

实验 7 基于 KMeans 聚类算法压缩图像颜色实习指导书

一、实验基本信息

项目	内容
实验名称	基于 KMeans 聚类算法压缩图像颜色
授课课程	大数据分析
上课日期	2025 年 10 月 21 日
实验学时	2 学时（90 分钟）
实验类型	算法应用型实验
实验目标	1. 掌握图像数据预处理操作（读取、数组转换）； 2. 掌握基于 KMeans 聚类算法压缩图像颜色的实操步骤； 3. 培养图像数据处理的细致性与结果验证的严谨性。

二、实验原理

KMeans 聚类算法是无监督学习算法，通过迭代划分簇类和更新簇中心，将数据分为指定数量（`n_clusters`）的簇。在图像颜色压缩中，利用 KMeans 将图像中数百万种颜色聚类为少量代表性颜色（如本实验的 4 种），每个簇的中心颜色替代该簇内所有像素颜色，从而实现颜色压缩，同时保留图像核心视觉特征。

三、实验环境准备

1. 软件要求

1. 操作系统: Windows 10/11 (64 位) ;
2. 软件套件: Anaconda3 (2024.02 及以上版本) ;
3. 开发工具: Spyder (Anaconda 自带, 确保 IPython 控制台可正常运行代码) ;
4. 依赖库:
 1. `scikit-learn` (提供 `KMeans` 聚类模型) ;
 2. `Pillow` (`PIL.Image`, 用于图像读取) ;
 3. `numpy` (数组处理) ;
 4. `matplotlib` (图像展示与保存) 。

2. 预检查步骤

打开 Spyder, 在 IPython 控制台输入以下代码, 验证库是否正常导入:

```
import numpy as np
import sklearn
from sklearn.cluster import KMeans
from PIL import Image
import matplotlib.pyplot as plt
print("NumPy 版本: ", np.__version__)
print("Scikit-learn 版本: ", sklearn.__version__)
print("Pillow 版本: ", Image.__version__)
print("库导入成功! ")
```

若提示 `ModuleNotFoundError`, 打开 Anaconda Prompt 执行:

```
conda install pillow scikit-learn matplotlib
```

3. 实验数据准备

将待压缩的图像文件 (如 “颜色压缩测试图像.jpg”) 放入 Spyder 当前工作目录 (可通过 “File → Working directory” 确认路径) 。

四、实验内容与操作步骤

模块 1：图像数据预处理（25 分钟）

操作目标

完成图像读取与数组格式转换，确保数据符合 KMeans 输入要求。

操作步骤

在 Spyder 中新建文件，命名为 `班级_学号_姓名_KMeansExp7.py`，输入以下代码：

```
import numpy as np
from sklearn.cluster import KMeans
from PIL import Image
import matplotlib.pyplot as plt

# 打开并读取原始图像中像素颜色值，转换为三维数组
imOrigin = Image.open('颜色压缩测试图像.jpg')
dataOrigin = np.array(imOrigin)

# 转换为二维数组，-1 表示自动计算该维度的大小（像素数×3，3 为 RGB 通道）
data = dataOrigin.reshape(-1, 3)

# 细节验证：检查图像数据形状
print("=== 图像数据预处理验证 ===")
print("原始图像三维数组形状：", dataOrigin.shape) # 预期：(高度, 宽度, 3)
print("转换后二维数组形状：", data.shape)          # 预期：(高度×宽度, 3)
```

注意事项

5. 若报 `FileNotFoundError`，检查图像文件名是否为“颜色压缩测试图像.jpg”且在工作目录中；
6. 若图像为灰度图，需先转换为 RGB 格式：`imOrigin = imOrigin.convert('RGB')`。

模块 2：KMeans 聚类模型构建与训练（30 分钟）

操作目标

完成 KMeans 模型训练，将图像像素颜色聚类为 4 类，掌握模型参数设置逻辑。

操作步骤

继续添加代码：

```
# 使用 KMeans 聚类算法把所有像素的颜色值划分为 4 类
kmeansPredictor = KMeans(n_clusters=4)
kmeansPredictor.fit(data)

# 验证聚类结果
print("\n=== 聚类模型训练验证 ===")
print("聚类中心数量：", len(kmeansPredictor.cluster_centers_)) # 预期：4
print("像素聚类标签数量：", len(kmeansPredictor.labels_))      # 预期：与 data 行数一致
```

注意事项

7. `n_clusters` 为聚类数量，本实验固定为 4，后续可尝试调整（如 2、6）并对比压缩效果；
8. 若图像像素数过多（如 4K 图像），可先缩小尺寸：`imOrigin = imOrigin.resize((width//2, height//2))`。

模块 3：图像颜色压缩重建与对比（35 分钟）

操作目标

完成压缩图像的重建与原始图像对比，培养结果可视化与验证的细致性。

操作步骤

继续添加代码：

```
# 使用每个像素所属类的中心值替换该像素的颜色
temp = kmeansPredictor.labels_
dataNew = kmeansPredictor.cluster_centers_[temp]
```

```
# 恢复图像形状
dataNew.shape = dataOrigin.shape
# 展示并保存压缩后的图像
plt.imshow(dataNew.astype(np.uint8)) # 转换为 uint8 类型以正确显示图像
plt.imsave('结果图像.jpg', dataNew.astype(np.uint8))
plt.show()
# 对比原始图像与压缩图像（可选）
plt.figure(figsize=(12, 6))
plt.subplot(1, 2, 1)
plt.imshow(imOrigin)
plt.title('原始图像')
plt.axis('off')
plt.subplot(1, 2, 2)
plt.imshow(dataNew.astype(np.uint8))
plt.title('KMeans 压缩后图像（4 种颜色）')
plt.axis('off')
plt.tight_layout()
plt.show()
```

细节验证

9. 检查“结果图像.jpg”是否生成在工作目录中；
10. 对比原始与压缩图像，确认压缩后仅含 4 种主要颜色，且图像轮廓保留。

五、实验注意事项

1. 图像数据类型

1. 必须转换为 `uint8`：图像像素值范围是 0-255 的整数，KMeans 聚类后的值为浮点数，需通过 `dataNew.astype(np.uint8)` 转换，否则 `plt.imshow` 会因数据类型错误导致图像显示异常（如全黑）。

1. 聚类参数调整

1. `n_clusters` 影响压缩效果：聚类数过少（如 2）会导致图像严重失真，过多（如 20）则压缩效率低。可尝试将 `n_clusters` 改为 2、6 等，对比不同聚类数的压缩结果。

1. 常见错误排查

- 1. “ValueError: input contains NaN”: 确保图像数据无缺失值，可通过 `np.isnan(data).any()` 检查；
- 2. “AttributeError: 'module' object has no attribute 'PILLOW_VERSION’”: Pillow 版本过高时该属性已弃用，可忽略此提示或升级代码为 `print(Image.__version__)`。

六、实验报告要求

1. 报告结构

需包含 “实验目的”“实验原理（KMeans 聚类与图像压缩逻辑）”“实验步骤与结果（附代码截图、原始与压缩图像对比截图）”“问题与解决方法”“实验总结” 5 个部分。

2. 关键截图要求

- 1. 必附截图：图像数据预处理的 `shape` 验证结果、原始与压缩图像对比图、“结果图像.jpg” 实际效果；
- 2. 截图标注规范：注明 “图 1 图像数组形状验证”“图 2 原始与压缩图像对比” 等，包含代码片段与对应输出。

1. “问题与解决方法” 要求

需记录至少 2 个真实问题及解决过程，示例：

- 1. “问题 1：图像显示全黑；解决：忘记将 `dataNew` 转换为 `uint8` 类型，添加 `dataNew.astype(np.uint8)` 后正常显示”；
- 2. “问题 2：聚类训练时间过长；解决：先缩小图像尺寸 `imOrigin.resize((300, 200))`，训练时间从 5 分钟缩短到 10 秒”。

七、实验考核标准

考核维度	考核要点（总分 100 分）	分值
代码正确性	图像预处理、KMeans 训练、图像重建代码与教材完全一致（语法错误扣 5 分 / 处）	40

结果有效性	压缩图像生成成功（4 种颜色，形状与原始一致），对比展示清晰	30
报告质量	结构完整，截图标注规范，“问题与解决”真实详细	20
科研品质	代码注释规范，有参数调试尝试（如改变 <code>n_clusters</code> 对比效果）	10

八、参考资料

[1] Scikit-learn 官方文档. KMeans 模块: <https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html>

[2] Pillow 官方文档。图像处理: <https://pillow.readthedocs.io/en/stable/>

[3] 董付国. Python 数据分析、挖掘与可视化（第二版）[M]. 北京：人民邮电出版社，2024.