《计算机视觉》（本科，2023）作业2

1. 将附带的彩色图像（I0）转为灰度图像（记为I1）。

【代码贴这里】

pic = cv2.imread("img/hw02-I0.jpg")  
row, col, channel = pic.shape  
print(row, col, channel)  
pic\_gray = numpy.zeros((row, col))  
print(pic\_gray.shape)  
for i in range(row):  
 for j in range(col):  
 pic\_gray[i][j] = 1 / channel \* numpy.sum(pic[i][j])  
cv2.namedWindow("pic\_gray", cv2.WINDOW\_NORMAL)  
cv2.imshow("pic\_gray", pic\_gray.astype("uint8"))  
cv2.waitKey()  
cv2.imwrite("img/hw02-I1.jpeg", img=pic\_gray)

【结果（灰度图像I1）贴这里】



1. 在灰度图像I1上增加不同类型（类型>=3）的噪声，分别生成噪声图像。

【代码贴这里】

def gauss(image):  
 # 给图片添加高斯噪声  
 shape = image.shape  
 mean = 0  
 sigma = 5  
 gauss = np.random.normal(mean, sigma, (shape[0], shape[1]))  
 noisy\_img = image + gauss  
 noisy\_img = np.clip(noisy\_img, a\_min=0, a\_max=255)  
 return noisy\_img  
def salt(image):  
 # 给图像添加椒盐噪声  
 shape = image.shape  
 s\_vs\_p = 0.01  
 amount = 0.01  
 num\_salt = np.ceil(amount \* image.size \* s\_vs\_p)  
 coords = [np.random.randint(0, i - 1, int(num\_salt)) for i in image.shape]  
 image[coords[0], coords[1]] = 255  
 num\_pepper = np.ceil(amount \* image.size \* (1. - s\_vs\_p))  
 coords = [np.random.randint(0, i - 1, int(num\_pepper)) for i in image.shape]  
 image[coords[0], coords[1]] = 0  
 return image  
def uniform(image):  
 # 给图像添加均匀噪声  
 shape = image.shape  
 mean, sigma = 1, 10  
 a = 2 \* mean - np.sqrt(12 \* sigma) # a = -14.64  
 b = 2 \* mean + np.sqrt(12 \* sigma) # b = 54.64  
 noise = np.random.uniform(a, b, img.shape)  
 noisy\_img = image + noise  
 noisy\_img = np.uint8(cv2.normalize(noisy\_img, None, 0, 255, cv2.NORM\_MINMAX))  
 return noisy\_img  
img = cv2.imread("./img/hw02-I1.jpeg", cv2.IMREAD\_GRAYSCALE)  
noisy\_img = uniform(img)  
# 保存图片  
cv2.namedWindow("pic", cv2.WINDOW\_NORMAL)  
cv2.imshow("pic", noisy\_img.astype("uint8"))  
cv2.waitKey()  
cv2.imwrite("./img/gray-uniform.jpg", noisy\_img)

【结果（噪声图像）贴这里】



1. 设计不同类型（类型>=3）的滤波器，对上述噪声图像分别进行去噪。对结果进行分析。

【滤波器贴这里】

def blur\_filter(image):  
 # 均值滤波  
 return cv2.blur(image, (3, 3))  
def median\_blur\_filter(image):  
 # 中值滤波  
 return cv2.medianBlur(image, 3)  
def gauus\_blur\_filter(image):  
 # 高斯滤波  
 return cv2.GaussianBlur(image, (3, 3), 1.5)

【结果（去噪后的图像）贴这里】



【分析贴在这里】

不同滤波器对于不同噪声的过滤效果有较大差异，以椒盐噪声为例，高斯滤波器对其的过滤效果较差，但中值滤波的效果较好，因为从过滤器底层原理来讲，中值过滤对于某个区域中较为突出的像素点，例如椒盐噪声添加的0或255，比较不敏感，因此效果有较为明显差异。

1. 尝试对彩色图像I0添加噪声，并设计滤波器进行去噪。对结果进行分析。

【代码贴这里】

image = cv2.imread("./img/hw02-I0.jpg")  
shape = image.shape  
s\_vs\_p = 0.01  
amount = 0.01  
num\_salt = np.ceil(amount \* image.size \* s\_vs\_p)  
coords = [np.random.randint(0, i - 1, int(num\_salt)) for i in image.shape]  
image[coords[0], coords[1], :] = [255, 255, 255]  
num\_pepper = np.ceil(amount \* image.size \* (1. - s\_vs\_p))  
coords = [np.random.randint(0, i - 1, int(num\_pepper)) for i in image.shape]  
image[coords[0], coords[1], :] = [0, 0, 0]  
cv2.imwrite("./img/rgb-salt.jpg", image)  
image = cv2.medianBlur(image, 5)  
cv2.namedWindow("pic", cv2.WINDOW\_NORMAL)  
cv2.imshow("pic", image.astype("uint8"))  
cv2.waitKey()  
cv2.imwrite("./img/rgb-filtered.jpg", image)

【结果（噪声图像+去噪结果）贴这里】



【分析贴在这里】

与之前的问题三类似，不过RGB图像的通道的噪声添加可以仅对一个通道添加，也可以对所有的添加，即添加白点，黑点，红点，绿点，蓝点，对之后的过滤效果也有影响。