

实验 2-A. 练习空间域图像处理函数的使用

1. 打开图像 Lenna.jpg，分别使用 OpenCV 的平滑滤波函数：`blur()`，`GaussianBlur()`和 `medianBlur()`对打开的图像进行平滑处理，调整各个函数的参数，通过 `imshow()`函数观察参数对结果的影响（注：请阅读 PPT 或自行百度上述函数的用法）。

2. 打开图像 Lenna.jpg，使用 OpenCV 的 Sobel 算子分别计算 X 和 Y 方向的梯度，使用 `imshow()`显示 X 和 Y 两个方向的梯度图像（注：可参阅已上传至钉钉群的 `blur_sharp.ipynb`），通过调整参数，观察参数对结果的影响；

3. 打开图像 Lenna.jpg，使用 OpenCV 的 Laplacian 算子对图像进行边缘计算，通过调整参数，观察参数对结果的影响（注：可参阅已上传至钉钉群的 `blur_sharp.ipynb`）。

4. 形如：`ker_emb = np.array([-4, -2, 0], [-1, 1, 1], [0, 2, 4]), dtype="float32")`定义一个自定义的滤波核，使用 OpenCV 的 `filter2D()`对 Lenna 图像的红色分量图像进行滤波处理，观察结果图像。

实验 2-B. 图像直方图均衡化处理

1. 打开 Lenna.jpg，取 Blue 分量图像，使用 `equalizeHist()`对图像作直方图均衡化处理；

2. 分别以窗体的形式显示均衡化处理之前和之后的直方图（显示请参考以下代码）

```
18
19 '''
20 matplotlib 的方法: plt.hist(x, bins, range, density, ...)
21 x -- 数组, bins -- bin 的数目, range -- bin 的下限和上限,...
22 im.ravel()-- 多维数组转一维数组
23 '''
24 plt.hist(im[:, :, 0].ravel(), 256, [0, 256])
25 plt.show()
```

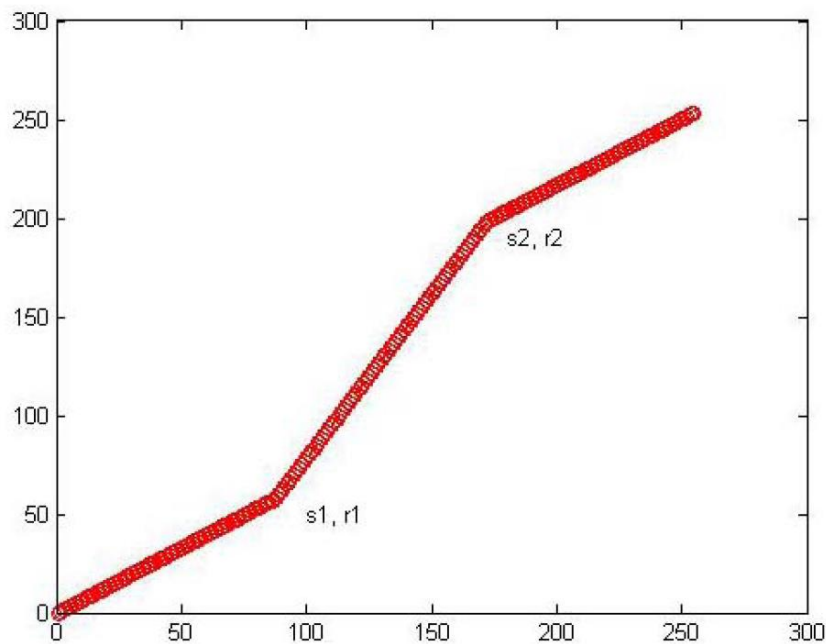
实验 2-C. 图像直方图规定化

1. 打开 Lenna.jpg，取绿色分量图像，使用 `equalizeHist()`对绿色分量图像作直方图均衡化处理；

2. 以均衡化处理后的绿色分量图像的直方图为目标直方图，对 **Lenna** 的蓝色分量图像作直方图规定化，分别显示原始和规定化处理后的蓝色分量图像，及各自的直方图（可参阅已上传至钉钉群的 **hist_match.ipynb**）。

实验 2-D. 图像直方图均衡化算法

编制程序实现如下所示的分段线性变换，横坐标 **r** 代表输入灰度级，纵坐标代表映射后的灰度级，其中 **r1=88**，**s1=60**（即输入灰度 **88** 经映射后为 **60**）；**r2=162**，**s2=200**。



实验 2-E. 图像直方图均衡化算法

如下为一个 **100x100**，灰度级为 **0--15** 的灰度图像的直方图，请编写程序实现直方图均衡化算法，且以画表方式给出均衡化处理前后的灰度映射关系（注二行表格，第一行为输入灰度级，第二行为均衡化以后对应的灰度级）。

灰度	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
N	2	57	704	1979	2452	1501	1128	1196	505	316	105	40	12	2	1	0

注：可将上述直方图定义成一个 **16x1** 的 **numpy** 数组，按 **PPT** 的算法步骤实现均衡化处理（即：归一化、求积聚直方图、乘最大灰度形成映射）。