哈爾濱工業大學

视听觉信号处理实验报告

实验 二

题　　目　视觉信号处理实验二 A

学　　院　

专　　业　

学　　号　

学　　生　

任 课 教 师　姚鸿勋

哈尔滨工业大学计算机科学与技术学院

2022 年秋季

实验一 报告

一、 实验内容（contents）

1. 选择合适的图像处理算法找到图 1.（源文件见附件）中的橘子和枣子的数量。

2. 在 1. 的基础上确定每个水果的外边界，并使用边界线或者 mask 将属于水果的像素点标注出来。

图 1 橘子们和枣子们

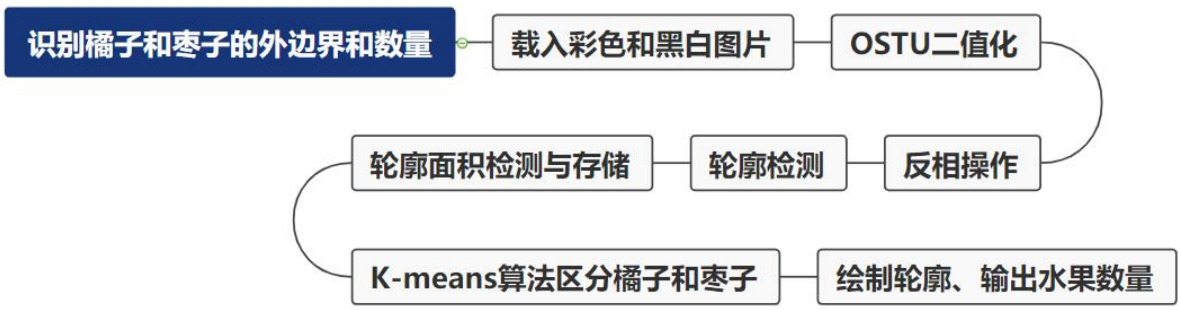
注意：只需要根据水果的大小差别区分出水果的不同种类即可

二、 实验目的（purposes）

综合运用图像处理中的知识解决实际问题，以及可能出现的多种多样的情况。通过实验，对理论知识加以理解和应用，并结合实际问题灵活选用可用的算法和处理流程。

三、 实验设计、算法和流程(Design, algorithm and procedure)

实验设计与流程：



算法：

①OSTU 二值化（最大类间方差法）

最大类间方差法是由日本学者大津(Nobuyuki Otsu)于 1979 年提出的,是一种自适应的阈值确定的方法,又叫大津法,简称 OTSU。它是按图像的灰度特性,将图像分成背景和前景两部分。因方差是灰度分布均匀性的一种度量,背景和前景之间的类间方差越大,说明构成图像的两部分的差别越大,当部分前景错分为背景或部分背景错分为前景都会导致两部分差别变小。因此,使类间方差最大的分割意味着错分概率最小。通过统计整个图像的直方图特性来实现全局阈值 T 的自动选取，其算法步骤为：

1) 先计算图像的直方图，即将图像所有的像素点按照 0~255 共 256 个 bin，统计落在每个 bin 的像素点数量

2) 归一化直方图，也即将每个 bin 中像素点数量除以总的像素点

3) i 表示分类的阈值，也即一个灰度级，从 0 开始迭代

4) 通过归一化的直方图，统计 0~i 灰度级的像素(假设像素值在此范围的像素叫做前景像素) 所占整幅图像的比例 w0，并统计前景像素的平均灰度 u0；统计 i~255 灰度级的像素(假设像素值在此范围的像素叫做背景像素) 所占整幅图像的比例 w1，并统计背景像素的平均灰度 u1；

5) 计算前景像素和背景像素的方差 g = w0*w1*(u0-u1) (u0-u1)

6) i++；转到 4)，直到 i 为 256 时结束迭代

7）将最大 g 相应的 i 值作为图像的全局阈值

得到全局阈值后，像素灰度值小于阈值的像素将被赋值为 0（黑色），大于阈值的像素将被赋值为 255（白色），实现二值化的目的。大津法二值化适用于图像直方图中存在双峰的图像（直方图中的双峰就是指背景像素和前景像素），最佳阈值就是双峰之间的某个参数，即将背景像素和前景像素分割开，其原理就是最大类间方差法。因此，OSTU 法对水果和浅色背景双峰图像相对比较适用。

②图像反相

“反相”即为图像的颜色色相反转。以前照相机的底片就是打印后的照片的反相。比如黑变白，蓝变黄、红变绿。

对灰度级范围为[0 , L-1]的一幅图像进行反转的操作为：s = L - 1 - r

其中 r 和 s 分别代表处理前后的像素值。使用这种方式反转一幅图像的灰度级，可得到等效的照片底片。

这种类型的处理特别适用于增强嵌入在一幅图像的暗区域中的白色和灰色细节，特别是当黑色面积在尺寸上占主导地位时。在本问题中，反相是为了下一步边缘检测做铺垫。

③轮廓检测

本实验中采用 OpenCV 中轮廓提取函数 findContours()检测轮廓，实现的是 Satoshi Suzuki and others. Topological structural analysis of digitized binary images by border following. Computer Vision, Graphics, and Image Processing, 30(1):32－46, 1985.这篇论文的算法。解决的是对于二值图像的轮廓提取问题。

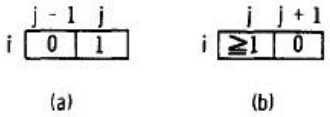
NBD:从边界开始点(i, j)以边界跟踪算法可以得到一条边界，为每条新找到的边界 B 赋予一个新的唯一的编号，NBD 表示当前跟踪的边界的编号。

LNBD：在光栅扫描的过程中，我们也保存最近遇到（上一个）的边界 B'的编号，记为 LNBD。

假设输入图像为 $F=\{f_{ij}\}$，将初始 NBD 设为 1（把 F 的 frame 看成第一个边界）。使用光栅扫描法扫描图像 F，当扫描到某个像素点(i, j)的灰度值 $f_{ij}\neq 0$ 时执行下面的步骤。每次当我们扫描到图片的新行的起始位置时，将 LNBD 重置为 1。

1）从下列情况选一种：

（a）如果 $f_{ij}=1$ 并且 $f_{i,j-1}=0$，即(a)所描述情况，则(i,j)是外边界开始点，NBD+=1，$(i_2,j_2)\leftarrow(i,j-1)$。



（b）如果 $f_{ij}\geq 1$ 并且 $f_{i,j+1}=0$，即(b)所描述情况，则(i,j)是孔边界开始点，NBD+=1，$(i_2,j_2)\leftarrow(i,j+1)$。如果 $f_{ij}>1$，则 $LNBD\leftarrow f_{ij}$。

（c）其他情况，到第（4）步

（2）根据上一个边界 B'和当前新遇到边界 B 的类型，我们可以从表 1 得到当前边界 B 的父边界。

TABLE 1

Decision Rule for the Parent Border of the Newly Found Border *B*

Type of the border *B′* with the sequential number LNBD / Type of *B*	Outer border	Hole border
Outer border	The parent border of the border *B′*	The border *B′*
Hole border	The border *B′*	The parent border of the border *B′*

（3）从边界开始点 (i,j) 开始按(3.1)到(3.5)进行边界跟踪。

（3.1）以(i,j)为中心，(i_2,j_2) 为起始点，按顺时针方向查找(i,j)的 4（8）邻域是否存在非 0 像素点。若找到非 0 像素点，则令(i_1,j_1)是顺时针方向的第一个非 0 像素点；否则令 $f_{ij}=-NBD$，转到（4）。

（3.2）$(i_2,j_2)\leftarrow(i_1,j_1)$，$(i_3,j_3)\leftarrow(i,j)$

（3.3）以 (i_3,j_3) 为中心，按逆时针方向，(i_2,j_2)的下一个点为起始点查找(i_3,j_3)的 4（8）邻域是否存在非 0 像素点，令(i_4,j_4)是逆时针方向的第一个非 0 像素点。

（3.4）

（a）如果(i_3,j_{3+1})是(3.3)中已经检查过的像素点且是 0 像素点，则 $f_{i3,j3}\leftarrow -NBD$。

（b）如果(i_3,j_{3+1})不是(3.3)中已经检查过的 0 像素点，并且 $f_{i3,j3}=1$，则 $f_{i3,j3}\leftarrow NBD$。

（c）其他情况，不改变 $f_{i3,j3}$。

（3.5）如果$(i_4,j_4)=(i,j)$且 $(i_3,j_3)=(i_1,j_1)$（回到了边界开始点），则转到(4)；否则令$(i_2,j_2)\leftarrow(i_3,j_3)$，$(i_3,j_3)\leftarrow(i_4,j_4)$，转到(3.3)

（4）如果 $f_{ij}\neq1$，则 LNBD←$|f_{ij}|$,从点 (i,j+1) 继续光栅扫描。当扫描到图片的右下角顶点时结束。

④轮廓面积计算

opencv 计算轮廓内面积函数 cv2.contourArea 使用的是格林公式计算轮廓内面积，公式如下：

$$\oint_L Pdx+Qdy=\iint_D(\frac{\partial Q}{\partial x}-\frac{\partial P}{\partial y})dxdy$$

该函数部分源码：

```
for( int i = 0; i < npoints; i++ )
    {
        Point2f p = is_float ? ptsf[i] : Point2f((float)ptsi[i].x, (float)ptsi[i].y);
        a00 += (double)prev.x * p.y - (double)prev.y * p.x;
        prev = p;
    }
    a00 *= 0.5;
    if( !oriented )
        a00 = fabs(a00);
```

其中，本实验通过计算轮廓面积的大小判断水果种类。

⑤k-means 算法

因为已知画面中有两张大小不同的水果，大的是橘子，小的是枣子。因此通过轮廓内面积(可以看作一维向量)的聚类为两个簇，令质心大的簇为橘子簇，质心小的簇为枣子簇，根据簇内样本数量可以判断不同水果的数量。

其中，K-means 算法又名 k 均值算法,K-means 算法中的 k 表示的是聚类为 k 个簇，means 代表取每一个聚类中数据值的均值作为该簇的中心，或者称为质心，即用每一个的类的质心对该簇进行描述。

其算法思想大致为：先从样本集中随机选取 k 个样本作为簇中心，并计算所有样本与这 k 个“簇中心”的距离，对于每一个样本，将其划分到与其距离最近的“簇中心”所在的簇中，对于新的簇计算各个簇的新的“簇中心”。

根据以上描述，我们大致可以猜测到实现 kmeans 算法的主要四点：

1）簇个数 k 的选择

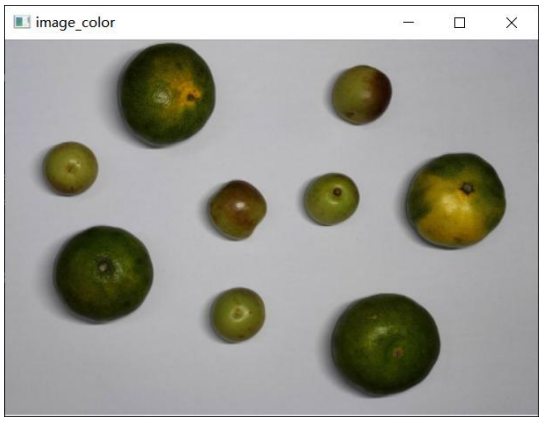
2）各个样本点到“簇中心”的距离

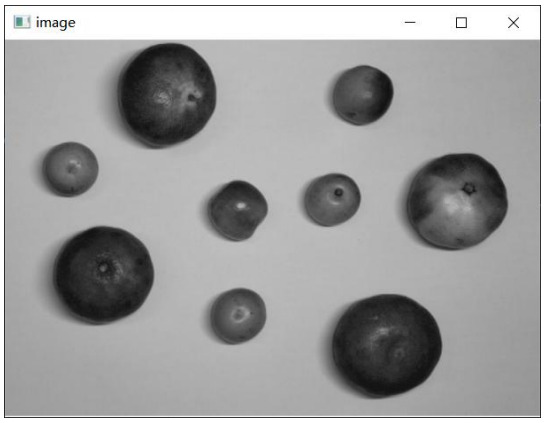
3）根据新划分的簇，更新“簇中心”

4）重复上述 2、3 过程，直至"簇中心"没有移动

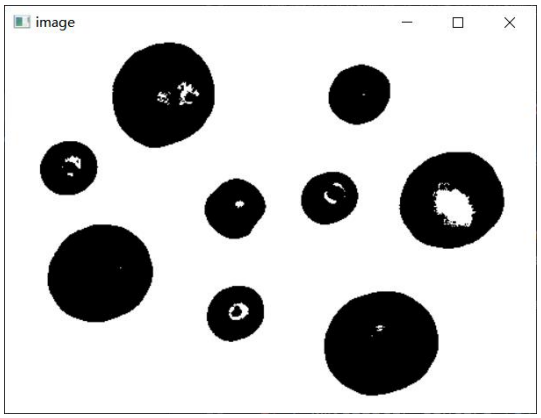
该算法优点：容易实现。缺点：可能收敛到局部最小值，在大规模数据上收敛较慢。显然，本实验我问题中只有个位数数量级的轮廓面积需要分类，适用于 k-means 算法。

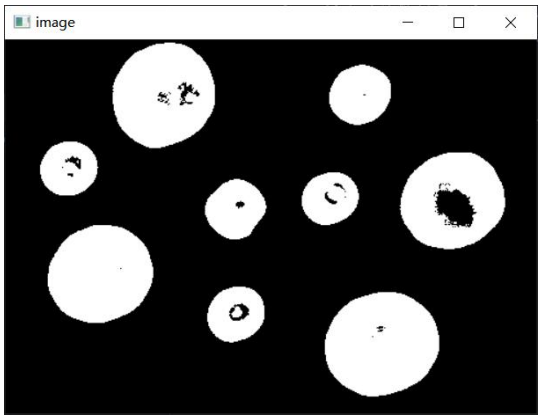
四、 实验结果(results)



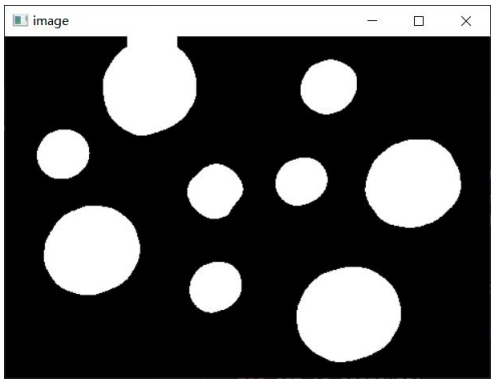


成功加载彩色和黑白图像。





直接对黑白图像进行 OSTU 二值化，可以根据图像灰度的方差自动区分前景和后景，并返回二值化的阈值。从中可以看出，OSTU 二值化虽然不能完全完美得区分背景和橘子的浅色部分，但是对于二值化后能够明显地界定外轮廓，有很好的效果。为了进一步进行轮廓提取，对该二值图像进行反相操作。



此外，一开始实验在处理尝试，通过多次膨胀、腐蚀的闭合操作将水果内部白色部分填满。可以构造 5*5 卷积核，在 5 次膨胀、5 次腐蚀后填补水果内部浅色部分。填补水果内部效果较好，但会导致靠近边缘的水果与边框粘连，故不进行闭合操作，而是通过后续的提取外轮廓避免水果内部斑纹轮廓的影响。

上图是提取外部轮廓（RETR_EXTERNAL）后的效果，根据轮廓的等级很好地识别了水果的边界。并计算出了各个轮廓的面积：

[7593. 2000.5 6541.5 2159. 1870.5 6420. 1967. 2305. 6833.5]

最后,对面积列表根据 k-means 算法聚类计算不同水果的个数，输出：

枣子的数量是： 5

橘子的数量是： 4

五、 结论(conclusion)

1. OSTU 二值化是处理前后景灰度差异大，分离物品和背景行之有效的算法，但是仅仅根据灰度信息,无法考虑到物品的斑点等属性,无法区分物品之间的关系。因此会受到物体噪声的影响，全局二值化还容易受到阴影的影响。

2. 多次膨胀、腐蚀操作能够闭合缝隙，但可能会与边缘粘连，应当通过配合其他方法予以克服。

3. opencv 中的 findcountours 算子对二值化图像边缘轮廓提取有很好的效果，而且能识别轮廓的等级关系，可以方便地只提取外部轮廓，效果很好。

六、 参考文献(reference)

[1]Rafael C. Gonzalez, Richard E. Woods, Digital Image Processing (Third Edition)

[2]Satoshi Suzuki and others. Topological structural analysis of digitized binary images by border following. Computer Vision, Graphics, and Image Processing, 30(1):32－46, 1985.

[3] Guenin B , Knemann J , Tunel L . A Gentle Introduction to Optimization. 2018.