**南京邮电大学**

**自动化学院、人工智能学院**

**实 验 报 告**

**课程名称：** 模式识别（双语）A

**实验名称：**  聚类实验

**指导教师：** 谢九成

**所在专业：**  人工智能

**学生姓名：** 单家俊

**班级学号：** B210416 B21080526

**2023 /2024 学年第 一 学期**

**实验名称：聚类实验**

**一、实验目的**

1. 掌握K-means聚类算法
2. 掌握将图像转换为NumPy数组的方法
3. 分析不同K值对分割效果的影响

**二、实验任务**

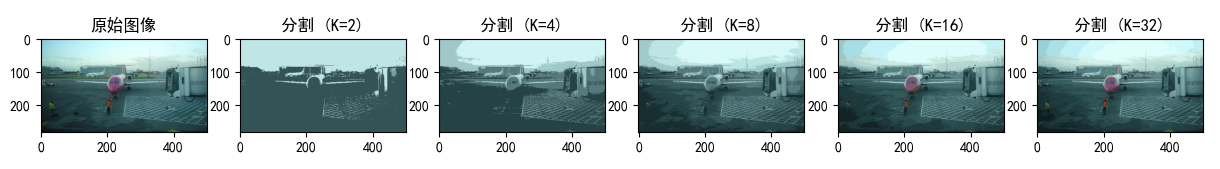
采用K-means聚类算法实现图像的分割，探究不同K值对分割结果的影响。

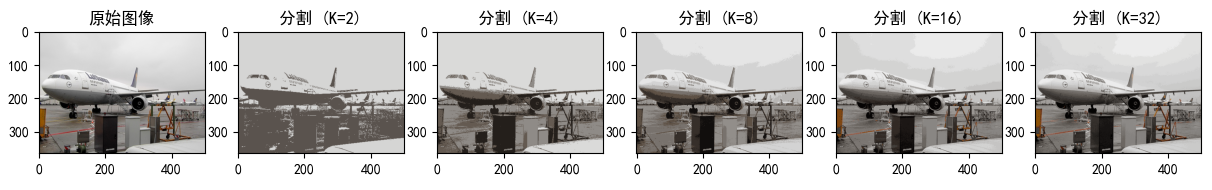
**三、实验内容**

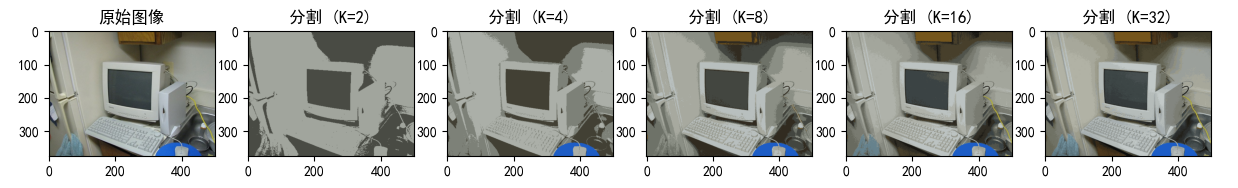
from PIL import Image  
import cv2  
import numpy as np  
import os  
import matplotlib.pyplot as plt  
  
plt.rcParams['font.sans-serif'] = ['SimHei']  
  
def load\_images(directory):  
 images = []  
 for filename in os.listdir(directory):  
 if filename.endswith(".jpg"):  
 img\_path = os.path.join(directory, filename)  
  
 *# 使用PIL库读取图片* img = Image.open(img\_path)  
  
 *# 将图片转换为NumPy数组* img\_array = np.array(img)  
  
 images.append(img\_array)  
 return images  
  
  
def kmeans\_segmentation(image, k):  
 *# 将图像重新形状为像素的2D数组* pixels = image.reshape((-1, 3))  
  
 *# 转换为float32类型* pixels = np.float32(pixels)  
  
 *# 定义标准并应用K均值算法* criteria = (cv2.TERM\_CRITERIA\_EPS + cv2.TERM\_CRITERIA\_MAX\_ITER, 100, 0.2)  
 \_, labels, centers = cv2.kmeans(pixels, k, None, criteria, 10, cv2.KMEANS\_RANDOM\_CENTERS)  
  
 *# 转换回8位值* centers = np.uint8(centers)  
  
 *# 将标签映射到中心* segmented\_image = centers[labels.flatten()]  
  
 *# 将图像重新形状为原始图像的形状* segmented\_image = segmented\_image.reshape(image.shape)  
  
 return segmented\_image  
  
  
def visualize\_segmentation(original, k\_values, segmented\_images):  
 num\_k\_values = len(k\_values)  
  
 *# 创建子图* fig, axes = plt.subplots(1, num\_k\_values + 1, figsize=(15, 5))  
  
 *# 在第一个子图上显示原始图像* axes[0].imshow(original)  
 axes[0].set\_title('原始图像')  
  
 *# 在每个后续子图上显示分割图像和对应的K值* for i in range(num\_k\_values):  
 axes[i + 1].imshow(segmented\_images[i])  
 axes[i + 1].set\_title(f'分割 (K={k\_values[i]})')  
  
 plt.show()  
  
  
def main():  
 image\_directory = r'D:\QQ\1413679561\FileRecv\exp\exp\VOCdevkit\VOC2012\JPEGImages'  
 k\_values = [2, 4, 8, 16, 32] *# 你可以调整K的值进行实验* images = load\_images(image\_directory)  
  
 for image in images:  
 segmented\_images = [kmeans\_segmentation(image, k) for k in k\_values]  
 visualize\_segmentation(image, k\_values, segmented\_images)  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

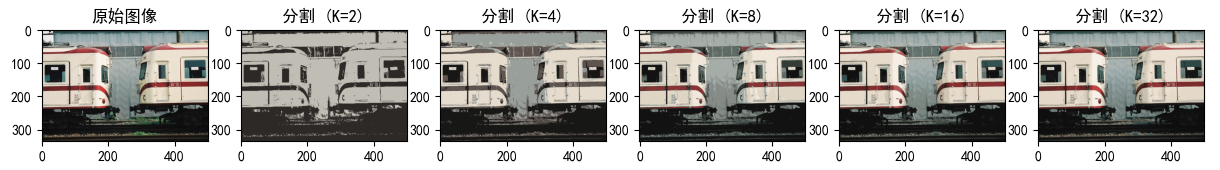
**四、实验过程描述**

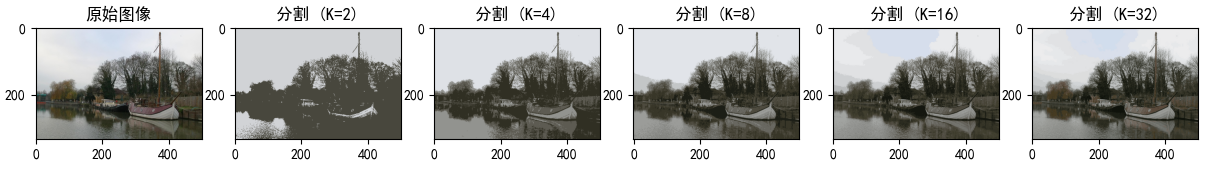
**实验结果分析：**

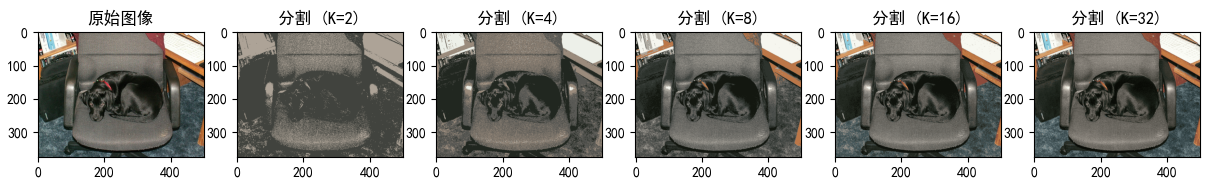
****

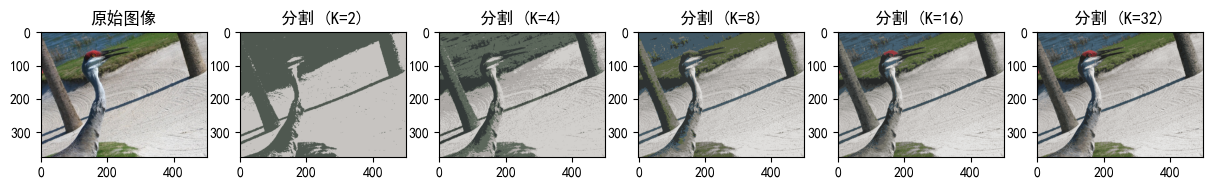
****

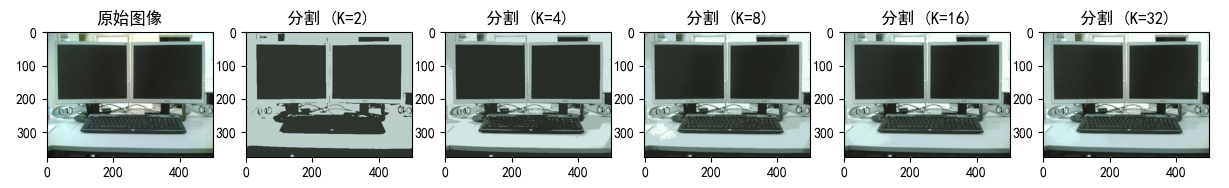
****

****

****

****

****

****

随着K值的增加，算法会尝试更细致地将图像划分为更多的簇。这导致更多的细节被捕捉，使得分割后的图像更清晰、颜色更鲜艳，与原始图像越来越接近。

**实验过程遇到的问题：**

1. 图像路径处理： 需要确保图像路径的正确性，以便成功读取图像。在代码中使用os.path.join函数拼接路径。
2. K-means参数调整： K-means算法有一些参数，如迭代次数、收敛标准等，需要合理调整以获得更好的分割效果。

**体会：**

通过这个实验，我深入了解了K-means聚类算法在图像分割中的应用。调整不同的K值，观察聚类效果，加深了我对于K-means在图像处理中的理解。同时，通过图像可视化，更直观地感受到K值选择对于聚类效果的重要性。这个实验提供了一个实际应用场景，使我更熟悉了图像处理和机器学习算法的结合。