import cv2

import numpy as np

import math

import matplotlib.pyplot as plt

def main():

path = "C:/Users/DELL 3468/Desktop/TY Shit/Python/Image Restoration Tool/Python-OpenCV3-master/Dataset/"

imgpath = path + "Damaged Image.tiff"

maskpath = path + "Mask.tiff"

img = cv2.imread(imgpath, 1)

img = cv2.cvtColor(img, cv2.COLOR\_BGR2RGB)

mask = cv2.imread(maskpath, 0)

height = np.size(img, 0)

width = np.size(img, 1)

createdMask = np.zeros([width, height])

grayimg = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY)

#temp = np.asarray(PIL.Image.open('test.jpg'))

#for j in grayimg:

# iii = np.asarray([[i[0], i[1]] for i in j])

for i in range(1,width-1):

for j in range(1,height-1):

#ii = iii[i,j]

if grayimg[i,j] > 0:

createdMask[i,j] = 0

else:

createdMask[i,j] = 255

gb\_kernel = cv2.getGaborKernel((8, 8), 0.01, np.pi/2, 20.0, 0.5, 0, ktype=cv2.CV\_32F)

gcreatedMask = cv2.filter2D(createdMask, cv2.CV\_8U, gb\_kernel.transpose())

mse = 0

for i in range(1,width):

for j in range(1, height):

mse = mse + (float(mask[i, j]) - float(gcreatedMask[i, j])) \*\* 2

mse = mse / (width \* height)

mse = math.sqrt(mse)

mse2 = 0

for i in range(1, width):

for j in range(1, height):

mse2 = mse2 + (float(mask[i, j]) - float(createdMask[i, j])) \*\* 2

mse2 = mse2 / (width \* height)

mse2 = math.sqrt(mse2)

print(mse)

print(mse2)

if \_\_name\_\_ == "\_\_main\_\_":

main()