数字图像第一次作业

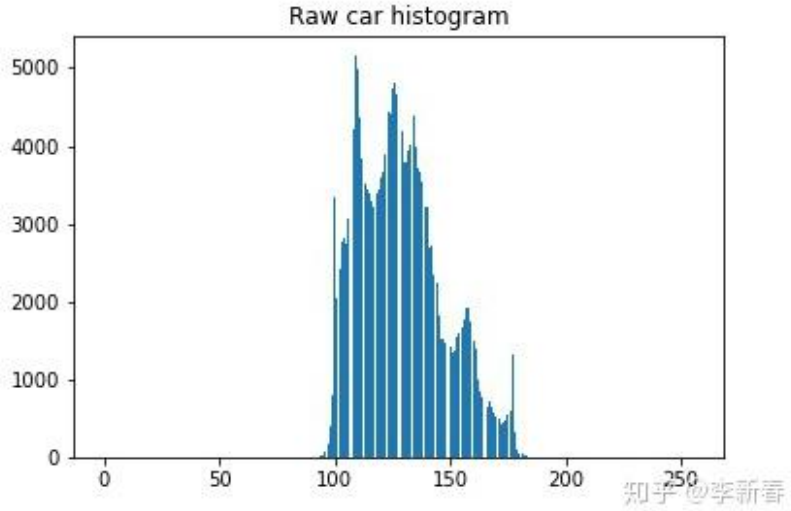
一、原理说明

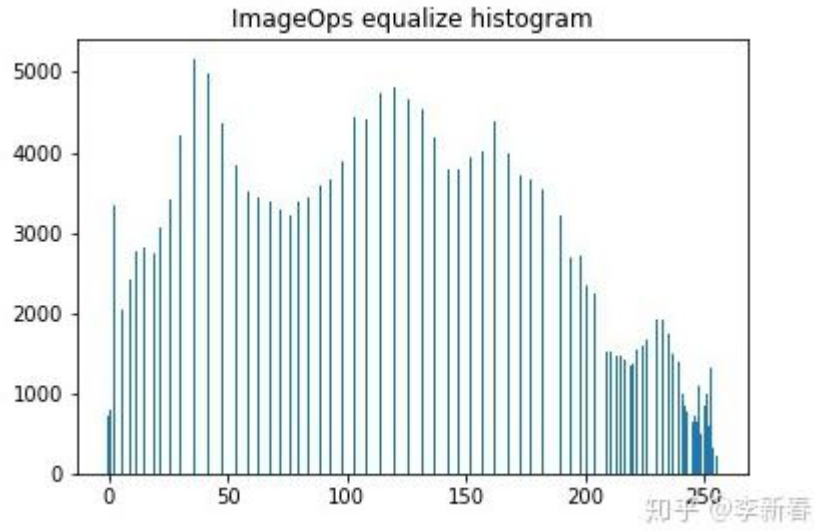
1.灰度直方图均衡原理说明：

灰度直方图均衡个人认为是图像增强的非常常见的方法，它进行图像增强的原理是将灰度直方图的灰度值分布变得更加均衡。具体来讲，灰度直方图是图像的统计特征，将每个灰度值有多少个进行直方图展示，优点是能看到整个图像的灰度分布特征。对于一些模糊图像或者过亮过暗图像来说，它们的灰度直方图分布会比较集中，如集中在灰度值较低的区域或者较高的区域。而这种灰度值集中体现在图像上就是像素点灰度值差异不够，造成图像模糊或质量不高。对灰度直方图进行均衡，可以让灰度值分布更加

均匀，使得像素点之间的灰度值差异变大，即让图像变得更清晰，轮廓更明晰。

(1) 图像说明 1：原图直方图和均衡后的直方图





2.彩色直方图均衡原理说明

由于直方图均衡这个操作一般只针对灰度值。如果要对彩色图做直方图均衡，需要更进一步的思考。有两种想法，一种是根据 RGB 的三个通道，

直接强制将图像分为 3 个通道（每个通道都是灰度图），对每个通道分别做直方图均衡。再组合起来。第二种想法是从网上查询资料得知，将

图像转换到如 YCbCr 颜色空间，再分离通道，分别做直方图均衡，再合并。

对于这两种做法，根据查询到的资料所知，第一种直接将 RGB 图像分离做直方图均衡再合并的做法会破坏图像内部关联。效果没有先转换

到其他颜色空间再做直方图均衡来得好。

*但根据我执行代码，对两个均衡的结果做分析，觉得第一种似

乎效果更好。原因还在研究，可能是 RGB 转换到 HSI 的函数写法需要修正。

二、程序说明：（代码中已经有了一些注释，这里只做简单说明）

程序文件夹中有 input 和 output、work101.py、work102.py、work103.py 这些内容。input 是选择的待操作图片，output 是均衡后的输出图片。几个 python 文件在下面具体说明。

1.work101.py:

是对灰度图做直方图均衡的代码，直接运行代码就能完成。运行结果会显示一个原图和转换后的图像的对比。并且会将转换后的图像保存在 output 文件夹。

2.work102.py:

是对彩色图做直方图均衡的代码，是直接将 RGB 图分成三个通道，分别做直方图均衡，再合并。直接运行代码就能完成。运行结果会显示一个原图和转换后的图像的对比。

并且会将转换后的图像保存在 output 文件夹。

3.work103.py:

是对彩色图做直方图均衡的代码，是将 RGB 图像先转换到 HSI 颜色空间，分别做直方图均衡，再合并。是对灰度图做直方图均衡的代码，直接运行代码就能完成。运行结果会显示一个原图和

转换后的图像的对比。并且会将转换后的图像保存在 output 文件夹。

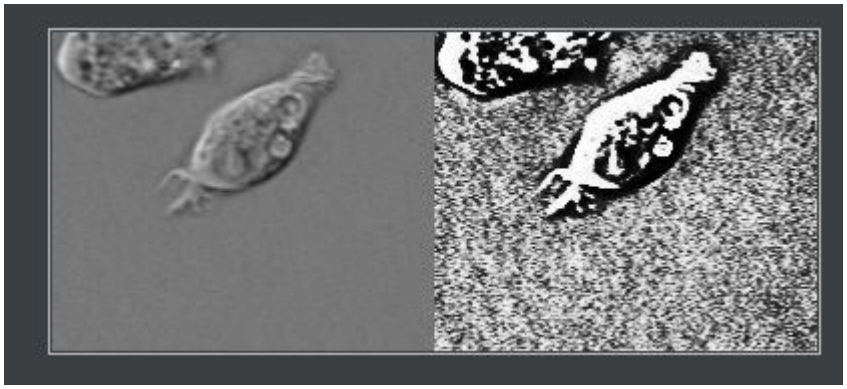
三、实验结果分析

1.灰度图直方图均衡

output 文件夹的 1.png 到 5.png 都是灰度图直方图均衡的结果。从结果来看，对比度比原图大了很多，看起来也亮了一些。这也和我在原理中说的一致，直方图均衡

使得灰度对比度加大，图像更加锐利、对比度增加。

(1) 图像说明：灰度图均衡前后



2.彩色图直方图均衡（直接 RGB 空间的版本）

output 文件夹的 6.png 到 10.png 都是灰度图直方图均衡的结果。从结果来看，对比度大了很多，颜色没有被非常大地被影响，但 8.png 有一些颜色变动

显得不太真实。

(1) 图像说明：彩色图直接 RGB 均衡前后

3.彩色图直方图均衡（先转换到 HSI 空间再处理的版本）

output 文件夹的 6_hsi.png 到 10_hsi.png 都是灰度图直方图均衡的结果。从结果来看，(根据分析，可能是由于 RGB 转 HSI 的代码不准确，导致均衡的效果不好)，效果不是太好，原因可能是代码实现的

原因，正在积极寻找解决方法。

(1) 图像结果