



黄冈师范学院
HUANGGANG NORMAL UNIVERSITY

人工智能与机器学习

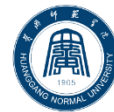
Artificial Intelligence and Machine Learning

章节：实验4-基于KMeans的图像压缩

教师：刘重

学院：计算机学院

厚德 博学 力行 致远



一、实验目的

- 1) 掌握KMeans算法的原理
- 2) 了解KMeans算法实现图片压缩的原理与过程
- 3) 掌握图片的打开和保存方法

- (1) 收集数据：一景jpg图片；
- (2) 准备数据：将三通道（RGB）图像（行，列）转为一个（行 \times 列，3）的数值型矩阵；
- (3) 测试算法：通过Kmeans聚类算法，指定k个聚类中心，那么k个聚类中心就构成了压缩的调色盘。由聚类中心的颜色来替换一张图片中一系列类似的颜色，以压缩代表颜色的比特数，通过图像重建达到图像压缩的目的。

三、实验原理

• K-Means算法

优点:

- 鲁棒性高;
- 速度快、易于理解、效率高;
- 计算成本低、灵活性高;
- 如果数据集是不同的, 则结果更好;
- 可以产生更紧密的簇;
- 重新计算质心时, 簇会发生变化。

缺点:

- 需要预先指定簇的数量;
- 如果有两个高度重叠的数据, 那么它就不能被区分, 也不能判断有两个簇;
- 欧几里德距离可以不平等的权重因素, 限制了能处理的数据变量的类型;
- 有时随机选择质心并不能带来理想的结果;
- 无法处理异常值和噪声数据;
- 不适用于非线性数据集;
- 对特征尺度敏感;
- 如果遇到非常大的数据集, 那么计算机可能会崩溃。

输入: 样本集 $D = \{x_1, x_2, x_3, \dots, x_m\}$; 聚类簇数 k 。

过程:

- 1: 从 D 中随机选择 k 个样本作为初始均值向量 $\{\mu_1, \mu_2, \mu_3, \dots, \mu_k\}$
 - 2: **repeat**
 - 3: 令 $C_i = \emptyset (1 \leq i \leq k)$
 - 4: **for** $j=1, 2, \dots, m$ **do**
 - 5: 计算样本 x_j 与各均值向量 $\mu_i (1 \leq i \leq k)$ 的距离: $d_{ji} = \|x_j - \mu_i\|_2$;
 - 6: 根据距离最近的均值向量确定 x_j 的簇标记: $\lambda_j = \operatorname{argmin}_{i \in \{1, 2, 3, \dots, k\}} d_{ji}$;
 - 7: 将样本 x_j 划入相应的簇: $C_{\lambda_j} = C_{\lambda_j} \cup \{x_j\}$;
 - 8: **end for**
 - 9: **for** $i=1, 2, \dots, k$ **do**
 - 10: 计算新均值向量: $\mu'_i = \frac{1}{|C_i|} \sum_{x \in C_i} x$;
 - 11: **if** $\mu'_i \neq \mu_i$ **then**
 - 12: 将当前均值向量 μ_i 更新为 μ'_i
 - 13: **else**
 - 14: 保持当前均值不变
 - 15: **end if**
 - 16: **end for**
 - 17: **until** 当前均值向量均未更新
- 输出: 簇划分 $C = \{C_1, C_2, \dots, C_k\}$

1.收集数据：一景jpg图片

- 黑白图像（单色图像、二值图像）
- 由黑色区域和白色区域组成。每个像素点只占一位，它的值只有0（黑）或1（白）。

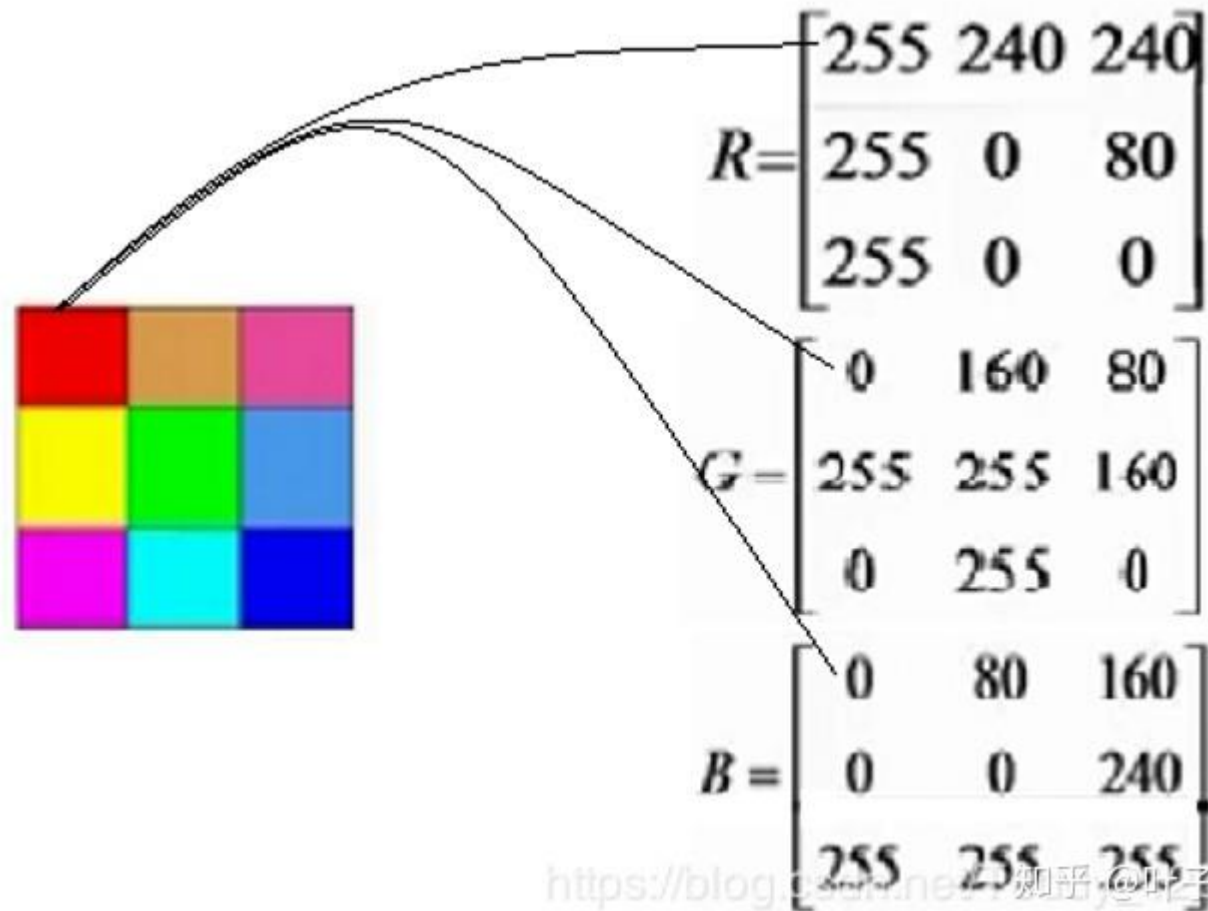


24位真彩色图像

- （1）每一个像素由RGB三个分量组成。
- （2）每个分量各占8位，其范围为0~255，每个像素24位。

1.收集数据：一景jpg图片

- 24位真彩色图像
- (1) 每一个像素由RGB三个分量组成。
- (2) 每个分量各占8位，其范围为0~255，每个像素24位。



2.准备数据:

- 图像中每个像素就相当于我们需要聚类的点，而这些点在R/G/B三个维度的取值就决定了其在特征空间的位置，换句话说，R/G/B三个色度空间上的取值即为聚类特征值。
- 数字图像是由 $M \times N$ 个像素构成的矩阵，所有像素即为待聚类的一个对象，因此，在数据处理的时候需要进行图像“扁平化”，将三通道（RGB）图像（行，列）转为一个（行 \times 列，3）的数值型矩阵；
- 因为24位真彩图像，在R/G/B通道上取值都为8位，取值范围为0~255，因此数据需要做归一化，原始值除以255。

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.utils import shuffle
from PIL import Image
import cv2 as cv

#读入图片并扁平化
original_img = np.array(Image.open('tree.jpg'), dtype = np.float64) / 255
original_dimensions = tuple(original_img.shape)
width,height,depth = tuple(original_img.shape)
image_flattened = np.reshape(original_img,(width*height,depth))
```


3.测试算法：使用KMmeans算法对像素进行聚类，以聚类中心为压缩调色盘重建图像。

- 通过Kmeans聚类算法，指定k 个（Demo中 K =64）聚类中心，
- 以k个聚类中心构成压缩的调色盘。
- 由聚类中心的颜色来替换一张图片中一系列类似的颜色，以压缩代表颜色的比特数，通过图像重建达到图像压缩的目的。

```
#使用kmeans从1000个随机选取的颜色样本中创建64个聚类，每个聚类都将称为压缩调色盘中的一个颜色
image_array_sample = shuffle(image_flattened,random_state=0)[:1000]
estimator = KMeans(algorithm='auto',copy_x= True, init='k-means++',max_iter=300,
                    n_clusters=64,random_state=0, tol = 0.0001, verbose= 0)
estimator.fit(image_array_sample)
#KMeans(algorithm='auto',copy_x= True, init='k-means++',max_iter=300,n_cluster=64,
# n_jobs=1, precompute_distances = 'auto',random_state=0, tol = 0.0001, verbose= 0)

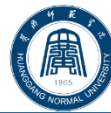
#为原始图像中每个像素分配聚类
cluster_assignments = estimator.predict(image_flattened)

#压缩调色盘和聚类分配来创建压缩图片
compressed_palette = estimator.cluster_centers_
compressed_img = np.zeros((width, height,compressed_palette.shape[1]))
label_idx = 0

for i in range(width):
    for j in range(height):
        compressed_img[i][j] = compressed_palette[cluster_assignments[label_idx]]
        label_idx +=1

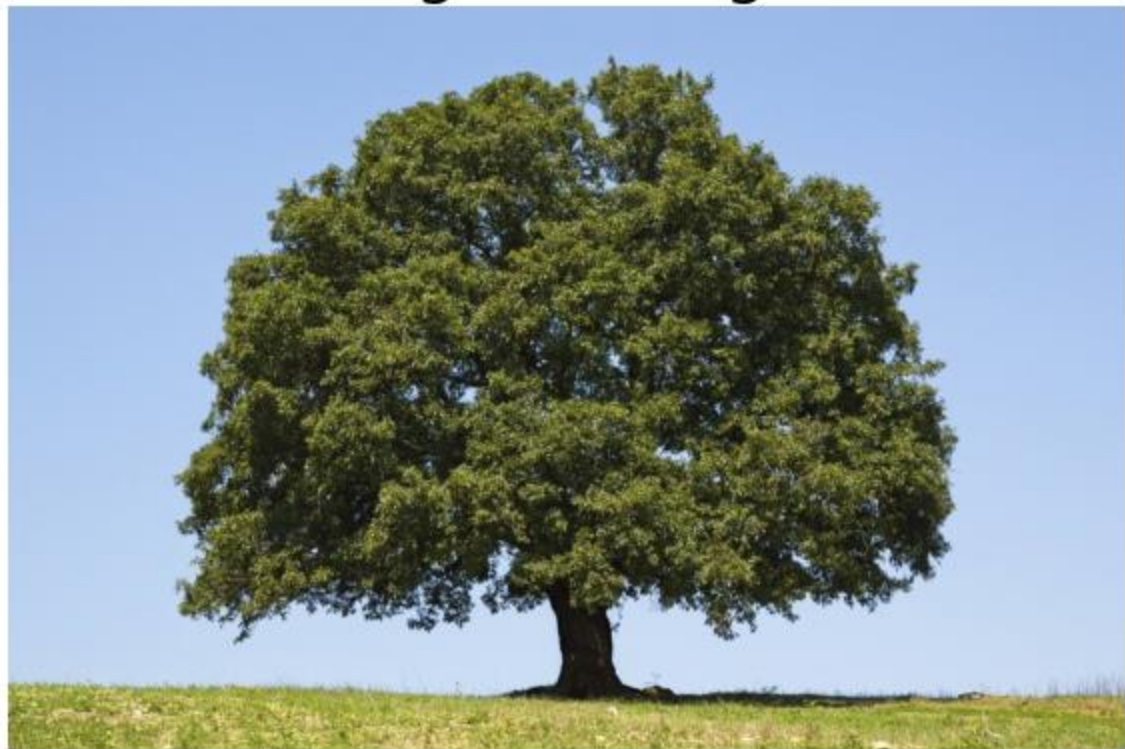
plt.subplot(121)
plt.title('original Image',  fontsize = 24)
plt.imshow(original_img)
plt.axis('off')
plt.subplot(122)
plt.title('compressed Image',  fontsize = 24)
plt.imshow(compressed_img)
plt.axis('off')
plt.show()
print(compressed_img.shape)
cv.imwrite('./tree_compressed.jpg',compressed_img)
```

五、实验结果

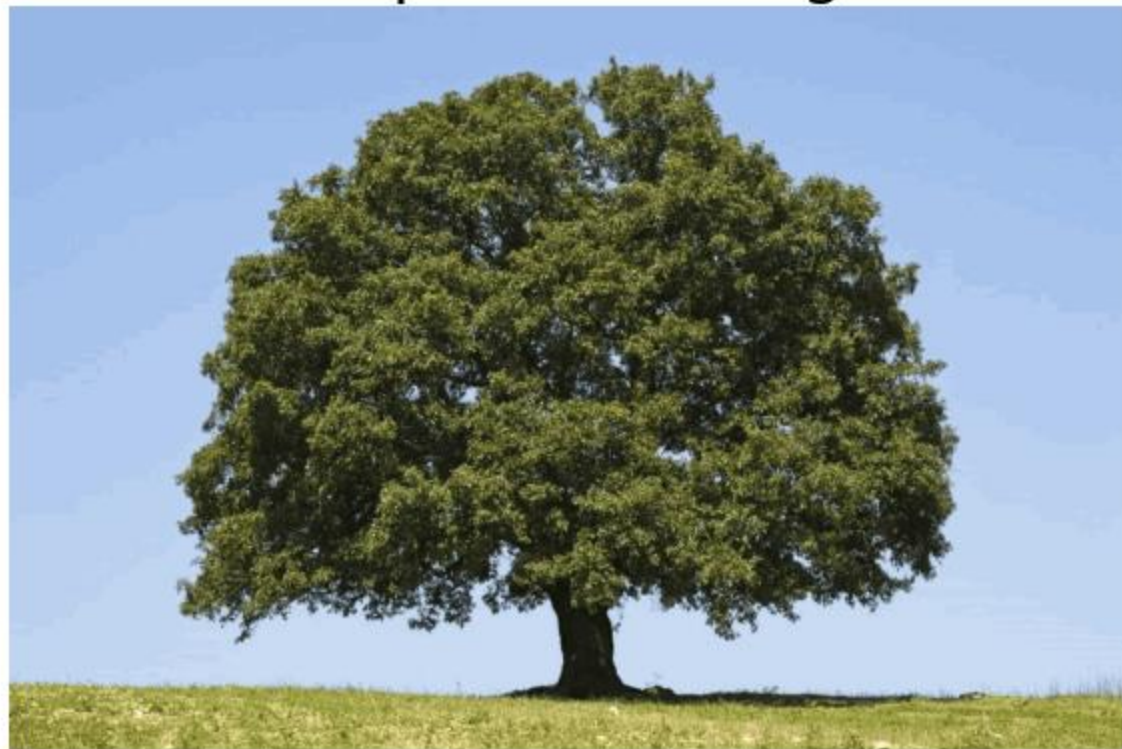


- 可视化压缩前后的图像:

original Image



compressed Image

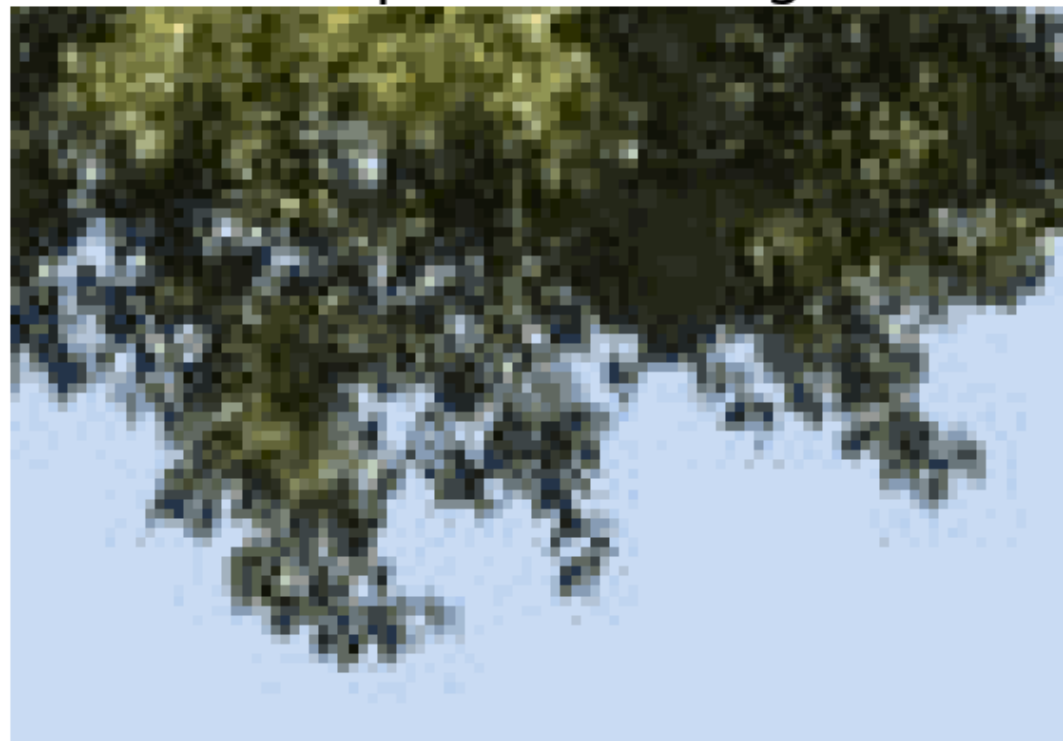


- 压缩效果:



original Image



compressed Image



- 压缩效果:

 tree.jpg	2022/4/27 17:04	JPG 文件	253 KB
 tree_compressed.jpg	2022/4/27 17:22	JPG 文件	20 KB

为什么文件大小变小?

这个大小是否可以无限压缩?

压缩是否有损?

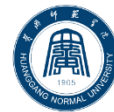
如何在数据信息保持与压缩之间权衡?

请在实验报告的 “5、运行结果与分析” 中回答下列1/2问题

- 1. 请分析 Kmeans算法参数对图像刻画能力的影响;

```
KMeans(algorithm='auto',copy_x= True, init='k-means++',  
        max_iter=300,n_cluster=64,random_state=0, tol = 0.0001, verbose= 0)
```

- 2. 图像压缩程序在本实验中有什么有关? 如何影响?
- 3. 按照模板, 完成实验报告。



六、实验报告要求

- 1、实验目的
 - 2、实验内容
 - 3、实验原理
 - 4、实验代码
 - 5、运行结果与分析
 - 6、实验小结
-
- 说明：每个学生都要交电子版的实验报告，命名格式：
 - 01/02-XXXX（学号）-XXX（姓名）



黄冈师范学院
HUANGGANG NORMAL UNIVERSITY

Q & A

> > > > > > > > > > > > > > > > >

< < < < < < < < < < < < < < < < <