

《计算机视觉》（本科，2023）作业 1

1. 将附带的彩色图像（I0）转为灰度图像（记为 I1）。

【代码贴这里】

```
import cv2
import numpy

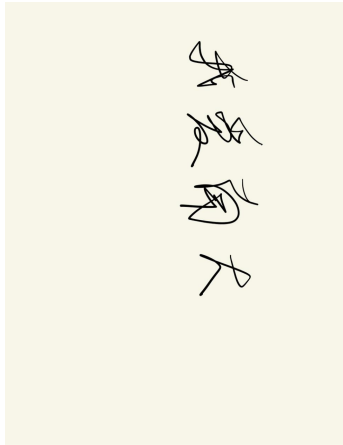
pic = cv2.imread("img/hw01-I0.jpeg")
row, col, channel = pic.shape
print(row, col, channel)
pic_gray = numpy.zeros((row, col))
print(pic_gray.shape)
for i in range(row):
    for j in range(col):
        pic_gray[i][j] = 1 / 3 * numpy.sum(pic[i][j])
print(pic_gray)
cv2.namedWindow("pic_gray", cv2.WINDOW_NORMAL)
cv2.imshow("pic_gray", pic_gray.astype("uint8"))
cv2.waitKey()
cv2.imwrite("img/hw01-I1.jpeg", img=pic_gray)
```

【结果（灰度图像 I1）贴这里】



2. 在白纸上手写“我爱南大”四个字，拍照，转为与 I1 分辨率相同的二值图像（记为 I2）。

【照片贴这里】



【代码贴这里】

```
from PIL import Image
import cv2
i2_init = Image.open("img/i2_init.png")
i1_size = Image.open("img/hw01-I1.jpeg").size
print(i2_init.size)
print(i1_size)
i2_out = i2_init.resize(i1_size, Image.ANTIALIAS)
"""
This can be
one
of :py:data:`PIL. Image.NEAREST`, :py:data:`PIL. Image.BOX`,
:py:data:`PIL. Image.BILINEAR`, :py:data:`PIL. Image.HAMMING`
,
:py:data:`PIL. Image.BICUBIC`
or :py:data:`PIL. Image.LANCZOS`.
"""
i2_binary = i2_out.convert("L").convert("1")
"""
When translating a color image to greyscale (mode "L"),
the library uses the ITU-R 601-2 luma transform::

    L = R * 299/1000 + G * 587/1000 + B * 114/1000
"""
i2_binary.show()
i2_binary.save("./hw01-I2.png")
```

【结果（二值图像 I2）贴这里】

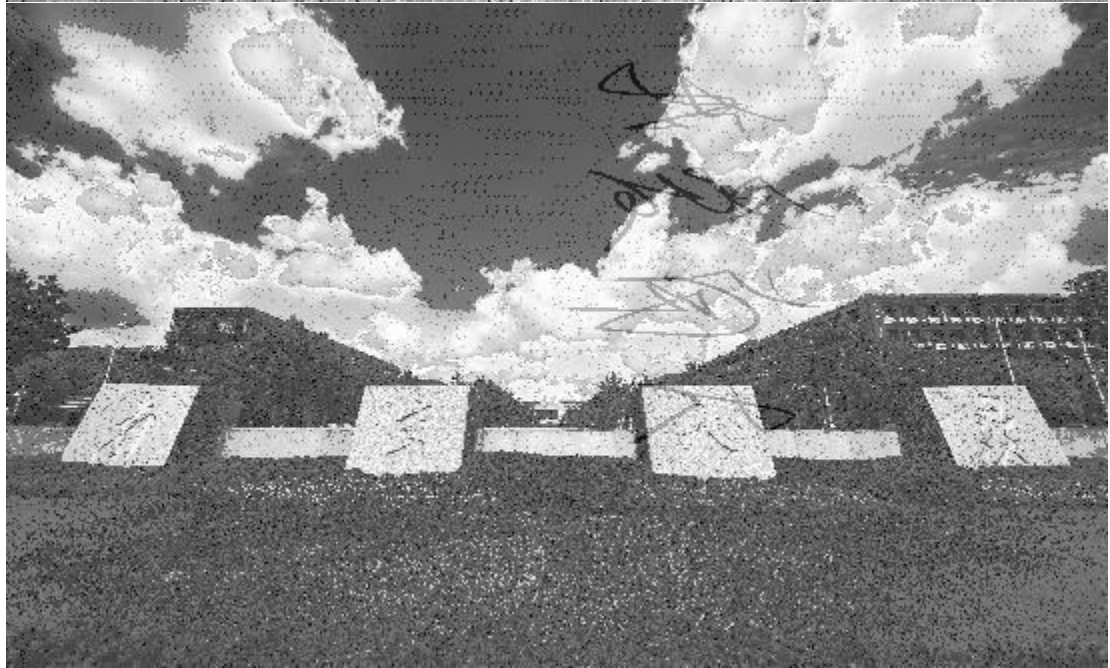


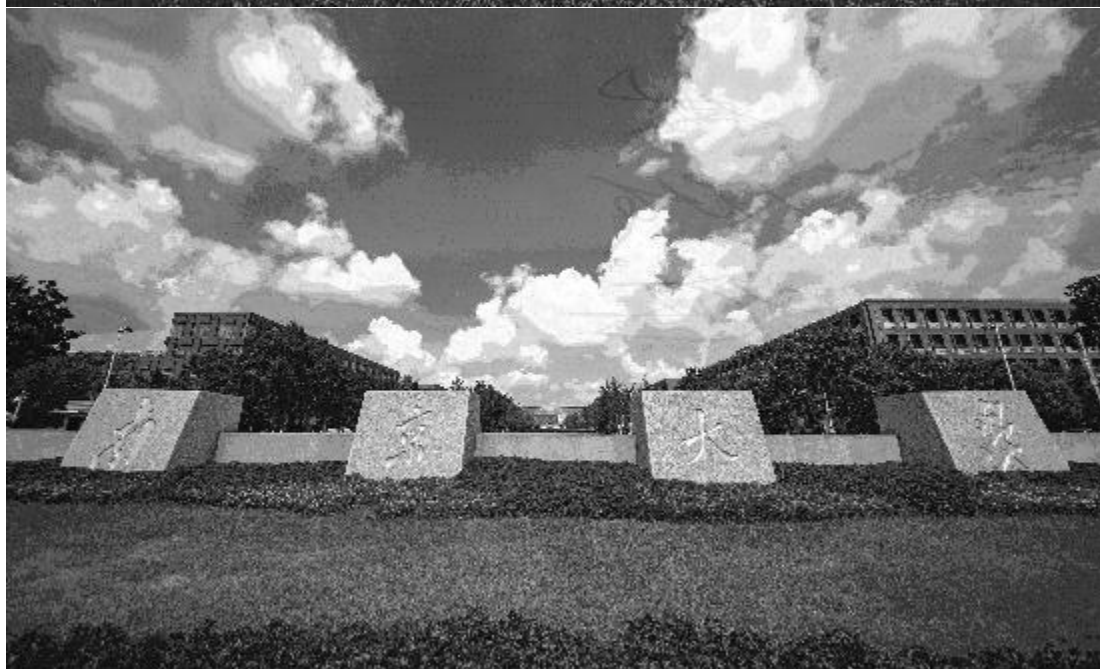
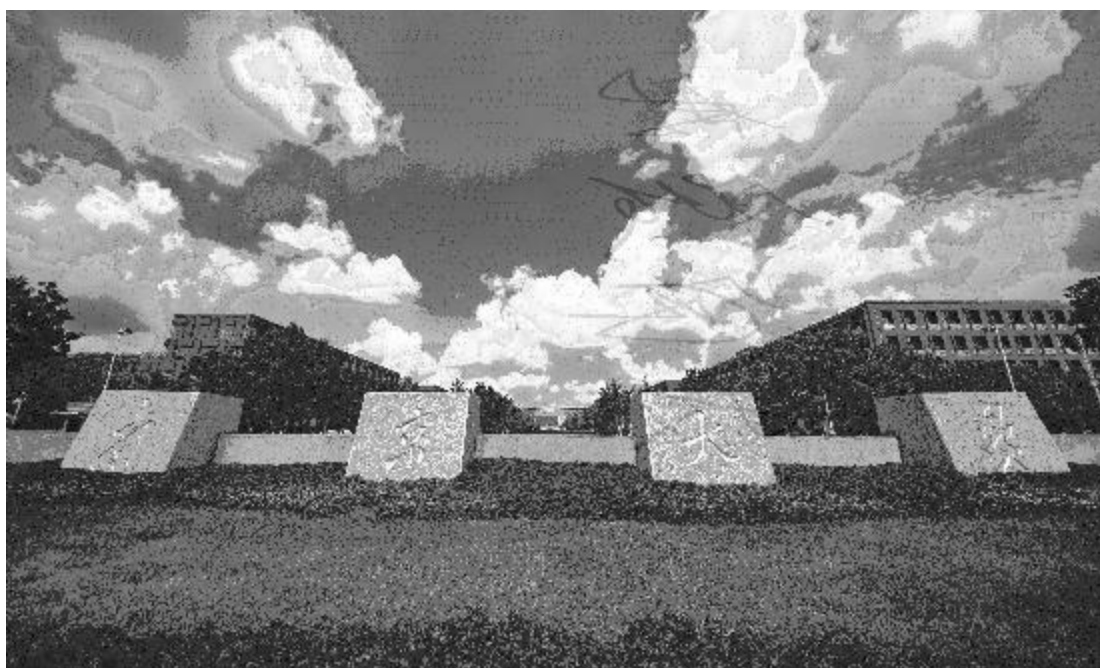
3. 灰度图像每个像素的灰度值为1个字节（8位），按照从低到高记为L1、L2、…、L8。
将I1中每个像素的L1、L2、…、L8分别用I2替换。对结果进行分析。

【代码贴这里】

```
def replace(first, second, index):  
  
    # print(bin(first))  
    # print(bin(second))  
    result = (first & ~(2 ** index)) | (second & (2 ** index))  
    # print(bin(result))  
    return result  
  
i1 = cv2.imread("img/hw01-I1.jpeg", cv2.IMREAD_GRAYSCALE)  
i2 = cv2.imread("img/hw01-I2.png", cv2.IMREAD_GRAYSCALE)  
assert i1.shape == i2.shape  
shape = i1.shape  
print(shape)  
cv2.namedWindow("pic", cv2.WINDOW_NORMAL)  
i3 = numpy.zeros(shape)  
for time in range(8):  
    for i in range(shape[0]):  
        for j in range(shape[1]):  
            i3[i][j] = replace(i1[i][j], i2[i][j], time)  
    print("Replace byte " + str(time))  
    cv2.imshow("pic", i3)  
    cv2.imwrite("./img/hw-replace-" + str(time) + ".png", i3)
```

【结果（L1、L2、…、L8 分别替换后生成的8张图像）贴这里】









【分析贴在这里】

将灰度值的字节的某一位替换为另一张图对应位置的灰度值的对应位置的值，起到了类似水印的效果，且越是替换字节高位的值，那么水印的效果就越强，若是添加判断条件，例如仅对我爱南大中的黑色像素的值进行对应位置的替换，那么就仅产生文字水印而没有整体图像的曝光效果。

4. 将附带彩色图像 I0 的 R、G、B 通道中某个或某几个通道做与问题 3 类似的处理。对结果进行分析。

【代码贴这里】

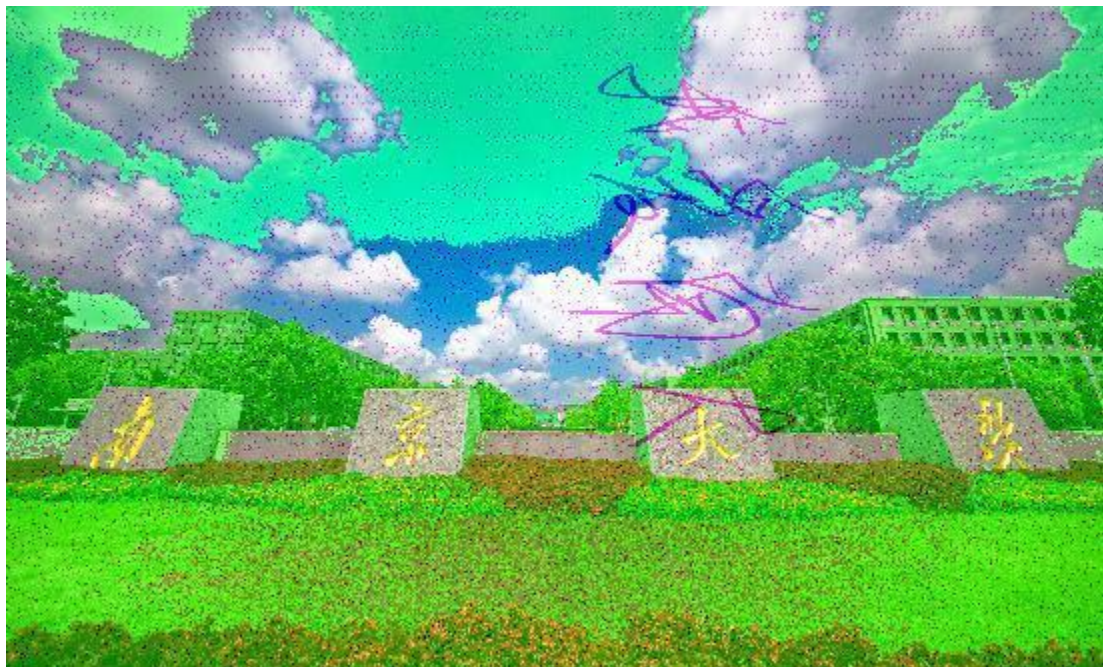
```
def replace(first, second, index):  
    return (first & ~(2 ** index)) | (second & (2 ** index))
```

```

i0 = cv2.imread("img/hw01-I0.jpeg")
i2 = cv2.imread("img/hw01-I2.png", cv2.IMREAD_GRAYSCALE)
shape = i0.shape
print(shape)
cv2.namedWindow("pic", cv2.WINDOW_NORMAL)
i3 = numpy.copy(i0)
for time in range(8):
    for i in range(shape[0]):
        for j in range(shape[1]):
            # Red channel
            i3[i][j][0] = replace(i0[i][j][0], i2[i][j], time)
            # Green channel
            i3[i][j][1] = replace(i0[i][j][1], i2[i][j], time)
            # Blue channel
            i3[i][j][2] = replace(i0[i][j][2], i2[i][j], time)
        print("Replace byte " + str(time))
    cv2.imshow("pic", i3)
    cv2.imwrite("./img/hw-all-replace-" + str(time) + ".png", i3)

```

【结果（替换后生成的若干张图像）贴这里】











【分析贴在这里】

列出了对于 RGB 三通道分别进行替换以及将整体通道全部进行替换的效果图，分别为第 8 位和第 6 位的效果图，共八张（低位替换效果不明显，未展示），起到的分别是对应色调/整体的水印效果。