Page| 06

**Practical No: 03**

**Aim:** Write program to demonstrate the following aspects of signal on Sound/image data.

**A:** Convolution operation.

Convolution is a simple mathematical operation which is fundamental to many common image processing operators. Convolution provides a way of `multiplying together' two arrays of numbers, generally of different sizes, but of the same dimensionality, to produce a third array of numbers of the same dimensionality.

**Program Code:**

im = rgb2gray(imread('../images/cameraman.jpg')).astype(float)

print(np.max(im))

print(im.shape) blur\_box\_kernel = np.ones((3,3)) / 9

edge\_laplace\_kernel = np.array([[0,1,0],[1,-4,1],[0,1,0]])

im\_blurred = signal.convolve2d(im, blur\_box\_kernel)

im\_edges = np.clip(signal.convolve2d(im, edge\_laplace\_kernel), 0, 1)

fig, axes = pylab.subplots(ncols=3, sharex=True, sharey=True, figsize=(18,6))

axes[0].imshow(im, cmap=pylab.cm.gray)

axes[0].set\_title('Original Image', size=20)

axes[1].imshow(im\_blurred, cmap=pylab.cm.gray)

axes[1].set\_title('Box Blur', size=20)

axes[2].imshow(im\_edges, cmap=pylab.cm.gray)

axes[2].set\_title('Laplace Edge Detection', size=20)

for ax in axes:

ax.axis('off')

pylab.show()

**Output:**



Page| 07

**B:** Template Matching.

Page 3

Template matching is the process of moving the template over the entire image and calculating the similarity between the template and the covered window on the image.

**Program Code:**

from PIL import Image

import numpy as np

from scipy import misc

from scipy import signal

from scipy.signal import correlate2d

from matplotlib import pyplot

import pylab as pb

face\_img=misc.face(gray=True)-misc.face(gray=True).mean()

temp\_img=np.copy(face\_img[300:365,670:750])

face\_img=face\_img+np.random.randn(\*face\_img.shape)\*50

cor=signal.correlate2d(face\_img,temp\_img,boundary='symm',mode='same')

y,x=np.unravel\_index(np.argmax(cor),cor.shape)

fig,(ax\_original,ax\_template,ax\_correlation)=pb.subplots(3,1,figsize=(6,15))

ax\_original.imshow(face\_img,cmap='gray')

ax\_original.set\_title('original\_image',size=20)

ax\_original.set\_axis\_off()

ax\_template.imshow(temp\_img,cmap='gray')

ax\_template.set\_title('template image',size=20)

ax\_template.set\_axis\_off()

ax\_correlation.imshow(cor,cmap='gray')

ax\_correlation.set\_title('correlated image',size=20)

ax\_correlation.set\_axis\_off()

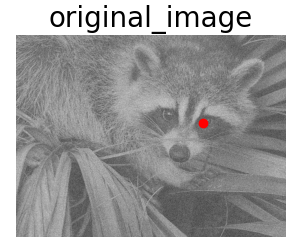
ax\_original.plot(x,y,'ro')

fig.show()

**Output:**







Page 7