**PAPER – III         PRACTICAL NO. : 04**

**AIM : WRITE A PROGRAM TO APPLY VARIOUS ENHANCEMENT ON IMAGE DERIVATIVES BY IMPLEMENTING GRADIENT AND LAPLACIAN OPERATION.**

**ROLL NO. : 02                                BATCH : M.SC PART-I**

**DATE : 17/10/22**

**CODE :**

import numpy as np

from scipy import signal,misc,ndimage

from skimage import filters,feature,img\_as\_float

from skimage.io import imread

from skimage.color import rgb2gray

from PIL import Image,ImageFilter

import matplotlib.pylab as pylab

import matplotlib.pylab as plt

from skimage.util import random\_noise

from skimage.filters import gaussian

from scipy import signal

import numpy as np

def plot\_image (image,title):

pylab.imshow(image),pylab.title(title,size=20),pylab.axis('off')

ker\_x=[[-1,1]]

ker\_y=[[-1],[1]]

im=rgb2gray(imread('elephant.jpg'))

im\_x=signal.convolve2d(im,ker\_x,mode='same')

im\_y=signal.convolve2d(im,ker\_y,mode='same')

im\_mag=np.sqrt(im\_x\*\*2+im\_y\*\*2)

im\_dir=np.arctan(im\_y/im\_x)

pylab.gray()

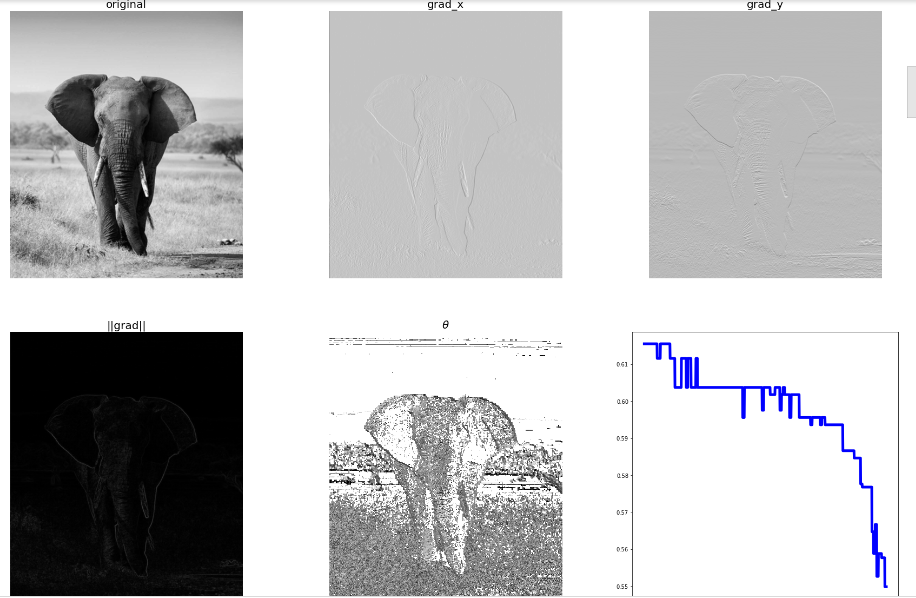
pylab.figure(figsize=(30,20))

pylab.subplot(231),plot\_image(im,'original'),pylab.subplot(232),plot\_image(im\_x,'grad\_x')

pylab.subplot(233),plot\_image(im\_y,'grad\_y'),pylab.subplot(234),plot\_image(im\_mag,'||grad||')

pylab.subplot(235),plot\_image(im\_dir,r'$\theta$'),pylab.subplot(236)

pylab.plot(range(im.shape[1]),im[0,:],'b-',label=r'$f(x,y)|\_{x=0}$',linewidth=5)

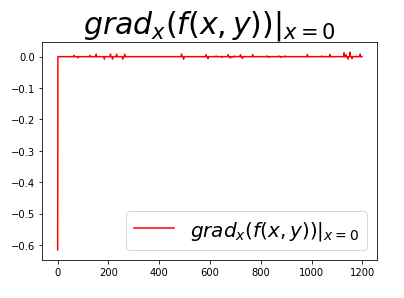


pylab.plot(range(im.shape[1]),im\_x[0,:],'r-',label=r'$grad\_x(f(x,y))|\_{x=0}$')

pylab.title(r'$grad\_x(f(x,y))|\_{x=0}$',size=30)

pylab.legend(prop={'size':20})

pylab.show()



ker\_x=[[-1,1]]

ker\_y=[[-1],[1]]

im=rgb2gray(imread('elephant.jpg'))

#im=random\_noise(im,var=sigma\*\*2)

#im=gaussian(im,sigma=0.25)

print(np.max(im))

im\_x=np.clip(signal.convolve2d(im,ker\_x,mode='same'),0,1)

im\_y=np.clip(signal.convolve2d(im,ker\_y,mode='same'),0,1)

im\_mag=np.sqrt(im\_x\*\*2+im\_y\*\*2)

im\_ang=np.arctan(im\_y/im\_x)

plt.gray()

plt.figure(figsize=(10,15))

plt.subplot(321)

plt.imshow(im)

plt.title('original',size=30)

plt.axis('off')

plt.subplot(322)

plt.imshow(im\_x)

plt.title('grad\_x',size=30)

plt.axis('off')

plt.subplot(323)

plt.imshow(im\_y)

plt.title('grad\_y',size=30)

plt.axis('off')

plt.subplot(324)

plt.imshow(im\_mag)

plt.title('||grad||',size=30)

plt.axis('off')

plt.subplot(325)

plt.imshow(im\_ang)

plt.title(r'$\theta$',size=30)

plt.axis('off')

plt.subplot(326)

im=np.zeros((im.shape[0],im.shape[1],3))

im[...,0]=im\_mag\*np.sin(im\_ang)

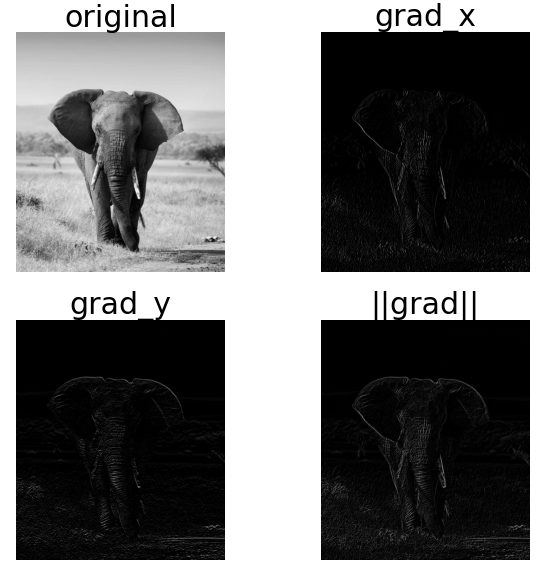
im[...,1]=im\_mag\*np.cos(im\_ang)

plt.imshow(im)

plt.title(r'||grad||+$\theta$',size=30)

plt.axis('off')

plt.show()





#Laplacian

ker\_laplacian=[[0,-1,0],[-1,4,-1],[0,-1,0]]

im=rgb2gray(imread('cgvr.png'))

im1=np.clip(signal.convolve2d(im,ker\_laplacian,mode='same'),0,1)

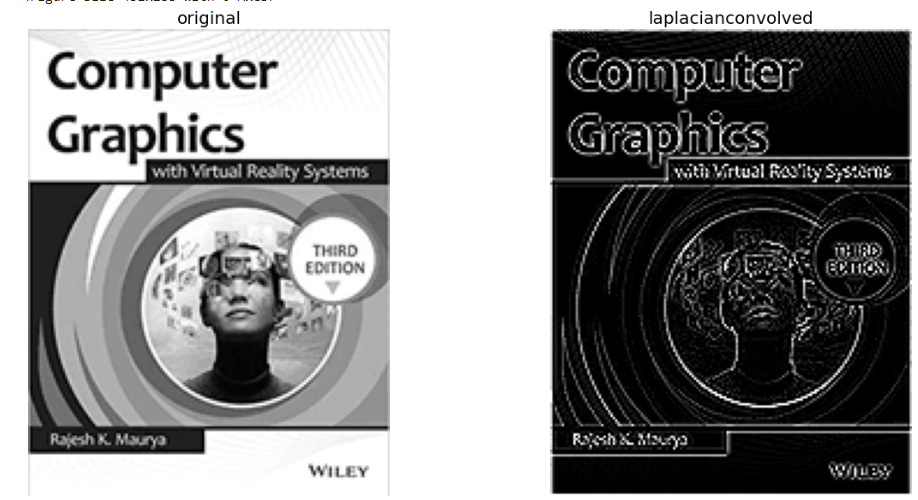
pylab.gray()

pylab.figure(figsize=(20,10))

pylab.subplot(121),plot\_image(im,'original')

pylab.subplot(122),plot\_image(im1,'laplacianconvolved')

pylab.show()



ker\_x=[[-1,1]]

ker\_y=[[-1],[1]]

im=rgb2gray(imread('board.PNG'))

sigma=0.25

sign=np.random.random(im.shape)

sign[sign<=0.5]=-1

sign[sign>0.5]=1

im=random\_noise(im,var=sigma\*\*2)

im=gaussian(im,sigma=0.25)

print(np.max(im))

im\_x=signal.convolve2d(im,ker\_x,mode='same')

im\_y=signal.convolve2d(im,ker\_y,mode='same')

im\_mag=np.sqrt(im\_x\*\*2+im\_y\*\*2)

im\_ang=np.arctan(im\_y/im\_x)

plt.gray()

plt.figure(figsize=(30,20))

plt.subplot(231)

plt.imshow(im)

plt.title('original',size=30)

plt.axis('off')

plt.subplot(232)

plt.imshow(im\_x)

plt.title('grad\_x',size=30)

plt.axis('off')

plt.subplot(233)

plt.imshow(im\_y)

plt.title('grad\_y',size=30)

plt.axis('off')

plt.subplot(234)

plt.imshow(im\_mag)

plt.title('||grad||',size=30)

plt.axis('off')

plt.subplot(235)

plt.imshow(im\_ang)

plt.title(r'$\theta$',size=30)

plt.axis('off')

plt.subplot(236)

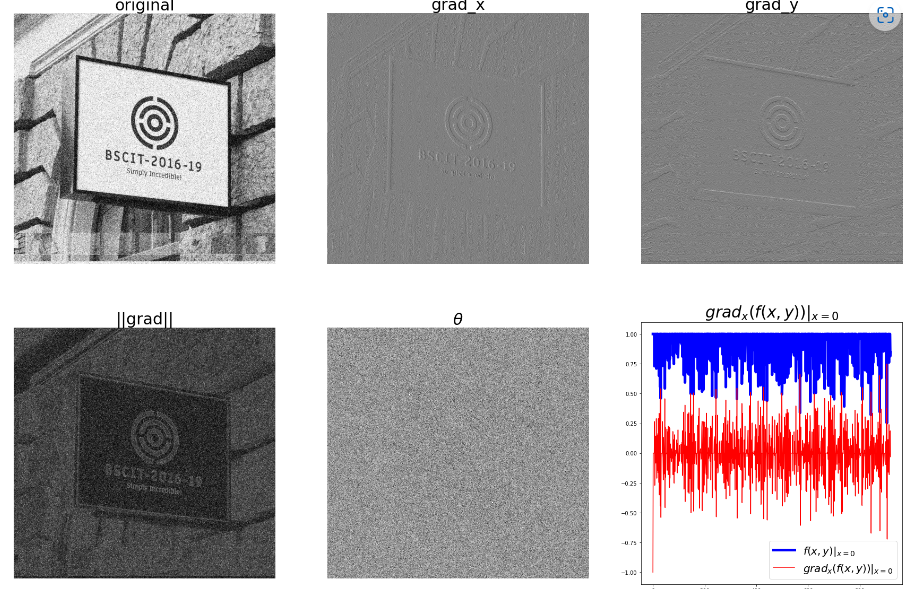
plt.plot(range(im.shape[1]),im[0,:],'b-',label=r'$f(x,y)|\_{x=0}$',linewidth=5)

plt.plot(range(im.shape[1]),im\_x[0,:],'r-',label=r'$grad\_x(f(x,y))|\_{x=0}$')

plt.title(r'$grad\_x(f(x,y))|\_{x=0}$',size=30)

plt.legend(prop={'size':20})

plt.show()



from skimage.filters import laplace

im=rgb2gray(imread('board.PNG'))

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

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

pylab.subplot(121),plot\_image(im,'originalimage')

pylab.subplot(122),plot\_image(im1,'sharpenedimage')

pylab.tight\_layout()

pylab.show()