

2023 年秋季学期 图像处理 编程作业 01

布置时间：2023 年 9 月 19 日

提交截止时间：2023 年 10 月 8 日晚 10 点(教育云课程网站)

问题 1 图像处理软件包比较 (20 分)

如果选择采用 MATLAB 完成编程作业, 请同学们熟悉其图像处理工具软件包。如果选择采用 Python, 请同学们熟悉 Pillow (<https://pillow.readthedocs.io/en/stable/>), Scikit-image (<https://scikit-image.org/>) or OpenCV (<https://opencv.org/>) 这三个软件包之一。

从上述四个软件包中选择至少三个进行比较。针对每个软件包, 请回答以下问题:

- 该软件包支撑什么图像类型和格式?
- 使用该软件包如何读取, 显示和写一个图像?
- 该软件包提供哪些类型的图像处理功能?

问题 2 黑白图像灰度扫描 (20 分)

实现一个函数 `s = scanLine4e(f, I, loc)`, 其中 `f` 是一个灰度图像, `I` 是一个整数, `loc` 是一个字符串。当 `loc` 为 'row' 时, `I` 代表行数。当 `loc` 为 'column' 时, `I` 代表列数。输出 `s` 是对应的相关行或者列的像素灰度值矢量。

调用该函数, 提取 `cameraman.tif` 和 `einstein.tif` 的中心行和中心列的像素灰度矢量并将扫描得到的灰度序列绘制成图。

问题 3 彩色图像转换为黑白图像(20 分)

图像处理中的一个常见问题是将彩色 RGB 图像转换成单色灰度图像, 第一种常用的方法是取三个元素 R, G, B 的均值。第二种常用的方式, 又称为 NTSC 标准, 考虑了人类的彩色感知体验, 对于 R,G,B 三通道分别采用了不同的加权系数, 分别是 R 通道 0.2989, G 通道 0.5870, B 通道 0.1140. 实现一个函数 `g = rgb2gray(f, method)`. 函数功能是将一幅 24 位的 RGB 图像 `f` 转换成灰度图像 `g`. 参数 `method` 是一个字符串, 当其值为 'average' 时, 采用第一种转换方法, 当其值为 'NTSC' 时, 采用第二种转换方法。将 'NTSC' 做为缺省方式。

调用该函数, 将提供的图像 `mandril_color.tif` 和 `lena512color.tiff` 用上述两种方法转换成单色灰度图像, 对于两种方法的结果进行简短比较和讨论。

问题 4 图像二维卷积函数 (20 分)

实现一个函数 `g = twodConv(f, w)`, 其中 `f` 是一个灰度源图像, `w` 是一个矩形卷积核。要求输出图像 `g` 与源图像 `f` 大小 (也就是像素的行数和列数) 一致。请注意, 为满足这一要求, 对于源图像 `f` 需要进行边界像素填补(padding)。这里请实现两种方案。第一种方案是像素复制, 对应的选项定义为 'replicate', 填补的像素拷贝与其最近的图像边界像素灰度。第二种方案是补零, 对应的选项定义为 'zero', 填补的像素灰度为 0. 将第二种方案设置为缺省选择。

问题 5 归一化二维高斯滤波核函数 (20 分)

实现一个高斯滤波核函数 $w = \text{gaussKernel}(\text{sig}, m)$ ，其中 sig 对应于高斯函数定义中的 σ ， w 的大小为 $m \times m$ 。请注意，这里如果 m 没有提供，需要进行计算确定。如果 m 已提供但过小，应给出警告信息提示。 w 要求归一化，即全部元素加起来和为 1。

问题 6 灰度图像的高斯滤波 (20 分)

调用上面实现的函数，对于问题 1 和 2 中的灰度图像 (cameraman, einstein, 以及 lena512color 和 mandril_color 对应的 NTSC 转换后的灰度图像) 进行高斯滤波，采用 $\sigma=1, 2, 3, 5$ 。任选一种像素填补方案。

对于 $\sigma=1$ 下的结果，与直接调用 MATLAB 或者 Python 软件包相关函数的结果进行比较 (可以简单计算差值图像)。然后，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。

作业报告撰写和提交要求:

- 请在报告中就每个问题对应的程序用简短文字描述如何运行，并展示主要代表性结果。
- 编写的代码应该有必要注释。
- 报告文件请存储为 PDF 格式以方便批改。
- 请将全部文件打包成一个压缩文件(zip 或者 rar 格式)提交。文件命名为“作业 01+学生姓名”