



北京航空航天大学
BEIHANG UNIVERSITY

图像阈值化分割 与边缘检测

实验报告

院（系）名称	自动化科学与电气工程学院
专业名称	模式识别与智能系统
学生姓名	张家奇
学号	15031204
任课老师	郑红

2018 年 04 月 24 日



1. 实验目的

- (1) 学习基本的图像分割算法、原理；以及常用的一阶、二阶边缘检测算子和边缘检测原理。
- (2) 编程实现大津阈值分割的实现程序，并与 MATLAB 自带的库函数的效果进行比较。
- (3) 自行编写 mask 与图像的卷积操作程序，并用不同的边缘检测算子形成的模板，对图像进行边缘检测，与自带的库函数进行比较。
- (4) 比较不同边缘检测算子的效果之间的差异。

2. 实验内容及算法流程

- (1) 利用大津阈值进行图像分割

a) 读取图片，构建图像的直方图和累计直方图。

b) 设图像的灰度值为 $1 \sim m$ 级，灰度值 i 的像素数为 n_i ，则像素总数为 $N = \sum_i n_i$ ，

各值的概率 $p_i = n_i / N$ ，用 T 将其分成两组 $C_0 = \{1 \sim T\}$ 和 $C_1 = \{T+1 \sim m\}$ ，各组产生的概率如下：

$$C_0 \text{ 产生的概率: } w_0 = \sum_{i=1}^T p_i = w(T), \quad C_0 \text{ 的均值: } \mu_0 = \frac{\sum_{i=1}^T i p_i}{w_0} = \frac{\mu(T)}{w(T)}$$

$$C_1 \text{ 产生的概率: } w_1 = \sum_{i=T+1}^m p_i = 1 - w_0, \quad C_1 \text{ 的均值: } \mu_1 = \frac{\sum_{i=T+1}^m i p_i}{w_1} = \frac{\mu - \mu(T)}{1 - w(T)}$$

其中， $\mu = \sum_{i=1}^m i p_i = w_0 \mu_0 + w_1 \mu_1$ 是整体图像的灰度平均值。

c) 计算两组之间的方差：

$$\delta^2(T) = w_0(\mu_0 - \mu)^2 + w_1(\mu_1 - \mu)^2 = w_0 w_1 (\mu_1 - \mu_0)^2$$

d) 从 $1 \sim m$ 之间改变 T ，求上式为最大值时的 T ，既是最大方差阈值。

e) 利用所求出的 T 值，遍历图像将小于 T 的像素值设为 1，大于 T 的像素值设为 0

- (2) 边缘检测实验

a) 选用不同的滤波器对图像进行平滑，抑制噪声。

b) 选用不同的边缘检测算子计算图像的梯度图。

c) 选取阈值进行边缘检测。

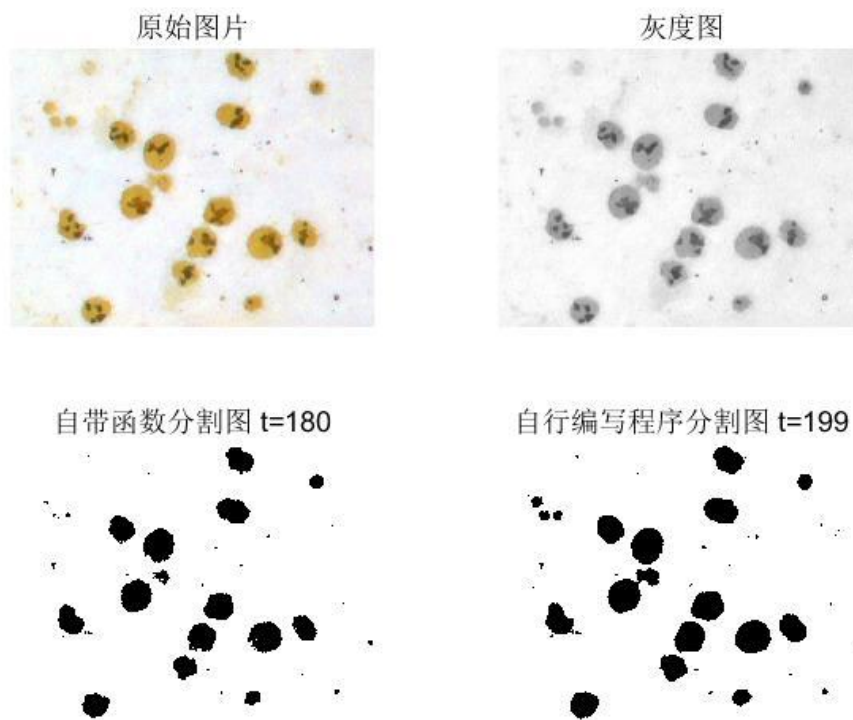
d) 对不同滤波器，不同阈值，不同算子的效果进行比较。



3. 实验过程及结果分析

(1) 大津阈值图像分割

使用大津阈值对图像进行二值化分割处理，得到了以下的实验结果：



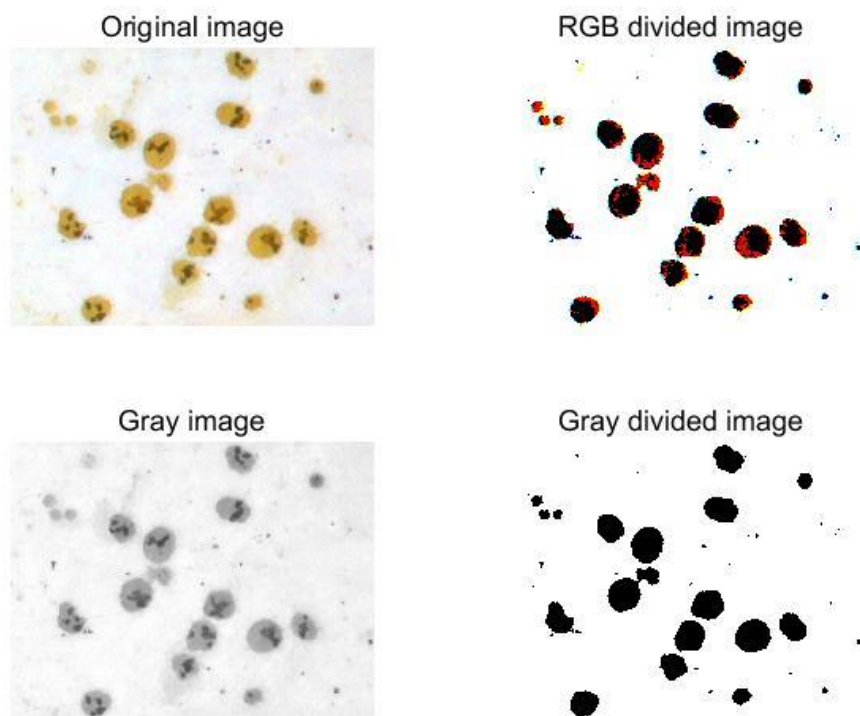
大津阈值实验结果

图中左上角为原始图像，左下角为 MATLAB 内置函数进行大津阈值的计算和分割得到的结果，右下角为自己编写的程序进行大津阈值的计算和图像分割的结果。

通过比较可以看出，分割效果较好，且对噪声有一定的抑制作用，但无法完全抑制噪声。

使用 Matlab 自带函数和自己编写的程序二者分割效果整体一致，但自带函数对噪声的抑制能力相对较强，而自编的大津阈值处理实现程序在目标边缘处的分割效果较好。

这样的图像分割只对灰度图进行了操作，如果对 RGB 图中不同层的图像进行大津阈值的计算，并根据大津阈值进行图像分割会得到怎样的结果呢？经过实验，效果如下：

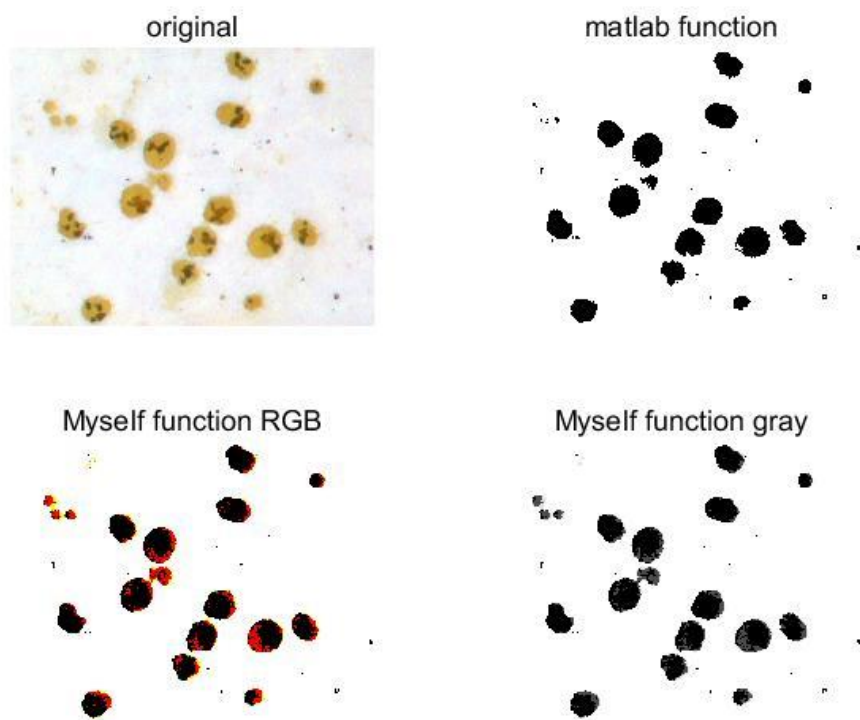


对 RGB 图像的不同层分别进行大津阈值计算及分割

可以看出，图像出现了失真，这是因为对不同层次通道而言，大津阈值的计算结果是不同的，以该实验为例，对 R，G，B 三个通道分别进行大津阈值的计算结果为：211，198，162。这表明在不同通道中，物体分割的感兴趣区域是不同的。这也就导致了对于有些像素而言，在 R 通道内被选为目标物体，而在其它通道中没有被选为目标趋于，导致颜色发生失真。

观察图像可以发现，希望被分出来的物体像素在 RGB 通道上的分布和噪声在 RGB 通道上的分布是不同的。以上图为例，被分割的细胞显然在 R 通道内分量较高，而噪声在 B 通道内噪声较高。因此我们可以根据这一特点，通过对 RGB 图像的不同通道进行阈值分割，然后将只在噪声通道（B）中被分出来的像素点进行剔除，便可以极大的抑制噪声。

编程实现之后，效果如下：



自行编写噪音抑制函数效果对比

通过观察可以发现，这种噪声抑制方法效果比 MATLAB 自带的函数还要好。这种方法不仅对噪声的抑制效果更好（图像中的噪点均被极大地减弱），更重要的是，它还保留了相当一部分的有用物体，没有将其误作为噪声而剔除（左上角部分）。

（2） 图像边缘检测

a. 首先对图像进行滤波处理，抑制噪声。本实验分别对图像进行了均值、中值滤波，比较后发现，使用均值滤波效果要普遍优于中值滤波，对图像的噪声抑制效果更好。虽然使用均值滤波也会导致图像缺失一些细微的弱边缘特征，但影响并不大。下图显示了不同阈值、不同滤波方式下，对图像进行 sobel 算子边缘提取效果的对比。

因此本实验中均采用均值滤波对图像进行处理。

Threshold=10



Threshold=20



Threshold=30



Threshold=40



Threshold=50



Threshold=60



Threshold=70



Threshold=80



Threshold=90



均值滤波

Threshold=10



Threshold=20



Threshold=30



Threshold=40



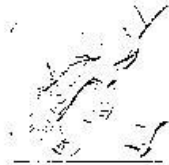
Threshold=50



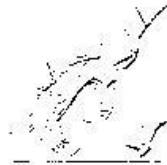
Threshold=60



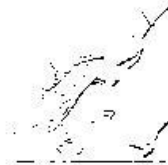
Threshold=70



Threshold=80

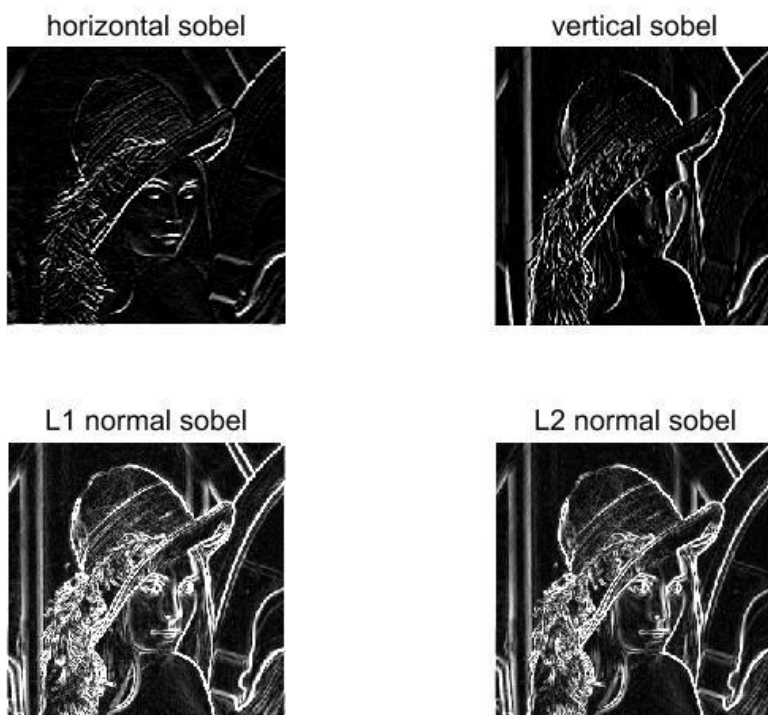


Threshold=90



中值滤波

b. 计算得到图像的梯度图 (sobel 算子)

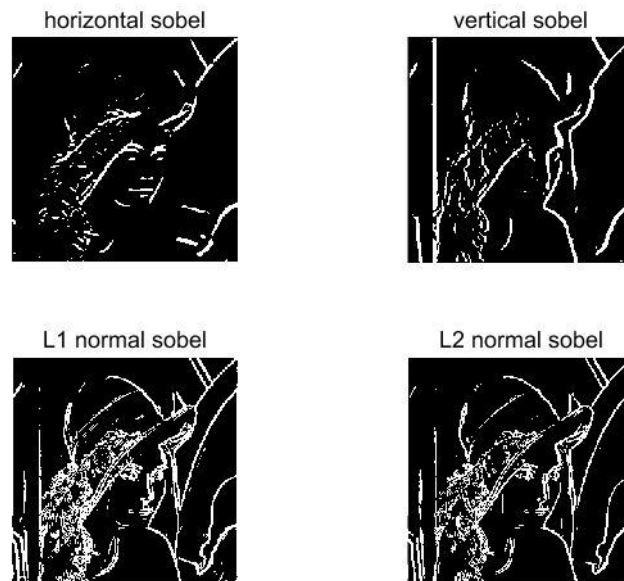
**Sobel 算子梯度图**

上图分别是采用纵向 sobel 算子、横向 sobel 算子、对两种 sobel 算子进行 L1 范数整合、L2 范数整合之后计算得到的图像梯度图。

可以看出，经过 L2 范数整合后的 sobel 算子对边缘的响应效果最好，能够充分提取图像中各个方向的纹理信息。

c. 选取合适的阈值对图像进行边缘提取。

效果如下所示：

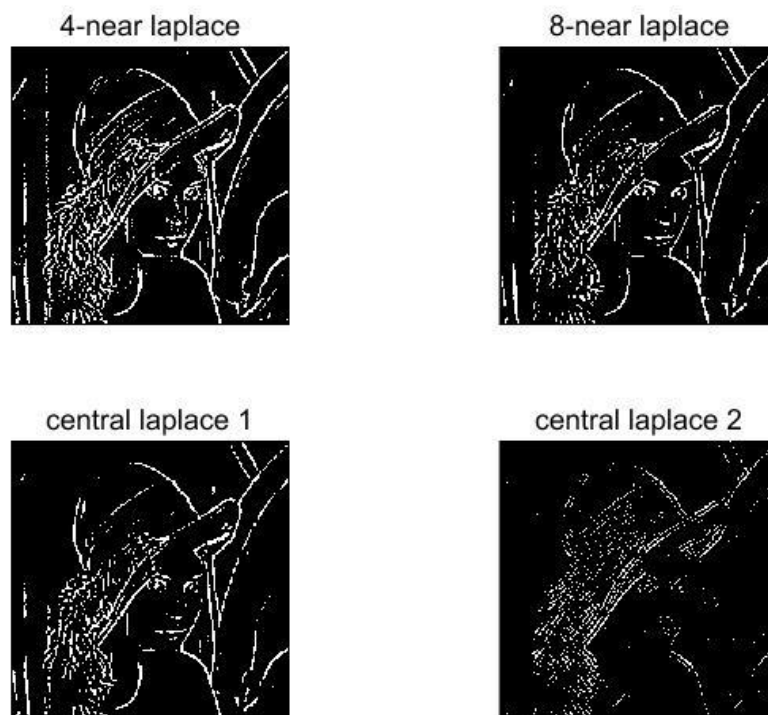


不同 Sobel 算子边缘提取效果

d. 选用不同边缘检测算子对图像进行边缘提取。

(1) Laplace 算子

选用不同的 Laplace 算子进行边缘提取，效果如图所示：



不同 Laplace 算子对图像边缘提取的效果对比

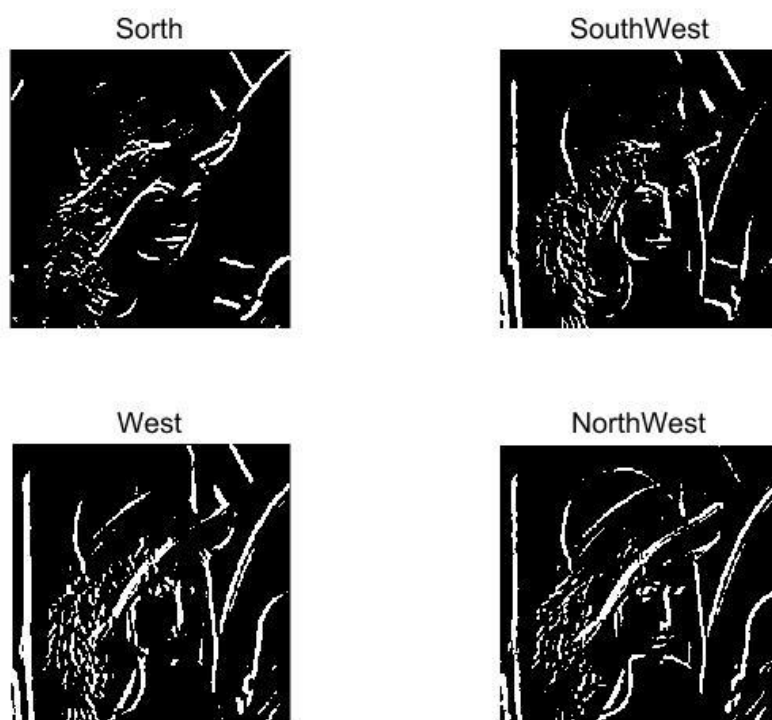
从图中可以看出，在对不同算子分别进行阈值选择，分别选择效果最好的阈值之后可以看出，4-领域 Laplace 算子和 8-领域 Laplace 算子提取的边缘特征更为丰富，但同时也引入了跟多噪声；另一方面它们对强边缘和弱边缘的提取效果都不错。与之相对的是，中心 Laplace 算子对复杂纹理、边缘的提取效果很好（如图片中的头发），对简单边缘（如直线）的响应很微弱，甚至没有。另一方面，Laplace 对噪声有很强的抑制作用，但同时也导致对强边缘的提取效果非常好，但对弱边缘提取效果并不好。原因有二：

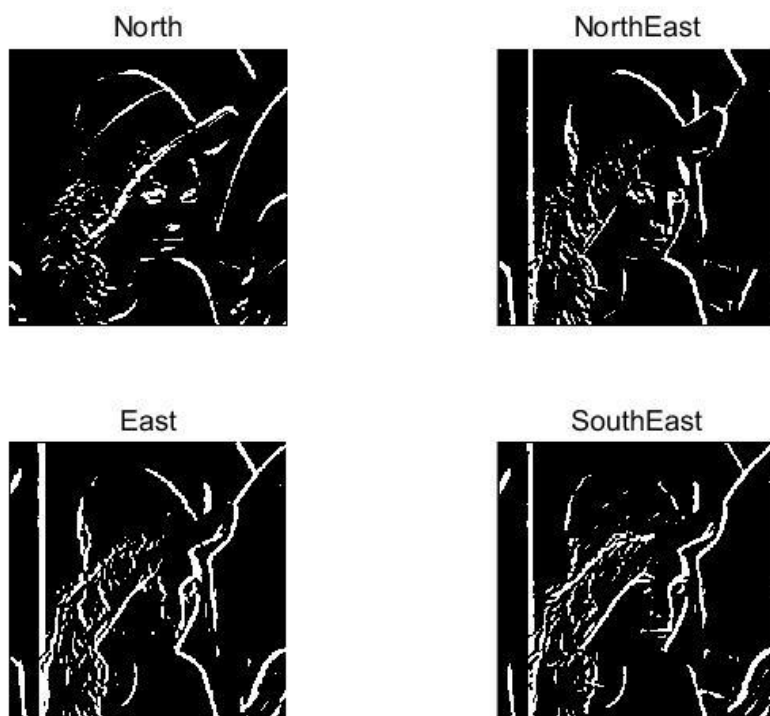
1. 中心 Laplace 算子对梯度的要求更为苛刻。观察中心 Laplace 算子可以发现，当一个像素点相对周围的八个像素点的梯度都很大时，Laplace 对该像素点的响应才会很大（且对周围像素点的梯度方向偏好也不一样，更增加了对像素点的响应难度）。因此会对复杂纹理的响应更强。

2. 本实验中对图像进行了均值滤波，使图像的弱边缘更不易被检测到。

（2）Kirsch 算子

运用针对不同方向的 Kirsch 算子进行边缘检测，比较各个 Kirsch 算子之间的区别。

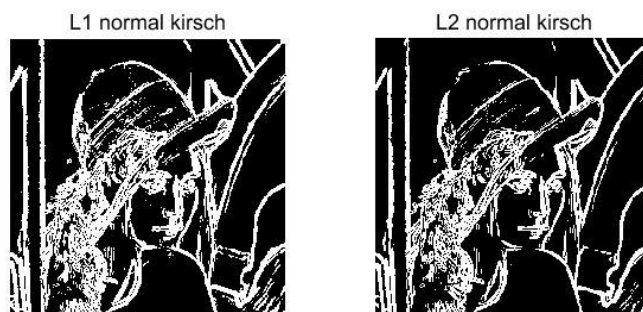




八个方向的 Kirsch 算子对图像的边缘提取效果对比

可以看出，各个 Kirsch 算子的边缘提取效果整体相似，均对某一方向的边缘有较强响应。不同方向的 Kirsch 算子对不同方向的边缘响应程度不同。这也导致仅仅使用单一方向的 Kirsch 算子会缺失很多其他方向的边缘信息。

因此，选取其中两个方向的 Kirsch 算子：South 和 West，进行 L1 范数整合和 L2 范数整合。效果如图所示。



South 和 West Kirsch 算子结合



可以看出，结合之后的边缘提取效果更好，对图像中几乎所有方向的边缘都有响应。且对噪声的抑制效果也不错。

4. 总结

本次实验中，我主要进行了以下程序的编程实现：

1. 大津阈值的计算程序并根据阈值进行图像分割。
2. 对阈值分割算法进行了优化，在保证图像准确分割的前提下达到噪声抑制效果。
3. 分别用均值滤波和中值滤波对图像进行抑制噪声处理。
4. 利用输入的图像和 `mask` 进行卷积操作。
5. 改变 `mask`，实现不同算子的边缘提取算法。
6. 选取阈值进行边缘提取。

以及以下的实验：

1. 实验不同的阈值对边缘提取效果的影响。
2. 比较 4-邻接，8-邻接，中心 Laplace 算子的效果。
3. 比较均值滤波和中值滤波对边缘提取效果的影响。
4. 比较八个不同方向的 Kirsch 算子边缘提取的效果并通过合并不同方向的 Kirsch 算子进行优化，比较优化之后的效果。
5. 比较不同 sobel 算子以及不同合并方式（L1 或 L2）的边缘提取效果。

通过本次实验，我亲自动手实现了大津阈值的图像分割程序，以及运用不同的边缘检测算子对图像进行边缘检测，并对不同算子的边缘检测效果进行了比较。在这个过程中，我还自行设计了阈值分割中噪声抑制的算法，取得了良好效果。

如何确定分割的标准以及分割效果的检验是其中的关键性问题。图像边缘检测对图像处理有着很重要的意义，在课上老师讲解了一些常用的图像边缘检测算子，但对于怎样具体实现检测过程还不是很懂。经过编程操作，一步步将老师讲解的理论变为现实，感觉收获很大。