import os  
  
import pandas as pd  
import cv2 as cv  
import numpy as np  
import random  
import matplotlib.pyplot as plt

Напишем функции тестирования изображений методами MSE – Среднеквадратического отклонения, замеряющего среднюю разницу между каждым отдельным пикселем. PSNR - отношение пикового значения сигнала к шуму означает соотношение между максимально возможной мощностью сигнала и мощностью искажающего шума, который влияет на точность его представления. SSIM - метод прогнозирования воспринимаемого качества цифрового телевидения и кинематографических изображений, а также других видов цифровых изображений и видео.

def PSNR(I1, I2):  
 s1 = cv.absdiff(I1, I2) s1 = np.float32(s1) s1 = s1 \* s1 sse = s1.sum() if sse <= 1e-10return 0 else:  
 shape = I1.shape  
 mse = 1.0 \* sse / (shape[0] \* shape[1] \* shape[2])  
 psnr = 10.0 \* np.log10((255 \* 255) / mse)  
 return psnr  
  
def MSSISM(i1, i2):  
 C1 = 6.5025  
 C2 = 58.5225  
I1 = np.float32(i1) I2 = np.float32(i2)  
 I2\_2 = I2 \* I2 I1\_2 = I1 \* I1 I1\_I2 = I1 \* I2 mu1 = cv.GaussianBlur(I1, (11, 11), 1.5)  
 mu2 = cv.GaussianBlur(I2, (11, 11), 1.5)  
 mu1\_2 = mu1 \* mu1  
 mu2\_2 = mu2 \* mu2  
 mu1\_mu2 = mu1 \* mu2  
 sigma1\_2 = cv.GaussianBlur(I1\_2, (11, 11), 1.5)  
 sigma1\_2 -= mu1\_2  
 sigma2\_2 = cv.GaussianBlur(I2\_2, (11, 11), 1.5)  
 sigma2\_2 -= mu2\_2  
 sigma12 = cv.GaussianBlur(I1\_I2, (11, 11), 1.5)  
 sigma12 -= mu1\_mu2  
 t1 = 2 \* mu1\_mu2 + C1  
 t2 = 2 \* sigma12 + C2  
 t3 = t1 \* t2 *# t3 = ((2\*mu1\_mu2 + C1).\*(2\*sigma12 + C2))* t1 = mu1\_2 + mu2\_2 + C1  
 t2 = sigma1\_2 + sigma2\_2 + C2  
 t1 = t1 \* t2 *# t1 =((mu1\_2 + mu2\_2 + C1).\*(sigma1\_2 + sigma2\_2 + C2))* ssim\_map = cv.divide(t3, t1) mssim = cv.mean(ssim\_map) return mssim  
  
def MSE(img1, img2):  
 squared\_diff = (img1 - img2) \*\* 2  
 summed = np.sum(squared\_diff)  
 num\_pix = img1.shape[0] \* img1.shape[1] *# img1 and 2 should have same shape* err = summed / num\_pix  
 return err

1. Подготовить несколько тестов по 4 изображения, из которых второе два будут похожими, с точки зрения человека (например, фотография сделанная с немного другого ракурса),а третье будет значительно отличаться от первого (фотография другой сцены), четвертое изображение получить, применив какие либо шумы или размытие к первому. Подготовить не менее 2 тестов, отличающихся характером изображений (темные, светлые, большие, малые и т.п.)img1 = cv.imread(**"cat1.jpg"**)  
*# img1 = cv.resize(img1, (960, 540), interpolation=cv.INTER\_AREA)*img2 = cv.imread(**"cat2.jpg"**)  
*# img2 = cv.resize(img2, (960, 540), interpolation=cv.INTER\_AREA)*img3 = cv.imread(**"122121.png"**)  
*# img3 = cv.resize(img3, (960, 540), interpolation=cv.INTER\_AREA)*img4 = img1.copy()  
img4 = cv.medianBlur(img4,7)

Использованные изображения:



Сравнить значения метрик MSE, PSNR и SSIM попарно вычисленных между первым изображением и остальными, в каждом тестеprint(**"MSE img1, img1 :"**,MSE(img1, img1))  
print(**"MSE img1, img2 :"**,MSE(img1, img2))  
print(**"MSE img1, img3 :"**,MSE(img1, img3))  
print(**"MSE img1, img4 :"**,MSE(img1, img4))  
  
print(**"PSNR img1, img1 :"**,PSNR(img1, img1))  
print(**"PSNR img1, img2 :"**,PSNR(img1, img2))  
print(**"PSNR img1, img3 :"**,PSNR(img1, img3))  
print(**"PSNR img1, img4 :"**,PSNR(img1, img4))  
  
print(**"MSSISM img1, img1 :"**,MSSISM(img1, img1))  
print(**"MSSISM img1, img2 :"**,MSSISM(img1, img2))  
print(**"MSSISM img1, img3 :"**,MSSISM(img1, img3))  
print(**"MSSISM img1, img4 :"**,MSSISM(img1, img4))

Результат работы программы:

MSE img1, img1 : 0.0

MSE img1, img2 : 260.23038532021604

MSE img1, img3 : 327.88830680941356

MSE img1, img4 : 77.93183545524691

PSNR img1, img1 : 0

PSNR img1, img2 : 20.174905134089364

PSNR img1, img3 : 6.358983905683342

PSNR img1, img4 : 32.085938607766536

MSSISM img1, img1 : (1.0, 1.0, 1.0, 0.0)

MSSISM img1, img2 : (0.6297633921260658, 0.6696239588316204, 0.6773239677906017, 0.0)

MSSISM img1, img3 : (0.4023127217485218, 0.4854457368548222, 0.6146369449641317, 0.0)

MSSISM img1, img4 : (0.7796047800896477, 0.7837975101107292, 0.7832668957661035, 0.0)

2. Оценить значения метрик MSE, PSNR и SSIM для искусственно зашумленного изображения. В качестве шума реализовать случайное заполнение заданного процента пикселей изображения случайными числами от 0 до 255 (для цветных изображений – в каждом канале). Исследовать влияние процента зашумленности изображения на значения метрик.def noiser(img, percent = 50):  
 noise = img.copy()  
  
 for y in range(noise.shape[0]):  
 for x in range(noise.shape[1]):  
  
 p = random.randint(0, 100)  
 if p < percent:  
 for c in range(noise.shape[2]):  
 noise[y,x,c] = np.clip(noise[y,x,c], random.randint(1,255), random.randint(1,255))  
 else:  
 pass  
 return noise  
  
  
noised1 = noiser(img1, 5)  
noised2 = noiser(img1, 30)  
noised3 = noiser(img1, 70)  
  
cv.imshow(**"noised 5% "**, noised1)  
cv.imshow(**"noised 30% "**, noised2)  
cv.imshow(**"noised 70% "**, noised3)  
  
print(**"MSE img1, noised 5% :"**,MSE(img1, noised1))  
print(**"MSE img1, noised 30% :"**,MSE(img1, noised2))  
print(**"MSE img1, noised 70% :"**,MSE(img1, noised3))  
  
print(**"PSNR img1, noised 5% :"**,PSNR(img1, noised1))  
print(**"PSNR img1, noised 30% :"**,PSNR(img1, noised2))  
print(**"PSNR img1, noised 70% :"**,PSNR(img1, noised3))  
  
print(**"MSSISM img1, noised 5% :"**,MSSISM(img1, noised1))  
print(**"MSSISM img1, noised 30% :"**,MSSISM(img1, noised2))  
print(**"MSSISM img1, noised 70% :"**,MSSISM(img1, noised3))

Обработанные фотографии:



Результат работы программы:

MSE img1, noised 5% : 12.479062017746914

MSE img1, noised 30% : 75.15738522376543

MSE img1, noised 70% : 175.43929157021606

PSNR img1, noised 5% : 24.25099615914473

PSNR img1, noised 30% : 16.462115946213007

PSNR img1, noised 70% : 12.788298128900689

MSSISM img1, noised 5% : (0.49494904148126384, 0.5521155418994521, 0.5594805170201617, 0.0)

MSSISM img1, noised 30% : (0.08103705325119989, 0.10529832756179298, 0.11093944538022878, 0.0)

MSSISM img1, noised 70% : (0.027402004004263358, 0.036804896466254663, 0.03933778307623698, 0.0)

3. Оценить значения MSE, PSNR и SSIM при JPEG сжатии с различными параметрами качества (с шагом 5, от 0 до 100).from PIL import Image  
im1 = Image.open(**"cat1.jpg"**)  
  
IMAGE\_5 = os.path.join(**'./cat5.jpg'**)  
im1.save(IMAGE\_5,**"JPEG"**, quality=5)  
IMAGE\_10 = os.path.join(**'./cat10.jpg'**)  
im1.save(IMAGE\_10,**"JPEG"**, quality=10)  
IMAGE\_20 = os.path.join(**'./cat20.jpg'**)  
im1.save(IMAGE\_20,**"JPEG"**, quality=20)  
IMAGE\_40 = os.path.join(**'./cat40.jpg'**)  
im1.save(IMAGE\_40,**"JPEG"**, quality=40)  
  
cat5 = cv.imread(**"cat5.jpg"**)  
cat10 = cv.imread(**"cat10.jpg"**)  
cat20 = cv.imread(**"cat20.jpg"**)  
cat40 = cv.imread(**"cat40.jpg"**)  
  
print(**"MSE img1, JPEG compression, quality=5 :"**,MSE(img1, cat5))  
print(**"MSE img1, JPEG compression, quality=10 :"**,MSE(img1, cat10))  
print(**"MSE img1, JPEG compression, quality=20 :"**,MSE(img1, cat20))  
print(**"MSE img1, JPEG compression, quality=40 :"**,MSE(img1, cat40))  
  
print(**"PSNR img1, JPEG compression, quality=5 :"**,PSNR(img1, cat5))  
print(**"PSNR img1, JPEG compression, quality=10 :"**,PSNR(img1, cat10))  
print(**"PSNR img1, JPEG compression, quality=20 :"**,PSNR(img1, cat20))  
print(**"PSNR img1, JPEG compression, quality=40 :"**,PSNR(img1, cat40))  
  
print(**"MSSISM img1, JPEG compression, quality=5 :"**,MSSISM(img1, cat5))  
print(**"MSSISM img1, JPEG compression, quality=10 :"**,MSSISM(img1, cat10))  
print(**"MSSISM img1, JPEG compression, quality=20 :"**,MSSISM(img1, cat20))  
print(**"MSSISM img1, JPEG compression, quality=40 :"**,MSSISM(img1, cat40))

Обработанные фотографии:



Результат работы программы:

MSE img1, JPEG compression, quality=5 : 219.5207277199074

MSE img1, JPEG compression, quality=10 : 131.05692756558642

MSE img1, JPEG compression, quality=20 : 66.65839891975308

MSE img1, JPEG compression, quality=40 : 42.097530381944445

PSNR img1, JPEG compression, quality=5 : 26.356645892931564

PSNR img1, JPEG compression, quality=10 : 30.712888267045617

PSNR img1, JPEG compression, quality=20 : 34.231973362051086

PSNR img1, JPEG compression, quality=40 : 36.44849670652234

MSSISM img1, JPEG compression, quality=5 : (0.7081735635066666, 0.7423376313840392, 0.7357959165488936, 0.0)

MSSISM img1, JPEG compression, quality=10 : (0.7720695503470992, 0.7956105878746803, 0.7909608832406944, 0.0)

MSSISM img1, JPEG compression, quality=20 : (0.8352323008994834, 0.8480132538608899, 0.8448243421346256, 0.0)

MSSISM img1, JPEG compression, quality=40 : (0.8757758148171648, 0.8824983314085353, 0.8803573776076528, 0.0)

4. Оценить деградацию изображения при его последовательном уменьшении и обратном увеличении, сравнить разные методы интерполяции при изменении размеров и разные коэффициенты масштабирования изображения.def resize (img, scale=0.3):  
 width = int(img.shape[1]\*scale)  
 height = int(img.shape[0]\*scale)  
 dimensions = (width, height)  
 return cv.resize(img, dimensions, interpolation=cv.INTER\_AREA)  
  
img50 = resize(resize(img1,0.5),2.0)  
img25 = resize(resize(img1,0.25),4.0)  
img01 = resize(resize(img1,0.1),10.0)  
*# cv.imshow("img01",img01)*cv.imwrite(**"D:/img01INTER\_AREA.png"**, img01)  
  
print(**"MSE img1, rescaling, quality=50% :"**,MSE(img1, img50))  
print(**"MSE img1, rescaling, quality=25% :"**,MSE(img1, img25))  
print(**"MSE img1, rescaling, quality=10% :"**,MSE(img1, img01))  
  
print(**"PSNR img1, rescaling, quality=50% :"**,PSNR(img1, img50))  
print(**"PSNR img1, rescaling, quality=25% :"**,PSNR(img1, img25))  
print(**"PSNR img1, rescaling, quality=10% :"**,PSNR(img1, img01))  
  
print(**"MSSISM img1, rescaling, quality=50% :"**,MSSISM(img1, img50))  
print(**"MSSISM img1, rescaling, quality=25% :"**,MSSISM(img1, img25))  
print(**"MSSISM img1, rescaling, quality=10% :"**,MSSISM(img1, img01))  
  
  
  
  
  
def resize (img, scale=0.3):  
 width = int(img.shape[1]\*scale)  
 height = int(img.shape[0]\*scale)  
 dimensions = (width, height)  
 return cv.resize(img, dimensions, interpolation=cv.INTER\_NEAREST)  
img50 = resize(resize(img1,0.5),2.0)  
img25 = resize(resize(img1,0.25),4.0)  
img01 = resize(resize(img1,0.1),10.0)  
cv.imwrite(**"D:/img01INTER\_NEAREST.png"**, img01)  
  
print(**"MSE img1, rescaling, quality=50% :"**,MSE(img1, img50))  
print(**"MSE img1, rescaling, quality=25% :"**,MSE(img1, img25))  
print(**"MSE img1, rescaling, quality=10% :"**,MSE(img1, img01))  
  
print(**"PSNR img1, rescaling, quality=50% :"**,PSNR(img1, img50))  
print(**"PSNR img1, rescaling, quality=25% :"**,PSNR(img1, img25))  
print(**"PSNR img1, rescaling, quality=10% :"**,PSNR(img1, img01))  
  
print(**"MSSISM img1, rescaling, quality=50% :"**,MSSISM(img1, img50))  
print(**"MSSISM img1, rescaling, quality=25% :"**,MSSISM(img1, img25))  
print(**"MSSISM img1, rescaling, quality=10% :"**,MSSISM(img1, img01))  
  
def resize (img, scale=0.3):  
 width = int(img.shape[1]\*scale)  
 height = int(img.shape[0]\*scale)  
 dimensions = (width, height)  
 return cv.resize(img, dimensions, interpolation=cv.INTER\_CUBIC)  
img50 = resize(resize(img1,0.5),2.0)  
img25 = resize(resize(img1,0.25),4.0)  
img01 = resize(resize(img1,0.1),10.0)  
cv.imwrite(**"D:/img01INTER\_CUBIC.png"**, img01)  
  
print(**"MSE img1, rescaling, quality=50% :"**,MSE(img1, img50))  
print(**"MSE img1, rescaling, quality=25% :"**,MSE(img1, img25))  
print(**"MSE img1, rescaling, quality=10% :"**,MSE(img1, img01))  
  
print(**"PSNR img1, rescaling, quality=50% :"**,PSNR(img1, img50))  
print(**"PSNR img1, rescaling, quality=25% :"**,PSNR(img1, img25))  
print(**"PSNR img1, rescaling, quality=10% :"**,PSNR(img1, img01))  
  
print(**"MSSISM img1, rescaling, quality=50% :"**,MSSISM(img1, img50))  
print(**"MSSISM img1, rescaling, quality=25% :"**,MSSISM(img1, img25))  
print(**"MSSISM img1, rescaling, quality=10% :"**,MSSISM(img1, img01))

Результат работы программы:

img01INTER\_AREA.png



MSE img1, rescaling, quality=50% : 29.206580825617284

MSE img1, rescaling, quality=25% : 49.90826485339506

MSE img1, rescaling, quality=10% : 79.49626543209877

PSNR img1, rescaling, quality=50% : 37.44017861323739

PSNR img1, rescaling, quality=25% : 34.08251305821453

PSNR img1, rescaling, quality=10% : 31.51086703468801

MSSISM img1, rescaling, quality=50% : (0.9074480913673171, 0.9075868983524076, 0.907813405276874, 0.0)

MSSISM img1, rescaling, quality=25% : (0.833117242553804, 0.8331448959838536, 0.8324507880445347, 0.0)

MSSISM img1, rescaling, quality=10% : (0.76613985117981, 0.7682423508772712, 0.7660539723741085, 0.0)

img01INTER\_NEAREST.png



MSE img1, rescaling, quality=50% : 47.88967013888889

MSE img1, rescaling, quality=25% : 84.55595196759259

MSE img1, rescaling, quality=10% : 131.14633391203705

PSNR img1, rescaling, quality=50% : 34.5876468132249

PSNR img1, rescaling, quality=25% : 30.78736050415692

PSNR img1, rescaling, quality=10% : 27.767439995367404

MSSISM img1, rescaling, quality=50% : (0.8479102984886375, 0.8483727188177963, 0.8488163370500524, 0.0)

MSSISM img1, rescaling, quality=25% : (0.7480571344876263, 0.749667414516575, 0.7491870417526947, 0.0)

MSSISM img1, rescaling, quality=10% : (0.697644879338233, 0.7029610789277213, 0.7008194496746286, 0.0)

img01INTER\_CUBIC.png



MSE img1, rescaling, quality=50% : 27.779532696759258

MSE img1, rescaling, quality=25% : 52.62715808256173

MSE img1, rescaling, quality=10% : 88.81839168595678

PSNR img1, rescaling, quality=50% : 37.96684643039542

PSNR img1, rescaling, quality=25% : 33.6892034311311

PSNR img1, rescaling, quality=10% : 30.482225209768092

MSSISM img1, rescaling, quality=50% : (0.9090518078106717, 0.909439767635439, 0.9097627605257246, 0.0)

MSSISM img1, rescaling, quality=25% : (0.8388158639068442, 0.8400647901648735, 0.8396435981604791, 0.0)

MSSISM img1, rescaling, quality=10% : (0.777373893712769, 0.780956207307267, 0.779455611888339, 0.0)

5. Оценить влияние размытия (по Гауссу, билатерального) на значение MSE, PSNR и SSIM метрик изображения. Для этого необходимо сравнить исходное изображение с размытым с различной степенью размытия.gauss = cv.GaussianBlur(img1, (3,3), cv.BORDER\_DEFAULT)  
bilateral= cv.bilateralFilter(img1, 100 ,25,300)  
  
print(**"MSE img1, GaussianBlur :"**,MSE(img1, gauss))  
print(**"MSE img1, bilateralFilter :"**,MSE(img1, bilateral))  
  
print(**"PSNR img1, GaussianBlur :"**,PSNR(img1, gauss))  
print(**"PSNR img1, bilateralFilter :"**,PSNR(img1, bilateral))  
  
print(**"MSSISM img1, GaussianBlur :"**,MSSISM(img1, gauss))  
print(**"MSSISM img1, bilateralFilter :"**,MSSISM(img1, bilateral))

Вывод программы:

MSE img1, GaussianBlur : 29.143645833333334

MSE img1, bilateralFilter : 39.20063850308642

PSNR img1, GaussianBlur : 37.488822992701586

PSNR img1, bilateralFilter : 36.91288100115427

MSSISM img1, GaussianBlur : (0.8968382917493197, 0.8977130885025412, 0.8979161777122023, 0.0)

MSSISM img1, bilateralFilter : (0.9323115904224508, 0.9406222365399748, 0.9404323116986851, 0.0)  
  
cv.waitKey(00)