Code:

**import** cv2

**import** numpy **as** np

**import** matplotlib.pyplot **as** plt

**def** apply\_smoothing\_filter(*image*, *kernel\_size*):

*# Apply smoothing filter to the image*

    smoothed\_image **=** cv2.blur(image, (kernel\_size, kernel\_size))

**return** smoothed\_image

**def** apply\_sharpening\_filter(*image*):

*# Create a sharpening kernel*

    kernel **=** np.array([[0, **-**1, 0], [**-**1, 5, **-**1], [0, **-**1, 0]])

*# Apply the sharpening kernel to the image*

    sharpened\_image **=** cv2.filter2D(image, **-**1, kernel)

**return** sharpened\_image

*# Load the input image*

image\_path **=** "D:\DSIP\codes\ex1\_2.png"

input\_image **=** cv2.imread(image\_path)

*# Apply smoothing filter*

smoothed\_image **=** apply\_smoothing\_filter(input\_image, *kernel\_size***=**5)

*# Apply sharpening filter*

sharpened\_image **=** apply\_sharpening\_filter(input\_image)

*# Display the original image and the filtered images side by side*

combined\_image **=** np.hstack((input\_image, smoothed\_image, sharpened\_image))

cv2.imshow("Original | Smoothed | Sharpened", combined\_image)

cv2.waitKey(0)

*# Save the filtered images (optional)*

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}")

Output:

