

My Answer 1. 黑白图像灰度扫描

运行代码 `scanLine4e.py`, 要求环境:

```
import numpy as np
```

```
import cv2
```

```
import matplotlib.pyplot as plt
```

```
import os
```

```
import math
```

输出结果如图 1 所示:

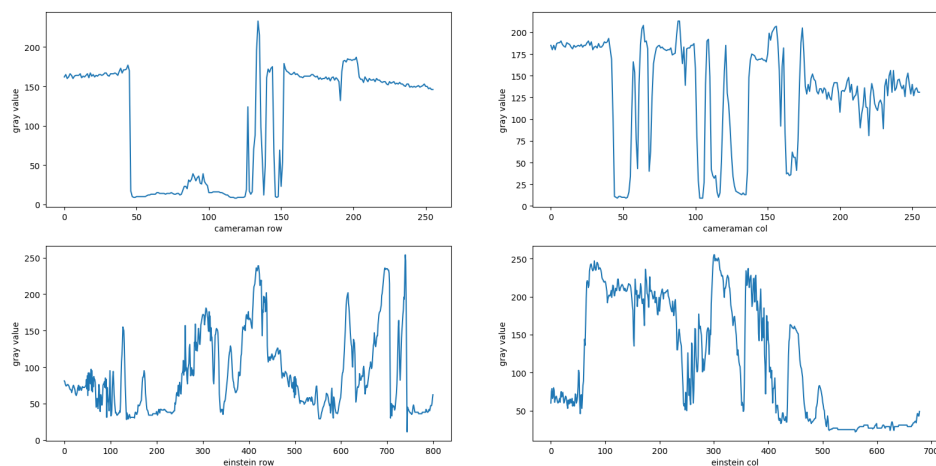


图 1: cameraman.tif 和 einstein.tif 的中心行和中心列的像素灰度矢量

My Answer 2. 彩色图像转换为黑白图像, 运行程序 `convolution.py` 要求环境和上面一致

输出结果如图 2 所示:

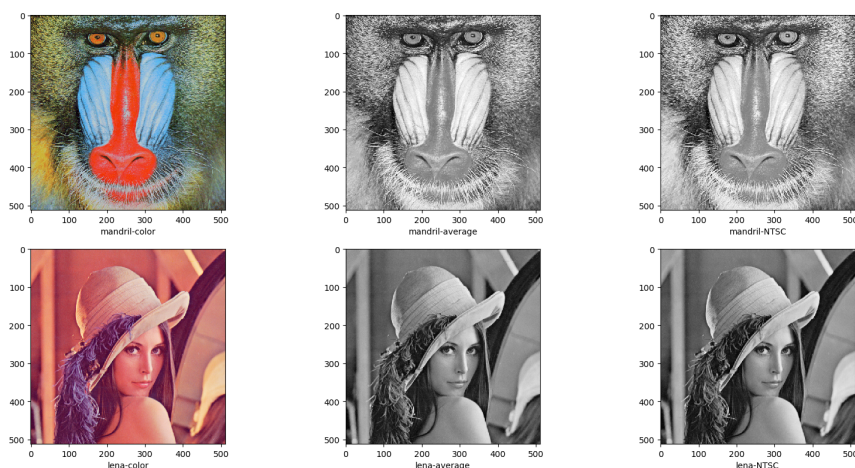


图 2: 将 `mandril_color.tif` 和 `lena_color.tif` 用均值和 NTSC 两种方法转换成单色灰度图像

My Answer 3. 灰度图像的高斯滤波，要求环境和上面一致

四张图片 (两张 NTSC 转换过) 在 $\sigma = 1, 2, 3, 5$ 下的高斯滤波输出结果如图 3 所示:



图 3: 四张图片 (两张 NTSC 转换过) 在 $\sigma = 1, 2, 3, 5$ 下的高斯滤波

在 $\sigma = 1$ 下的结果, 与直接调用 `CV2.GaussianBlur` 的情况下比较, 其中高斯核的设置和 `gaussKernel` 函数中计算的结果一样, 按照 *PPT* 中的进行计算: $m = \text{roundup}(3 \times \sigma) \times 2 + 1$, 对比输出包括差值如图 4 所示:

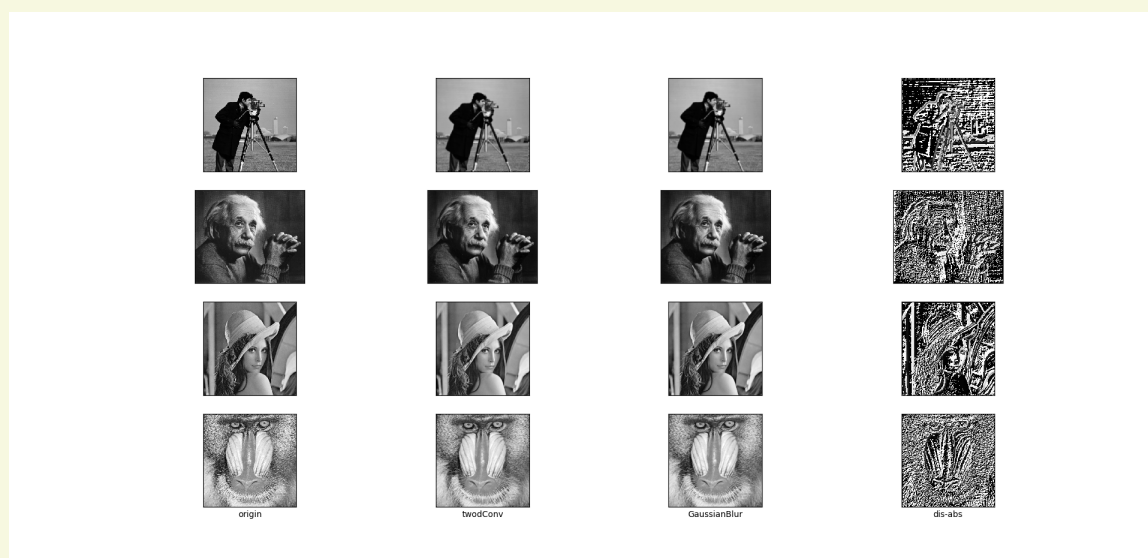


图 4: 四张图片 (两张 NTSC 转换过) 在 $\sigma = 1$ 下的高斯滤波, 和 `cv2` 的高斯滤波图像做差值对比

My Answer 4. (接上面) 在 $\sigma = 1$ 的情况下, 任选两幅图像, 比较其他参数条件不变的情况下像素复制和补零下滤波结果在边界上的差别, 输出图像如图 5 所示:

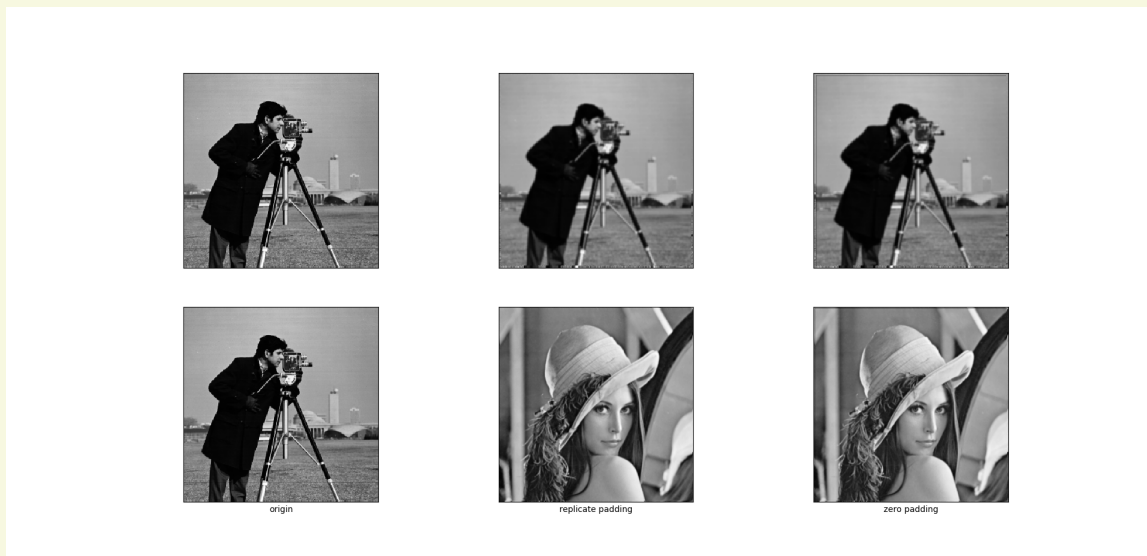


图 5: camerman 和 lena 图像在 $\sigma = 1$ 下的高斯滤波, 对比两边的边框

从图中可以看出, 使用 *zero* 的方式 *padding* 之后, 会出现比较窄的黑边, 这是因为补充的是 0, 导致在最后加权平均的时候, 会使图像变暗一点。在其卷积核更大一些之后, 即 $\sigma = 2, 3, 5 \dots$ 之后, 黑边会相对而言大一点, 或者更加明显一点。具体如图 6 所示:



图 6: camerman 和 lena 图像在 $\sigma = 1, 2, 3, 5$ 情况下的高斯滤波, 对比两边的边框