**Practical No: 9**

Page | 25

**Aim** Write a program to apply various image enhancement using image derivatives by implementing smoothing, sharpening, and unsharp masking filters for generating suitable images for specific application requirements.

**● Smoothing with scipy ndimage**

The scipy ndimage module provides a function named percentile\_filter(), which is a generic version of the median filter.The following code block demonstrate how to use this filter.

**Code:**

lena **=** misc**.**imread('../images/lena.jpg')

* *add salt-and-pepper noise to the input image* noise **=** np**.**random**.**random(lena**.**shape) lena[noise **>** 0.9] **=** 255

lena[noise **<** 0.1] **=** 0

plot\_image(lena,'noisy image') pylab**.**show()

fig **=** pylab**.**figure(figsize**=**(20,15))

i **=** 1

**for** p **in** range(25, 100, 25):

**for** k **in** range(5, 25, 5):

pylab**.**subplot(3,4,i)

filtered **=** ndimage**.**percentile\_filter(lena, percentile**=**p, size**=**(k,k,1))

plot\_image(filtered, str(p) **+** ' percentile, ' **+** str(k) **+** 'x' **+** str(k) **+** ' kernel')

i **+=** 1

pylab**.**show()

**output:**



Page | 26

**● Sharpening with Laplacian**

**from** skimage.filters **import** laplace

im **=** rgb2gray(imread('../images/me8.jpg'))

im1 **=** np**.**clip(laplace(im) **+** im, 0, 1)

pylab**.**figure(figsize**=**(10,15))

pylab**.**subplot(211), plot\_image(im,'original image')

pylab**.**subplot(212), plot\_image(im1,'sharpened image')

pylab**.**tight\_layout()

pylab**.**show()





Page | 27

**● Unsharp masking**

**def** rgb2gray(im):

'''

the input image is an RGB image

with pixel values for each channel in [0,1]

'''

**return** np**.**clip(0.2989 **\*** im[**...**,0] **+** 0.5870 **\*** im[**...**,1] **+** 0.1140 **\*** im[**...**,2], 0, 1)

im **=** rgb2gray(img\_as\_float(misc**.**imread('../images/me4.jpg')))

im\_blurred **=** ndimage**.**gaussian\_filter(im, 5)

im\_detail **=** np**.**clip(im **-** im\_blurred, 0, 1)

pylab**.**gray()

fig, axes **=** pylab**.**subplots(nrows**=**2, ncols**=**3, sharex**=True**, sharey**=True**, figsize**=**(15, 15))

axes **=** axes**.**ravel()

axes[0]**.**set\_title('Original image', size**=** 15), axes[0]**.**imshow(im) axes[1]**.**set\_title('Blurred image, sigma=5', size**=** 15), axes[1]**.**imshow(im\_blurred)axes[2]**.**set\_title('Detail image', size**=** 15), axes[2]**.**imshow(im\_detail)

alpha **=** [1, 5, 10]

**for** i **in** range(3):

im\_sharp **=** np**.**clip(im **+** alpha[i]**\***im\_detail, 0, 1)

axes[3**+**i]**.**imshow(im\_sharp), axes[3**+**i]**.**set\_title('Sharpened image, alpha=' **+** str(alpha[i]), size**=** 15) **for** ax **in** axes:

ax**.**axis('off')

fig**.**tight\_layout()

pylab**.**show()



Page | 28

