

图像分割各经典方法的对比研究实验报告

实验目的

- 1、通过实验，了解图像分割常用的方法，掌握各经典方法背后的数学原理；
- 2、通过 Python 的编程，掌握 numpy 库、OpenCV 库等图像处理常用的库函数。

实验内容

- 1、学习阈值类图像分割(最大类间方差法、基于最大熵的阈值分割、迭代阈值分割法)数学原理；
- 2、学习边缘检测 (以 Canny 为例)数学原理；
- 3、学习基于马尔可夫随机场(MRF)图像分割的数学原理；
- 4、根据上述三种图像分割方法的数学原理，基于 Python 语言，完成程序的编写，对比自写代码的实现效果与参考代码的实现效果并分析。

一、实验背景

图像处理技术发展到今天，已然成为了各种智能识别技术的必要性基础性研究，在国家建设和百姓生活的方方面面都已得到了广泛应用。而图像分割是计算机视觉领域中的基础技术之一，起步于上世纪七十年代，已经有几十年的历史，是指利用图像的某些特性，将一幅图像分割成若干个独立的有共同像素属性的对象，其实质是一个按照像素属性进行聚类的过程，是图像识别、场景解析、对象检测等任务的预处理^[1]。研究者们从图像的灰度、颜色、纹理、形状等这些特征出发，利用各种数学理论和工具，使用不同的模型，对灰度以及彩色图像进行分割处理，形成了错综复杂的图像分割方法。各种图像分割方法相互交叉，相互影响，相互促进，形成了一个庞大的体系，为后来的计算机视觉领域的学术创新发展奠定了良好的基础。

二、图像分割各方法原理介绍

所谓图像分割，就是指根据图像中各内容的意义，将图像分割成不同区域的方法。而分割成的区域必须遵循互不相交、每一区域都满足特定区域一致性的原则^[2]，即将图像 $g(x, y), (0 \leq x \leq \max(x), 0 \leq y \leq \max(y))$ 分割成满足以下条件的子区域 $g_1, g_2, g_3 \dots$ ：

(1) $\bigcup_{k=1}^N g_k(x, y) = g(x, y)$ ，即所有子区域组成了整个图像；

(2) g_k 是连通的区域；

(3) $g_i(x, y) \cap g_j(x, y) = \emptyset$ ，即任意两个子区域不存在公共元素；

(4) 区域 g_k 满足一定的均匀性条件，均匀性一般指同一区域内的像素点之间的灰度值差异较小或灰度值的变化较缓慢。

2.1 阈值类图像分割方法原理介绍

2.1.1 最大类间方差法(OTSU)原理简介

OTSU 法，又称最大类间方差法，是目前公认的，对阈值分割选择相对合理，且分割效果良好的办法。其理论依据为：按照图像的灰度特性，将图像分为背景和物体两部分，通过计算得到一个阈值，该阈值满足：使背景和物体的像素差异最大，即类间方差最大^[3]。

OTSU 法的基本思想：设定分割阈值，将像素 $f(a, b)$ 分为对象和背景两类；通过最大化类间方差并最小化类内方差得到目标阈值。

假设一幅大小为 $M \times N$ 图像的像素分为 L 个灰度级 $\{0, 1, 2, \dots, L-1\}$, n_i 为灰度级为 i 的像素个数，则像素总数公式为：

$$MN = n_0 + n_1 + \dots + n_{L-1} \quad (1)$$

各灰度级的概率为：

$$P_i = \frac{n_i}{MN} \quad (2)$$

其中，概率满足 $\sum_{i=0}^{L-1} P_i = 1$ 且 $P_i \geq 0$ 。若目标阈值为 k ，将所有像素分为灰度值为 $[0, k]$ 的背景类 C_1 和 $[k+1, L-1]$ 的对象类 C_2 。 $P_1(k)$ 和 $P_2(k)$ 为 C_1 和 C_2 的概率，可由式(3)(4)计算得到：

$$P_1(k) = \sum_{i=0}^k P_i \quad (3)$$

$$P_2(k) = \sum_{i=k+1}^{L-1} P_i = 1 - P_1(k) \quad (4)$$

其中 $m_1(k)$ 和 $m_2(k)$ 分别为 C_1 和 C_2 像素的灰度平均值，可由式(5)和式(6)计算得到：

$$m_1(k) = \sum_{i=0}^k i \cdot P(i | C_1) = \frac{1}{P_1(k)} \sum_{i=0}^k i \cdot P_i \quad (5)$$

$$m_2(k) = \sum_{i=k+1}^{L-1} i \cdot P(i | C_2) = \frac{1}{P_2(k)} \sum_{i=k+1}^{L-1} i \cdot P_i \quad (6)$$

m_G 为图像平均灰度值， θ 为类间平方差，可由式(7)得到：

$$\begin{aligned} \theta &= P_1(k)(m_1(k) - m_G)^2 + P_2(k)(m_2(k) - m_G)^2 \\ &= P_1(k)P_2(k)(m_1(k) - m_2(k))^2 \end{aligned} \quad (7)$$

由式(7)可得式(8)：

$$\theta(T) = \max_{0 \leq k \leq L-1} \theta(k) \quad (8)$$

其中， T 即为目标阈值，输入图像就可以按照阈值 T 进行分割，像素具体灰度显示由式(9)得到：

$$f(x, y) = \begin{cases} 255, & f(x, y) > T \\ 0, & f(x, y) \leq T \end{cases} \quad (9)$$

2.1.2 基于最大熵的阈值分割原理简介

由于 OTSU 法在分割图像之时只考虑将背景和对象分割最大化，而忽略了边缘和弱对象等信息，故本文提出与最大熵分割相结合，使分割时能够将对象最大化分割的同时保留更多的信息^[4]。最大熵原理求阈值与 OTSU 类似，OTSU 是求取最佳阈值将图像分为对象和背景求取最大方差，最大熵是求取最佳阈值使对象和背景 2 个部分熵之和最大。熵的公式为：

$$H(I) = \sum_{u,v} p(I(u,v)) \cdot \log_b \left(\frac{1}{p(I(u,v))} \right) = - \sum_{u,v} p(g) \cdot \log_g(p(g)) \quad (10)$$

约束条件为 $\sum_{i=0}^{L-1} P_i = 1, P_i \geq 0, P_n = \sum_{i=0}^t P_i$, b 常取 2, $I(u,v)$ 为灰度值, 假设阈值为 t , A 为 $\{0,1,\dots,t\}$ 的灰度分布, B 为 $\{t+1,t+2,\dots,L-1\}$ 的灰度分布, 则两个概率密度相关的熵分别为 $H(A)$ 和 $H(B)$, 由式(11)和式(12)计算得到:

$$H(A) = - \sum_{i=0}^t \frac{P_i}{P_n} \ln \left(\frac{P_i}{P_n} \right) \quad (11)$$

$$H(B) = - \sum_{i=t+1}^{L-1} \frac{P_i}{1-P_n} \ln \left(\frac{P_i}{1-P_n} \right) \quad (12)$$

定义函数 $\varphi(t)$ 为 $H(A)$ 和 $H(B)$ 的和:

$$\varphi(t) = H(A) + H(B) \quad (13)$$

则最佳阈值为:

$$t = \arg_t \max \varphi(t) = \arg_t \left(- \sum_{i=0}^t \frac{P_i}{P_n} \ln \left(\frac{P_i}{P_n} \right) - \sum_{i=t+1}^{L-1} \frac{P_i}{1-P_n} \ln \left(\frac{P_i}{1-P_n} \right) \right) \quad (14)$$

2.1.3 迭代阈值分割原理简介

迭代法选择初始的估计阈值, 按某种原则, 通过迭代不断改变这一估计值, 直到满足给定的准则为止。迭代法实现了在实际处理中, 图像之间有较大变化时, 对每一幅图片进行自动估计阈值, 达到良好分割效果。

统计图像灰度直方图, 求出图像的最大灰度值和最小灰度值, 分别记为 Z_{max} 和 Z_{min} , 令初始阈值:

$$T_0 = \frac{Z_{max} + Z_{min}}{2} \quad (15)$$

根据阈值 T_k 将图象分割为前景和背景, 计算小于 T_0 所有灰度的均值 Z_0 和大于 T_0 的所有灰度的均值 Z_B , 求出新阈值:

$$T_{k+1} = \frac{Z_0 + Z_B}{2} \quad (16)$$

若 $T_k = T_{k+1}$, 则所得即为阈值, 否则转(15)进行迭代计算, 直到求出最佳阈值^[5]。

2.2 边缘检测(以 Canny)原理简介

1986 年, John.Canny 把边缘检测问题转换为检测单位函数的极大值问题, 其基本思想是在图像中找出具有局部最大梯度幅值的像素点, 检测平滑图像的二阶方向导数的过零点(zero-crossings), 通过非最大化抑制方法对边缘进行定位, 得到候选边缘点, 然后用双阈值

法进行边缘二值化，最终的边缘检测结果主要由高阈值确定^[6]。Canny 认为一个优良的边缘检测算子应具有以下 3 个特性：好的检测性能、好的定位性能、对同一边缘要有低的响应次数，并以此提出了判定边缘检测算子的三个准则^[7]：

①信噪比准则：信噪比准则要求输出信号的信噪比越大，则非边缘点判断为边缘点或边缘点判断为非边缘点的概率最小，即误检率越低；

②定位精度准则：检测出的边缘点与实际边缘点距离最小，使定位精度最高；

③单边响应准则：单个边缘产生多个响应的概率要低，虚假边缘的响应要得到最大的抑制。

Canny 算子边缘检测采用二维高斯函数的任一方向上的一阶方向导数为平滑滤波器，通过与图像卷积进行滤波，然后对滤波后的图像寻找图像梯度的局部极大值，以确定图像边缘。

(1) 图像平滑

设一维高斯函数为

$$G = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{x^2}{2\sigma^2}} \quad (17)$$

其中 σ 是高斯函数的标准差，也是尺度参数。用上面的高斯函数构造滤波器，分别按照行和列对原始图像 $I(x, y)$ 进行卷积操作，得到平滑图像 $K(x, y)$ 。通过平滑滤波可以起到抑制噪声的作用，这里高斯滤波函数的标准差 σ 控制着平滑的程度。当 σ 较小时，滤波器也较短。卷积运算量小，定位精度高，但信噪比低；当 σ 较大时，情况恰好相反。因此，要根据实际需要适当选取高斯滤波器参数。

(2) 计算梯度的幅值和方向

根据邻域一阶偏导的有限差分计算平滑后的图像 $K(x, y)$ 的梯度幅值 $M(x, y)$ 和梯度方向 $Gd(x, y)$ ：

$$M(x, y) = \sqrt{f_x(x, y)^2 + f_y(x, y)^2} \quad (18)$$

$$Gd(x, y) = \arctan \frac{f_y(x, y)}{f_x(x, y)} \quad (19)$$

其中 $f_x(x, y) = K(x, y + 1) - K(x, y - 1)$, $f_y(x, y) = K(x + 1, y) - K(x - 1, y)$ ，下标 x 和 y 分别代表 $K(x, y)$ 沿着 x 方向和 y 方向的方向偏导数。

(3) 非极大值抑制

分别对每个方向进行非极大抑制(Canny 算法中，方向为 4)，若像素点 (x, y) 的梯度幅值 $M(x, y)$ 不小于梯度方向上两个相邻像素点的梯度幅值，则判断该点为可能的边缘点，否则，把它从边缘点的集合中剔除。而由于图像中， x 轴水平向右， y 轴竖直向下，因此梯度方向角

度从 x 轴顺时针向下计算(见图 2-1):

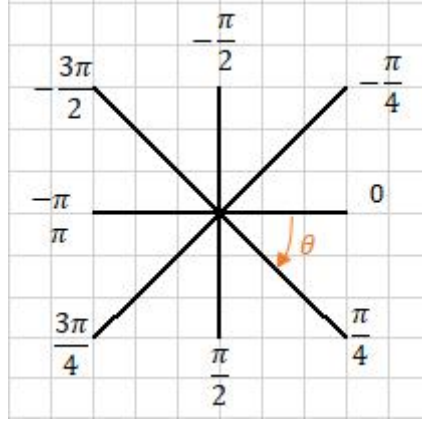


图 2-1 梯度方向角度

(4) 双阈值选取

使用累计直方图计算两个阈值(低阈值)和(高阈值), 对梯度图像进行双阈值化处理, 可得到一个高阈值检测结果和低阈值检测结果, 再在前者中连接边缘轮廓, 连接到端点时, 到后者寻找弱边缘点, 弥补前者的边缘间隙, 一直到找不到与它相邻接的弱边缘点为止^[8]。

2.3 基于马尔可夫随机场(MRF)的图像分割方法原理简介

实际中在用马尔可夫随机场(MRF)对视觉问题进行建模时, 首先要根据问题的特点定义相应的邻域系统、簇和势函数, 然后得到它的联合概率分布。精确的分割一幅图像中有几个部分所对应的图像内容是图像聚类的目的, 聚类结果可被视为具有不同均值和方差的高斯混合模型。每类对应于一个高斯分布:

$$P(s_i|W) = \frac{1}{\sqrt{2\pi\sigma_i}} e^{-\frac{(s_i-\mu_i)^2}{2\pi\sigma_i}} \quad (20)$$

其中 s_i 是一个图像类的类内像素值, W 是图像标记场, 即为图像分割各类结果。假设整个图像 S 条件独立, 有条件密度 $P(S|W)$:

$$P(S|W) = \prod_{i \in L} P(s_i|W) \quad (21)$$

其中, L 为图像聚类范围, 即 $w = 1, 2, \dots, L$ 。 $P(S|W)$ 就表示已知像素点标记 w 的情况下, 它是真实的像素点 s 的概率

由贝叶斯定理可以得知:

$$P(W|S) = \frac{P(S|W) \cdot P(W)}{P(S)} \quad (22)$$

其中, $P(W)$ 是标记结果 W 的先验概率。由 Hammersley-Clifford 定理, 可以得知马尔可夫随机场与 Gibbs 随机场等价, 因此标记结果 W 的先验概率 $P(W)$ 可以用 Gibbs 随机场的概率等

价替换^[9]，即：

$$P(W) = \frac{1}{z} e^{-\frac{1}{T} U_2(W)} \quad (23)$$

$$U_2(W) = \sum_{c \in C} V_c(w_c) \quad (24)$$

其中， $z = \sum_W e^{-\frac{1}{T} U_2(W)}$ 是归一化常数，可用参数 T 控制 $P(W)$ 。 C 为所有势团的集合， $V_c(w_c)$ 为邻域子团势能量函数^[10]：

$$V_c(w_c) = V_{s,r}(w_s, w_r) = \begin{cases} -\beta, w_s = w_r \\ \beta, w_s \neq w_r \end{cases} \quad (25)$$

通过寻找最优标记场 W_{opt} ，使后验概率达到最大值，即：

$$W_{opt} = \max \arg(P(W|S)) = \max \arg \left[\frac{P(S|W) \cdot P(W)}{P(S)} \right] \quad (26)$$

三、实验结果对比

基于第二章所提到的方法原理，利用 Python 语言进行编程，将程序输出结果与参考代码中使用 Python 内部 OpenCV 库函数实现结果对比。首先，绘制 lena 图的灰度直方图，据此大致分析 lena 图的灰度分布情况(见图 3-1)：

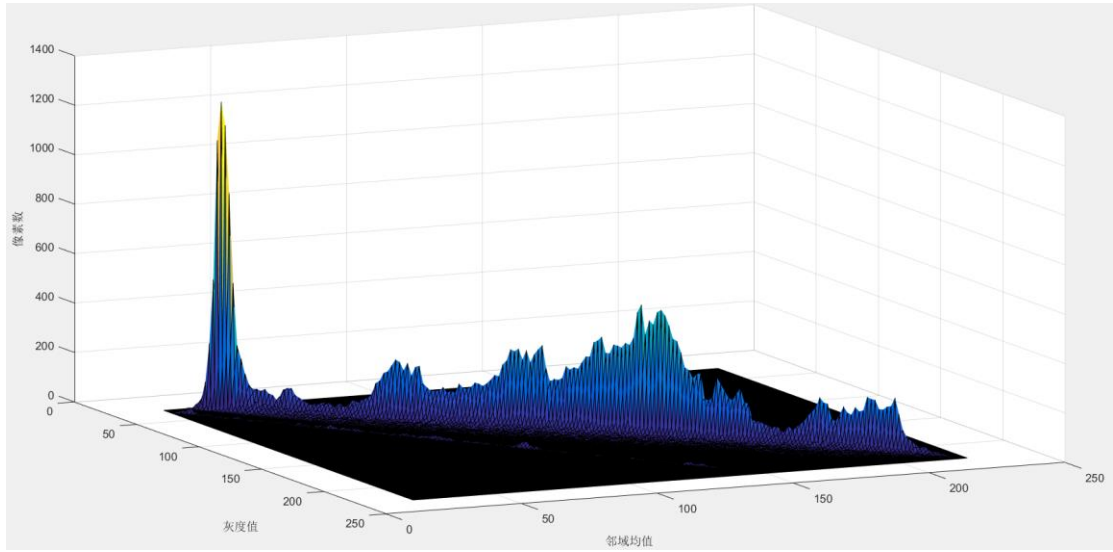
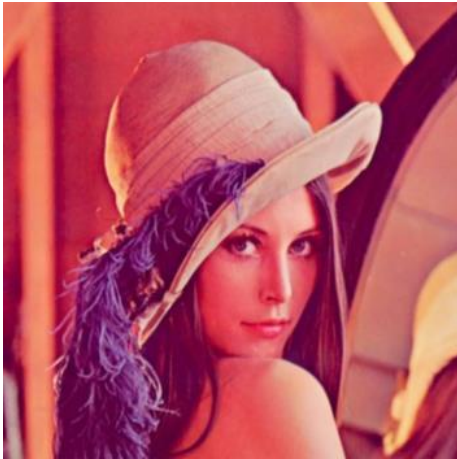


图 3-1 lena 灰度直方图

由图 3-1 可以看出，lena 图在低灰度较集中范围的像素数最多，说明 lena 图存在较多的深色区域，可以判断是在人物头发区域；而中灰度范围灰度跨度很广，且像素数相差不大，可以判断为图像的背景区域；在高灰度范围存在少量像素的浅色区域，可以判断是在人物的脸部区域。

3.1 阈值类图像分割代码实现效果



(a) 原图像



(b) 灰度图



(c) 最大类间方差 OTSU 按原理编写结果



(d) 最大类间方差 OTSU 库函数编写结果



(e) 基于最大熵按原理编写结果



(f) 基于最大熵库函数编写结果

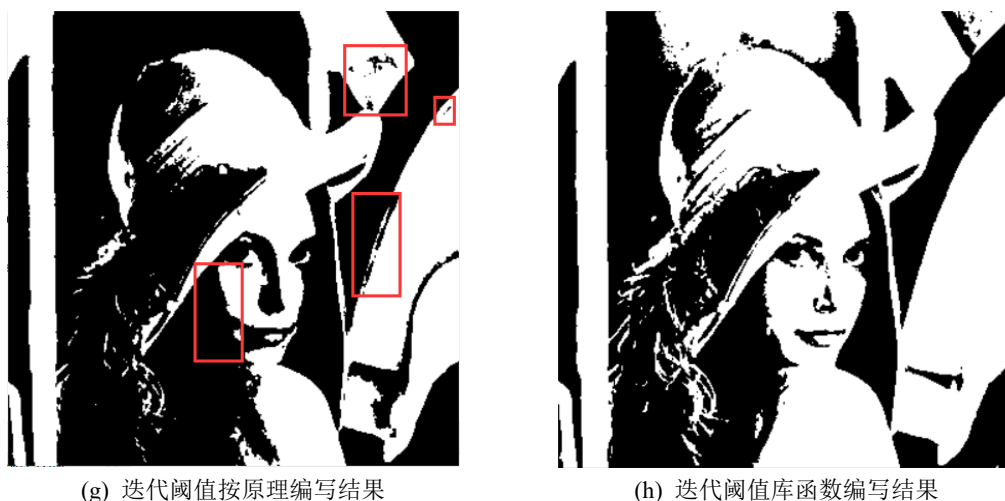


图 3-2 以 lena 图为例的代码实现效果

图 3-2 的(c)(d)(e)(f)(g)(h)为代码实现效果，左列均为按算法原理编写(自写)，右列均为运用 Python 的 OpenCV 库函数编写(参考代码)。

(c)(d)显示了最大类间方差法按原理编写和使用 OpenCV 库函数实现的效果几乎无异，但是与后面几个方法对比来看，二者都缺少了右下角的图片信息，可以认为是 OTSU 方法导致的信息丢失；

(e)(f)显示了基于最大熵的按原理编写和使用 OpenCV 库函数实现的效果也几乎一样，与最大类间方差法 OTSU 相比，右侧的红框显示分割结果仍存在少许噪声，并没有分割干净，而且 lena 人像面部也没有完全分割出来，不过利用库函数的分割效果也同样存在这些问题。但最大熵的分割结果保留了图像右下角的信息，这是优于 OTSU 的地方；

(g)(h)显示出的迭代阈值法的两种代码实现效果则存在一定程度的区别，按迭代阈值原理编写的代码偏向基于最大熵的阈值分割，存在没有分割干净、面部细节分割不充分的问题，库函数编写的效果则有些偏向最大类间方差法，面部分割的很清楚但是图片右下角的信息有所丢失。

由此可见，三种阈值分割类的图像分割方法各有利弊，具体使用时可按实际需要选择方法。

3.2 Canny 边缘检测代码实现效果

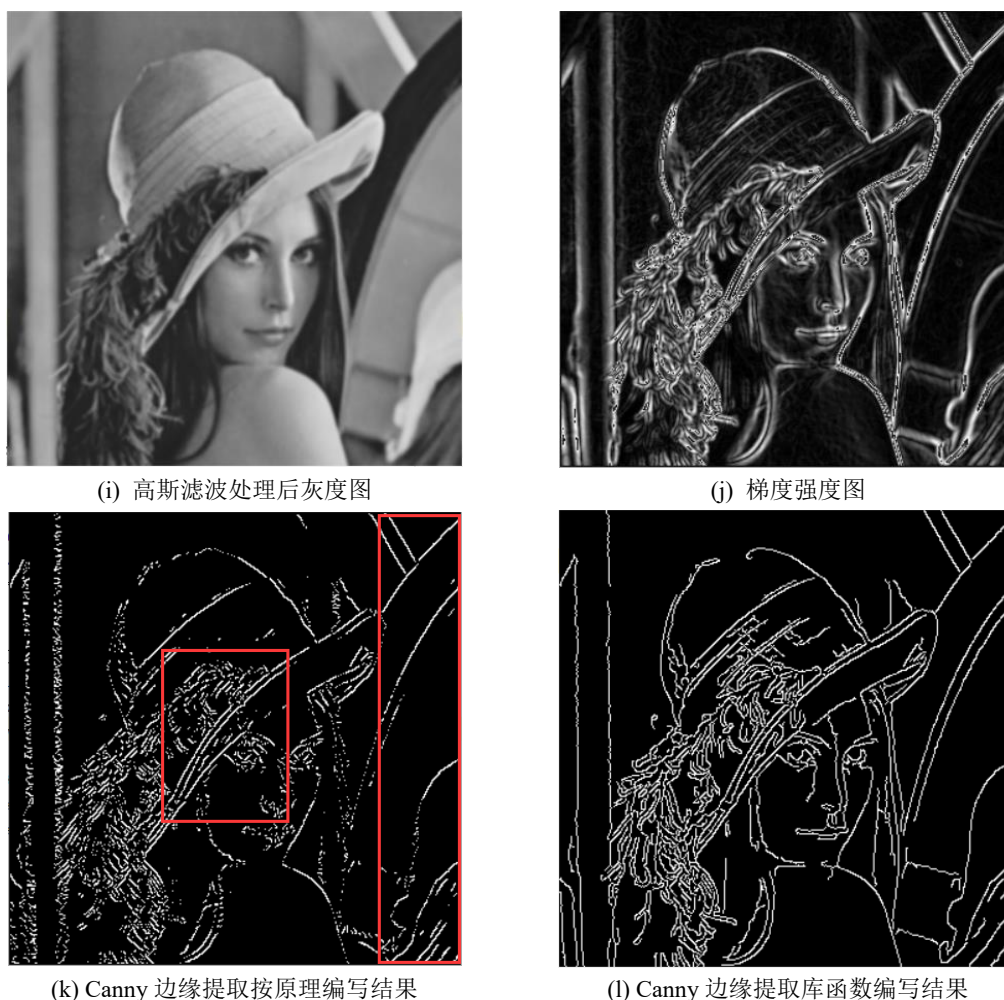


图 3-3 Canny 边缘提取代码实现效果

图 3-3 显示了 Canny 边缘提取的代码实现效果，(k)为按照 Canny 原理编写的代码(自写)效果，(l)是使用 Python 中的 OpenCV 库函数 `cv2.Canny` 的实现效果。

(i)为使用高斯滤波处理过的灰度图像，与未滤波处理的灰度图(b)对比，可以发现(i)的清晰度较低但是噪声更少。滤波去噪处理是边缘提取的预处理，目的是减少图像自带的噪声，方便后续的 Sobel 算子的卷积计算；

(j)为卷积计算得出的梯度强度图，本文选用 Sobel 算子进行卷积运算。其中，卷积运算过程也是按照原理手动编写；

(k)为根据卷积得到的梯度强度及各像素的梯度方向的基础上，进行非极大值抑制和双阈值提取边缘而得的边缘提取结果，与 OpenCV 库函数 `cv2.Canny` 的结果(l)相对比还是存在较大差距的：根据 Canny 原理自写程序所得的边缘提取结果大部分呈散点形式，并不像 `cv2.Canny` 的结果那样呈清晰连续的边缘线，仅图像右边的背景和人物帽沿部分是连续的线(红框所示)。但纵观整个图像，和库函数的结果相对比，基本上没有出现特别重大的边缘判

断错误或边缘丢失的情况，可以认为基本提取准确，需要解决的难点在于散点无法连成线、呈现边缘不清晰的问题。

而且二者在程序复杂度和运行速度方面也有着较大差距：原理自写程序共 76 行，共耗时 8.0714 秒；而 cv2.Canny 程序共 7 行，耗时 0.0353 秒，从这个角度结合边缘提取效果，二者差距甚远。

3.3 基于马尔可夫随机场(MRF)的图像分割



图 3-4 马尔可夫随机场代码实现效果

(m)(n)分别为马尔可夫随机场方法按原理编写和参考代码的图像分两类的 10 次循环结果，(o)(p)分别为两种代码的图像分两类 50 次循环结果。

将(m)(n)与阈值类图像分割结果对比可以看到，马尔可夫随机场方法的分割效果有些类似最大类间方差法(OTSU)的分割效果，都存在人物头顶处背景与人物帽子未分割开和图像右下角信息丢失的问题(如红框所示)，但是与基于最大熵的方法相比，马尔可夫随机场方法将人物的面部分割的更加清楚。

对比(m)(n)可以看到，按原理编写比参考代码分割的结果虽然多了些噪声，但是保留了

更多的细节(如人物嘴部、鼻子和帽沿)。

而对比(o)(p)和(m)(n)就能发现，按照马尔可夫随机场原理编写的代码的 50 次循环结果和自写程序的 10 次循环结果几乎没有区别，但参考代码的 50 次循环结果则比其 10 次循环的结果少了更多细节(人物嘴部、帽沿)。

但从程序运行方面考虑，参考代码从数据结构的角度入手，经过算法优化，比根据原理用 for 循环实现 所占内存和所耗时间都要小得多：按原理编写代码循环 10 次运行时间共 31.8143 秒，而参考代码循环 10 次运行时间仅 0.4448 秒；同样的，循环 50 次时，按原理编写代码运行时间共 145.6898 秒，参考代码运行时间仅 1.7904 秒，相差两个数量级，如此看来原理编写代码在数据结构方面优化改进的空间还有很多。

下面考虑基于马尔可夫随机场将图像分为 4 类的情形：



图 3-5 基于马尔可夫随机场将图像分为 4 类结果

由图 3-5 根据马尔可夫随机场的原理，将 lena 图分为 4 类，以灰度深浅表示结果可以看出，人物头顶的背景与帽子上部仍被分成一类，而且在最大类间方差 OTSU 和最大熵分割产生明显差别的人物侧面部和人物鼻子则是被单独分出一类，这说明基于马尔可夫随机场的图像分割可以同时兼顾最大类间方差 OTSU 和最大熵对人物面部的分割。同时，图像右下角被丢失的边缘信息也被很清晰的提取出来，并没有被忽略。由此可见，马尔可夫随机场是一种很合适的图像分割方法。

四、参考文献

- [1]. 周莉莉,姜枫.图像分割方法综述研究[J].计算机应用研究,2017,34(07):1921-1928.
- [2]. 罗希平,田捷,诸葛婴,王靖,戴汝为.图像分割方法综述[J].模式识别与人工智能,1999,12(03):300-312.
- [3]. 景晓军,蔡安妮,孙景鳌.一种基于二维最大类间方差的图像分割算法[J].通信学报,2001(04):71-76.
- [4]. 周德龙,潘泉,张洪才,戴冠中.最大熵阈值处理算法[J].软件学报,2001(09):1420-1422.
- [5]. 邓林华,许骏,程向明.基于迭代阈值的太阳像分割算法的应用研究[J].计算机与现代化,2010(10):72-74.
- [6]. 刘洋,薛向阳,路红,郭跃飞.一种基于边缘检测和线条特征的视频字符检测算法[J].计算机学报,2005(03):427-433.
- [7]. 曾俊. 图像边缘检测技术及其应用研究[D].华中科技大学,2011.
- [8]. 王丹. 基于各向异性高斯滤波的图像边缘检测方法[D].西安电子科技大学,2010.
- [9]. 李旭超,朱善安.图像分割中的马尔可夫随机场方法综述[J].中国图象图形学报,2007(05):789-798.
- [10]. 刘伟强,陈鸿,夏德深.基于马尔可夫随机场的快速图象分割[J].中国图象图形学报,2001(03):26-31.