**PAPER – III         PRACTICAL NO. : 01**

**AIM : WRITE PROGRAM TO DEMONSTRATE THE FOLLOWING ASPECTS OF SIGNAL PROCESSING ON SUITABLE DATA :**

1. **UNSAMPLING AND DOWNSAMPLING ON IMAGE / SPEECH SIGNAL**
2. **FAST FOURIER TRANSFORM TO COMPUTE DFT**

**ROLL NO. : 16                                BATCH : M.SC PART-I**

**DATE : 13/08/22**

**CODE :**

import numpy as np

import matplotlib.pyplot as plt

image\_filename = "E:\Msc Roll no. 02\Image processing\cgvr.png"

def calculate\_2dft(input):

ft = np.fft.ifftshift(input)

ft = np.fft.fft2(ft)

return np.fft.fftshift(ft)

def calculate\_2dift(input): #this function is not used

ift = np.fft.ifftshift(input)

ift = np.fft.ifft2(ift)

ift = np.fft.fftshift(ift)

return ift.real

# Read and process image

image = plt.imread(image\_filename)

image = image[:, :, :3].mean(axis=2) # Convert to grayscale

# Array dimensions (array is square) and centre pixel

array\_size = len(image)

centre = int((array\_size - 1) / 2)

# Get all coordinate pairs in the left half of the array,

# including the column at the centre of the array (which

# includes the centre pixel)

coords\_left\_half = (

(x, y) for x in range(array\_size) for y in range(centre+1)

)

plt.set\_cmap("gray")

ft = calculate\_2dft(image)

plt.subplot(121)

plt.imshow(image)

plt.axis("off")

plt.subplot(122)

plt.imshow(np.log(abs(ft)))

plt.axis("off")

plt.show()

