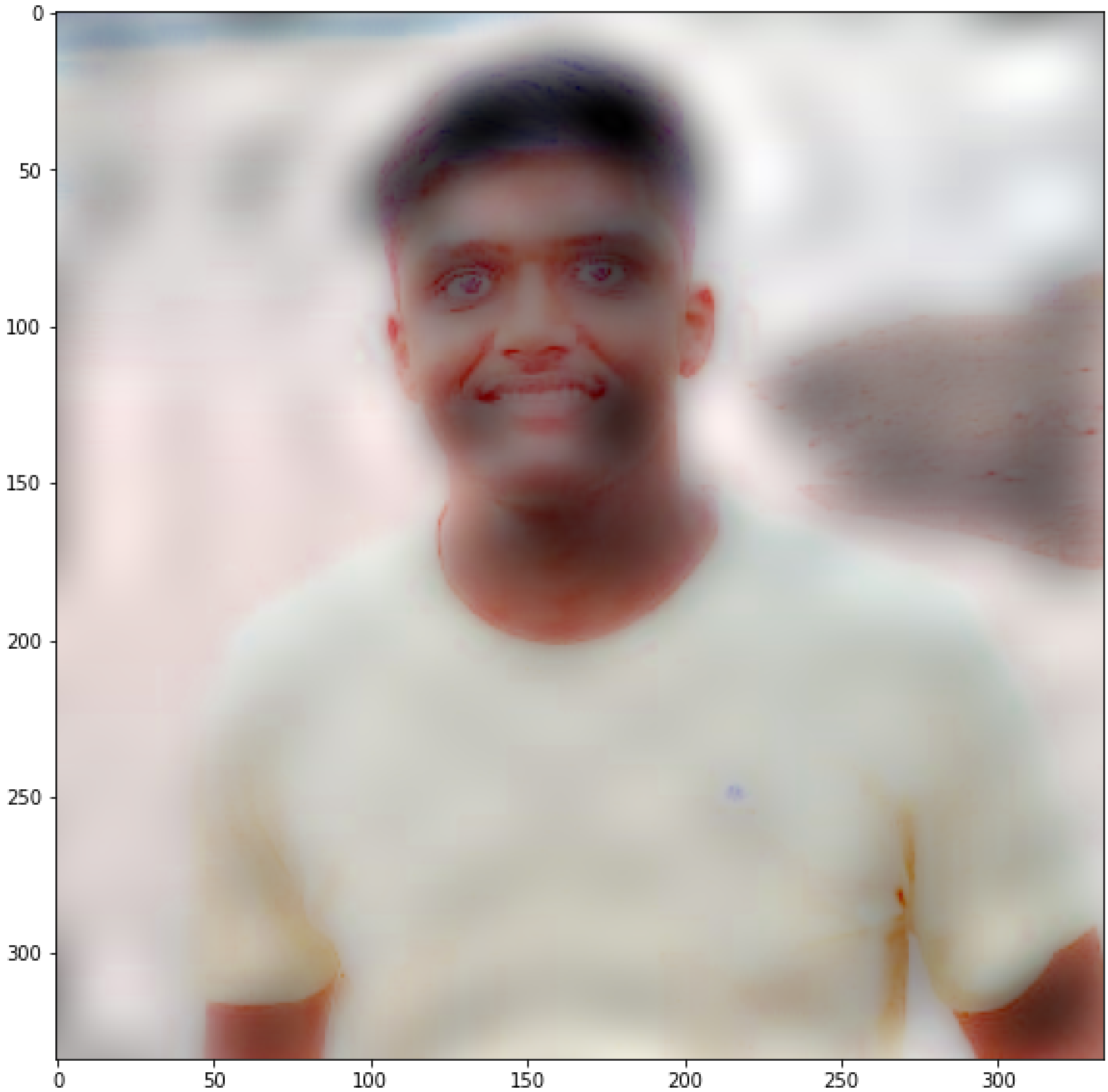
```
# Return Back to rgb color space
finalImage = col.hsv_to_rgb(hsvImage)
plt.figure("Final Image", figsize=(10,10))
plt.imshow(finalImage)

HPF = generateFilter(ShiftedFT,0.025, 0.025, "HPF")
plt.figure("Ideal High Pass Filter in frequency domain",
figsize=(10,10))
plt.imshow(HPF)
filteredVChannel = np.abs(fftpack.ifft2(HPF * ShiftedFT))
plt.figure("Filtered Value Channel of the Image", figsize=(10,10))
plt.imshow(filteredVChannel)
# Covert Image to hsv
hsvImage = col.rgb_to_hsv(image)
filteredVChannel = filteredVChannel/np.max(filteredVChannel)
# Add filtered value channel to hsv image
hsvImage[...,2] = filteredVChannel
# Return Back to rgb color space
finalImage = col.hsv_to_rgb(hsvImage)
plt.figure("Final Image", figsize=(10,10))
plt.imshow(finalImage)
```

OUTPUT:

1. Original image



2. After Low pass filtering (LPF)



3. After High pass filtering (HPF)

CONCLUSION :

In this lab assignment we learnt about filtering in frequency domain and successfully wrote a program to read images and apply low pass filtering (LPF) and high pass filtering (HPF).