

数字图像处理课程设计

****

**题 目** 感兴趣目标提取

学生姓名 张瑞晨

学 号 202283290159

学 院 计算机学院、网络空间安全学院

专 业 计算机科学与技术

指导教师 孙玉宝

**二Ｏ四零年十二月**

目 录

[任务目的与要求 4](#_Toc4506)

[一、图像预处理 5](#_Toc13360)

[1.1.对比度增强 5](#_Toc8100)

[1.2锐化处理 6](#_Toc27987)

[1.3预处理结果比较 7](#_Toc30481)

[二、直方图分析与阈值分割方法尝试 8](#_Toc22691)

[2.1直方图绘制分析和阈值分割 8](#_Toc29179)

[三、超像素分割 9](#_Toc31869)

[3.1 超像素分割方法选取 9](#_Toc26439)

[3.2 超像素分割结果比较 11](#_Toc7721)

[3.3 SLIC超像素分割参数选择 12](#_Toc4402)

[四、超像素特征表示 14](#_Toc14525)

[4.1特征选取 14](#_Toc10522)

[4.2超像素特征聚类 15](#_Toc10294)

[五、感兴趣目标提取 15](#_Toc27324)

[5.1超像素合并 15](#_Toc4678)

[5.2感兴趣目标提取结果 15](#_Toc581)

[六、课程设计总结与体会 18](#_Toc8870)

[参考文献 19](#_Toc17368)

[致谢 20](#_Toc7922)

感兴趣目标提取

张瑞晨

南京信息工程大学计算机与网络安全学院，江苏 南京 210044

摘要：本项目基于超像素分割和特征聚类方法，实现了对图像中感兴趣目标区域的自动提取。首先，使用 SLIC（Simple Linear Iterative Clustering） 算法对输入图像进行初步的超像素分割，将图像划分为多个连贯的像素区域。

接着，为每个超像素区域提取多维特征向量，包括颜色均值、纹理特征（灰度共生矩阵的对比度和同质性）、边缘信息和形状特征，并对这些特征进行加权处理，以提高关键特征的重要性。在特征聚类阶段，采用 KMeans 聚类算法 对超像素特征进行分类，进而实现对图像的语义分割。随后，通过特征相似性计算和阈值控制，进行超像素区域的合并，以减少过度分割的影响，提升目标区域的连贯性。

最终，利用 形态学操作 和 连接组件分析 提取图像中的最大连通区域，生成感兴趣区域（ROI）的二值掩码，并将其应用于原图中。实验结果表明，该方法能够有效地分离图像中的前景目标和背景区域，为图像分割和目标识别任务提供了可靠的技术支持。

关键词：超像素分割、SLIC算法、KMeans聚类、特征相似性、感兴趣区域提取

# 任务目的与要求

主要内容：

对输入的自然图像进行感兴趣目标提取，给出所有10副图的结果（图像教师提供）。

参考步骤：

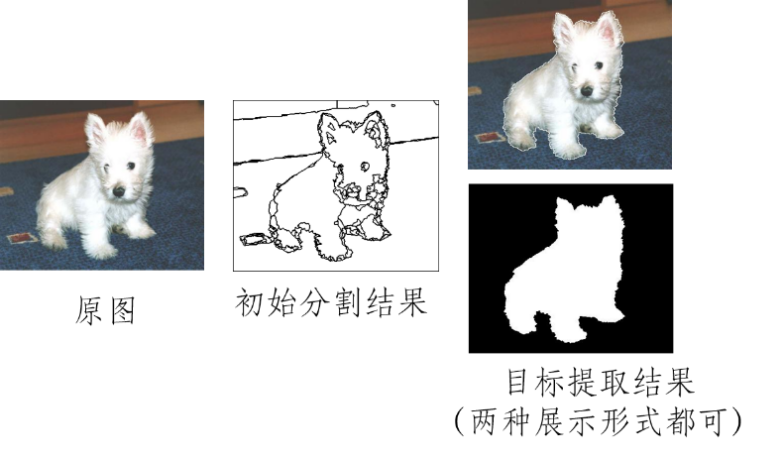
对图像进行初始分割，可选择的初始分割方法包括分水岭方法（Meanshift），SLIC超 像素算法，标准分割N-cut方法等；

对每个分割块（或称超像素）进行特征表示；

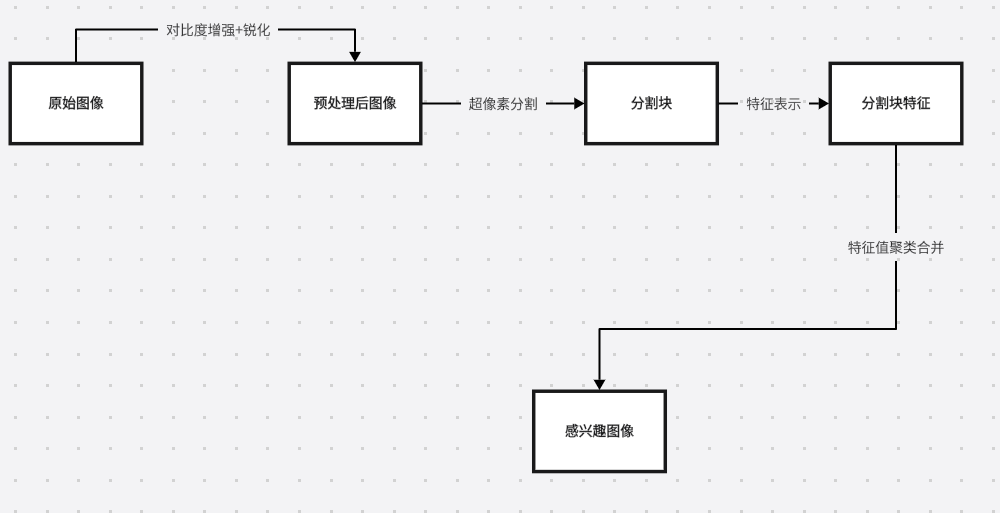
基于特征相似性进行超像素合并，给出感兴趣目标提取结果。

可以设计友好简洁的人机互动界面，设计图像显示框、按钮等，并将原始图像、过分割 结果、目标提取结果图像等，显示在界面中。

参考结果：



系统流程图：



# 一、图像预处理

## 1.1.对比度增强

观察得到图像中存在低对比度的区域，分割算法可能难以区分不同区域时，增强对比度可以帮助提高分割效果。



图 1 对比度较低的图像

使用python pillow库中的ImageEnhance.Contrast 可以调整图像的对比度，使图像中的亮部更亮、暗部更暗，从而增强图像的视觉效果，帮助图像处理算法更好地提取特征。通过增强对比度，可以使边界更加明显，进而提高图像分割的精度。同时在预处理同时，进行不同方法增强的比较，如亮度，颜色等。

enhancer = ImageEnhance.Contrast(image)# 对比度增强

image\_contrast = enhancer.enhance(1.3)





图 2 不同方法增强效果比较

上述图片分别使用对比度，亮度，锐化，颜色等不同角度的增强。不难发现对比度增强应会对后续分割结果有更好的提升。

## 1.2锐化处理

在图像分割之前，为了识别器能够更好得识别图像轮廓。采用锐化将图形边界增强。使用一些方法，有效增强图像的 细节和边缘，而不至于过度模糊或增加噪声。

此处选取双边滤波（Bilateral Filter）、拉普拉斯算子（Laplacian） 和 巴特沃斯高通滤波（Butterworth Highpass Filter）方法进行对比实验。

1. 双边滤波（Bilateral Filter）

双边滤波是一种非线性滤波方法，它能够在平滑图像的同时保留边缘信息。这是通过在空间域和颜色域上同时进行加权平均来实现的。它使用一个 空间权重（通常是一个高斯核）来平滑图像，同时使用一个 颜色权重 来保留边界区域，使得图像的细节得以保持，而不会模糊边缘。

去噪和保持边缘：在图像处理过程中，特别是对于纹理较复杂或噪声较多的图像，双边滤波能够有效去除噪声，同时保留重要的边缘信息，因此它在图像预处理阶段非常有用。

此处选取参数：

邻域半径 d=9，决定了在滤波过程中每个像素影响的邻域范围。

sigmaColor=75 和 sigmaSpace=75：控制颜色和空间的权重，影响平滑效果的程度。

2.拉普拉斯算子（Laplacian）

拉普拉斯算子是一种边缘检测算子，用于突出图像的变化部分，即 高频信息。它通过计算图像像素点的二阶导数来增强图像中的边缘和细节。

这对于图像的边缘检测和细节增强非常有效。

边缘增强：拉普拉斯算子用于从图像中提取边缘信息，对于图像分割、特征提取等任务非常有帮助。通过增强图像的高频部分，能够更清晰地展示图像中的重要结构。

去除低频背景：它帮助去除图像中的低频信息，保留重要的边缘，适用于图像中细节较为模糊或需要突出细节的情况。

3. 巴特沃斯高通滤波（Butterworth Highpass Filter）

巴特沃斯高通滤波是一种在频域中进行的滤波方法，用于增强图像中的 高频成分。它在频域内对图像进行处理，通过滤除低频信息来强调高频部分（例如边缘和纹理）。

巴特沃斯滤波器的特点是其平滑的过渡特性，避免了理想高通滤波器产生的硬边界效应。

增强高频信息：巴特沃斯高通滤波器能有效增强图像中的边缘和细节，突出图像中的高频成分，特别是在图像中有微小变化或纹理的区域。

避免过度增强噪声：相较于其他频域滤波器，巴特沃斯高通滤波在增强高频部分时，提供了更平滑的过渡，避免了由于频率切断产生的不自然效果。

此处选取参数：

d0 = 20：截止频率，定义低频和高频的分界点。

n = 2：巴特沃斯滤波器的阶数，决定了频率响应的平滑度。

使用对比度增强后的图像转换为灰度图后进行锐化处理。

bilateral\_filtered\_image = cv2.bilateralFilter(image\_cv, d=9, sigmaColor=75, sigmaSpace=75)

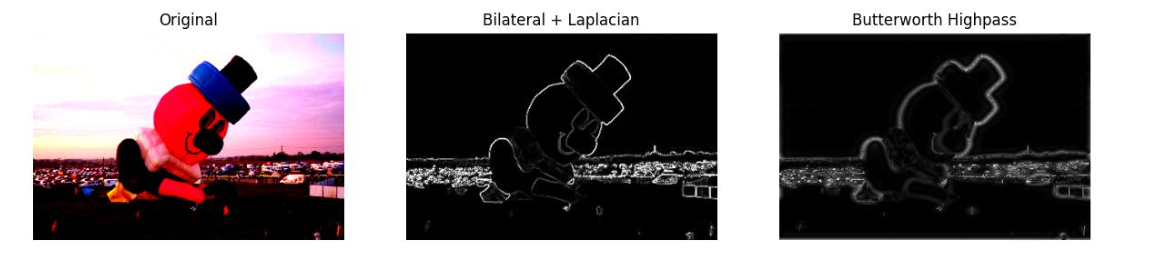
    # 拉普拉斯算子

    laplacian\_image = cv2.Laplacian(bilateral\_filtered\_image, cv2.CV\_64F)

    laplacian\_image = cv2.convertScaleAbs(laplacian\_image)

    # 将拉普拉斯图像叠加到原图上

    strengthened\_image = cv2.addWeighted(image\_cv, 1, laplacian\_image, 0.5, 0)



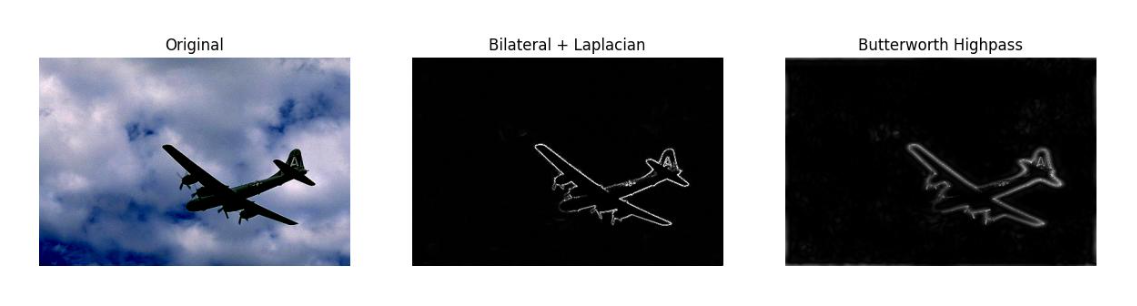


图 3 不同锐化方法比较

能够发现对于边界处理上，双边滤波＋拉普拉斯算子锐化的效果轮廓清晰并保留了大部分细节。而巴特沃斯虽然更加平滑但在图像分割上边缘并不适合。因此此处使用双边滤波＋拉普拉斯算子锐化的方法。

## 1.3预处理结果比较





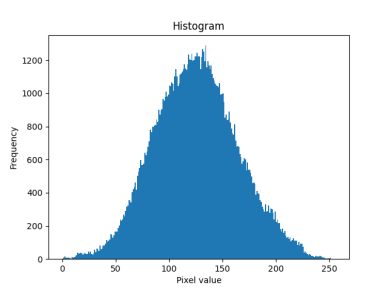
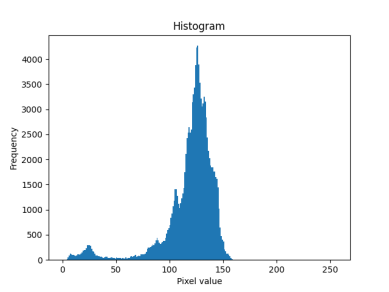
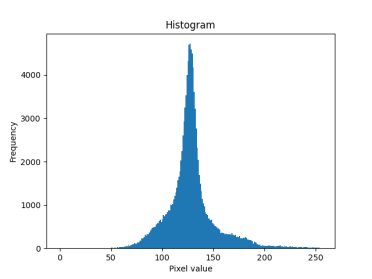
图 4 图像预处理结果

能够看出，通过对比度增强和锐化处理之后，在保留图像纹理细节同时，有效增加了图像的对比度和边缘清晰程度，为图像分割作好充足准备。

# 二、直方图分析与阈值分割方法尝试

## 2.1直方图绘制分析和阈值分割

对于阈值分割，首先需要分析其直方图，对十张待处理图片，进行灰度直方图的绘制。



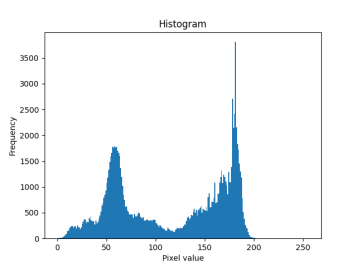
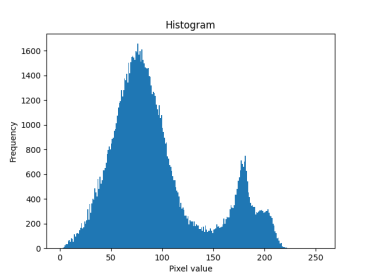
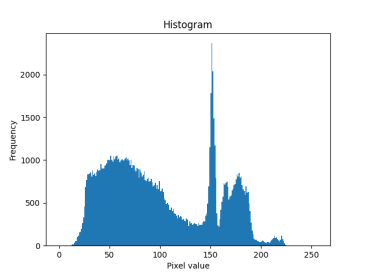


图 5 灰度直方图

能够观察到大部分灰度直方图呈单峰或多峰，尝试进行三角阈值分割和自动阈值分割并观察效果。此处使用三角阈值分割，OSTU阈值分割进行比较。

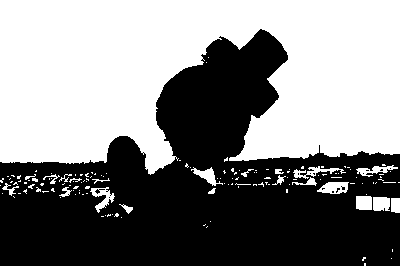


图 6 OSTU阈值分割 图 7 三角阈值分割

能够发现，对于一部分图片，灰度图呈多峰状只能使用自动阈值法分割处理，但其效果并不佳，部分细节被忽略并错误划分。对于三角阈值分割，大部分图片能够被正确分割。此处需要提取感兴趣图像需要具有普适性。因此放弃这一方法。

# 三、超像素分割

超像素概念是2003年Xiaofeng Ren提出和发展起来的图像分割技术，是指具有相似纹理、颜色、亮度等特征的相邻像素构成的有一定视觉意义的不规则像素块。它利用像素之间特征的相似性将像素分组,用少量的超像素代替大量的像素来表达图片特征,很大程度上降低了图像后处理的复杂度，所以通常作为分割算法的预处理步骤。

使用步骤一所示的图像预处理方法即对比度增强+双边滤波＋拉普拉斯算子边缘增强后图像如图所示，具有较高的对比度和边缘轮廓识别率。便于进行超像素分割。



图 8 预处理后的图像

## 3.1 超像素分割方法选取

超像素初始分割中，初始分割的目标是将图像划分成具有相似特征的区域（例如颜色、纹理、亮度等），从而为后续的特征提取、目标检测等任务提供基础。这些分割方法的核心思想是根据图像中的像素或超像素的相似性来划分区域。

常见的超像素分割方法包括： Graph-based 、NCut 、Turbopixel 、 Quick-shift 、 Graph-cut a、Graph-cut b 以及 SLIC 。下述为对分割方法的介绍选取：

(1) 分水岭方法（Watershed / Meanshift）

分水岭算法将图像视为地形图，将灰度值较低的区域看作“谷底”，将灰度值较高的区域看作“山峰”。算法通过模拟水从低处流向高处的过程，将图像划分成不同的区域。

另一种类似的方法是 Meanshift，它基于像素特征空间的密度估计，通过向密度更高的方向移动数据点来寻找区域的中心，从而实现图像分割。

优势：

能够很好地分割 具有明显边界 的图像，适合处理 灰度图像。

边界检测效果好，能够精确找到区域之间的边界。

适合 对象轮廓清晰 的场景。

劣势：

对 噪声敏感，可能产生过分割现象（即将一个对象分割成多个小块）。

对于 纹理复杂 的图像表现不佳。

(2) SLIC超像素算法

特点：

SLIC（Simple Linear Iterative Clustering）超像素算法是一种将图像分割成若干个小块（超像素）的算法。每个超像素是由相邻、相似的像素组成的区域，能够保留图像的局部特征。

优势：

计算效率高，可以快速生成超像素块。

空间和颜色特征结合，能够平衡区域的紧密度和颜色一致性。

超像素块通常是规则的、连贯的，更加符合人类的视觉感知。

劣势：

依赖于 初始化参数，如超像素数量和紧密度因子，参数选择不当可能导致分割不理想。

对于 复杂纹理或非均匀区域，可能产生较大的误差。

(3) 标准分割方法：N-cut（Normalized Cuts）

N-cut 是一种基于 图论 的图像分割方法，将图像表示为一个加权图，节点代表像素，边的权重表示像素之间的相似度。通过对图进行划分，使得划分后的子图之间的相似度最小化，而子图内部的相似度最大化。

优势：

能够实现 全局最优的分割，避免了局部最优问题。

适合处理 复杂图像，能够分割出具有不规则形状的区域。

能够同时考虑 局部和全局信息，实现更精确的分割。

劣势：

计算复杂度高，特别是在图像规模较大时，计算成本较高。

依赖于图像特征选择，不同的特征选择会影响分割结果。

segments = slic(image, n\_segments=350, compactness=20, sigma=1)  # 减少n\_segments和 增加compactness来合并超像素

    slic\_colored = color.label2rgb(segments, image, kind='avg')

    # N-cut方法（模拟）

    ncut\_segments = segmentation.slic(image, n\_segments=350, compactness=20)  # 减少 n\_segments来减少分割的数量

    ncut\_colored = color.label2rgb(ncut\_segments, image, kind='avg')

    # 分水岭算法 (基于标记的分水岭)

    # 应用阈值以获得二值图像

    ret, thresh = cv2.threshold(gray, 0, 255, cv2.THRESH\_BINARY\_INV + cv2.THRESH\_OTSU)

表 1 不同超像素分割方法比较

| **分割方法** | **适用场景** | **优势** | **劣势** |
| --- | --- | --- | --- |
| 分水岭（Watershed） | 医学影像、物体轮廓检测 | 精确的边界检测效果 | 对噪声敏感，可能产生过分割现象 |
| SLIC超像素算法 | 自然图像分割、前景/背景分离 | 计算效率高、符合人类视觉感知 | 对复杂纹理区域的分割效果可能不佳 |
| N-cut（标准分割） | 复杂图像分割、模糊边界分割 | 全局最优分割效果，适合复杂场景 | 计算复杂度高，对大图像处理较慢 |

## 3.2 超像素分割结果比较

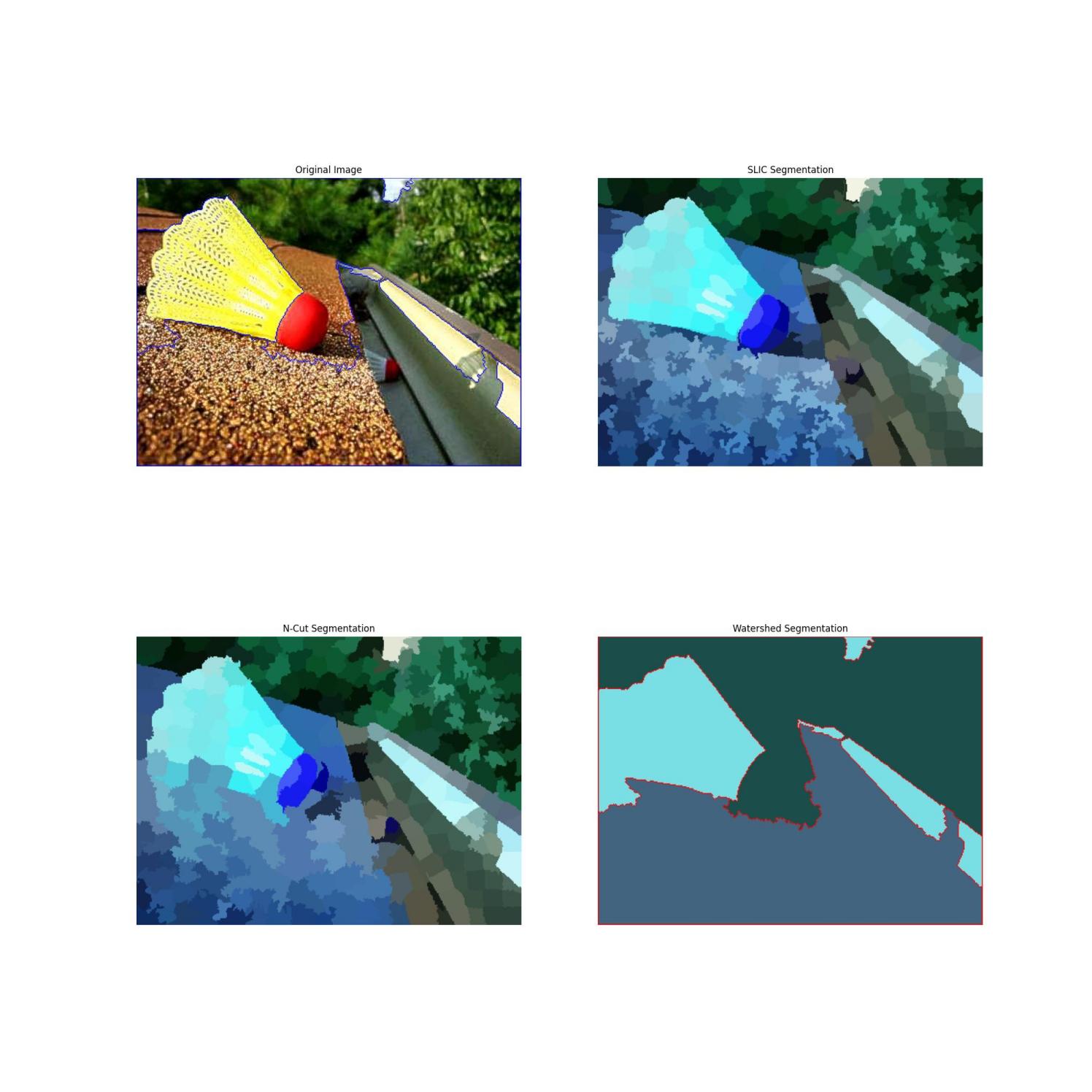


图 9 超像素分割结果比较

如图9所示，对选取经过预处理增强后的图片分别使用SLIC超像素算法，N-cut（标准分割），分水岭（Watershed）进行超像素分割。能够明显观察到分水岭（Watershed）对轮廓清晰程度要求较高，在非专业图像基础上的效果表现并不好,没有将其轮廓完整分割。而SLIC超像素算法和N-cut（标准分割）则做到了较好的分割。

对比SLIC和N-CUT，发现SLIC在前景/背景分离上表现更佳，而N-CUT方法则对上图中羽毛球进行了过度分割。因此此处选择SLIC进行超像素分割。

## 3.3 SLIC超像素分割参数选择

SLIC（simple linear iterativeclustering），即 简单线性迭代聚类 。

它是2010年提出的一种思想简单、实现方便的算法，将彩色图像转化为CIELAB颜色空间和XY坐标下的5维特征向量，然后对5维特征向量构造距离度量标准，对图像像素进行局部聚类的过程。

SLIC主要优点如下：

生成的超像素如同细胞一般紧凑整齐，邻域特征比较容易表达。这样基于像素的方法可以比较容易的改造为基于超像素的方法。

不仅可以分割彩色图，也可以兼容分割灰度图。

需要设置的参数非常少，默认情况下只需要设置一个预分割的超像素的数量。

相比其他的超像素分割方法，SLIC在运行速度、生成超像素的紧凑度、轮廓保持方面都比较理想。

SLIC(simple linear iterative clustering）原理分析

初始化种子点（聚类中心）：按照设定的超像素个数，在图像内均匀的分配种子点。假设图片总共有 N 个像素点，预分割为 K 个相同尺寸的超像素，那么每个超像素的大小为N/ K ，则相邻种子点的距离（步长）近似为S=sqrt(N/K)。

在种子点的n\*n邻域内重新选择种子点（一般取n=3）。具体方法为：计算该邻域内所有像素点的梯度值，将种子点移到该邻域内梯度最小的地方。这样做的目的是为了避免种子点落在梯度较大的轮廓边界上，以免影响后续聚类效果。

在每个种子点周围的邻域内为每个像素点分配类标签（即属于哪个聚类中心）。和标准的k-means在整张图中搜索不同，SLIC的搜索范围限制为2S2S，可以加速算法收敛，如下图。在此注意一点：期望的超像素尺寸为SS，但是搜索的范围是2S\*2S。

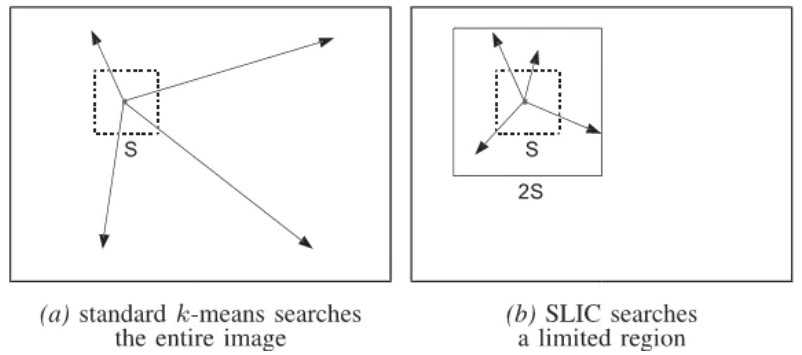


图 10 SLIC 原理图

理论上上述步骤不断迭代直到误差收敛（可以理解为每个像素点聚类中心不再发生变化为止），实践发现10次迭代对大部分图片都可以得到较理想效果，因此一般迭代次数取10。

经过上述迭代优化可能出现以下瑕疵：出现多连通情况、超像素尺寸过小，单个超像素被切割成多个不连续超像素等，这些情况可以通过增强连通性解决。主要思路是：新建一张标记表，表内元素均为-1，按照“Z”型走向（从左到右，从上到下顺序）将不连续的超像素、尺寸过小超像素重新分配给邻近的超像素，遍历过的像素点分配给相应的标签，直到所有点遍历完毕为止。

SLIC参数如下：

n\_segments: 分割块的个数。可能最后分割出的块数与实际设置并不一样，可能是slic算法做了后续处理，将小的超像素合并到大的超像素中。

compactness:分割块的边界是否压缩，压缩会使分割块的边沿更光滑。

经过参数调整发现当n\_segments=350，compactness=20时有最好效果。





图 11 SLIC 超像素分割结果

如图11所示，在n\_segments=350，compactness=20时，图像中绝大部分细节均能被超像素分割结果保留，并且清晰地分割出了感兴趣图像。

# 四、超像素特征表示

## 4.1特征选取

超像素特征表示可以分为 **低层视觉特征**、**中层语义特征** 和 **高层深度特征**：

| **特征类型** | **描述** | **适用场景** | **推荐使用情况** |
| --- | --- | --- | --- |
| 颜色特征 | 表示超像素的颜色分布 | 颜色差异明显的图像 | 推荐使用 |
| 纹理特征 | 表示超像素的纹理模式 | 复杂场景、自然图像 | 推荐使用 |
| 形状特征 | 表示超像素的边缘和轮廓 | 对目标形状敏感的任务 | 根据任务需求使用 |
| 空间位置特征 | 表示超像素在图像中的位置 | 目标位置有一定规律时 | 视任务需求选择 |

此处使用了

1. 颜色特征

颜色是图像中非常基本且直观的特征，能够很好地反映出图像中物体的外观和区域分布。通过计算每个超像素的平均颜色，可以有效地描述每个区域的色彩特征。

计算方法：计算每个超像素区域的 颜色均值，即每个区域所有像素的RGB值的平均值。 权重调整：颜色均值设置了加权系数 2。

2. 纹理特征

对比度 描述了图像灰度变化的强度，通常反映了区域的粗糙度。同质性 描述了图像纹理的平滑程度，反映了纹理的均匀性。纹理特征能够反映图像的细节和模式，帮助识别复杂的纹理结构。对比度有助于描述区域内部的变化，而同质性则有助于描述区域的平滑度。

权重调整：对比度和同质性设置了加权系数，但将它们的权重乘以 0.5

计算方法：使用了 灰度共生矩阵（GLCM） 来提取纹理特征，计算两个特征：对比度 和 同质性。

3. 边缘信息

使用了 Sobel算子 来提取每个超像素区域的边缘信息。Sobel算子通过计算图像的梯度来检测边缘，得到每个区域的 边缘幅度，即在该区域内的边缘强度。边缘信息有助于描述图像中物体的 轮廓 和 形状，是目标检测中常用的特征。边缘可以帮助区分不同的物体和背景，尤其是当物体具有明确的轮廓时。

权重调整:对边缘幅度特征设置了加权系数 0.8。

4. 形状特征

提取了两个形状特征：

(1)区域面积：即超像素区域内的像素数量。

(2)周长：即该区域的边界长度，通过轮廓检测（cv2.findContours）计算。

形状特征能有效地描述区域的几何属性，如大小和形状复杂度。面积和周长是常用的形状特征，有助于区分不同形状的物体或背景。

权重调整：面积特征加权系数 1.2，对周长特征的加权系数是 0.5。表示在任务中面积比周长更重要，而周长对特征表示影响较小。

## 4.2超像素特征聚类

使用 KMeans 聚类算法 对超像素特征进行分类：每个超像素的特征向量被视为一个样本。使用 KMeans 算法将这些超像素分成多个类别。聚类结果用于标识每个超像素区域属于哪个类别，从而实现图像的 语义分割。通过聚类，超像素区域被归为不同类别，例如 前景 和 背景。

# 五、感兴趣目标提取

## 5.1超像素合并

为了消除过度分割的影响，设计一个超像素合并函数 merge\_superpixels：比较每个超像素的特征向量之间的欧氏距离。如果两个超像素的特征距离小于设定的阈值，就将它们合并为同一个区域。合并后，通过形态学闭操作确保区域的连通性，避免出现分散的噪声区域。

在合并后的超像素基础上，通过以下步骤提取感兴趣目标区域：

1.找到最大连通区域：

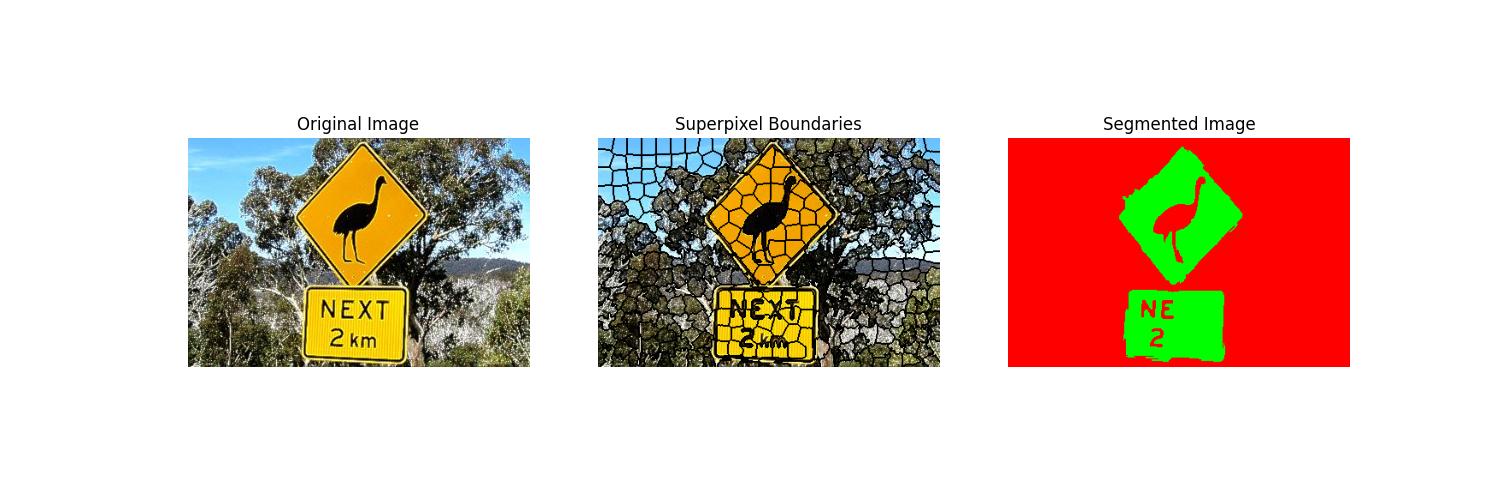
使用 连接组件分析（Connected Component Analysis）来标识图像中的连通区域。找到面积最大的连通区域，视为感兴趣目标。

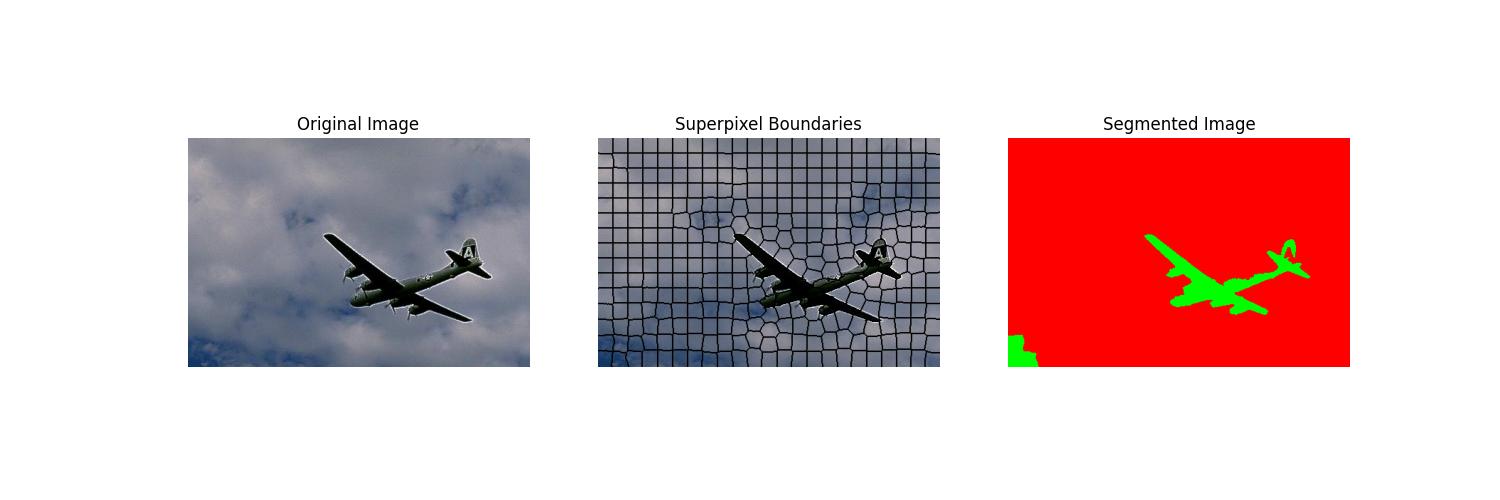
2.生成ROI图像：

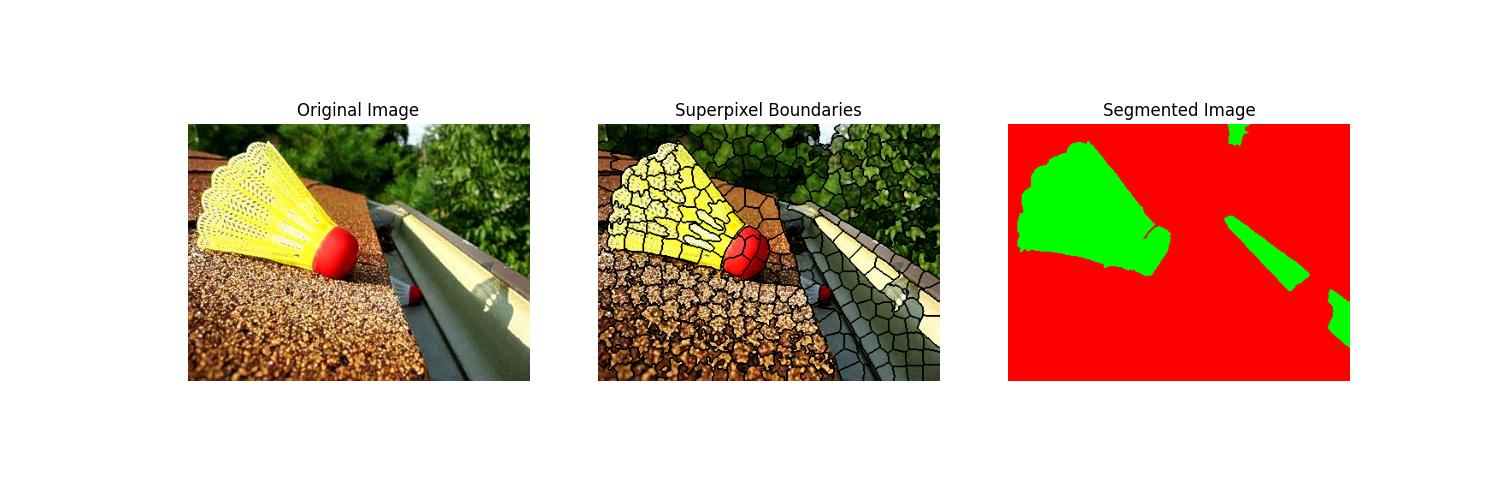
根据最大连通区域的掩码，将原图中对应的目标区域提取出来，生成最终的 ROI 图像。

## 5.2感兴趣目标提取结果

对于部分清晰，目标明确的样本，该方法具有良好的效果。能够清晰识别出感兴趣图像。







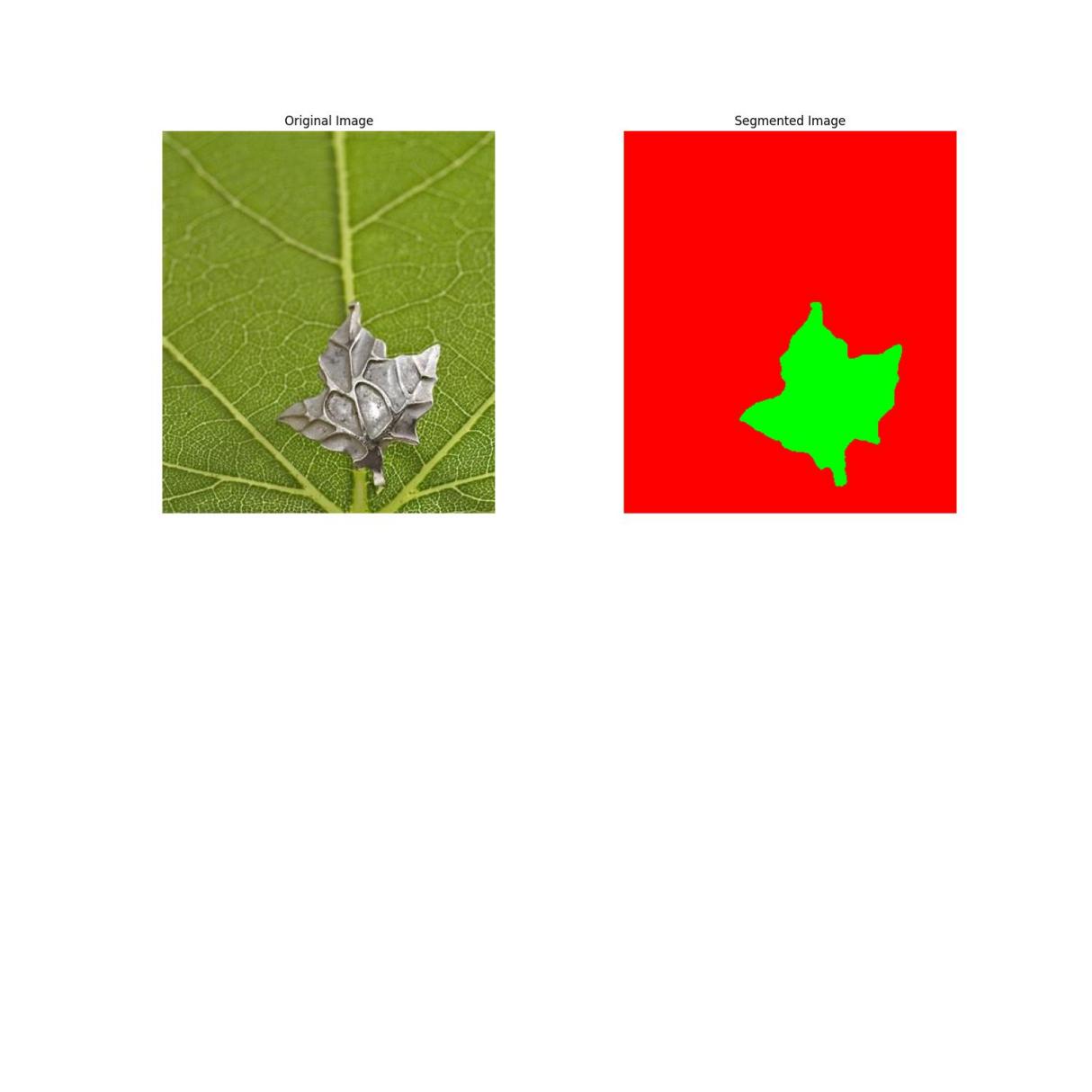


图 12 识别效果良好的图像

观察上述图像的超像素分割和合并的结果，能够观察到图像中主要目标均能被识别，并且没有过多的特征丢失。说明该方法效果良好。

但对于部分图像，分割效果并非如此。

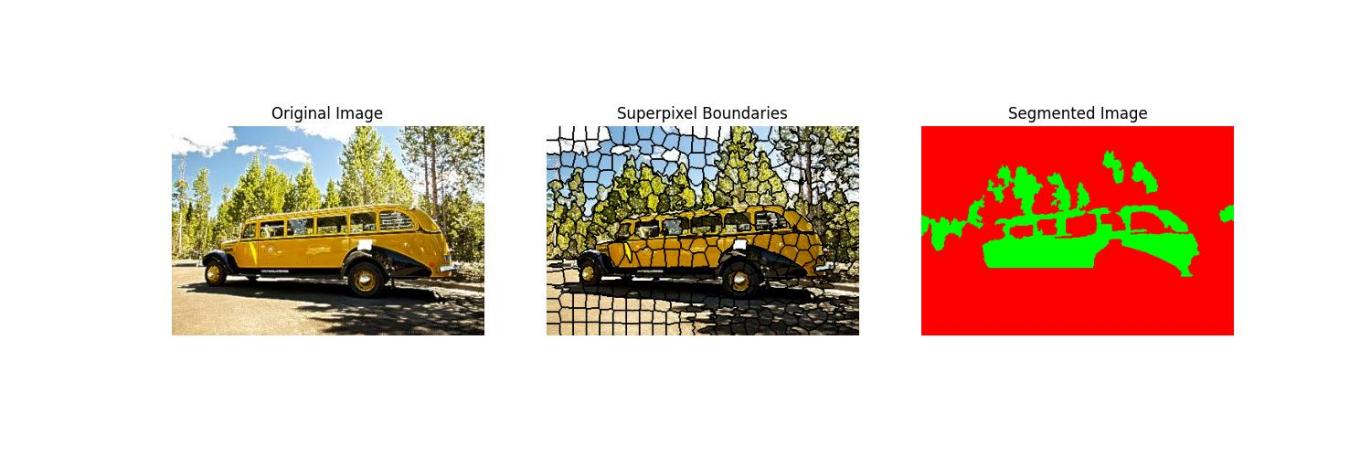
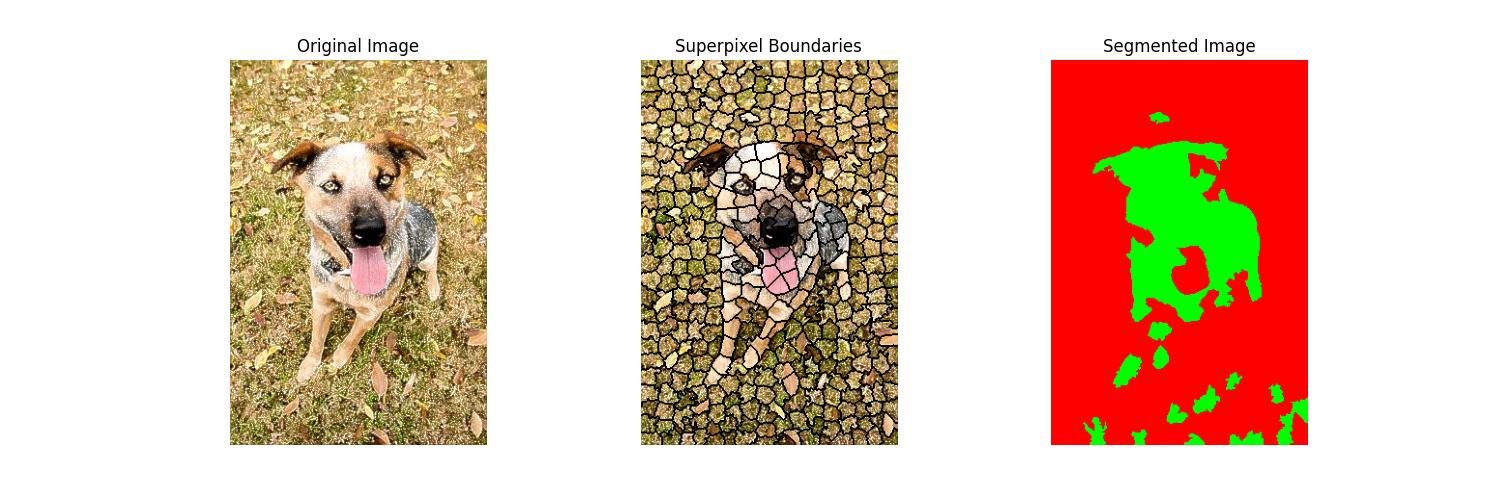
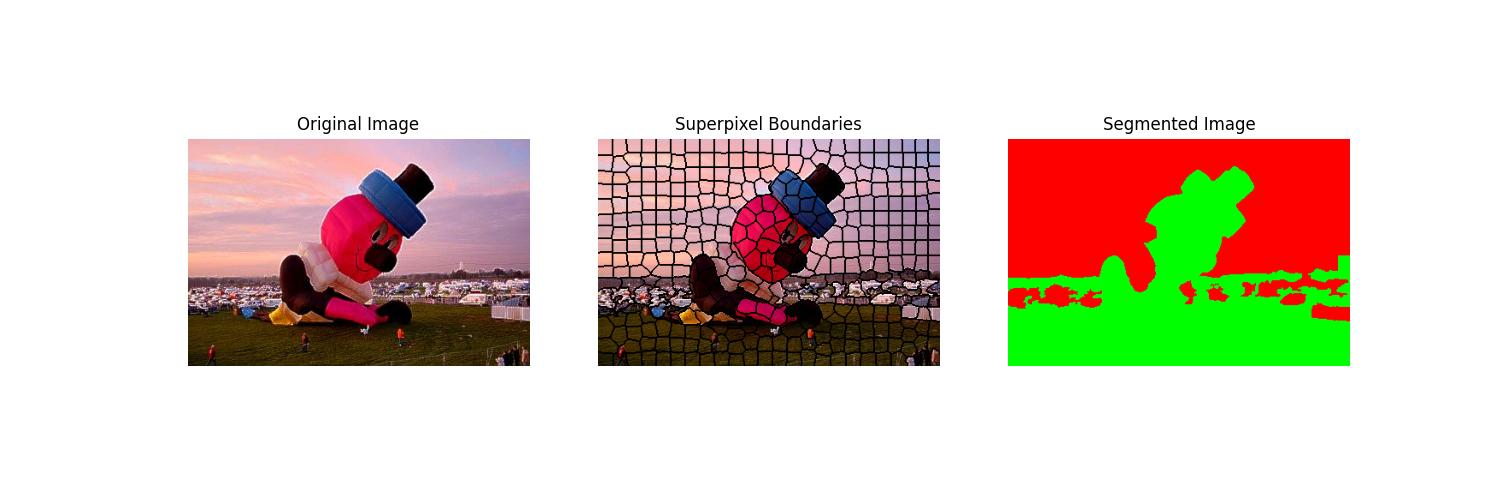


图 13 效果欠佳的图像

如图13所示，部分图像在合并超像素的过程种遗漏或错误合并了某些细节，导致最终的合并结果欠佳，该现象是由于超像素通过聚类方法合并的过程中未能考虑某些人为视觉上的因素，例如图中本应识别到人缺只识别其衣物，识别汽车整体时因其轮胎颜色原因未能识别将其识别为一个整体。对于该现象应使用基于深度学习的语义分割进行解决，运用yolo等成熟模型也许会有更好的效果。

# 六、课程设计总结与体会

通过本次课程设计，我在以下几个方面有了显著的提升：

1. 理论与实践结合

课程设计不仅加深了我对图像分割和特征提取理论知识的理解，还让我在实践中掌握了OpenCV、scikit-image等工具的使用方法。我逐渐认识到，只有将理论与实践结合，才能真正解决实际问题。

2. 算法调参与优化能力

在实验过程中，我遇到了一些问题，比如超像素区域过于零散、目标区域不完整等。这促使我不断调整算法参数（如SLIC的n\_segments和compactness参数、KMeans的聚类类别数等），以提升分割效果。

3. 解决问题的能力

在实验过程中，我学会了如何定位和解决代码中的问题。在调试和优化代码的过程中，日志分析、错误排查等技能也得到了提升。这让我意识到，编程能力不仅仅是写代码，更重要的是解决问题的能力。

4. 团队合作与交流

在与老师和同学的交流中，我学习到了很多新的想法和方法。通过讨论，我不仅能够及时发现自己的问题，还能够借鉴他人的经验，不断完善自己的设计方案。

# 参考文献

[1]唐真,庄怡.基于超像素分割算法的景观边缘提取仿真[J].计算机仿真,2024,41(09):228-232.

[2][Achanta, Radhakrishna](https://infoscience.epfl.ch/items/111df507-1d06-4406-b7f9-245fc03a6994),[Shaji, Appu,](https://infoscience.epfl.ch/items/dbb4ae92-9d06-4d37-ae80-3344f0b930cd)[Smith, Kevin,](https://infoscience.epfl.ch/items/883f4674-f7ea-4129-b967-3284f8eea84c)[Lucchi, Aurélien](https://infoscience.epfl.ch/items/549b00ed-5743-438f-a48a-f554f911507f),[Fua, Pascal](https://infoscience.epfl.ch/items/26ee9199-5a82-4097-90f0-e9eb052de875)[Süsstrunk, Sabine](https://infoscience.epfl.ch/items/92a168c1-ed7f-44d0-bb15-2b931a33b047)SLIC Superpixels Compared to State-of-the-art Superpixel Methods,[2012.120](https://doi.org/10.1109/TPAMI.2012.120" \t "https://infoscience.epfl.ch/entities/publication/605ab010-dcfa-4dcf-bb55-ddf0eda015ba/_blank)

[3]K. Harris, 1998, IEEE TIP, Hybrid image segmentation using watershed and fast region merging

[4]Haris K, Efstratiadis S N, Maglaveras N, et al. Hybrid image segmentation using watersheds and fast region merging.[J]. IEEE Transactions on Image Processing A Publication of the IEEE Signal Processing Society, 1998, 7(12):1684-1699.

# 致谢

在本次实验与课程设计的整个过程中，我受益良多，顺利完成本项目的各项研究与实现。这不仅是我个人努力的结果，更离不开老师和同学们的帮助与指导。在此，我想向我的指导老师 孙老师 表达我最真诚的感谢。

感谢孙老师在整个学习过程中给予的悉心指导和耐心解答。无论是算法的选择、实验方案的设计，还是代码实现中的细节问题，孙老师都给予了我宝贵的建议和支持。尤其是在图像分割与特征聚类的难点上，您的指导帮助我理清了思路，提升了我的技术水平与科研能力。

此外，我还要感谢同学们在实验过程中与我一起讨论问题、分享经验。正是因为这些交流与合作，我才能不断完善我的项目思路，顺利完成实验目标。

在本次实验的开展过程中，我不仅掌握了图像分割和特征提取的相关知识，还进一步增强了编程实践能力和解决实际问题的能力。这段学习经历将成为我今后学习与科研道路上的宝贵财富，再次感谢孙老师的辛勤付出和无私帮助。