**基于马尔可夫随机场的图像去噪**

**实验报告**

1. **问题描述**

给定一张具有噪声的任意二值图，尝试着将其恢复为对应的无噪声原始二值图。

1. **模型描述**

将图片抽象为二维矩阵，矩阵上的每一个点对应着图像上的一个像素点。把我们观测到的具有噪声的图像对应的二维矩阵定义为，猜测的无噪声图像对应的二维矩阵定义为。矩阵上的元素只有两个值：和，对应原图片上的两种不同灰度值。

考虑到去噪后的图像上的像素点应该与实际观测到的图像上的相应像素点具有一定的相关性，我们引进关系：。

考虑到一张图片上，一个像素点应该与其周围的像素点具有一定的相关性，我们引进关系：。

除此之外，我们引入作为一个偏移量来表示图像的倾向性。综上，可以将一张图像的能量值定义为：

我们的目标是在合理的范围内尽量最小化对应的图像的能量值，最后得到的结果便是去噪后的图像。为此我们将采取两种不同的策略:贪心法和模拟退火。

1. **算法描述**
2. **预处理**

我们采用两个分别具有一百张不同图片的图片集来进行实验，这些图片具有个灰度级别。首先将图片进行量化，将大于的灰度值分别映射到 (白色)，将小于等于的灰度值映射到 (黑色)。随后将映射为，将映射为，得到图像对应的二维矩阵。

遍历该矩阵，对于每一个元素有的概率使其值取反，比如变成。完成这个过程后便得到了我们想要的矩阵。

1. **算法一：贪心法**
2. 算法流程

初始化，在每次迭代中，我们遍历上的每个元素，对于某个，将它乘以，固定其它元素不变，计算图像的总能量是否降低，如果降低了，我们则接收这种改变，否则将还原。

1. 能量处理

注意到我们无须每次都重新计算整张图像的能量值，因为的改变只会影响到其中的一部分元素，考虑中的每一项：

对于，取反后它的改变量为;

对于，取反后它的改变量为

对于，情况较为复杂。我们将在上下左右的四个点定义为在周围的点，设这四个点中在图像边界范围内的点的集合为，则取反后能量的改变量为

。

但由于在计算的能量值时，改变的影响会被再考虑一次，这一部分对总能量值的影响也为。总的影响为

。

由此，将某元素取反后图像总能量的改变值为：

1. 算法分析

采用该方法后，除了第一次计算图像的总能量外，以后每次计算图像能量仅需的时间复杂度，可大大降低计算能量值所需要的时间。

一开始我们设置迭代停止的条件为经过一轮迭代后图像的能量值不再改变，但实际运行中发现图像的能量会不停降低，如果迭代次数过多反而会造成反效果。实践后得出结论，只进行一轮迭代可以取得时间和去噪效果的最佳平衡。对于我们所采用的两个具有不同的一百张图片的图片集，平均可以去掉和的噪声----即图像可还原为原始图片的和。在有不错去噪效果的前提下，可以在十秒内完成一百张图片的去噪。

1. **算法二：模拟退火**
2. 算法流程：
3. 初始化

初始化，设置初始温度、终止温度、降低速率和随机数产生器。

1. 计算状态能量值

最底层循环每次遍历矩阵中每个元素，首先计算当前能量值，再将当前乘以-1，计算新的能量值以及能量差。

1. 判断是否接受变化

若能量降低，则接受该变化；若能量升高，则利用随机数产生接受概率，使用判别公式判断是否接受该变化，不接受时将还原。

1. 判断终止

循环结束时，初始温度乘以获得新的温度值，判断当该温度值低于终止温度时，循环终止。

1. 能量评估函数

能量评估函数分为两个，基于马尔可夫随机场**（Markov random fields）**，设置好三个参数，公式为：

1. 局部能量函数

该函数为运算过程中计算局部改变能量值，具体操作是对点计算能量，再加上对周围四个方向的点计算能量，通过递归实现。因为注意到，每次改变时，有能量公式第二部分可知，矩阵中其他点的能量值会因为改变而改变的只有其四个方向的点，故添加四部分能量值，每次计算能量范围如下图。这样运算过程中可以大大减少没必要的遍历开销。

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

1. 全局能量函数

第二个为计算整个矩阵的能量值，即每次循环后维护更新当前全局最优能量值。

1. **实验结果及分析**

经过参数训练后，选择使得平均复原率最高的一组参数**作**为最终结果，并对这一组数据进行分析。

下面是按照复原程度分别从最差，中等以及最好三个等级中选取的4张图片进行的对比分析：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **去噪效果** | **去噪前** | **去噪后** | **复原率** | **去噪率** |
| **最差** | **C:\Users\xb\Desktop\worst\noise\3.png** | **C:\Users\xb\Desktop\worst\denoise\3_91.4175.png** | **91.4175%** | **14.175%** |
| **C:\Users\xb\Desktop\worst\noise\4.png** | **C:\Users\xb\Desktop\worst\denoise\4_91.8935.png** | **91.8935%** | **18.935%** |
| **C:\Users\xb\Desktop\worst\noise\1.png** | **C:\Users\xb\Desktop\worst\denoise\1_91.9393.png** | **91.9393%** | **19.393%** |
| **C:\Users\xb\Desktop\worst\noise\2.png** | **C:\Users\xb\Desktop\worst\denoise\2_92.4744.png** | **92.4744%** | **24.744%** |
| **中等** | **C:\Users\xb\Desktop\average\noise\2.png** | **C:\Users\xb\Desktop\average\denoise\2_96.3613.png** | **96.3613%** | **63.613%** |
| **C:\Users\xb\Desktop\average\noise\1.png** | **C:\Users\xb\Desktop\average\denoise\1_96.4935.png** | **96.4935%** | **64.935%** |
| **C:\Users\xb\Desktop\average\noise\4.png** | **C:\Users\xb\Desktop\average\denoise\4_97.0194.png** | **97.0194%** | **70.194%** |
| **C:\Users\xb\Desktop\average\noise\3.png** | **C:\Users\xb\Desktop\average\denoise\3_97.7946.png** | **97.7946%** | **77.946%** |
| **最好** | **C:\Users\xb\Desktop\best\noise\2.png** | **C:\Users\xb\Desktop\best\denoise\2_99.1211.png** | **99.1211%** | **91.211%** |
| **C:\Users\xb\Desktop\best\noise\3.png** | **C:\Users\xb\Desktop\best\denoise\3_99.2218.png** | **99.2218%** | **92.218%** |
| **C:\Users\xb\Desktop\best\noise\4.png** | **C:\Users\xb\Desktop\best\denoise\4_99.3022.png** | **99.3022%** | **93.022%** |
| **C:\Users\xb\Desktop\best\noise\1.png** | **C:\Users\xb\Desktop\best\denoise\1_99.7223.png** | **99.7223%** | **97.223%** |

对比分析可以得出如下结论：

图片的复原率，去噪率和图片的复杂程度基本上是负相关的。

当图片比较简单，没有什么细节的时候图片的复原率可以达到以上，随着图片逐渐变复杂，细节部分越来越多，图片的复原率会逐渐下降到左右，此时去噪率仅为。最终使用最佳参数对两个分别拥有一百张不同图片的图片集进行去噪的平均复原率为和

图片的复杂程度指的是图片中黑白像素点的集中程度，图片的细节部分所占比重，共同决定的。

出现这种结果是因为我们所采用的去噪算法有关，在算法中我们使用如下公式：

来定义一张图片的能量。其中 用到了一个点与其周围的像素点的关系，体现在图片上就是如果一个像素点周围大都是一样的颜色，那么这一个像素点就很可能是一个噪点，不过前提是这个点在原始图片中不是一个边缘点。

而一个像素点为边缘点的可能性正是与图片的复杂程度成正相关的，所以，当一张图片越复杂时，图片中的细节部分会越多，从而导致图片的边缘地带越多，即边缘点越多，于是去噪时一个像素点为边缘点的可能性会大大增加。即使通过训练参数只能在一定程度上削弱这种干扰，并不能完全抵消。