Marwadi University	**Marwari University** **Faculty of Technology** **Department of Information and Communication Technology**	
Subject: Digital Signal and Image Processing(01CT0513)	**Aim:** Simulate smoothing and sharpening operations on images using frequency domain filters.	
Experiment No: 10	**Date:**	**Enrollment No: 92200133030**

Aim: Simulate smoothing and sharpening operations on images using frequency domain filters.

Theory:-

- Smoothing and sharpening operations can also be performed in the frequency domain using filters such as the Gaussian filter and the Laplacian filter.

Smoothing in Frequency Domain:-

- To perform smoothing in the frequency domain, we apply a low-pass filter that attenuates high-frequency components. This helps to blur the image and reduce noise. One common approach is to use a Gaussian filter in the frequency domain.

Sharpening in Frequency Domain:-

- To perform sharpening in the frequency domain, we apply a high-pass filter that enhances high-frequency components. This amplifies the edges and fine details in the image. One common approach is to use a combination of a high-pass filter and the original image.

Programm:-

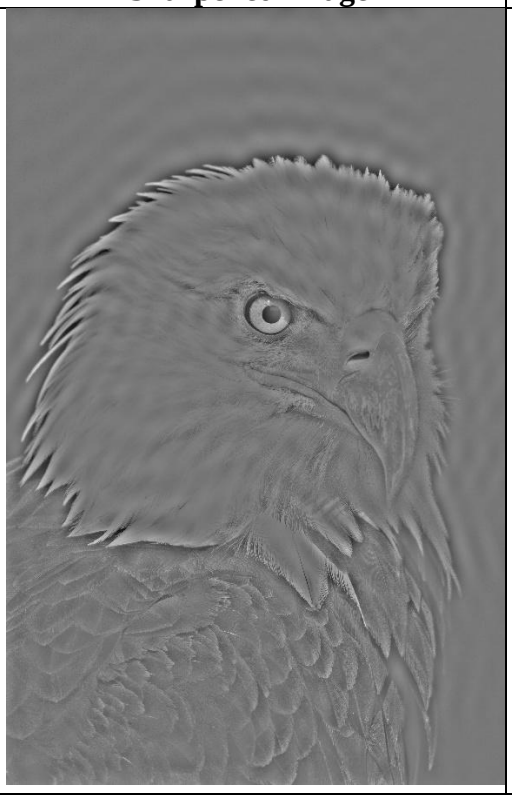
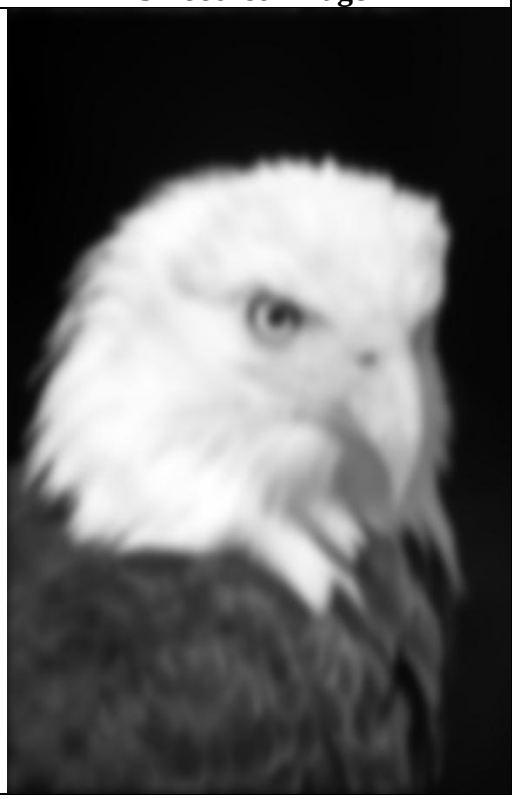
```
import cv2
import numpy as np
import matplotlib.pyplot as plt

def apply_gaussian_filter(image, sigma):
    image = np.float32(image)
    frequency_domain = cv2.dft(image, flags=cv2.DFT_COMPLEX_OUTPUT)
    shifted_frequency_domain = np.fft.fftshift(frequency_domain)
    rows, cols = image.shape
    crow, ccol = rows // 2, cols // 2
    mask = np.zeros((rows, cols, 2), np.float32)
    for i in range(rows):
        for j in range(cols):
            distance_squared = (i - crow) ** 2 + (j - ccol) ** 2
            gaussian_value = np.exp(-distance_squared / (2 * sigma**2))
            mask[i, j] = [gaussian_value, gaussian_value]
    filtered_frequency_domain = shifted_frequency_domain * mask
    shifted_filtered_frequency_domain = np.fft.ifftshift(filtered_frequency_domain)
    filtered_image = cv2.idft(
        shifted_filtered_frequency_domain, flags=cv2.DFT_SCALE | cv2.DFT_REAL_OUTPUT
    )
    filtered_image = cv2.normalize(filtered_image, None, 0, 255, cv2.NORM_MINMAX)
    return np.uint8(filtered_image)

def apply_sharpening_filter(image, strength):
    image = np.float32(image)
    frequency_domain = cv2.dft(image, flags=cv2.DFT_COMPLEX_OUTPUT)
    shifted_frequency_domain = np.fft.fftshift(frequency_domain)
```

Marwadi University	**Marwari University** **Faculty of Technology** **Department of Information and Communication Technology**	
Subject: Digital Signal and Image Processing(01CT0513)	**Aim:** Simulate smoothing and sharpening operations on images using frequency domain filters.	
Experiment No: 10	**Date:**	**Enrollment No: 92200133030**

```
    rows, cols = image.shape
    crow, ccol = rows // 2, cols // 2
    mask = np.ones((rows, cols, 2), np.float32)
    mask[crow - strength : crow + strength, ccol - strength : ccol + strength] = 0
    filtered_frequency_domain = shifted_frequency_domain * mask
    shifted_filtered_frequency_domain = np.fft.ifftshift(filtered_frequency_domain)
    filtered_image = cv2.idft(
        shifted_filtered_frequency_domain, flags=cv2.DFT_SCALE | cv2.DFT_REAL_OUTPUT
    )
    filtered_image = cv2.normalize(filtered_image, None, 0, 255, cv2.NORM_MINMAX)
    return np.uint8(filtered_image)

image_path = "./Images.jpg"
input_image = cv2.imread(image_path, 0)
if input_image is None:
    raise FileNotFoundError(
        f"The image '{image_path}' could not be loaded. Check the file path."
    )

sigma = 20
smoothed_image = apply_gaussian_filter(input_image, sigma)

strength = 20
sharpened_image = apply_sharpening_filter(input_image, strength)

combined_image = np.hstack((input_image, smoothed_image, sharpened_image))
plt.imshow(combined_image, cmap="gray")
plt.title("Original | Smoothed | Sharpened")
plt.axis("off")
plt.show()

smoothed_path = "smoothed_image.jpg"
sharpened_path = "sharpened_image.jpg"
cv2.imwrite(smoothed_path, smoothed_image)
cv2.imwrite(sharpened_path, sharpened_image)

print(f"Smoothed image saved at: {smoothed_path}")
print(f"Sharpened image saved at: {sharpened_path}")
```

Marwadi University	**Marwari University** **Faculty of Technology** **Department of Information and Communication Technology**	
Subject: Digital Signal and Image Processing(01CT0513)	**Aim:** Simulate smoothing and sharpening operations on images using frequency domain filters.	
Experiment No: 10	**Date:**	**Enrollment No: 92200133030**

Output:-

Original Image	Sharpened Image	Smoothed Image
		

Conclusion :-