聚类分割实验

班级： 21计科4班 姓名：陈昊天

一、实验原理

本实验通过聚类方法对图像进行分割，研究图像处理中无监督学习的实际应用。实验采用 K-Means 聚类算法对图像像素颜色进行聚类，以达到分割图像中不同区域的目的。K-Means 是一种将数据划分为 K 个簇的经典算法，其目标是最小化簇内数据点到簇中心的平方和。在图像分割中，图像可以看作由像素构成的三维数据，通过将像素点聚类到不同簇中，实现图像颜色区域的划分。

在车牌定位实验中，结合颜色过滤和形状分析，实现图像中车牌区域的自动定位。利用 HSV 色彩空间中蓝色车牌的颜色范围进行掩膜处理，过滤出可能是车牌的蓝色区域；然后通过轮廓检测和形状特征进一步筛选，最终识别出最可能的车牌区域。在提取出的车牌图像中继续应用 K-Means 聚类，实现车牌图像的分割。

二、实验代码与运行结果

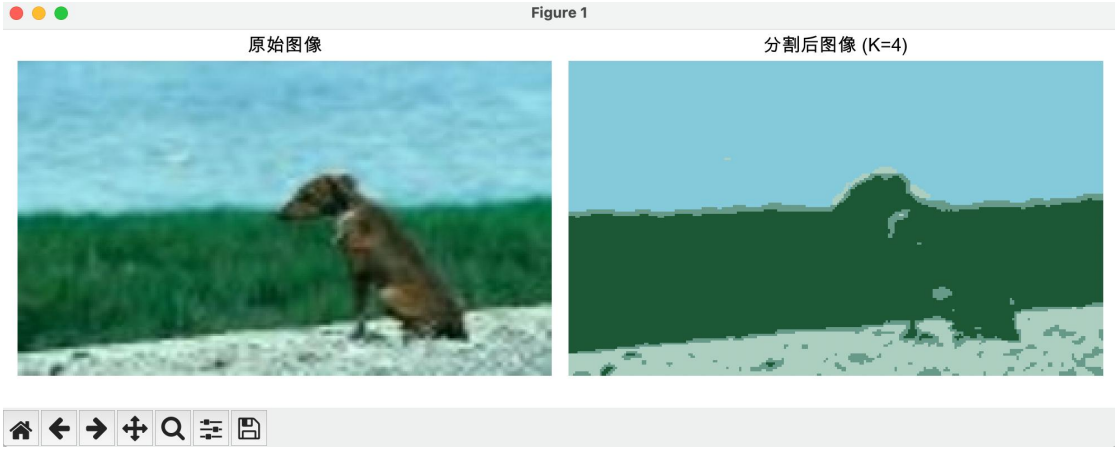
2.1 图像分割的简单应用

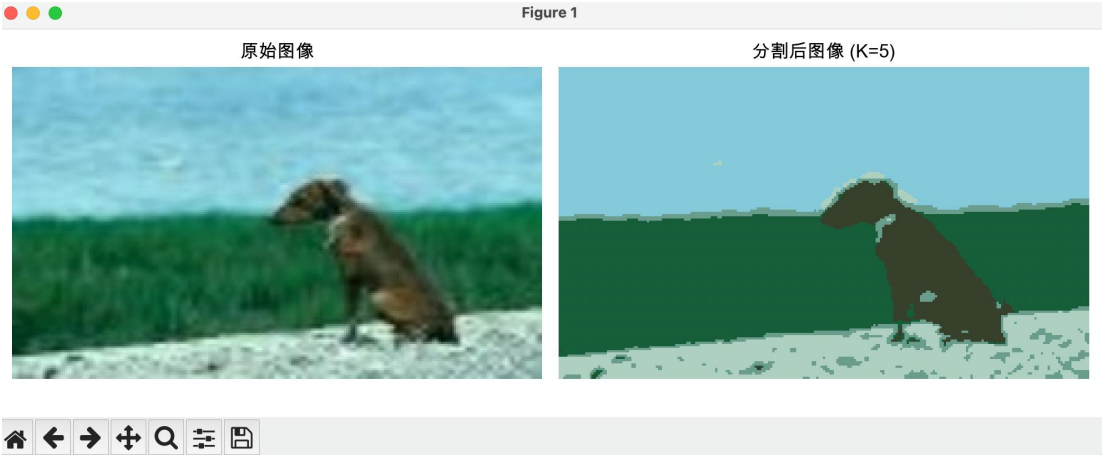
实验代码

```
import cv2
import numpy as np
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt

plt.rcParams["font.sans-serif"] = ["Arial Unicode MS"]
plt.rcParams["axes.unicode_minus"] = False

image_path = "T1.png"
k = 4
img = cv2.imread(image_path)

if img is None:
    print(f"无法加载图像: {image_path}")
else:
    img_rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
    pixel_values = img_rgb.reshape((-1, 3))
    pixel_values = np.float32(pixel_values)
    kmeans = KMeans(n_clusters=k, n_init=20, random_state=50,
```

```
max_iter=300)
    kmeans.fit(pixel_values)
    centers = kmeans.cluster_centers_
    labels = kmeans.labels_
    centers = np.uint8(centers)
    segmented_image_flat = centers[labels.flatten()]
    segmented_image = segmented_image_flat.reshape(img_rgb.shape)
    plt.figure(figsize=(10, 5))
    plt.subplot(1, 2, 1)
    plt.imshow(img_rgb)
    plt.title("Original Image")
    plt.axis("off")
    plt.subplot(1, 2, 2)
    plt.imshow(segmented_image)
    plt.title(f"Segmented Image (K={k})")
    plt.axis("off")
    plt.tight_layout()
    plt.show()
```

实验结果





2.2 车牌分割

实验代码

```
import cv2
import numpy as np
import os


def find_license_plate_and_cluster(image_path, k=2):
    img = cv2.imread(image_path)
    if img is None:
        print(f"无法加载图像: {image_path}")
        return None, None, None

    original_image_with_box = img.copy()
    hsv = cv2.cvtColor(img, cv2.COLOR_BGR2HSV)
    lower_blue = np.array([100, 40, 40])
    upper_blue = np.array([130, 255, 255])
    mask = cv2.inRange(hsv, lower_blue, upper_blue)
    kernel = np.ones((3, 3), np.uint8)
    mask = cv2.morphologyEx(mask, cv2.MORPH_OPEN, kernel, iterations=1)
    contours, _ = cv2.findContours(mask, cv2.RETR_EXTERNAL,
cv2.CHAIN_APPROX_SIMPLE)
    possible_plates = []
    min_area = 500
    max_area = 30000
    min_aspect_ratio = 2.0
    max_aspect_ratio = 5.0
    for contour in contours:
        area = cv2.contourArea(contour)
        if area > min_area and area < max_area:
            x, y, w, h = cv2.boundingRect(contour)
            aspect_ratio = float(w) / h

            if aspect_ratio > min_aspect_ratio and aspect_ratio <
max_aspect_ratio:
                rect_area = w * h
                extent = float(area) / rect_area
```

```
                if extent > 0.6:
                    possible_plates.append((x, y, w, h))

    best_plate_rect = None
    if possible_plates:
        possible_plates.sort(key=lambda p: p[2] * p[3], reverse=True)
        best_plate_rect = possible_plates[0]

    if best_plate_rect is None:
        print("未检测到车牌")
        return original_image_with_box, None, None

    x, y, w, h = best_plate_rect
    padding = 2
    x_start = max(0, x - padding)
    y_start = max(0, y - padding)
    x_end = min(img.shape[1], x + w + padding)
    y_end = min(img.shape[0], y + h + padding)
    license_plate_roi = img[y_start:y_end, x_start:x_end]

    if license_plate_roi.size == 0:
        print("提取的车牌区域为空")
        return original_image_with_box, None, None

    cv2.rectangle(
        original_image_with_box, (x_start, y_start), (x_end, y_end),
(0, 0, 255), 2
    )

    pixel_values = license_plate_roi.reshape((-1, 3))
    pixel_values = np.float32(pixel_values)
    criteria = (cv2.TERM_CRITERIA_EPS + cv2.TERM_CRITERIA_MAX_ITER,
100, 0.2)
    compactness, labels, centers = cv2.kmeans(
        pixel_values, k, None, criteria, 10, cv2.KMEANS_RANDOM_CENTERS
    )
    centers = np.uint8(centers)
    segmented_data = centers[labels.flatten()]
```

```
    segmented_plate = segmented_data.reshape(license_plate_roi.shape)
    color1 = centers[0]
    color2 = centers[1]
    is_blue_white = False
    if (color1[0] > 100 and color1[1] < 150 and color1[2] < 150) or (
        color2[0] > 100 and color2[1] < 150 and color2[2] < 150
    ):
        if (color1[0] > 150 and color1[1] > 150 and color1[2] > 150)
or (
            color2[0] > 150 and color2[1] > 150 and color2[2] > 150
        ):
            is_blue_white = True
    return original_image_with_box, license_plate_roi, segmented_plate


if __name__ == "__main__":
    image_file = "T2.png"
    if not os.path.exists(image_file):
        print(f"文件 '{image_file}' 不存在")
    else:
        original_with_box, plate_roi, segmented_result =
find_license_plate_and_cluster(
            image_file, k=2
        )
        if plate_roi is not None and segmented_result is not None:
            cv2.imshow("Original", original_with_box)
            cv2.imshow("Extracted", plate_roi)
            cv2.imshow("Segmented", segmented_result)
            cv2.waitKey(0)
            cv2.destroyAllWindows()
        elif original_with_box is not None:
            cv2.imshow("Original - Plate not found", original_with_box)
            cv2.waitKey(0)
            cv2.destroyAllWindows()
```

实验结果

Original为识别到车牌的原始图像；Extracted为提取的车牌区域；Segmented为聚类分割后的车牌

三、结果分析与心得体会

对 2.1 图像分割的简单应用的分析

需运行多次才有可能得到较好的效果的原因：

K-Means 算法的执行过程对初始聚类中心的选择非常敏感。由于初始中心是随机选取的，不同的初始值可能会导致算法收敛到不同的结果。K-Means 可能会陷入局部最优解，而不是全局最优解。找到的聚类结果在局部看起来是好的，但不是所有聚类中最好的。为了增加找到全局最优解的可能性，运行多次算法，每次使用不同的随机初始中心，选择所有运行中簇内平方和最低的结果作为最终输出。

k=4时无法达成目标效果（将图片分割为合适的背景区域（3个）和前景区域（小狗）），而k=5时可以：

K=4 时，聚类数量不足以区分图像中所有目标区域的代表性颜色。图像中存在四个颜色差异相对较大的区域（天空、草地、沙地、小狗）。当 K 被强制设为 4 时，K-Means 算法为了最小化簇内距离，会将颜色上相对接近的两个区域合并到同一个簇中，导致无法将小狗作为独立区域分割出来。而 K=5 提供了足够的聚类数量，使得算法能够为这四个主要区域各自指定一个簇。

对 2.2 车牌分割的分析

如何对车牌进行定位：

根据先按颜色找，再按形状筛的思想，利用车牌通常具有的特定颜色蓝色和固定尺寸范围和长宽比例的矩形这两个特征。通过颜色过滤，筛选出图像中所有可能是蓝色车牌的区域。在蓝色区域中，通过分析形状的尺寸、长宽比和规整度，排除掉那些颜色符合但形状不像车牌的区域，最终确定最符合车牌特征的矩形区域。

心得体会

通过本次实验，我了解 K-Means 聚类在图像处理中的应用原理和实践效果，掌握图像分割的基本流程。实验提升我对无监督学习算法的理解，加深对图像数据结构和颜色空间转换的认识。在调试过程中，我体会到参数选择对聚类结果有显著影响，对我今后的模型调优能力有很大帮助。

在车牌定位部分，我感受到传统图像处理技术在实际任务中的重要性。通过图像处理与机器学习算法的结合，较为准确地实现车牌定位与分割，激发我对智能图像分析的兴趣。

这次实验提升我对图像聚类理论的理解，锻炼我从数据处理到模型应用的综合能力，为日后从事计算机视觉或人工智能相关方向的研究和开发打下了良好基础。