

BIT 图像视频处理五个小作业

环境

Python 版本: python 3.6.9

Opencv 版本: python-opencv 3.4.2

系统: windows 10

所有代码都放在 github:

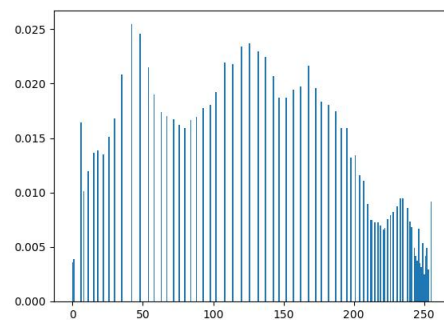
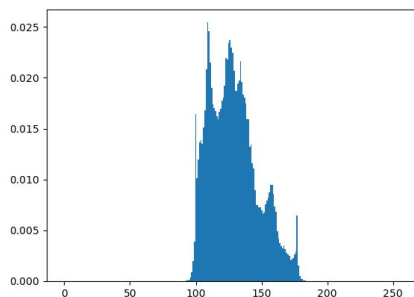
实验说明及效果

实验一：直方图均衡化

定义：如果一幅图像的灰度直方图几乎覆盖了整个灰度的取值范围，并且除了个别灰度值的个数较为突出，整个灰度值分布近似于均匀分布，那么这幅图像就具有较大的灰度动态范围和较高的对比度，同时图像的细节更为丰富。已经证明，仅仅依靠输入图像的直方图信息，就可以得到一个变换函数，利用该变换函数可以将输入图像达到上述效果，该过程就是直方图均衡化。除了使用 matplotlib 包中 `plt.imshow()` 进行数据展示外，处理后的图像保存在相对路径 `result\His\` 目录下。

运行命令： `python histogramEqualization.py --img img\car.jpg --Level 256`

实验效果：

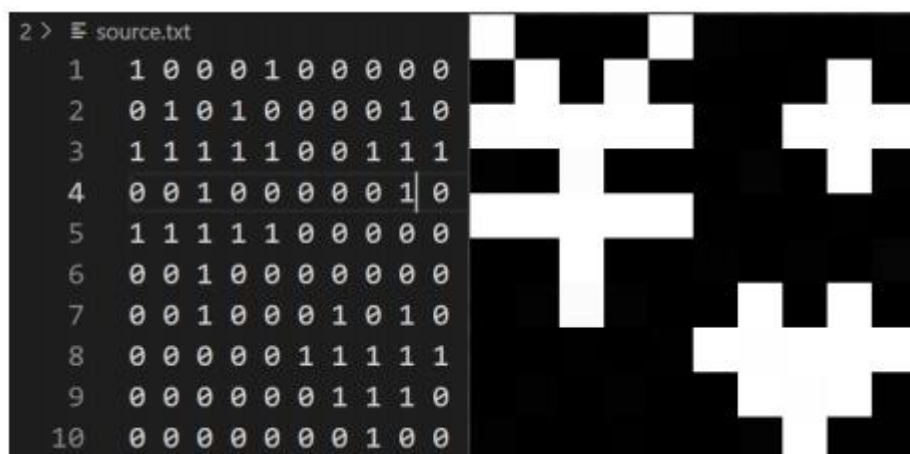


实验二：区域编号和区域面积计算

定义：二值图像分析最重要的方法就是连通区域标记，它是所有二值图像分析的基础，它通过对二值图像中白色像素（目标）的标记，让每个单独的连通区域形成一个被标识的块，进一步的我们就可以获取这些块的轮廓、外接矩形、质心、不变矩等几何参数。这里因为是实现一个区域的算法，为了简化并没有使用外部的图片，采用的是自己定义的一个矩阵，只需要运行代码即可。结果保存在路径 `result` 中

运行命令：`python main.py`

实验效果：



```
[[1. 0. 0. 0. 1. 0. 0. 0. 0. 0.]
 [0. 1. 0. 1. 0. 0. 0. 0. 2. 0.]
 [1. 1. 1. 1. 1. 0. 0. 2. 2. 2.]
 [0. 0. 1. 0. 0. 0. 0. 0. 2. 0.]
 [1. 1. 1. 1. 1. 0. 0. 0. 0. 0.]
 [0. 0. 1. 0. 0. 0. 0. 0. 0. 0.]
 [0. 0. 1. 0. 0. 0. 3. 0. 3. 0.]
 [0. 0. 0. 0. 0. 3. 3. 3. 3. 3.]
 [0. 0. 0. 0. 0. 0. 3. 3. 3. 0.]
 [0. 0. 0. 0. 0. 0. 0. 3. 0. 0.]]
```

总像素：100

区域编号 1.0 像素面积 17 像素

区域编号 0.0 像素面积 67 像素

区域编号 2.0 像素面积 5 像素

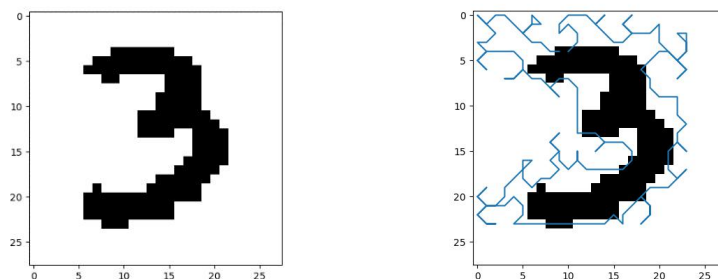
区域编号 3.0 像素面积 11 像素

实验三：链码

定义：链码（又称为 freeman 码）是用曲线起始点的坐标和边界点方向代码来描述 曲线或边界的方法，常被用来在图像处理、计算机图形学、模式识别等领域中表示曲线和区域边界。该代码主要使用 `plt.imshow()` 和 `plt.plot()` 展示链码划分的边缘，没有保存最后的结果图片，仅显示。注意：本次图片使用了较为简单的手写数字识别图片，在 `img\train.csv` 中每行对应一张图片的 784 维像素值和标签，但是将这些像素点使用 `opencv` 保存图片再读取后会出现一些像素点的值变化，因此会导致结果不是很好，如果想展示较好结果，可以修改代码直接从 `img\train.csv` 中读取一行进行展示

运行命令： `python chainCode.py --img img\three.jpg --threshold 70`

运行结果：



实验四：频域滤波

定义：傅里叶变换是一种函数在空间域和频率域的变换，从空间域到频率域的变换是傅里叶变换，而从频率域到空间域是傅里叶的反变换。本实验为实现图像的傅里叶变换以及生成经过频域锐化，高斯高通滤波器和频域平滑，高斯低通滤波器的图片。除了使用 `matplotlib` 包中 `plt.imshow()` 进行数据展示外，高通处理后的图像保存在相对路径 `result\Frequency\high\` 目录下，低通处理后的图像保存在相对路径 `result\Frequency\low\` 目录下。

运行命令：

高斯高通：

`python frequencyDomainFiltering.py --img img\car.jpg --d 10 --mode high``

高斯低通

`python frequencyDomainFiltering.py --img img\car.jpg --d 10 --mode low`

运行结果：



高斯高通：



高斯低通：



实验五：拉普拉斯图像增强 s

定义：利用拉普拉斯算子进行图像增强本质是利用图像的二次微分对图像进行锐化，在图像领域中微分是锐化，积分是模糊，利用二次微分对图像进行锐化即利用邻域像素提高对比度。本次实验中采用的卷积核有两种：

$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$ 和 $\begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ 。除了使用 matplotlib 包中 `plt.imshow()` 进行数据展示外，拉普拉斯图像增强 sharp 模式处理后的图像保存在相对路径 `result\Lap\sharp\` 目录下，拉普拉斯图像增强 edge 模式处理后的图像保存在相对路径 `result\Lap\edge\` 目录下。

运行命令：

```
python laplacian.py --img img\car.jpg --mode sharp (sharp 是  $\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$  的核)  
python laplacian.py --img img\car.jpg --mode edge (edge 是  $\begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$  核)
```

运行结果：



sharp:



edge:

