**任课教师：黄文清**

**《智能信息处理及应用》**

**（ 2020 -- 2021 学年 第一学期）**

**实**

**验**

**报**

**告**

**学号：2018329621262**

**2018329621277**

**姓名：刘恒玮**

**朱若晨**

**班级：智能科学与技术（2）班**

**实验：基于模糊C均值聚类算法的图像分割**

**一、实验目的**

1. 掌握模糊C聚类算法的基本原理。
2. 了解图像分割的目的和基本要求。
3. 能将模糊C聚类算法应用于图像分割中，并编程实现。
4. 复习python语言，能熟练使用python语言编写程序。

**二、实验环境与设备**

本次实验以小组形式进行。

实验环境：python语言。

实验设备：win10 Intel i7 2.21GHz。

**三、****实验要求**

1. 理解并编写模糊*C*均值算法.
2. 基本要求**（必做）**：以灰度图像中各像素的灰度值为聚类时的特征数据，对图像按灰度进行分割。
3. 提高要求1**（选做）**：以灰度图像中各像素的灰度值和像素坐标为聚类时的特征数据，对图像按灰度和位置进行分割。
4. 提高要求2**（选做）**：以彩色图像中各像素的RGB三色值为聚类时的特征数据，对图像按颜色进行分割。
5. 提高要求3**（选做）**：以彩色图像中各像素的RGB三色值以及像素坐标为聚类时的特征数据，对图像按颜色和位置进行分割。

**四、实验原理**

1）模糊C均值算法的基本原理和流程分析

**模糊原理**：模糊控制是自动化控制领域的一项经典方法。其原理则是模糊数学、模糊逻辑。1965，L. A. Zadeh发表模糊集合“Fuzzy Sets”的论文， 首次引入隶属度函数的概念，打破了经典数学“非0即 1”的局限性，用[0,1]之间的实数来描述中间状态。很多经典的集合（即：论域U内的某个元素是否属于集合A，可以用一个数值来表示。在经典集合中，要么0，要么1）不能描述很多事物的属性，需要用模糊性词语来判断。模糊数学和模糊逻辑把只取1或0二值（属于/不属于）的普通集合概念推广0~1区间内的多个取值，即隶属度。用“隶属度”来描述元素和集合之间的关系。

**基本原理**：模糊C均值聚类算法fuzzy c-means algorithm( FCM)，在众多模糊聚类算法中，应用最广泛且较成功。它通过优化目标函数得到每个样本点对所有类中心的隶属度，从而决定样本点的类属以达到自动对样本数据进行分类的目的。然后通过机器学习中提到的相关的距离开始进行相关的聚类操作经过一定的处理之后可以得到相关的cluster，而cluster之间的元素或者是矩阵之间的距离相对较小，从而可以知晓其相关性质与参数较为接近。

模糊c均值聚类融合了模糊理论的精髓。因为大部分情况下，数据集中的对象不能划分成为明显分离的簇，指派一个对象到一个特定的簇有些生硬，也可能会出错。故，对每个对象和每个簇赋予一个权值，指明对象属于该簇的程度。当然，基于概率的方法也可以给出这样的权值，但是有时候我们很难确定一个合适的统计模型，因此使用具有自然地、非概率特性的模糊c均值就是一个比较好的选择。抛开复杂的算式，这个算法的意思就是：给每个样本赋予属于每个簇的隶属度函数。通过隶属度值大小来将样本归类。

算法流程：

 1)  设置目标函数的精度e，模糊指数m（m通常取2）和算法最大迭代次数；

2)  初始化隶属度矩阵或聚类中心；

3)  由式(2)(3)更新模糊划分矩阵和聚类中心；

4)  若目标函数则迭代结束；否则，跳转执行第三步；

5)  根据所得到的隶属度矩阵，取样本隶属度最大值所对应类作为样本聚类的结果，聚类结束。

2）图像分割结果的评价指标

  图像分割就是把图像细分为构成它的对象或子区域，这些区域是互不相交的，每个区域都满足特定区域的一致性。分割的程度主要取决于人们想要解决的问题，当感兴趣的区域或对象已经被区分出来，分割就算完成。

目前，图像分割算法一般是围绕亮度值的两个基本特性设计的：不连续性和相似性。亮度值的不连续性的应用途径主要是基于像素点特性（如灰度值）的不连续变化分割图像，如最常用的边缘检测。而利用亮度值的相似性可以形成一套机制，即依据事先指定的准则将图像分割为相似的区域。一些实例包括门限处理、区域分离、区域生长和聚类等。而采用模糊C均值聚类及其扩展算法进行图像分割的好处是避免了阈值的设定问题，聚类的过程不需要人工干预，只需输入预想的分类数目即可实现自动化的图像分割。

1. **实验内容**
2. **新建一个FCM类，初始化隶属度矩阵**

def Initial\_U(self, sample\_num, cluster\_n): # 初始化隶属度矩阵U

        U = np.random.rand(sample\_num, cluster\_n)  # sample\_num为样本个数即像素点的个数, cluster\_n为分类数

        row\_sum = np.sum(U, axis=1)  # 按行求和 row\_sum: sample\_num\*1

        row\_sum = 1 / row\_sum    # 该矩阵每个数取倒数

        U = np.multiply(U.T, row\_sum)  # 确保U的每列和为1

        return U   # cluster\_n\*sample\_num

1. **类中心矩阵的迭代**

def Cen\_Iter(self, data, U, cluster\_n):# 计算类中心

        c\_new = np.empty(shape=[0, self.dim])  # self.dim为样本矩阵的最后一维度

        for i in range(0, cluster\_n):          # 灰度图片像素值的dim为1，如散点的dim为2，彩色图片的dim为3 加上坐标后则dim+2

            u\_ij\_m = U[i, :] \*\* m  # (sample\_num,)

            sum\_u = np.sum(u\_ij\_m)

            ux = np.dot(u\_ij\_m, data)

            ux = np.reshape(ux, (1, self.dim))  # (1,dim)

            c\_new = np.append(c\_new, ux / sum\_u, axis=0)   # 按列的方向添加类中心到类中心矩阵

        return c\_new  # cluster\_num\*dim

1. **迭代隶属度的矩阵**

def U\_Iter(self, U, c):# 隶属度矩阵迭代

        for i in range(0, self.cnum):

            for j in range(0, self.sample\_num):

                sum = 0

                for k in range(0, self.cnum):

                    temp = (np.linalg.norm(self.data[j, :] - c[i, :]) /np.linalg.norm(self.data[j, :] - c[k, :])) \*\* (2 / (m - 1))

                    sum = temp + sum

                U[i, j] = 1 / sum

        return U

1. **进行迭代**

for i in range(0, iter\_num): # 迭代次数默认为10

            C = self.Cen\_Iter(self.data, U, self.cnum)

            U = self.U\_Iter(U, C)

            print("第%d次迭代" %(i+1) ,end="")

            print("聚类中心\n",C)

            J = self.J\_calcu(self.data, U, C)  # 计算目标函数

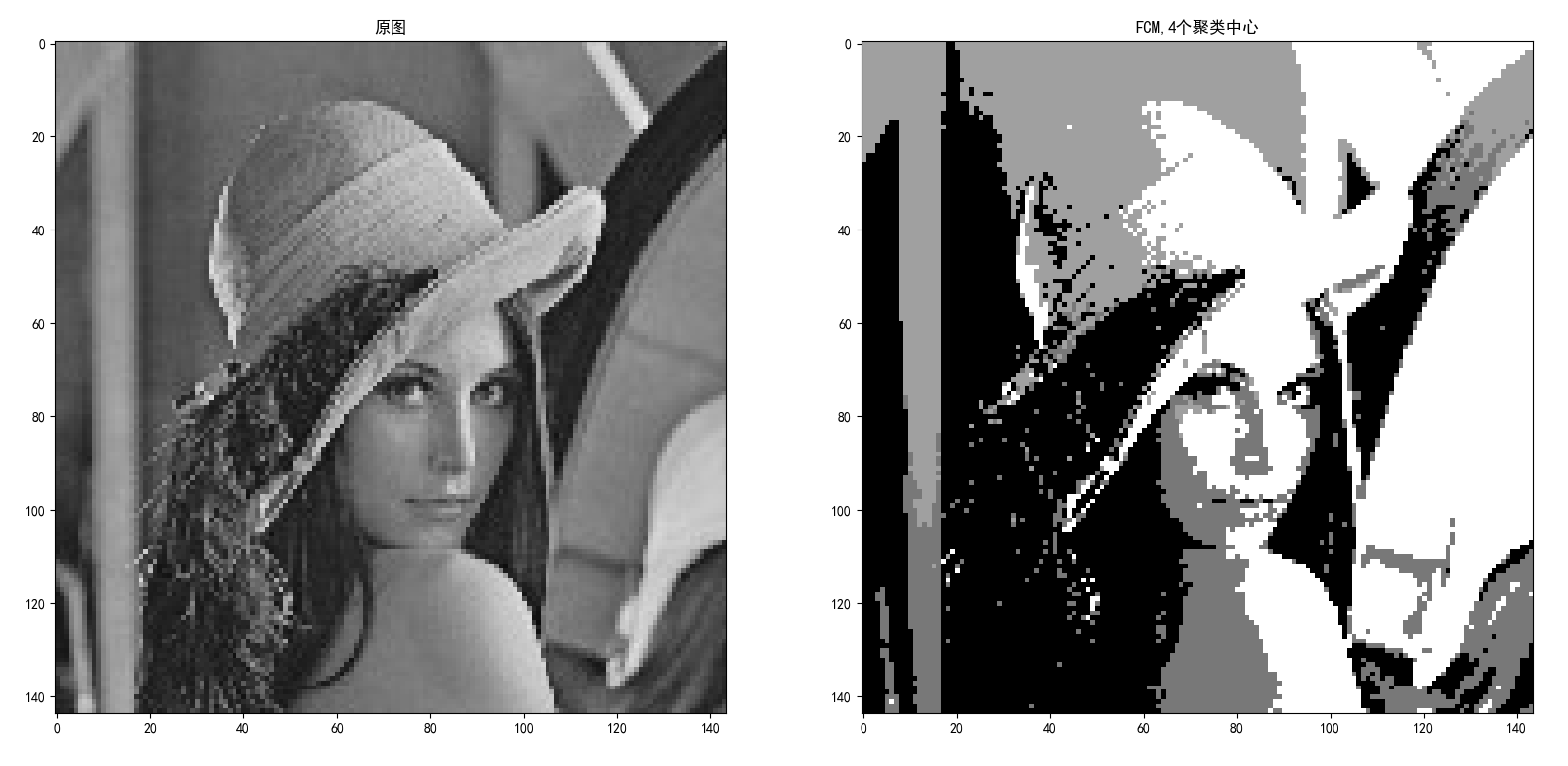
        self.label = np.argmax(U, axis=0)  # 所有样本的分类标签

        self.Clast = C    # 最后的类中心矩阵

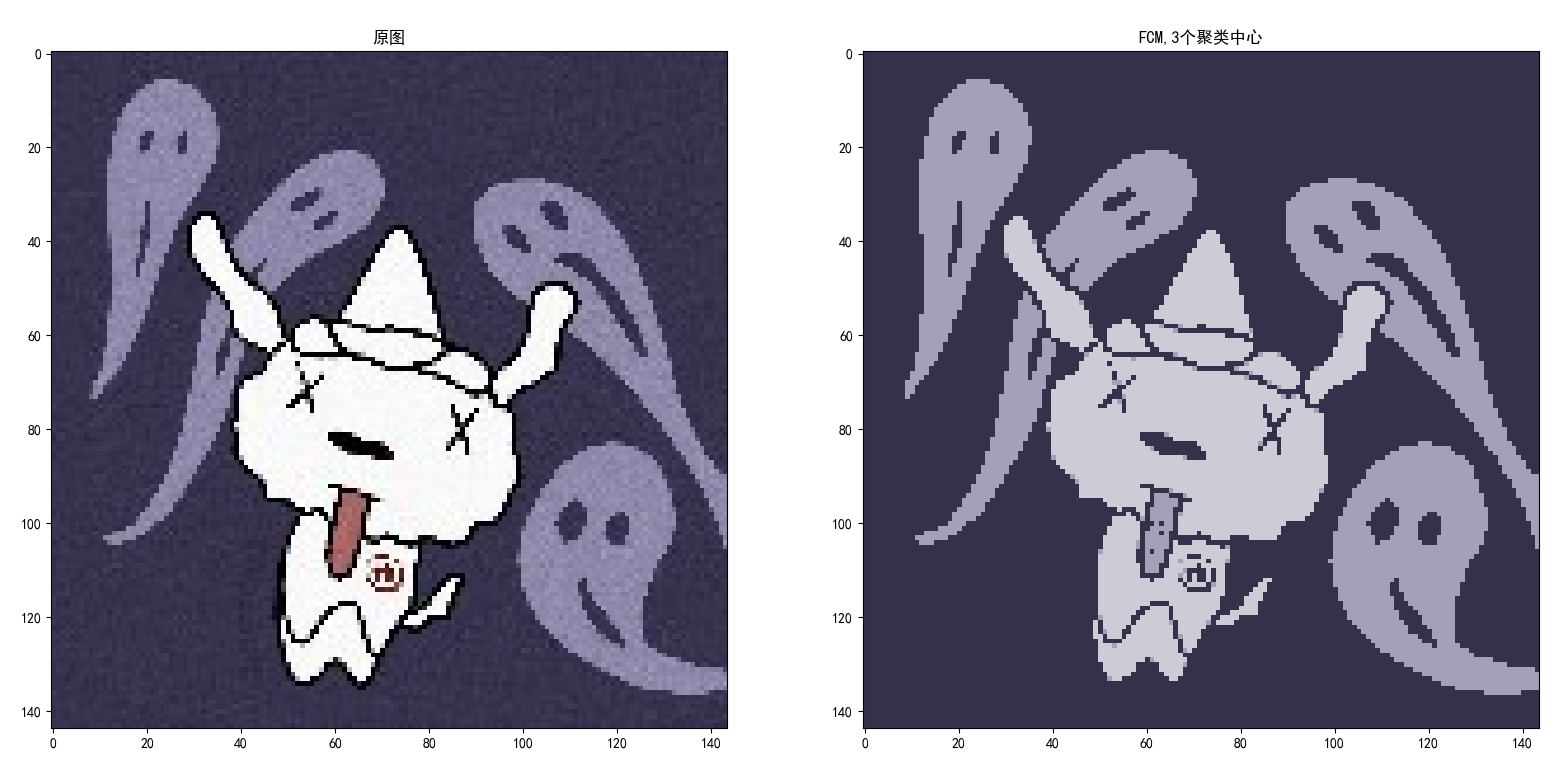
**六、实验结果**

**实验参数设置：**设置隶属度因子m=2,默认迭代次数iter\_num，分类数clust\_num，初始化隶属度矩阵U，计算目标函数值J。

**实验结果截图：**



图一：灰度图像以灰度值和坐标聚类结果



图二：彩色图像以RGB值和坐标聚类结果

**实验讨论：**不同初始化分会对聚类的迭代次数产生影响，我们的初始化分是随机产生的，并没有对其进行优化。关于将一张图片具体分为几类，我们主要是根据实验中选择的图片的颜色的丰富程度，手动调节的，在颜色较为丰富时，一定得将图片分为多类，才能达到理想的积累效果。我们在实验过程中，对比了没有加入位置信息和加入位置信息的聚类结果，发现加入位置信息的聚类结果更加理想，不会出现将颜色相近的背景分为两类的情况。

**七、实验总结**

通过本次实验，我理解了C均值聚类算法的具体过程以及如何运用C均值聚类算法去分割图像的问题。本次实验也不是一帆风顺的，实验过程中出现了一些问题，通过思考、查阅资料和上机实践，有些问题得以解决，有些问题还存有疑问。

**问题1**：实验中若实验选择一般大小的图片，则程序需要运行较长时间。

**解决方案**：先对图像进行压缩，尽量控制在200\*200以内测试代码聚类效果，也可以手动收入迭代次数以达到控制。

**问题2**：如何将位置信息加入同RGB三色值一起聚类。

**解决方案**：先将读取的img 先转为list 然后再将位置信息加入，加入之后在转回数组，聚类结束之后再将后两列的位置信息删除，再重新输出为新的图像即可。

**参考资料：**

1. 《基于模糊C均值聚类算法的图像分割》实验报告指导书
2. 《智能信息处理技术原理与应用》
3. 《人工智能》（第3版）