import numpy as np

import cv2 as cv

import math

def gaussian(m,n,sigma):

gauss = np.zeros((m,n))

pi=3.1416

m //=2

n //=2

for x in range(-m,m+1):

for y in range(-n,n+1):

r = (x\*\*2 + y\*\*2)/(2\*sigma\*\*2)

gauss[x+m][y+n] = math.exp(-r)/(2\*pi\*sigma\*\*2)

return gauss

def bilateral(img,filt,sigma):

w1 = filt.shape[0]

h2 = filt.shape[1]

w = w1//2

h = h2//2

img = img/255

m = img.shape[0]

n = img.shape[1]

out = np.zeros((m,n),np.float32)

pi = 3.1416

for i in range(m):

for j in range(n):

rs = 0.0

factor = 0.0

range\_domain = np.zeros((w1,h2),np.float32)

for x in range(-w,w+1):

for y in range(-h,h+1):

if i-x>=0 and i-x<m and j-y>=0 and j-y<n:

v = img[i][j] - img[i-x][j-y]

v = (math.exp(-(v\*v)/(2\*sigma\*sigma))/((math.sqrt(2\*pi))\*sigma))

range\_domain[x+w][y+h] = filt[x+w][y+h] \* v

factor += v

for x in range(-w,w+1):

for y in range(-h,h+1):

if i-x>=0 and i-x<m and j-y>=0 and j-y<n:

rs += (range\_domain[x+w][y+h] \* img[i-x][j-y])

rs = (rs/factor)

out[i][j] = rs

out = out\*255

out = cv.normalize(out, None, 0, 1.0,cv.NORM\_MINMAX, dtype=cv.CV\_32F)

return out

img = cv.imread("Lena.jpg",cv.IMREAD\_GRAYSCALE)

cv.imshow("original",img)

g\_s = int(input("Enter value of gaussian sigma:\n"))

p = (2\*g\_s) + 1

gauss\_filt = gaussian(p,p,g\_s)

w = int(input("Enter kernel width:\n"))

h = int(input("Enter kernel height:\n"))

g\_r = int(input("Enter value of sigma:\n"))

avg\_filt = np.ones((w,h))/(w\*h)

cv.imshow("final\_guass\_bilateral",bilateral(img,gauss\_filt,g\_r))

cv.imshow("final\_average\_bilateral",bilateral(img,avg\_filt,g\_s))

cv.waitKey()

cv.destroyAllWindows()