

图像处理与分析课后作业一

李一帆

October 17, 2020

1 问题 1

输入为三个参数，f: 图像路径，I: 提取图像的第 I 行或第 I 列，loc: ‘row’ 或 ‘col’，指定提取行或列。

执行 Q1.py 文件后会产生图像第 I 行以及第 I 列像素灰度的折线图，并将结果保存到 figures 目录所在的文件夹下。

另外，如果图像路径不正确会产生提示信息 “Image path is invalid, please input again!” 如果图像索引超出范围会提示信息 “The pixel vector index is invalid, please input again!” 如果 loc 参数不正确，会提示信息 “The loc parameter is invalid, please input again!”

最终的结果如 Figure 1 所示。

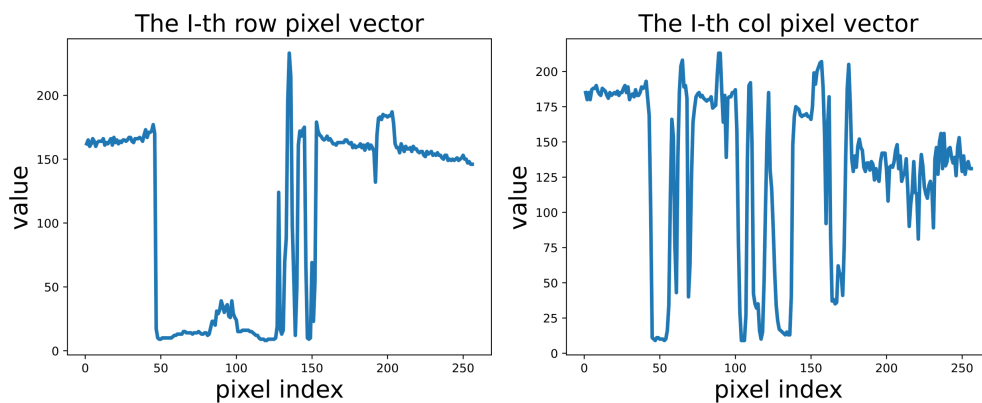


Figure 1: 第 i 行以及第 i 列的灰度折线图

2 问题 2

输入为三个参数 f, method (NTSC 为缺省) 以及 showflag (默认为 True)。其中 f 表示输入图像的路径，method 为 average 时为计算三个通道均值模式；

method 为 NTFC 时，为采用不同加权系数加权后的灰度图；showflag 为是否显示原图和生成的灰度图。

输出为生成的灰度图。执行 Q2.py 文件后会产生 mandril 和 lena 的两种形式的灰度图, 并将结果保存到 figures 文件夹下。

另外，如果路径不正确会产生提示信息，参数不正确也会产生提示信息。

最终的结果如 Figure 2 所示。

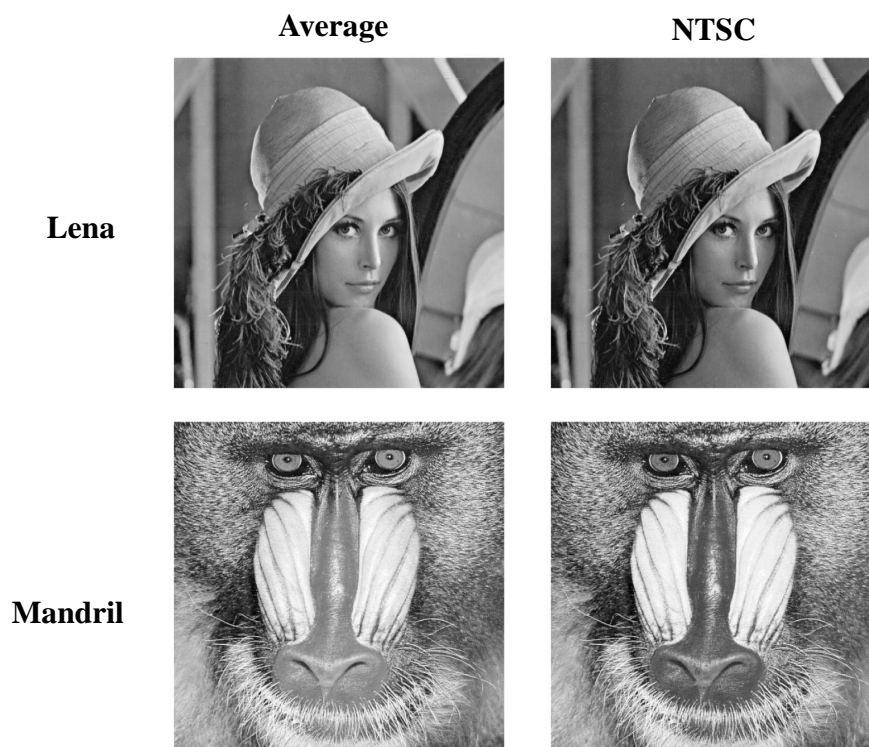


Figure 2: Average 与 NTSC 方法生成的灰度图

由 Figure 2 可以看出，由 Average 生成的灰度图的效果相对于 NTSC 来说偏淡一些，层次感相对来说没有那么强。而由 NTSC 生成的灰度图颜色偏深，对比度更加明显，更有利于辨认。

3 问题 3

由两个函数构成, 其中 twodConv 可以产生两种形式的 padding 并调用 conv2d 函数进行卷积运算, conv2d 为 2 维卷积函数。twodConv 函数的输入为 2 维图像矩阵 f, kernel 以及 padding 方式（默认为 zero 形式），输出为二维卷积后的图像。

直接运行 Q3.py 文件后，会输出卷积后的结果。

另外，如果输入的 padding 形式不正确，会提示重新输入。

整理后的结果如图 Figure 3 所示。

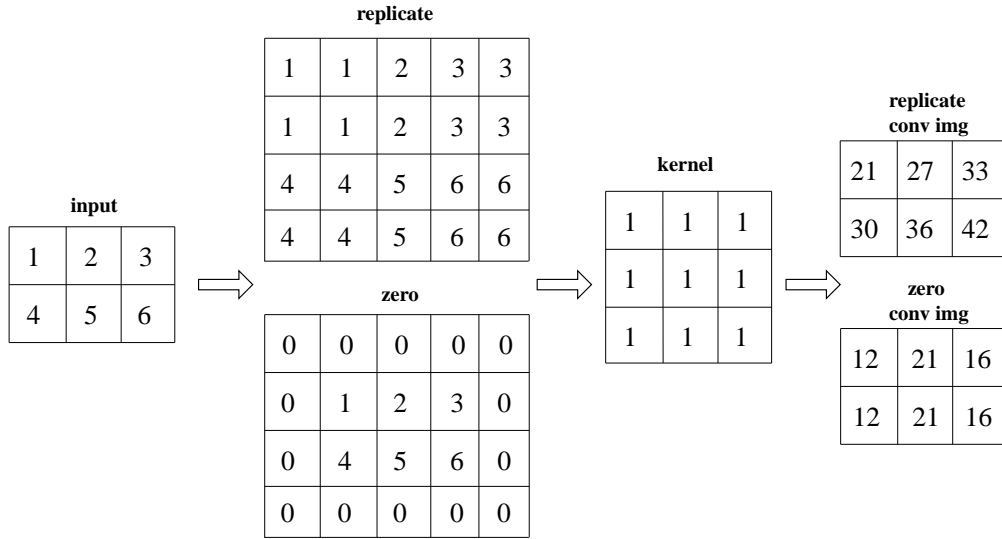


Figure 3: 不同 padding 下的卷积图

4 问题 4

主要由一个产生高斯核的函数 `gaussKernel` 构成，输入为两个参数 `sig` 以及 `m`（默认为 0）。输出为高斯核。其中 `sig` 为高斯核函数的标准差，`m` 为高斯核的大小。

直接运行 `Q4.py` 文件后，会输出在缺省 `kernel` 大小后标准差为 1 的高斯核。

另外，如果 `kernel` 大小小于 3 会有提示信息。

最终结果如 Figure 4 所示。

```
[[1.96519161e-05 2.39409349e-04 1.07295826e-03 1.76900911e-03 1.07295826e-03 2.39409349e-04 1.96519161e-05]
[2.39409349e-04 2.91660295e-03 1.30713076e-02 2.15509428e-02 1.30713076e-02 2.91660295e-03 2.39409349e-04]
[1.07295826e-03 1.30713076e-02 5.85815363e-02 9.65846250e-02 5.85815363e-02 1.30713076e-02 1.07295826e-03]
[1.76900911e-03 2.15509428e-02 9.65846250e-02 1.59241126e-01 9.65846250e-02 2.15509428e-02 1.76900911e-03]
[1.07295826e-03 1.30713076e-02 5.85815363e-02 9.65846250e-02 5.85815363e-02 1.30713076e-02 1.07295826e-03]
[2.39409349e-04 2.91660295e-03 1.30713076e-02 2.15509428e-02 1.30713076e-02 2.91660295e-03 2.39409349e-04]
[1.96519161e-05 2.39409349e-04 1.07295826e-03 1.76900911e-03 1.07295826e-03 2.39409349e-04 1.96519161e-05]]
```

Figure 4: sigma=1 时的高斯 kernel

5 问题 5

直接运行文件，会产生四幅图像在高斯核参数 `sig` 分别为 1, 2, 3, 5 下的卷积图像。最后会产生本程序编写的高斯核和 `opencv` 产生的高斯核的对比的结果，输出两者的差值。

最终结果如 Figure 5 和 Figure 6 所示。

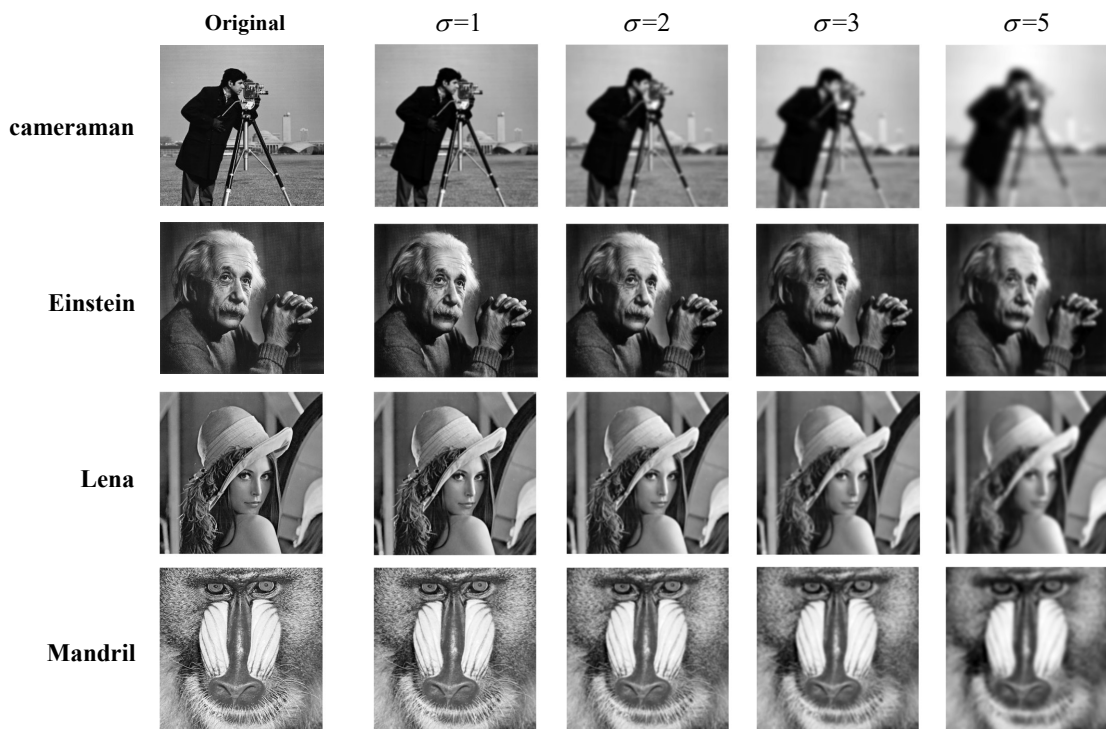


Figure 5: 不同 sigma 下高斯 kernel 卷积后的对比图

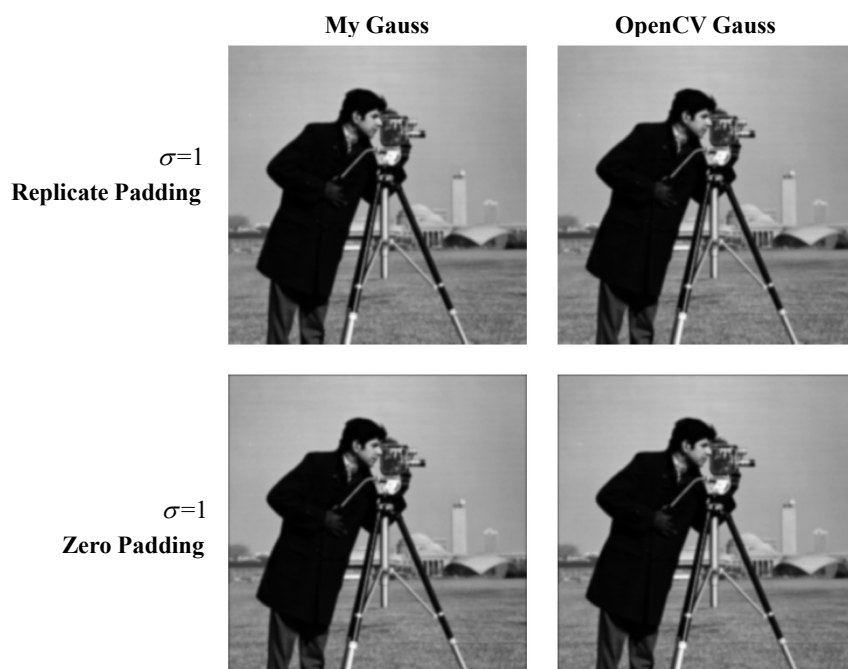


Figure 6: 不同 padding 方式 (sigma=1) 与 opencv 的对比图

在 padding=replicate 情况下 (diff1) 和 padding=zero 情况下 (diff2), 本文的高斯卷积出来的图像和 OpenCV 卷积出来图像差值 (加绝对值) 的图像如 Figure 7 所示。由 Figure 6 可以看出, 在 padding 形式为 replicate 情况下, 图像边缘比较自然, 而在 padding 为 zero 情况下, 边缘偏黑一些。

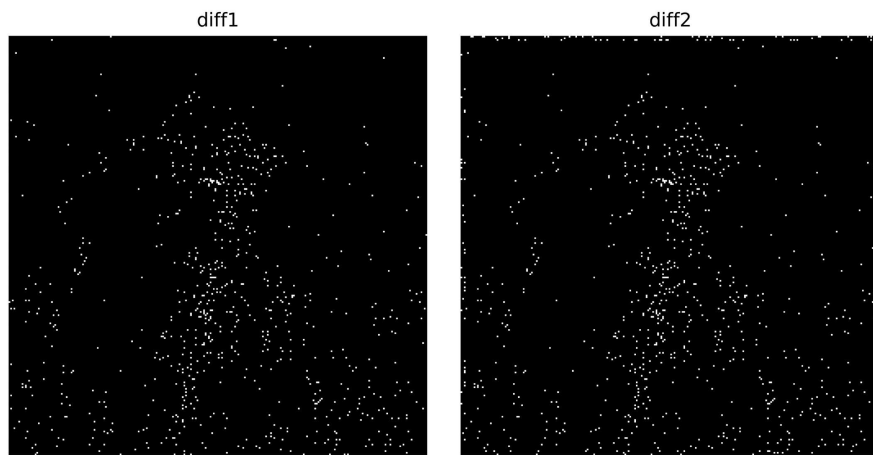


Figure 7: opencv 卷积与本文卷积图像差值