import matplotlib.pylab as pylab

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

def plot\_image (image,title):

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

im=Image.open('rajesh.png').convert('L')

pylab.gray()

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

pylab.subplot(3,2,1),plot\_image(im,'original')

edges=filters.roberts(im)

pylab.subplot(3,2,2),plot\_image(edges,'roberts')

edges=filters.scharr(im)

pylab.subplot(3,2,3),plot\_image(edges,'scharr')

edges=filters.sobel(im)

pylab.subplot(3,2,4),plot\_image(edges,'sobel')

edges=filters.prewitt(im)

pylab.subplot(3,2,5),plot\_image(edges,'prewitt')

edges=np.clip(filters.laplace(im),0,1)

pylab.subplot(3,2,6),plot\_image(edges,'laplace')

pylab.subplots\_adjust(wspace=0.1,hspace=0.1)

pylab.show()

im=Image.open('rajesh.png').convert('L')

pylab.gray()

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

pylab.subplot(2,2,1)

plot\_image(im,'original')

pylab.subplot(2,2,2)

edges\_x=filters.sobel\_h(im)

plot\_image(np.clip(edges\_x,0,1),'sobel\_x')

pylab.subplot(2,2,3)

edges\_y=filters.sobel\_v(im)

plot\_image(np.clip(edges\_y,0,1),'sobel\_y')

pylab.subplot(2,2,4)

edges=filters.sobel(im)

plot\_image(np.clip(edges,0,1),'sobel')

pylab.subplots\_adjust(wspace=0.1,hspace=0.1)

pylab.show()

import numpy as np

import matplotlib.pyplot as plt

from scipy import ndimage as ndi

from skimage.util import random\_noise

from skimage import feature

#Generatenoisyimageofasquare

image = np.zeros((128,128),dtype=float)

image[32:-32,32:-32]=1

image=ndi.rotate(image,15,mode='constant')

image=ndi.gaussian\_filter(image,4)

image=random\_noise(image,mode='speckle',mean=0.05)

#ComputetheCannyfilterfortwovaluesofsigma

edges1=feature.canny(image)

edges2=feature.canny(image,sigma=3)

#displayresults

fig,ax=plt.subplots(nrows=1,ncols=3,figsize=(8,3))

ax[0].imshow(image,cmap='gray')

ax[0].set\_title('noisyimage',fontsize=10)

ax[1].imshow(edges1,cmap='gray')

ax[1].set\_title(r'Cannyfilter,$\sigma=1$',fontsize=10)

ax[2].imshow(edges2,cmap='gray')

ax[2].set\_title(r'Cannyfilter,$\sigma=3$',fontsize=10)

for a in ax:

a.axis('off')

fig.tight\_layout()

plt.show()