课堂练习4

姓名：骆禹松　　班级：2018211129　　学号：2018210071

1 实验目的

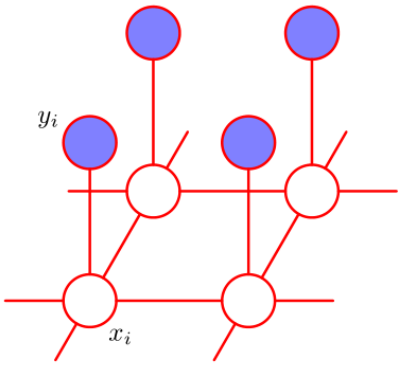
- 参考教材8.3.3，自行选取一张灰度图像（可通过对彩色图像处理得到），每个像素由8比特表示。
- 对图像加噪声。
- 使用ICM算法进行图像去噪。

2 实验原理

噪声图像可以用一组二进制像素值描述，即$y_i \in \{-1, +1\}$，其中$i = 1, \ldots, D$包括了所有像素值。噪声图像可以通过对无噪声图像进行处理得到，比如令无噪声图像中的像素点随机以小概率翻转。无噪声图像也可以用一组二进制像素值描述，即$x_i \in \{-1, +1\}$。我们的目的是将噪声图像恢复为原始无噪声图像。

由于噪声级别较小，x_i与y_i之间有很强的相关性，并且一张图片中的相邻像素点x_i与x_j也有很强的相关性。这些先验条件可以通过马尔科夫随机场获得，其无向图结构如下图所示。图中有两种团，每种都包含两个变量。$\{x_i, y_i\}$形式的团可以用能量函数表示变量之间的相关性。可以选择非常简单的公式$-\eta x_i y_i$来作为能量函数，其中η是一个正数。即当x_i与y_i具有相同的符号时，能量较低，反之能量较高。

相似地，$\{x_i, x_j\}$形式的团可以用能量函数$-\beta x_i y_i$来表示变量间的相关性，其中β是一个正数。



由于势函数是极大团上的任意非负函数，我们可以将其与团的子集的任意非负函数相乘，或者我们也可以将对应的能量相加。因此可以给无噪声图像中的每个像素点i增加一个额外的项hx_i，使处理过的图像整体偏向某一个像素值。

综上所述，完整的能量函数表达式为

$$E(x, y) = h \sum_i x_i - \beta \sum_{\{i,j\}} x_i x_j - \eta \sum_i x_i y_i \tag{1}$$

能量越低，降噪过的图像与原图一致的概率越高。

假设x是降噪后的图像矩阵，y是带噪声的图像矩阵。对于二值图像降噪来说，一种简单的方法是先将x初始化为y，然后遍历每个元素，对每个元素分别尝试1和-1两种状态，选择能够得到更低的能量的那个，实际上相当于一种贪心的策略，这种方法称为Iterated Conditional Modes (ICM)。

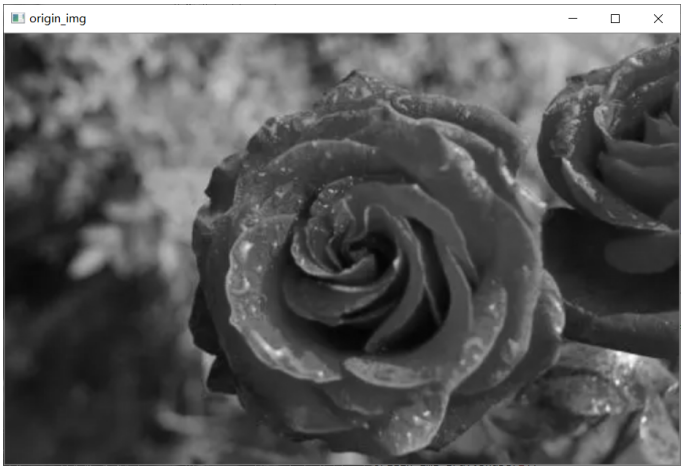
对于本次实验对灰度图像的降噪问题，我们可以将每个像素点的颜色用8bit表示，对应十进制的0~255。和二值图像降噪类似，我们也可以采用相同的能量函数表达式。假设某个像素点的颜色值为071，对应的二进制为01000111，由于我们仍是以颜色翻转的形式添加噪声，故该元素需要分别尝试01000111和10111000两种状态。至于能量函数的计算，则考虑采用二进制对应位相乘并加权后相加的方式，这里用-0.5表示二进制中的0，用0.5表示二进制中的1，从高位到低位分别赋予128，64，32，16，8，4，2，1的权重，这样可以使降噪后的元素与相邻元素颜色相近的概率更大，从而使图像去噪性能更好。

3 实验步骤

3.1 灰度图像获取

本次实验通过对彩色图像进行处理得到灰度图像，彩色图像如下图所示：

通过如下代码可读取灰度图像 ```python origin_img = cv2.imread("img3.png") #读取图片 origin_img_gray = cv2.cvtColor(origin_img, cv2.COLOR_BGR2GRAY) #转换了灰度化 cv2.imshow('origin_img', origin_img_gray) #显示图片 ``` 最终得到的灰度图像如下图所示：



3.2 图像加噪声

本次实验采用像素颜色值反转的方式加噪声，在生成噪声图片时采用了10%的误差率，即整张图片有10%的像素发生了反转。如某像素点颜色值是071，对应的二进制是01000111，则将其转化为噪声点后的颜色值是10111000。对应代码块如下：

```
for i in range(noise_img_bin.shape[0]):
    for j in range(noise_img_bin.shape[1]):
        r = np.random.rand()
        if r < 0.1:
            noise_img_bin[i,j] = '{:08b}'.format(255-int(noise_img_bin[i,j],2))
```

加噪声后的图像如下图所示：

3.3 灰度图像ICM去噪算法

本次实验采用函数$E(x,y)=h\sum_i x_i-\beta\sum_{\{i,j\}}x_ix_j-\eta\sum_i x_iy_i$来计算能量。其中$x$表示的是去噪后图像，$y$表示的是噪声图像。

假设噪声图中某个像素点的颜色二进制值为$a_1a_2a_3a_4a_5a_6a_7a_8$，与其相邻的四个像素点的颜色二进制分别为$b_1b_2b_3b_4b_5b_6b_7b_8$、$c_1c_2c_3c_4c_5c_6c_7c_8$、$d_1d_2d_3d_4d_5d_6d_7d_8$和$e_1e_2e_3e_4e_5e_6e_7e_8$，需要分别尝试用$a_1a_2a_3a_4a_5a_6a_7a_8$和$a_1'a_2'a_3'a_4'a_5'a_6'a_7'a_8'$来计算能量函数的值并比较大小，其中$a_1a_2a_3a_4a_5a_6a_7a_8+a_1'a_2'a_3'a_4'a_5'a_6'a_7'a_8'=11111111$。PRML书中在计算能量函数时将255表示为1，将0表示为-1，相似地，我们将二进制数中的1映射为0.5，将0映射为-0.5，这样只需对二进制数的每位减去0.5即可得到，方便编程。综上，能量函数的第二项为

$$-\beta\sum_{\{i,j\}}x_ix_j=-\beta\Big[\sum_{i=1}^{8}(2^{8-i}(a_ib_i)+\sum_{i=1}^{8}(2^{8-i}a_ic_i)+\sum_{i=1}^{8}(2^{8-i}a_id_i)+\sum_{i=1}^{8}(2^{8-i}a_ie_i)\Big]$$

其中2^{8-i}表示权重，从二进制数的高位到低位分别乘上128，64，32，16，8，4，2，1的权重，这样可以使降噪后的元素与相邻元素颜色相近的概率更大，从而使图像去噪性能更好。

能量函数的第三项为

$$-\eta\sum_i x_iy_i=-\eta\sum_{i=1}^{8}(2^{8-i}a_i^{(')}a_i)$$

其中$a_i^{(')}$表示$a_i^{'}$或a_i两种情况。

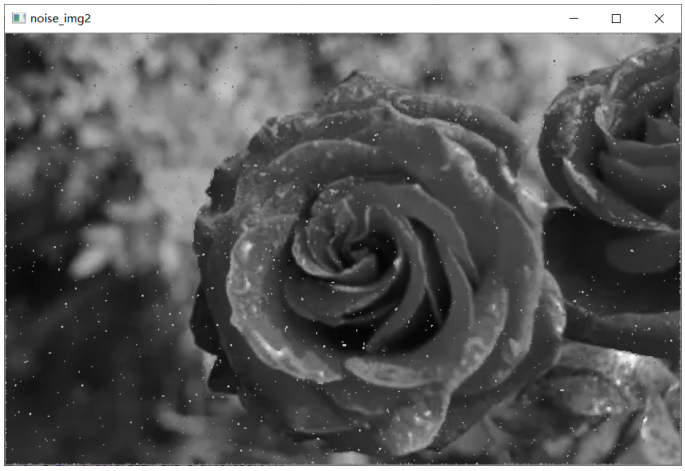
分别考虑$a_1a_2a_3a_4a_5a_6a_7a_8$和$a_1^{'}a_2^{'}a_3^{'}a_4^{'}a_5^{'}a_6^{'}a_7^{'}a_8^{'}$两种情况可以得到能量函数$E_1$和$E_2$。若$E_1 < E_2$，则该处像素降噪后的颜色二进制值为$a_1a_2a_3a_4a_5a_6a_7a_8$，反之则取$a_1^{'}a_2^{'}a_3^{'}a_4^{'}a_5^{'}a_6^{'}a_7^{'}a_8^{'}$。

本次实验设置的参数为$h = 0$，$\beta = 3.5$，$\eta = 0.1$。

4 实验结果

4.1 去噪后图像

使用章节3.3中的去噪算法对章节3.2中的噪声图像进行去噪处理后可以得到如下图所示图像：



4.2 去噪效果分析

本次实验还计算了噪声所占百分比及降噪图像与原图像的相似度，输出结果如下图所示：

```
In [1]: runfile('E:/Study/机器学习/ICM/ICM_Final2.py', wdir='E:/Study/机器学习/
ICM')
加噪声百分比为 0.09911899346442864
降噪图像与原图像的相似度为 0.9801428036737396
```

可以看出噪声所占百分比为9.91%，降噪图像与原图像的相似度为98.01%，降噪效果较为理想。从视觉上看，降噪过后的图像依然存在一些明显的噪点，这是因为ICM采取的类似贪心的策略使得它容易陷入局部最优。

5 实验感悟

通过本次实验，我更深入地理解了马尔科夫随机场与ICM算法，学会了使用上述理论对二值图像和灰度图像进行去噪处理。