PAPER – III         PRACTICAL NO. : 07

AIM :  WRITE A PROGRAM TO EXTRACT IMAGE FEATURES BY IMPLEMENTING METHODS

LIKE CORNER AND BLOB DETECTOR, HOG AND HAAR FEATURES.

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

DATE : 15/11/22

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

from scipy.signal import convolve2d

#fromscipy.miscimportimread

from scipy.ndimage import gaussian\_filter

from numpy import pi

def plot\_kernel (kernel,s,name):

pylab.imshow(kernel,cmap='YlOrRd')

def LOG(k=12,s=3):

n=2\*k+1

kernel=np.zeros((n,n))

for i in range(n):

for j in range(n):

kernel[i,j]=-(1-((i-k)\*\*2+(j-k)\*\*2)/(2.\*s\*\*2))\*np.exp(-((i-k)\*\*2+(j-k)\*\*2)/(2.\*s\*\*2))/(pi\*s\*\*4)

kernel=np.round(kernel/np.sqrt((kernel\*\*2).sum()),3)

return kernel

def DOG(k=12,s=3):

n=2\*k+1

s1,s2=s\*np.sqrt(2),s/np.sqrt(2)

kernel=np.zeros((n,n))

for i in range(n):

for j in range(n):

kernel[i,j]=np.exp(-((i-k)\*\*2+(j-k)\*\*2)/(2.\*s1\*\*2))/(2\*pi\*s1\*\*2)-np.exp(-((i-k)\*\*2+(j-k)\*\*2)/(2.\*s2\*\*2))/(2\*pi\*s2\*\*2)

kernel=np.round(kernel/np.sqrt((kernel\*\*2).sum()),3)

return kernel

s=3

#sigmavalueforLoG

img=rgb2gray(imread('rkm.jpg'))

kernel=LOG()

outimg=convolve2d(img,kernel)

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

pylab.subplot(221),pylab.title('LOGkernel',size=20),plot\_kernel(kernel,s,'DOG')

pylab.subplot(222),pylab.title('outputimagewithLOG',size=20)

pylab.imshow(np.clip(outimg,0,1),cmap='gray')

#clipthepixelvaluesinbetween0and1

kernel=DOG()

outimg=convolve2d(img,DOG())

pylab.subplot(223),pylab.title('DOGkernel',size=20),plot\_kernel(kernel,s,'DOG')

pylab.subplot(224),pylab.title('outputimagewithDOG',size=20)

pylab.imshow(np.clip(outimg,0,1),cmap='gray')

pylab.show()