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 operation on images using spatial filters.	
Experiment No: 08	**Date:**	**Enrollment No: 92200133030**

Aim: Simulate smoothing and sharpening operation on images using spatial filters.

Theory:-

- Smoothing and sharpening are common image enhancement techniques used to modify the spatial characteristics of an image.

Smoothing:-

- Smoothing, also known as blurring, is used to reduce noise and remove fine details in an image. It works by applying a low-pass filter that averages the pixel values within a neighborhood, thus producing a blurred effect.

Sharpening:-

- Sharpening is used to enhance the edges and fine details in an image. It works by applying a high-pass filter that amplifies the differences between neighboring pixel values, thus increasing the contrast and emphasizing edges.

Programm:-

```
import cv2  # type: ignore
import numpy as np

# Load the image
image = cv2.imread('./Images.jpg', 0)

# Define a low-pass filter kernel (Gaussian)
kernel_size = 5
sigma = 1.5
low_pass_filter = cv2.getGaussianKernel(kernel_size, sigma)
low_pass_filter = low_pass_filter * low_pass_filter.T

# Apply the low-pass filter to the image
smoothed_image = cv2.filter2D(image, -1, low_pass_filter)

# Define a high-pass filter kernel (Laplacian)
high_pass_filter = np.array([[0, -1, 0],
                             [-1, 4, -1],
                             [0, -1, 0]], dtype=np.float32)

# Apply the high-pass filter to the image
sharpened_image = cv2.filter2D(image, -1, high_pass_filter)

# Save the images
cv2.imwrite("original_image.jpg", image)
```

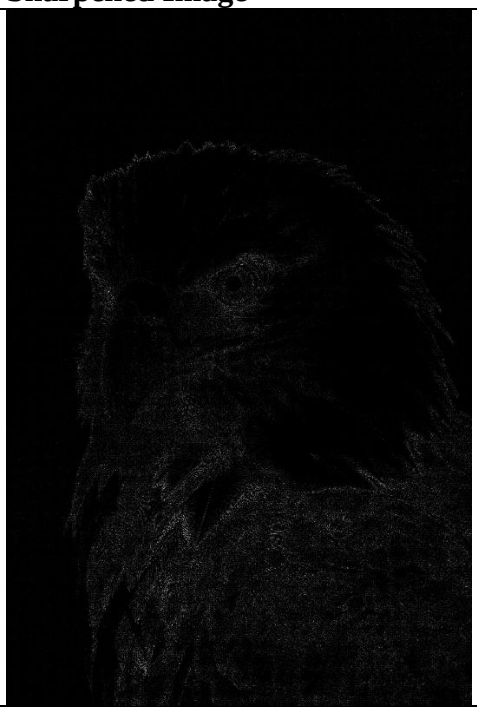
Marwadi University	**Marwari University** **Faculty of Technology** **Department of Information and Communication Technology**	
Subject: Digital Signal and Image Processing(01CT0513)	**Aim:** Simulate smoothing and sharpening operation on images using spatial filters.	
Experiment No: 08	**Date:**	**Enrollment No: 92200133030**

```
cv2.imwrite("smoothed_image.jpg", smoothed_image)
cv2.imwrite("sharpened_image.jpg", sharpened_image)

# Wait for a key press and then close the windows
cv2.waitKey(0)
cv2.destroyAllWindows()

print("Images saved as 'original_image.jpg', 'smoothed_image.jpg', and
'sharpened_image.jpg'")
```

Output :-

Original Image	Smoothed Image	Sharpened Image
		

Conclusion :-