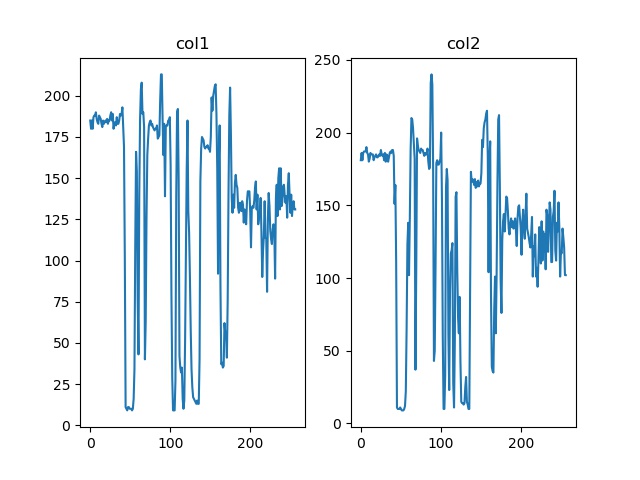
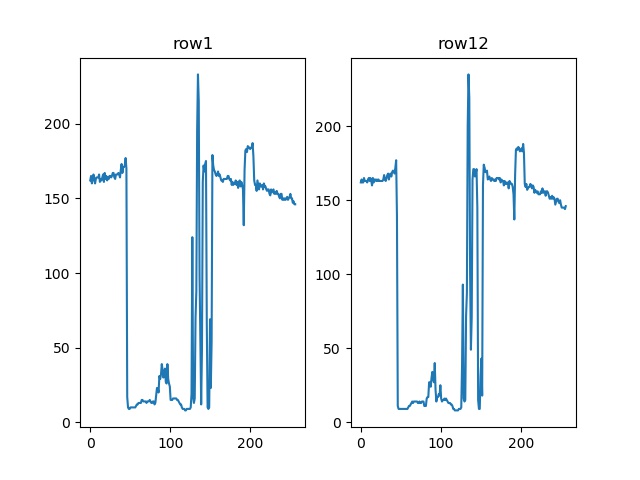
图像处理作业

题目一、

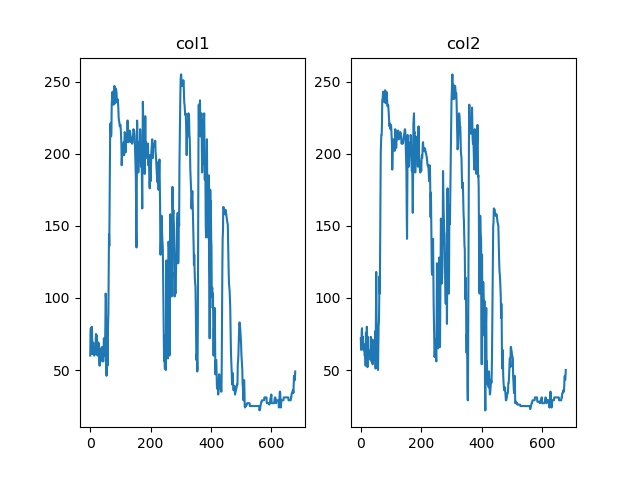
代码运行：运行session1.py文件中的代码，程序自动显示结果，同时将结果图保存在./session1/文件夹内。

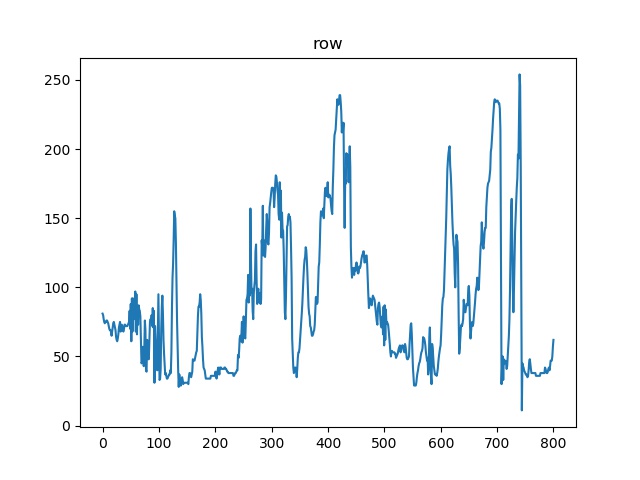
效果图：cameraman.tif





Einstein.tif：

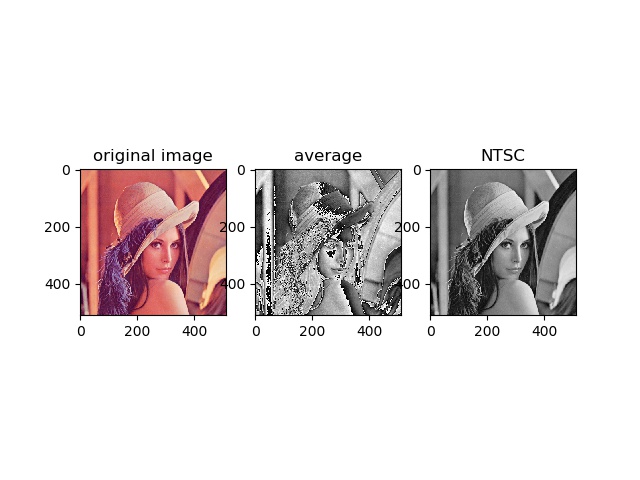


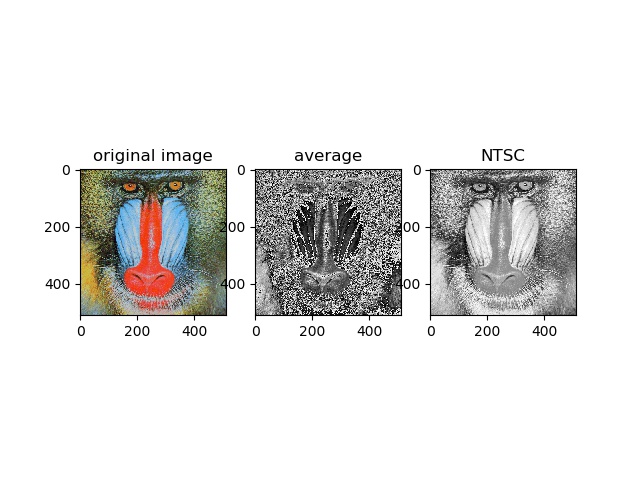


题目二、

代码运行：运行session.py文件代码，程序自动显示结果并将其保存至session2文件夹

效果：从效果上看，我们可以观察到average方法会有失真情况，而NTSC方法失真更小，效果更好，能够较好地保存图像信息。





题目三、四、五

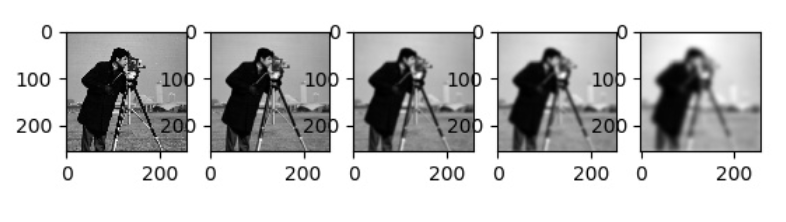
代码：全部代码在session345.py文件中，运行test\_5\_1()函数，显示题目（**调用上面实现的函数，对于问题 1 和 2 中的灰度图像**（**cameraman, einstein, 以及 lena512color 和 mandril\_color 对应的 NTSC 转换后的灰度图像）进行高斯滤波，采用σ=1，2，3，5。任 选一种像素填补方案。**）结果并保存在session345文件夹中。

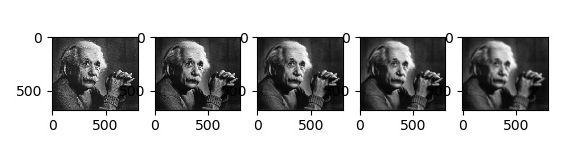
运行test5\_2()函数，显示题目(**对于σ=1 下的结果，与直接调用相关函数的结果进行比较（可以简单计算差值图像）**）结果并保存在session345文件夹中。

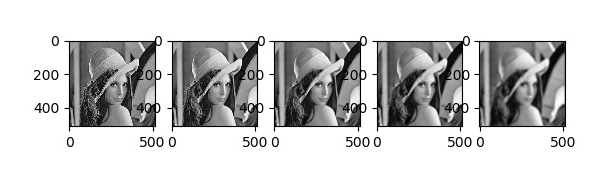
运行test5\_3()函数，显示题目（**任选两幅图像，比较其他参数条件不变的情况下像素复制和补零下滤波结果在边界上的差 别**）结果并保存在session345文件夹中。

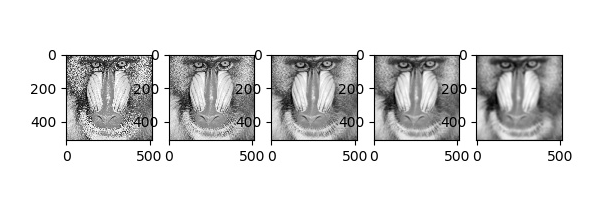
结果：

问题：**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**



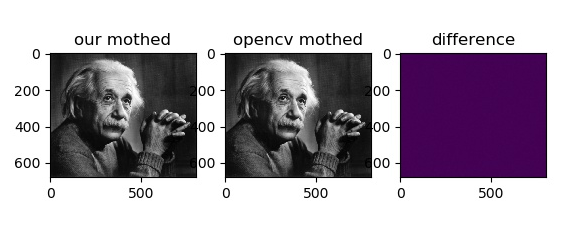


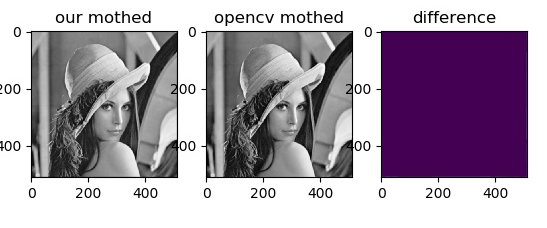


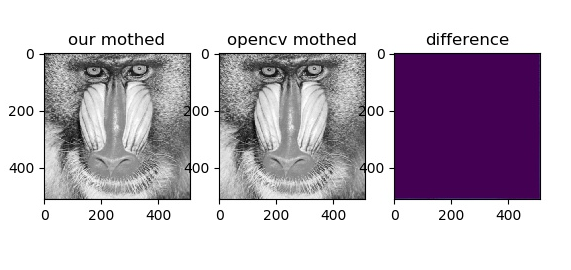


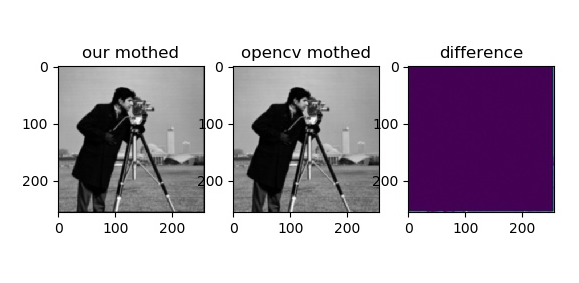
从结果上看，高斯滤波低通滤波的特点很明显，而写随着高斯滤波核方差越大，滤波效果越明显。

问题：**对于σ=1 下的结果，与直接调用相关函数的结果进行比较（可以简单计算差值图像）**



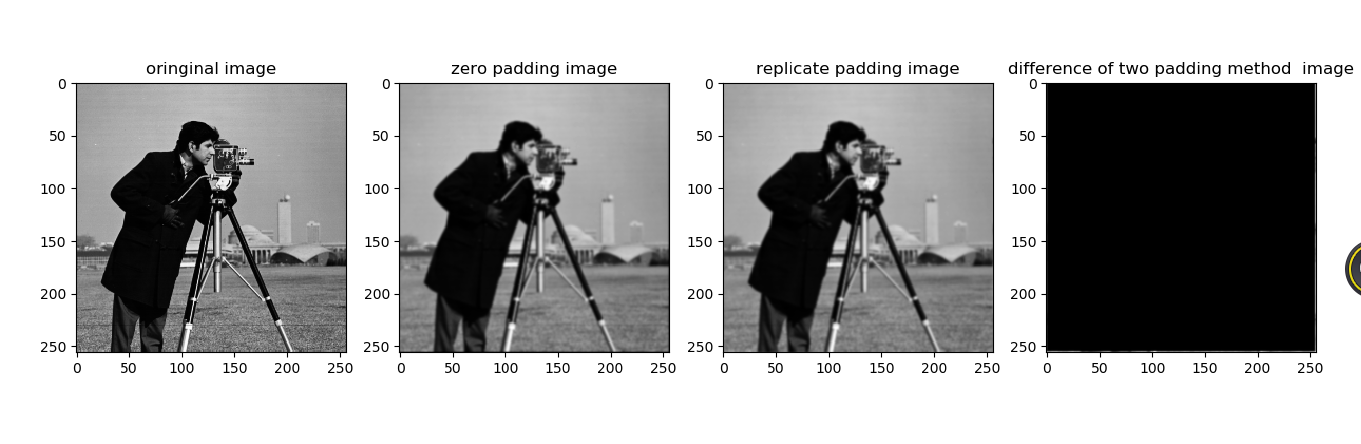


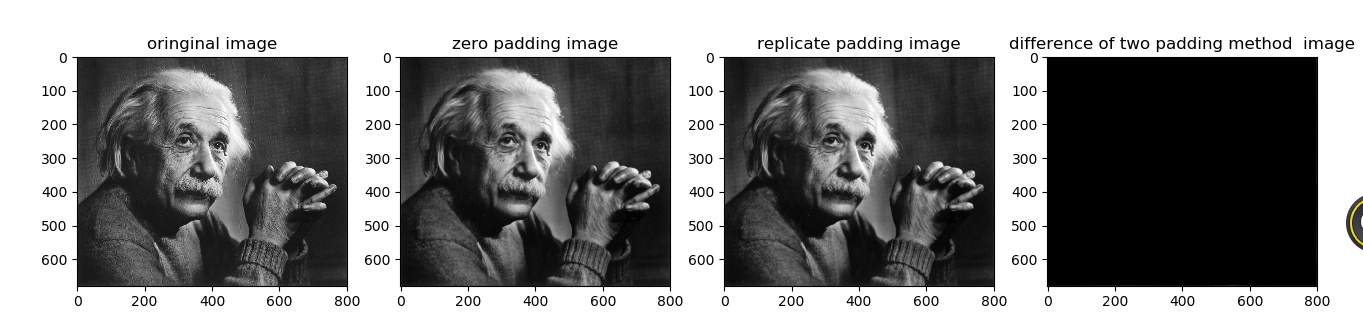




从效果上看，我们设计的高斯滤波核和卷积函数与opencv所用基本上没有区别，差分图中间像素大豆为0，同时只有边沿区域有差值，知识由于padding方式不同导致，说明我们所编写的卷积函数和高斯滤波核函数基本正确。

问题:**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**





Padding方式不同，对于滤波效果而言，在视觉上并没有多大不同，但是在边沿区域，不同padding方式在数值上还有有差异。