**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)

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 :-**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_