

青島大學

数字图像处理课程作业

## 摘要

本文主要介绍了传统图像分割方法和基于深度学习的分割方法。传统方法包括阈值分割、区域生长、区域分裂与合并、边缘检测、分水岭分割、活动轮廓模型、图割算法、K 均值聚类、高斯混合模型和马尔科夫随机场等 11 类分割算法，这些方法在不同应用场景中各有其独特的优势与局限性。传统方法往往依赖于图像的基本特征，如灰度、颜色和纹理等，适用于一些简单的图像分割任务，但在面对复杂图像时可能表现出一定的局限性。

近年来，深度学习方法在图像分割领域取得了显著进展。本文着重介绍了全卷积网络（FCN）和 U-Net 等典型深度学习方法。深度学习方法通过卷积神经网络自动提取图像特征，显著提升了分割的精度和泛化能力，尤其在处理复杂图像时表现出色。与传统方法相比，深度学习方法具备更强的学习能力和适应性，但也对计算资源和大规模标注数据有较高要求。

实验部分通过实际数据集测试这些方法的性能，分析其优缺点。实验结果表明，深度学习方法在复杂场景中的表现优于传统方法，但传统方法在特定应用中仍具有不可替代的优势。本文的综述和实验分析为今后的研究和应用提供了重要参考。

**关键词：**图像分割；阈值分割；深度学习

# Overview of Image Segmentation Methods

## Abstract

This paper primarily introduces traditional image segmentation methods and deep learning-based segmentation methods. Traditional methods include 11 types of segmentation algorithms such as thresholding, region growing, region splitting and merging, edge detection, watershed segmentation, active contour models, graph cuts, k-means clustering, Gaussian mixture models, and Markov random fields. These methods each have their unique advantages and limitations in different application scenarios. Traditional methods often rely on basic image features such as grayscale, color, and texture, making them suitable for simple image segmentation tasks but potentially limited when dealing with complex images.

In recent years, deep learning methods have made significant advancements in the field of image segmentation. This paper focuses on representative deep learning methods such as Fully Convolutional Networks (FCN) and U-Net. Deep learning methods automatically extract image features through convolutional neural networks, significantly improving segmentation accuracy and generalization capabilities, particularly in handling complex images. Compared to traditional methods, deep learning methods have stronger learning capabilities and adaptability but also require substantial computational resources and large-scale labeled data.

The experimental section tests the performance of these methods on actual datasets and analyzes their advantages and disadvantages. The experimental results indicate that deep learning methods outperform traditional methods in complex scenarios, though traditional methods still hold irreplaceable advantages in certain applications. This review and experimental analysis provide important references for future research and applications.

**Keywords:** image processing; image segmentation; deep learning

# 目 录

摘 要.....	II
Abstract.....	III
0 引言 .....	5
1 传统图像分割方法 .....	5
1.1 阈值分割.....	5
1.2 区域生长.....	7
1.3 区域分裂与合并.....	8
1.4 边缘检测.....	9
1.5 分水岭分割.....	10
1.6 活动轮廓模型.....	11
1.7 图割算法.....	12
1.8 K 均值聚类.....	13
1.9 高斯混合模型.....	13
1.10 马尔科夫随机场.....	14
2 深度学习分割方法 .....	15
2.1 深度学习方法与传统方法的对比.....	15
2.2 FCN.....	15
2.3 U-Net.....	16
3 实验 .....	17
3.1 数据集介绍.....	17
3.2 实验详细.....	18
3.3 各模型分割效果分析.....	20
3.4 分割性能的对比.....	28
结 论.....	31
参考文献.....	31

## 0 引言

图像分割是计算机视觉和数字图像处理中的一个重要研究课题。其目标是将图像划分为具有独特特征的区域，以便进行进一步的分析和理解。图像分割方法的研究对医学影像分析、目标检测、视频监控、自动驾驶等领域有着重要意义。传统的图像分割方法主要依赖于图像的灰度、纹理、颜色等特征，通过阈值分割、区域生长、边缘检测和分水岭算法等技术实现<sup>[1]</sup>。然而，这些方法在处理复杂图像和应对多变场景时往往表现出一定的局限性。

随着计算能力的提升和大数据的普及，深度学习在图像处理领域展现出强大的优势<sup>[2]</sup>。深度学习方法特别是卷积神经网络（CNN），能够自动提取多层次的图像特征，大幅提高了图像分割的精度和鲁棒性。全卷积网络（FCN）<sup>[3]</sup>和 U-Net<sup>[4]</sup>等模型，通过端到端的训练方式，解决了传统方法在特征提取和组合上的不足。FCN 通过全卷积层替代全连接层，实现任意尺寸输入的图像分割，而 U-Net 则通过编码器-解码器结构增强了对高分辨率特征的捕捉。

本文将从传统方法和基于深度学习的方法两个方面，对图像分割技术进行全面综述，并通过实验对比不同方法的性能。通过分析这些方法的优缺点和适用场景，旨在为研究人员和工程师提供有价值的参考，为未来图像分割技术的发展方向提供指导。

## 1 传统图像分割方法

传统图像分割方法在计算机视觉领域有着悠久的发展历史，主要依赖于图像的灰度、纹理、颜色等基础特征。这些方法通常通过一些经典的算法将图像划分为若干具有相似特征的区域，以便进行进一步的分析和处理。传统图像分割方法包括阈值分割、区域生长、区域分裂与合并、边缘检测、分水岭分割、活动轮廓模型、图割算法、K 均值聚类、高斯混合模型和马尔科夫随机场等。这些方法各具特色，适用于不同的应用场景，对于一些简单的图像分割任务有着出色的表现，在如今的图像处理领域仍然发挥着重要作用。

这些方法各具特色，适用于不同的应用场景，对于一些简单的图像分割任务有着出色的表现，在如今的图像处理领域仍然发挥着重要作用。

### 1.1 阈值分割

阈值分割是一种基于区域的图像分割技术，其原理是通过设定不同的特征阈值，将图像像素点分为若干类。阈值分割法是一种传统的图像分割方法，因

其实现简单、计算量小、性能较稳定而成为图像分割中最基本和应用最广泛的分割技术。

它特别适用于目标和背景占据不同灰度级范围的图像。图像阈值化的目的是按照灰度级，对像素集合进行划分，得到的每个子集形成一个与现实景物相对应的区域，各个区域内部具有一致的属性。

通过设定特征阈值（如灰度或彩色特征），将图像像素点分为具有不同灰度级的目标区域和背景区域的若干类。

阈值的选取是图像阈值分割方法中的关键技术。常见的阈值选取方法有最大类间方差法（OTSU法）、基于最大熵的阈值分割法、迭代阈值分割法等<sup>[5]</sup>。其中，OTSU法是目前公认的，对阈值分割选择相对合理且分割效果良好的办法。

设计了一种基于模拟退火的阈值选取方法，如下所示。

#### 初始化:

- 设置初始阈值 `threshold`（可以是随机值或图像灰度直方图的某个特征值）。
- 设置初始温度 `T`（较高的温度值），初始温度决定了搜索的广度。
- 设置温度下降策略（如按固定比例下降，或根据迭代次数逐渐下降）。
- 设置终止条件（如温度低于某个阈值、达到最大迭代次数等）。
- 设定评价函数  $f(\text{threshold})$ ，该函数根据当前阈值计算分割后图像的质量度量（如类间方差）。

#### 当前状态评估

- 计算当前阈值 `threshold` 下的评价函数值  $f_{\text{current}} = f(\text{threshold})$ 。

#### 邻域搜索

- 生成当前阈值的邻域值（如随机扰动产生的新阈值），记作 `new_threshold`。
- 计算新阈值下的评价函数值  $f_{\text{new}} = f(\text{new\_threshold})$ 。

#### 接受准则

- 如果  $f_{\text{new}}$  优于  $f_{\text{current}}$ （即新阈值下的图像质量更好），则无条件接受 `new_threshold` 作为新的当前阈值。
- 如果  $f_{\text{new}}$  不优于  $f_{\text{current}}$ ，则根据模拟退火的概率接受准则决定是否接受 `new_threshold`。这个概率通常是一个随温度 `T` 减小的函数，如  $\exp((f_{\text{current}} - f_{\text{new}}) / T)$ 。

### 降温

- 根据降温策略降低温度  $T$ 。

### 终止条件检查

- 如果满足终止条件（如温度低于某个阈值、达到最大迭代次数等），则算法结束，输出当前阈值作为最优阈值。
- 否则，回到步骤 2 继续迭代。

## 1.2 区域生长

区域生长（Region Growth）算法<sup>[6]</sup>是一种基于区域的传统图像分割算法。区域生长可以根据预先定义的生长规则将像素或者小区域不断组合为更大区域的过程。具体地，区域生长是从一组初始种子点出发，通过预先定义的区域生长规则，将与种子点性质相似的领域像素不断添加到每个种子点上，并且满足区域生长的终止条件时形成最终生长区域的过程。

选择不同的种子点，对区域分割算法的结果影响较大。对于同一幅图像，我们对比如下（图 1.3.1-1.3.3）：



图 1.3.1 原始图像 图 1.3.2 选中白色区域作为种子 图 1.2.3 选中黑色区域作为种子

一般选择种子的方法包含人工选择和自动选择。人工选择可以根据图像的特点和需要分割的区域，直接选择一个或多个像素点作为种子点。自动选择可以采用自动选择种子的方法。例如，可以通过检测图像中的特定特征（如边缘、角点、纹理等）来自动选择种子点。具体流程以及分个示例如图 1.3.4-1.3.7 所示。

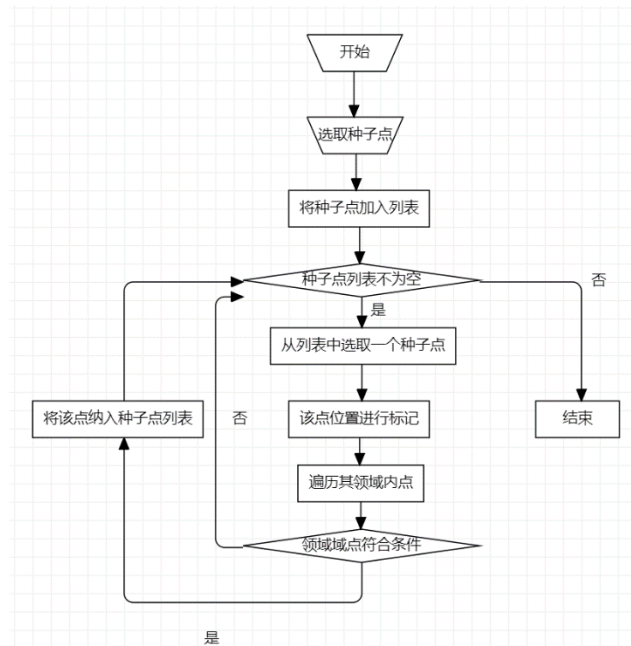


图 1.3.3 区域生长的流程图



图 1.3.4 待分割的图像



图 1.3.5 期望的结果



图 1.3.6 实际的结果

### 1.3 区域分裂与合并

区域分裂合并是一种基于区域的图像分割方法，其主要思想是通过递归地将图像分割成更小的区域（分裂）并根据某些准则（如相似性）合并相邻的区域，直到满足停止条件。该算法通常采用四叉树结构进行图像分割与合并。

算法的步骤包括：（1）初始分裂：将图像作为根节点开始，依据预定义的准则（如均匀性）递归地将图像区域分裂成子区域，直至所有子区域满足准则或达到最小尺寸。（2）合并步骤：在分裂完成后，检查相邻的区域，根据相似性准则（如颜色或纹理相似度）合并相邻的、相似的区域。

区域分裂合并算法是较为基础的图像分割算法，它实现简单，计算复杂度不高，而对于复杂纹理的图像处理效果一般。因此适合处理区域内部相对均匀的图像和具有明显区域划分的场景，如遥感图像中的土地分类，但不适用于过于复杂或纹理丰富的图像（图 1.4.1、图 1.4.2）。





图 1.4.1 原始图像

图 1.4.2 分割结果

## 1.4 边缘检测<sup>[7]</sup>

图像的边缘是图像最基本的特征。在图像中若某个像素点与相邻像素点的灰度值差异较大，则认为该像素点可能处于边界处像素点，并将它们连接起来，就可以形成边缘轮廓，从而将图像划分成不同的区域。通过计算一阶导数或二阶导数可以检测出图像中每个像素在其邻域内的灰度变化，从而检测出边缘。

在实际应用中，边缘检测法可以借助微分算子进行卷积实现分割。常用的边缘检测微分算子有：Roberts、Sobel、Prewitt、LoG、Canny 等（图 1.5.1-图 1.5.6）。分别使用不同的微分算子对相同的图像进行处理。从图中可以看出，经边缘检测算子处理后，物体的边缘轮廓相对清晰，实现了图像分割的目的。不同算子实现的边缘检测效果也有所不同，Roberts 算子丢失了很多边缘；Prewitt 算子和 Sobel 算子的抗噪声性能比 Roberts 算子好一些，有一定的抑制噪声的能力；LOG 算子和 Canny 算子检测的边缘清晰连贯。在实验中，我们通过应用 Canny 算子实现图像分割。

由于噪声等因素的影响，在通过 Canny 算子完成边缘检测后，其检测结果通常会是一些分散的边缘，没有形成分割区域所需的闭合边界。为此，我们应用数学形态学的操作实现边界连接。具体方法是：先在得到的边缘检测后的图像上进行图像膨胀操作，之后填充图像空洞区域，然后进行图像腐蚀操作，之后再一次填充图像空洞区域。通过实验，我们发现经过以上操作后，得到的分割结果最好。



图 1.5.1 原始图像

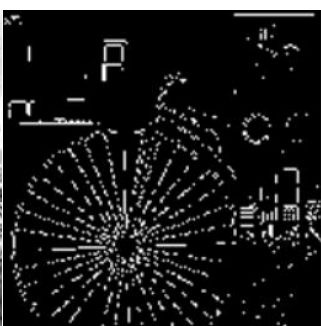


图 1.5.2 sobel 算子

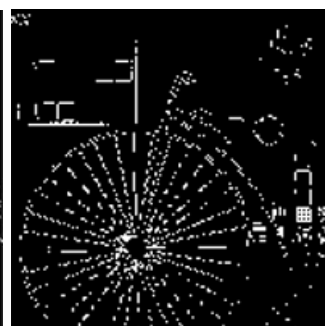


图 1.5.3 prewitt 算子

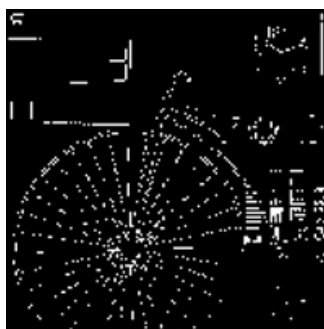


图 1.5.4 roberts 算子



图 1.5.5 log 算子

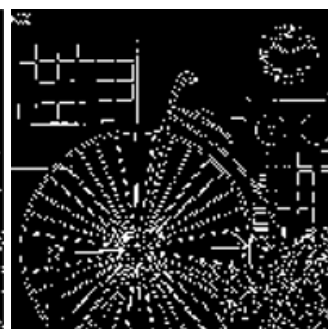


图 1.5.6 canny 算子

## 1.5 分水岭分割

分水岭分割算法将地形学和水文学的概念引入到基于区域的图像分割中，适合粘连区域的分割。分水岭分割法将图像视为地形图，其中图像的亮度和颜色信息对应于地形中的高度。通过寻找图像中的局部最小值点（即分水岭），算法将图像分割成不同的区域。

在图像中，每个像素点被视为地形图中的一个点，其灰度值代表该点的海拔高度。局部极小值及其影响区域被称为集水盆地，而集水盆地之间的边界则形成分水岭。分水岭分割算法的主要目的就是找到集水盆地之间的分水岭。

在实践中，我们通过计算图像的梯度信息，得到图像中的边缘信息，用于寻找分水岭。根据梯度图像（图 1.6.1-图 1.6.6）的局部最小值点确定种子点，这些种子点对应于图像中的分水岭。根据种子点，将图像中的像素点标记为不同的区域，并进行融合，得到最终的分割结果。

通过实验，我们发现分水岭分割算法对噪声比较敏感，容易产生过分割或欠分割的情况。因此，我们在分水岭分割算法中应用其他分割方法来提高分水岭分割法的效果。



图 1.6.1 原始图像

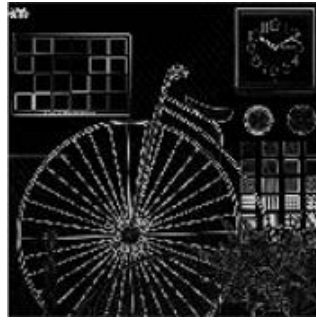


图 1.6.2 梯度幅度

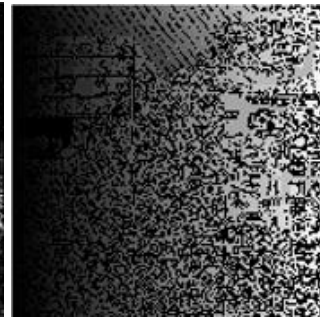


图 1.6.3 第一次分水岭

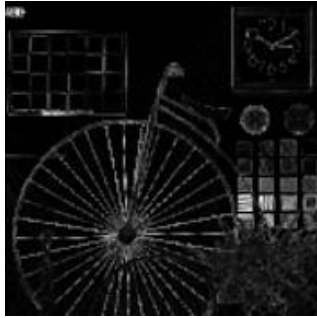


图 1.6.4 平滑后梯度幅度

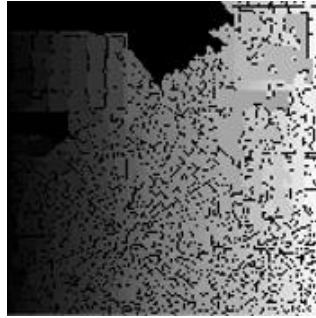


图 1.6.5 第二次分水岭

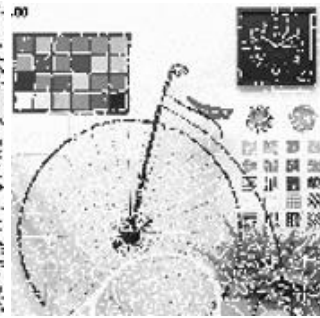


图 1.6.6 分水岭边界标记

## 1.6 活动轮廓模型<sup>[8]</sup>

活动轮廓模型，也称为蛇（Snake），是一种用于图像分割和边界检测的能量最小化方法。其基本思想是通过在图像中定义一个初始轮廓，然后通过迭代优化能量函数，使轮廓逐渐逼近目标物体的边界。能量函数通常包含两个部分：内部能量和外部能量。内部能量控制轮廓的光滑性，防止其过度变形；外部能量则利用图像的梯度信息，吸引轮廓向目标边界移动。

内部能量和外部能量的总和构成了整体能量函数：

$$E_{snake} = \int E_{internal} + E_{external} ds$$

其中， $E_{internal}$  包含一阶和二阶导数项，用于控制轮廓的拉伸和弯曲：

$$E_{internal} = \alpha \left| \frac{dv(s)}{ds} \right|^2 + \beta \left| \frac{d^2v(s)}{ds^2} \right|^2$$

而  $E_{external}$  则通常定义为图像梯度的负值，以吸引轮廓向边界移动：

$$E_{external} = -|\nabla I(v(s))|$$

初始轮廓能够准确的变形直至最终贴合到目标物体的边界.活动轮廓模型通过迭代优化能量函数，能够在复杂背景下准确地检测目标边界。这种方法在目



图 1.7.1 原始图像



图 1.7.2 分割结果

标边界清晰且连续的图像中表现尤为突出（图 1.7.1、图 1.7.2）。正因为该算法的高精度边界检测并能够处理复杂的目标形状，因此适合于需要高精度边界检测的场景，如医学图像中的器官轮廓检测。但是其对初始轮廓敏感，并且计算复杂度高，因此不适用于大规模的图像集。

## 1.7 图割算法<sup>[9]</sup>

图割算法是一种基于图论的图像分割方法。其主要思想是将图像像素表示为图中的节点，通过构建一个包含源点和汇点的流网络，并在图上寻找最小割，从而实现图像分割。

算法首先将图像表示为带权无向图  $G=(V,E)$  其中  $V$  是图像的像素， $E$  是相邻像素间的边。每条边  $(i,j)$  赋予权重  $w_{ij}$ ，表示像素  $i$  和  $j$  之间的相似性。源点  $s$  和汇点  $t$  分别代表前景和背景。图割算法通过求解最小割问题来分割图像：

$$\min_{S,T} \sum_{i \in S, j \in T} w_{ij}$$

其中  $S$  和  $T$  分别代表分割后的前景和背景集合，目标是找到最小割，使得前景和背景之间的边权总和最小。



图 1.8.1 原始图像

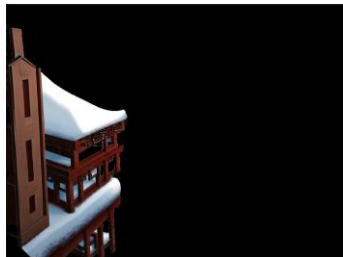


图 1.8.2 分割建筑

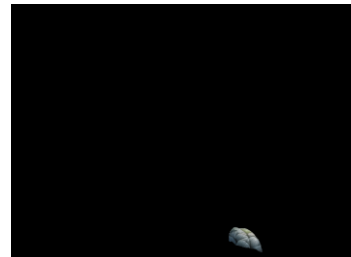


图 1.8.3 分割石头

图 1.8.1-1.8.3 展示了图割算法的分割效果。通过最小割优化，图像中的像素被分为前景和背景，前景物体的边界被准确地检测出来。图割算法在复杂背景下仍能有效地分割目标区域。图割算法是一种较为复杂的图像分割算法，计算复杂度高且对参数设置敏感，但同时具有优越的图像分割效果。适合复杂背景下的目标检测。

## 1.8 K 均值聚类<sup>[10]</sup>

K 均值聚类是一种迭代求解的聚类分析算法，旨在将数据集中的数据点划分为 K 个聚类。在图像分割的上下文中，数据点通常代表图像的像素，而像素的特征（如颜色、亮度等）则用于确定它们之间的相似性。K 均值聚类的主要流程步骤为：（1）初始化聚类中心。随机选择 K 个像素作为初始的聚类中心。这些中心点的选择会影响最终的聚类结果，因此在实际应用中，可能需要尝试不同的初始中心点或使用一些启发式方法来获得更好的结果。（2）分配像素到聚类：对于图像中的每个像素，计算它与各个聚类中心之间的距离。将每个像素分配给距离其最近的聚类中心所在的聚类。（3）对于每个聚类，计算其中所有像素的特征值的平均值将这个平均值作为新的聚类中心。（4）重复步骤（2）和（3），直到满足某个终止条件。终止条件可以是没有（或最小数目）像素被重新分配给不同的聚类，没有（或最小数目）聚类中心再发生变化，或者达到预定的迭代次数。

## 1.9 高斯混合模型<sup>[11]</sup>

高斯混合模型（Gaussian Mixture Model, GMM）是一种基于概率统计的方法，用于描述一个包含多个高斯分布的混合概率分布。其主要思想是将图像的像素值看作由若干个高斯分布生成，每个高斯分布对应图像中的一个类别或区域。GMM 通过最大化数据的似然函数，估计每个高斯分布的参数（均值向量、协方差矩阵和混合系数），从而实现图像的分割。

GMM 假设图像像素值  $\mathbf{x}$  服从 K 个高斯分布的加权和，即：

$$p(\mathbf{x}) = \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k)$$

其中， $\pi_k$  为第 k 个高斯分布的混合系数，满足

$$\sum_{k=1}^K \pi_k = 1;$$

$\mathcal{N}(\mathbf{x}|\mu_k, \Sigma_k)$  为第 k 个高斯分布， $\mu_k$  为均值向量， $\Sigma_k$  为协方差矩阵。



图 1.10.1 原始图像（洪水灾害 1）

图 1.10.1 GMM 分割结果（三类别）

GMM 在处理复杂图像时表现出色（图 1.10.1、图 1.10.2），特别是能够建模多模态分布的数据，利用概率模型处理不确定性和噪声。然而，GMM 对初始参数非常敏感，容易陷入局部最优解。虽然 EM 算法可以有效地估计参数，但其计算复杂度较高，尤其在大规模图像数据上。高斯分布的假设有时并不符合实际数据的分布，限制了 GMM 在某些应用中的有效性。总的来说，GMM 在图像分割领域仍然是一种重要的工具，但其应用需要结合具体场景和数据特点进行权衡。

## 1.10 马尔科夫随机场<sup>[12]</sup>

马尔科夫随机场（Markov Random Field, MRF）是一种利用概率图模型描述像素间空间依赖关系的图像分割方法。MRF 假设图像像素形成一个无向图，像素之间的依赖关系仅与其邻域中的像素有关。通过引入局部相互作用模型，MRF 能够捕捉图像中的空间结构信息，实现更精确的分割。

MRF 模型假设图像  $\mathbf{X}$  的像素标签  $\mathbf{Y}$  满足马尔科夫性质，即给定像素  $\mathbf{Y}_i$  的邻域  $\mathbf{N}_i$ ， $\mathbf{Y}_i$  的条件概率仅依赖于  $\mathbf{N}_i$  中的标签：

$$P(\mathbf{Y}_i | \mathbf{Y}_{-i}) = P(\mathbf{Y}_i | \mathbf{Y}_{\mathbf{N}_i})$$

通过定义能量函数  $U(\mathbf{X}, \mathbf{Y})$ ，MRF 模型将标签分配问题转换为能量最小化问题：

$$P(\mathbf{Y} | \mathbf{X}) = \frac{1}{Z} \exp(-U(\mathbf{X}, \mathbf{Y}))$$

其中， $Z$  为归一化常数。

马尔科夫随机场能够有效地捕捉图像的空间结构信息，从而提升分割精度，这使得其在处理噪声和复杂背景的图像时表现出色（图 1.10.3、图 1.10.4）。通过利用像素间的空间依赖性，MRF 可以更好地维护区域的连贯性。然而，MRF 的计算复杂度较高，尤其在高维图像数据上，计算代价昂贵。其性能在很大程度上依赖于模型参数的设定，参数选择不当可能导致分割效果





图 1.10.3 原始图像（洪水灾害 2）

图 1.10.4 MRF 分割结果（三类别）

不佳。此外，优化过程较慢，这对大规模图像数据的处理提出了挑战。尽管如此，MRF 在图像分割中的应用仍然具有重要意义，特别是在需要考虑空间依赖性和局部结构的场景中。

## 2 深度学习分割方法

### 2.1 深度学习方法与传统方法的对比

传统的图像分割方法主要依赖于图像的表层信息，无法满足需要大量语义信息的分割任务的需求。而随着深度学习的发展及引入，卷积神经网络为图像分割带来了新的可能。在图像分割领域中，卷积神经网络可以充分利用图像的语义信息，实现图像的语义分割，不仅提升了分割的精准度，还提高了处理的效率，使得图像分割的应用范围得到了进一步的拓展。本节将重点介绍 2 种基于深度学习的经典分割方法：FCN、U-Net。

### 2.2 FCN

全卷积网络（FCN）是深度学习用于语义分割的开创之作，确立了图像语义分割通用网络模型框架。通常，卷积神经网络（CNN）经过多层卷积之后接入若干个全连接层，将卷积层产生的特征图（feature map）映射成固定长度的特征向量进行分类。但 FCN 与 CNN 不同，如图 2.2.1 所示，FCN 采用“全卷积”方式，在经过 8 层卷积处理后，对特征图进行上采样实现反卷积操作，然后通 SoftMax 层进行分类，最后输出分割结果。

全卷积网络模型在图像分割领域具有重要的意义。它是首个将深度学习用于语义分割的模型，通过端到端的方式，可以接受任意尺寸的输入图像，并保持像素级别的精度。图 2.2.2 所示，全卷积网络模型对不同类别（人、车、羊、船等）目标的分割效果。实验表明，全卷积网络模型对图像的细节信息不够敏

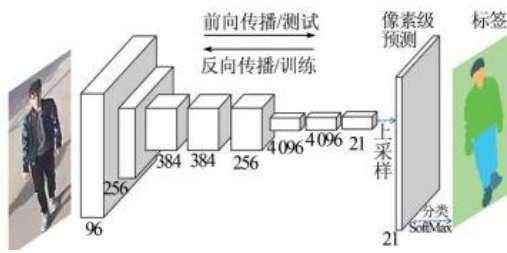


图 2.2.1 语义分割模型 FCN

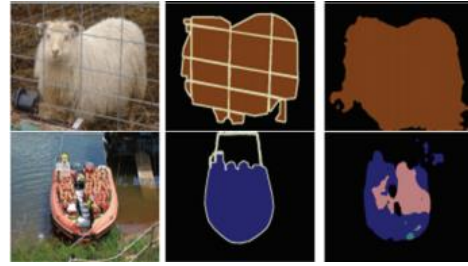


图 2.2.2 FCN 模型的分割效果

感，且由于像素点之间的关联性较低，导致目标边界模糊，前景目标的轮廓分割得不够细致。

## 2.3 U-Net

U-Net 是一种在医学图像分割领域广泛应用的深度学习模型。主要用于生物医学图像分割。U-Net 的结构形似字母“U”，如图 2.3.1 所示，主要由两部分组成：下采样和上采样。下采样主要用于捕获图像的上下文信息，而上采样则用于精确定位。这两个部分的结合使得 U-Net 能够在保持高分辨率特征的同时，有效地利用上下文信息。

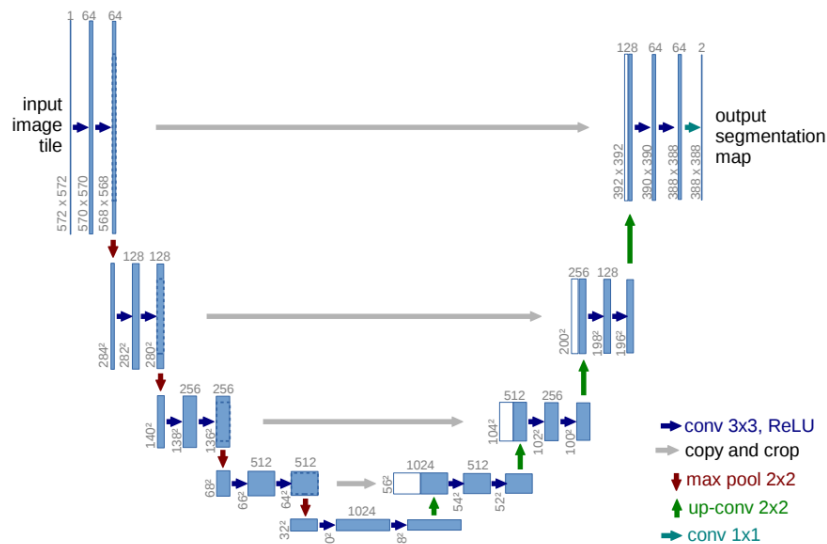


图 2.3.1 U-Net 模型结构

在下采样过程中，U-Net 使用了连续的两个 3x3 卷积操作，然后进行 2x2 的最大池化操作进行下采样。每次下采样后，特征通道的数量加倍。在上采样过程中，U-Net 首先使用 2x2 的上采样操作，然后将结果与下采样过程中相应级别的特征图进行拼接，再进行两个 3x3 的卷积操作。U-Net 的一个重要特点是其在上采样过程中使用了跳跃连接，将下采样过程中的特征图与上采样过程中



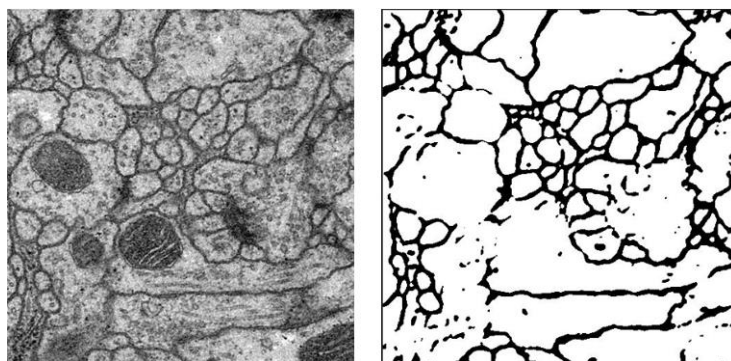


图 2.3.2 U-Net 分割效果

的特征图进行拼接。这种设计使得 U-Net 能够在进行分割时，同时利用浅层的细节信息和深层的语义信息。

U-Net 模型以其优秀的分割性能和有效的结构设计，在医学图像分割领域取得了广泛的应用。如图 2.3.2 所示，U-Net 能够在保持高分辨率特征的同时，有效地利用上下文信息。同时，U-Net 使用跳跃连接，使得其在进行分割同时利用浅层的细节信息和深层的语义信息，进一步提高分割的性能。但由于卷积和池化操作，U-Net 在处理小目标时，可能会导致小目标在特征图中消失，对小目标的分割效果不佳。

### 3 实验

#### 3.1 数据集介绍

本文所使用的数据集来源于“Skin Lesion Analysis Towards Melanoma Detection”挑战赛，该挑战赛的数据集由 ISIC（International Skin Imaging Collaboration）提供。数据集包含了来自 ISIC 档案的标注皮肤病变图像，涵盖了恶性和良性皮肤病变的代表性混合图像。数据集在发布前已随机划分为训练集和测试集，其中训练集约有 900 张图像，测试集约有 350 张图像。因为传统图像分割算法大多是无监督模型，所以我们仅选择了其中的 900 张训练集图像作为我们的数据集，其中大多如图 3.1.1 所示。

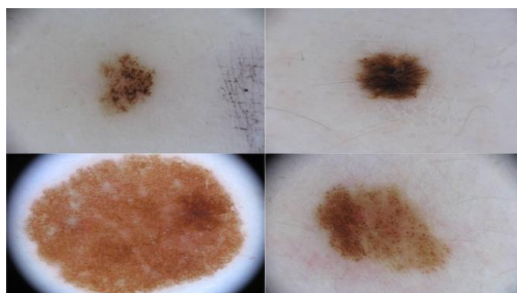


图 3.1.1 数据集示例

数据集的标注包括三个参考标准注释：1. 分割：皮肤病变由专业皮肤科医生在正常皮肤和其他结构之间进行了分割。2. 皮肤镜特征：皮肤病变的局部临床皮肤镜特征由专家进行了标注。3. 疾病状态：皮肤病变的诊断金标准为病理学诊断。ISIC 档案中的图像均来自具有专业病理学认证的中心，可以作为金标准。档案中未经过病理诊断的良性病变由多位专家进行评审，只有在一致性临床诊断的情况下才会被纳入。

我们将该数据集的特点归纳为一下两点：

- 任务为单目标分割，但部分原始图像存在多目标干扰
- 目标物体边缘不明显，过度较为光滑

## 3.2 实验详细

**阈值分割、区域生长和 K 均值聚类：**为三种方法依次设计了分割函数并且适配了接口。我们的分割函数接口分别定义如下：

```
function segmented_image = K_minth(image)
function J = region_grow(I)
function mask = threshold_seg(image, threshold)
```

这三个函数分别接受图片参数，完成对图片的分割任务。其中 `region_grow` 采取了一种基于图像特色的简单有效的种子选择方案，`threshold_seg` 的改进版本中，我们引入了模拟退火方案来确定阈值。

**区域分裂合并算法**的主要流程设置：首先，将整个图像视为一个初始区域。然后，递归地分裂不满足特定均匀性准则的区域，将其划分为更小的子区域。每次分裂时，评估每个子区域的均匀性，继续分裂不均匀的区域。分裂过程持续进行，直到所有区域都满足均匀性准则或达到最小区域尺寸。接着，检查相邻的子区域，对满足合并准则的区域进行合并。合并时，评估相邻区域的相似度，将相似的区域合并为一个更大的区域。迭代进行分裂和合并过程，直到没有进一步的分裂或合并。最终，将得到的区域标签映射回原图像尺寸，生成分割掩码。

**边缘检测**的主要流程设置：（1）采用 `canny` 算子，使用 `edge` 函数将原始灰度图像，将阈值设置为 0.3，进行边缘检测，得到的结果是不连续的边缘。

（2）使用 `imdilate` 函数对图像进行膨胀操作，在其中采用菱形作为结构元素，参数设置为 3。（3）使用 `imfill` 函数填充孔洞。（4）使用 `imerode` 函数对图像进行腐蚀操作，在其中采用菱形作为结构元素，参数设置为 3。（5）使用 `bwperim` 函数进行轮廓跟踪，找到图像的轮廓。参数设置为 8，8 方向轮廓跟踪。（6）使用 `imfill` 函数填充孔洞。

**分水岭分割**的主要流程设置：（1）使用 `fspecial` 函数创建滤波算子，算子类型设置为 `sobel`，使用 `imfilter` 函数计算图像的梯度幅度。（2）使用 `imclose`

函数进行闭运算对图像进行处理。（3）使用 `watershed` 函数进行分水岭变换。

（4）找到分水岭变换结果中为 0 的位置（即分水岭的边界）。（5）将分水岭边界设置为白色（255），得到分水岭分割后的图像。（6）将原图像转化为二值图像。（7）将分水岭分割后的图像与转换的二值图像进行与运算，得到二者综合的结果。（8）进行闭运算得到最终分割的结果。

**活动轮廓模型**的主要流程设置：首先，初始化一个初始轮廓，可以是手动绘制的或者通过其他方法自动生成。然后，根据图像的梯度信息和轮廓的内外外部能量，定义活动轮廓模型的能量函数。接着，迭代地优化能量函数，通过最小化能量来驱动轮廓向图像的边缘移动。在每次迭代中，轮廓点的位置更新，通过计算新的能量值来调整轮廓形状。迭代过程持续进行，直到能量函数收敛或达到预定的迭代次数。最终，得到的轮廓即为分割结果，将其映射到原图像中，生成分割掩码。

**图割算法**的主要流程设置：首先，将图像表示为一个图，图的节点表示像素，边表示像素间的相似度或距离。然后，定义两个源节点和汇节点，分别代表前景和背景。接着，根据像素间的相似度和像素与源汇节点的连接权重，构建一个流网络。使用最大流最小割算法，通过最小化割断边的总权重，将图分割为前景和背景两部分。在计算完成后，前景和背景的像素分别对应于源节点和汇节点的连通分量。最后，将前景和背景的像素标签映射回原图像尺寸，得到分割掩码。

**高斯混合模型（GMM）**的主要流程设置：首先，将原始灰度图像矩阵展平为向量形式。然后，利用 `fitgmdist` 函数拟合高斯混合模型，指定要分割的聚类数（`cluster_num`）和正则化值（`RegularizationValue`）以确保计算的稳定性。在模型拟合完成后，计算每个像素属于每个高斯成分的后验概率，进而选取最大后验概率对应的高斯成分作为每个像素的类别。最后，将得到的标签重塑回原图像尺寸，并将标签映射到 0-255 的灰度范围，得到最终的分割掩码。

**马尔科夫随机场（MRF）**的主要流程设置：首先，将原始灰度图像转换为双精度类型，并初始化标签矩阵，随机分配每个像素的初始类别标签。然后，在每次迭代中，基于像素及其邻域的标签信息，计算先验概率。接着，根据先验概率和似然函数（结合像素灰度值与各类别的高斯分布参数），更新每个像素的类别标签。迭代过程持续到达到预设的最大迭代次数。最终，将标签重塑回原图像尺寸，并映射到 0-255 的灰度范围，得到最终的分割掩码。

**U-Net** 的主要流程设置：首先，加载包含原始图像和像素标签的数据集，并将数据集划分为训练集、验证集和测试集。然后，定义 U-Net 网络结构，指定输入图像的大小和类别数量。设置训练选项，包括初始学习率、最大迭代次数（20）和批次大小（8）。在这些准备工作完成后，利用训练数据进行模型训

练。训练完成后，对测试集中每张图像进行预测，生成分割掩码。最后，将预测结果保存，并对模型进行评估。

### 3.3 各模型分割效果分析

**阈值分割、K 均值聚类和区域生长：**在图像分割领域，阈值分割算法的优点在于其简单和计算速度快。它通过设置一个阈值将像素分为前景和背景，适用于图像对比度高且目标与背景差异明显的情况。然而，当图像中的斑点和毛发与背景对比度不足时，阈值分割很难区分它们（图 3.3.1）。K 均值聚类算法通过将像素点分配到最近的聚类中心来进行分割，优点是能够在多维空间中进行聚类，更好地处理颜色和纹理信息，它比阈值分割更能滤除毛发等干扰因素（图 3.3.2）。但是，K 均值聚类的缺点包括对初始聚类中心的选择敏感，以及在处理大小和密度差异较大的区域时可能会出现问题。在实践中，区域生长算法表现出了较好的分割效果，然而，该算法的成功很大程度上依赖于合适的种子点选择；不恰当的种子点可能导致分割失败（图 3.3.3）。

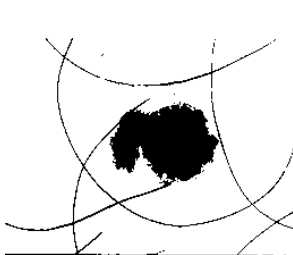


图 3.3.1 阈值分割

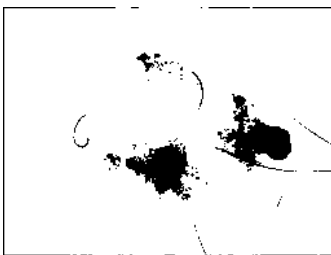


图 3.3.2 K 均值聚类

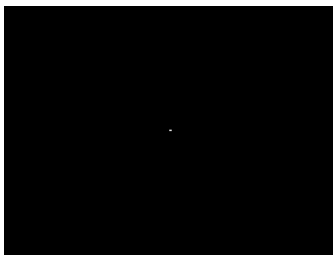


图 3.3.3 区域生长

**区域分裂合并算法 (SplitAndMerge)：**区域分裂合并算法通过递归地分裂和合并图像区域，能够有效地识别和分割出具有相似属性的区域，从而提高分割效果。它利用分裂过程将不均匀的区域划分为更小的子区域，然后通过合并过程将相似的子区域结合起来，形成最终的分割结果。区域分裂合并算法在处理边界清晰、区域间差异明显的图像时表现尤为出色。然而，区域分裂合并算法也存在一定的局限性，例如在处理纹理复杂、噪声较多或区域边界不明显的图像时，可能会导致过度分裂或合并不足，从而影响分割精度。同时，由于合并的局限性，算法的结果必然包含不少孔洞和其他噪声。使用开闭运算去除噪声之后结果会好得多（表 3.1、图 3.3.4、图 3.3.5）。

表 3.1 区域分裂合并算法的消融实验

方法	Dice	IOU
SplitAndMerge	0.4875	<b>0.3839</b>
SplitAndMerge+开闭运算	<b>0.5139</b>	0.3745



图 3.3.4 阈值分割



图 3.3.5 K 均值聚类

**边缘检测：**我们对不同的微分算子进行实验比较，如图一通过对目标图像的直观比较，我们认为选用 canny 算子实现的效果最好。

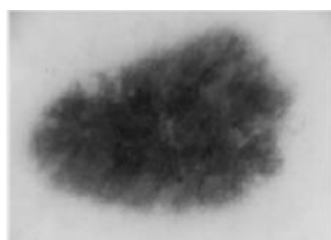


图 3.3.6 原始图像



图 3.3.7 sobel 算子



图 3.3.8 prewitt 算子



图 3.3.9 roberts 算子



图 3.3.10 log 算子

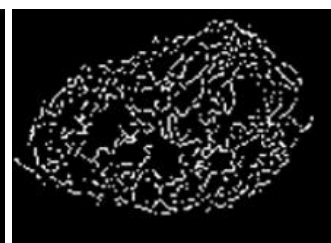


图 3.3.11 canny 算子

通过边缘检测得到的是不连续的边缘，为得到连续的边界，我们运用数字形态学处理。对得到的图像进行闭运算。得到图像的连续边界。在实验中，我们发现在进行膨胀操作后添加一次填充孔洞的操作比没有添加填充孔洞操作要好，如图 2 所示，因此我们在实验中设置了分别在膨胀、腐蚀操作后两次填充孔洞的操作。

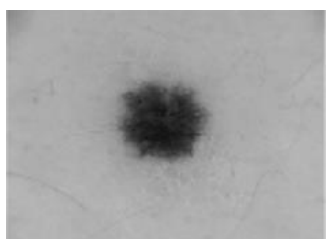


图 3.3.12 原始图像



图 3.3.13 膨胀后填孔洞



图 3.3.14 膨胀后不填孔洞

**分水岭分割**对噪声比较敏感，而我们进行分水岭操作后，得到的分割效果并不好，得到的并不是我们想要的分割结果，而是分割出大量的碎片（图 3.3.13-图 3.3.18），所示，因此，我们将分水岭分割与二值图像进行结合，来实现效果的改善。

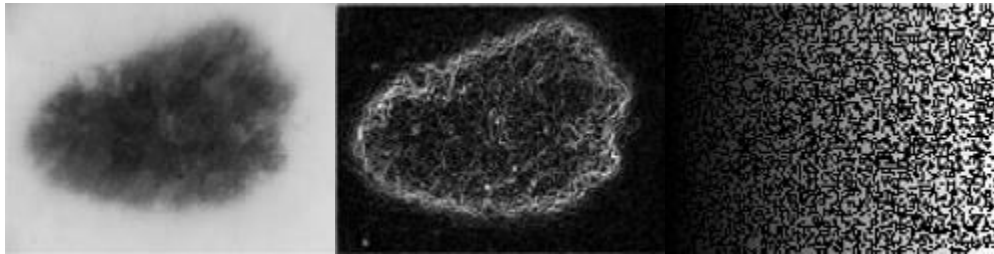


图 3.3.13 原始图像

图 3.3.14 梯度幅度

图 3.3.15 第一次分水岭



图 3.3.16 平滑后梯度幅度

图 3.3.17 第二次分水岭

图 3.3.18 分水岭边界标记

**活动轮廓模型 (ActiveContour)**：尤其适用于具有清晰边界的目标（图 3.3.19）。通过初始化一个轮廓，并利用图像的梯度信息迭代调整轮廓形状，活动轮廓模型能精准捕捉目标边缘，保证分割的精确度。它通过平衡内外部能量函数，使轮廓在边缘处停留，适用于边缘信息丰富的图像。然而，对于纹理复杂、噪声较多或边界模糊的图像，活动轮廓模型可能会收敛不准，影响分割效果（图 3.3.20）。此外，其计算复杂度较高，需要多次迭代优化，耗时较长。此外，该模型对初始边界较为敏感。



图 3.3.19 清晰边缘图像的分割结果



图 3.3.20 未能收敛的分割结果

**图割算法（GrabCut）：**在图像分割中有显著优势，尤其在处理前景与背景对比明显的图像时。它通过将图像像素表示为图的节点，并根据像素间相似度设置边的权重，然后利用最大流最小割算法实现分割，确保分割的准确性和连贯性。图割算法擅长处理复杂背景下的目标分割，能全局优化分割结果，避免局部最优。然而，图割算法对初始边界敏感，效果可能不佳。此外，计算复杂度较高，处理大规模图像时耗时较长。因此，通常需要结合其他方法，本次实验中，采用了几种分割方法产生初始掩膜结合图割算法的方法，表现出色（表 3.2）。

表 3.2 图割算法的消融实验

方法	Dice	IOU
GrabCut	0.6258	0.5098
GrabCut + ActiveContour	0.6258	0.5088
GrabCut + SplitAndMerge	0.6042	0.4938
<b>GrabCut + GMM</b>	<b>0.6540</b>	<b>0.5538</b>

其中 GrabCut 的初始轮廓为距离上下左右边界各是 20px 的矩形边框。其表现相较于其他模型较为出色。

**GrabCut + ActiveContour** 没有任何提升，究其原因因为 GrabCut 天生对初始轮廓敏感，ActiveContour 亦是如此。这就导致了 ActiveContour 无法确定初始轮廓的，再使用 GrabCut 之后和直接使用矩形边框无异。因此 GrabCut 不能结合其他对初始轮廓敏感模型使用。

**GrabCut + SplitAndMerge** 反而是下降，这是由于 SplitAndMerge 本身的限制，SplitAndMerge 本身的效果一般，很可能使得初始轮廓小于实际轮廓。因此，GrabCut 不宜结合效果普通的模型。

综上所述，想要 GrabCut 发挥更好的效果，需要更好地确定初始轮廓的方法。需要确定的初始轮廓的性质为：（1）唯一性。初始 mask 只存在唯一的白色区域。初始掩膜上多个白色区域会显著影响效果。（2）包围性。初始轮廓必须包裹住实际的初始轮廓，宁愿比初始轮廓大得多，也决不能小于实际轮廓。（3）无孔洞。唯一的初始轮廓中不能有孔洞。

**GrabCut + GMM：**在使用效果较好的 GMM（见下，高斯混合模型）之后，得到的初始 mask 也并不能确保满足以上所有性质，因此，需要经过处理（图 3.3.21 为对应流程图，图 3.3.22 为对应图像效果图）。首先，算法开始并使用高斯混合模型（GMM）生成初始掩膜。接着，对生成的掩膜进行开运算和

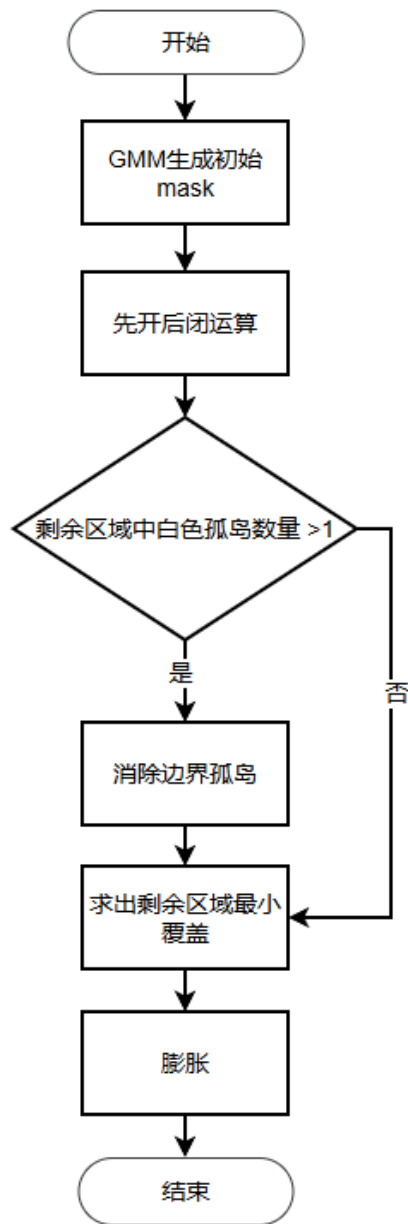


图 3.3.21 GrabCut+GMM 算法流程图

闭运算，这两个操作用于去除噪声和填补小孔洞，从而平滑图像。然后，算法检查剩余区域中孤岛的数量（孤岛指的是在二值图像中，与其他区域分离的单独像素或小区域，在我们的数据集中，孤岛指的就是与目标区域分离的干扰区域，这些干扰区域被误认为是目标区域，而这些干扰区域往往位于图像边界），如果超过一个，则会将边界孤岛消除。反之，继续进行后续操作。接下来，计算剩余区域的最小覆盖矩形。最后，对该区域进行膨胀操作，以完成处理，得到最后的分割掩膜。



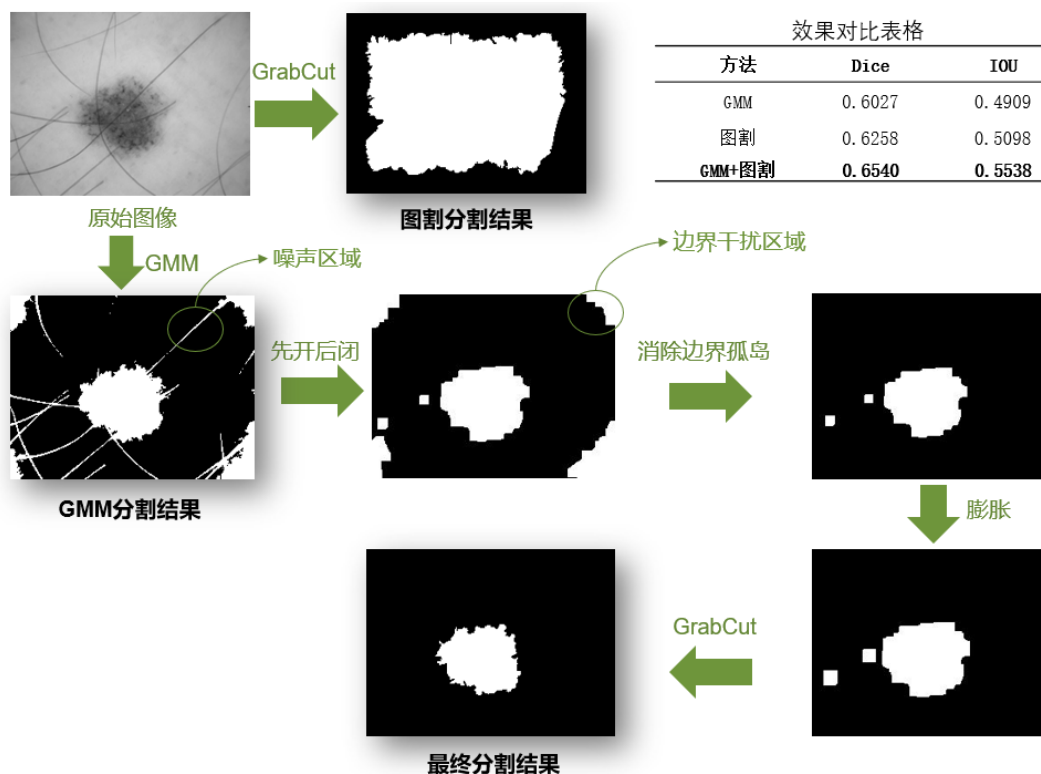


图 3.3.22 GrabCut+GMM 效果对比图

**高斯混合模型（GMM）：**在图像分割领域中表现出了一定的优势，特别是在处理具有统计特性明显的数据集时。它能够有效地模拟数据点的分布，并且通过调整混合成分的数量，可以灵活地适应不同复杂度的图像。然而，GMM 也有其局限性，例如在处理具有复杂纹理或噪声较多的图像时，可能会导致分割效果不佳。在实际应用中，GMM 在图像背景和目标区域的颜色或亮度对比较大时表现较好，而在颜色或亮度分布相似、边界不明显的场景下表现不佳。此外，GMM 对初始参数的选择较为敏感，不当的初始化可能会导致收敛到局部最优解，在我们的实验中，由于初始化的随机性，就导致了对于目标区域划分成背景的情况（图 3.3.23、图 3.3.24）。

要解决这个问题，可以通过中心点初始化、额外加入空间信息和掩膜修正等操作，以减弱或消除初始化随机性带来的误差。因此，在选择 GMM 作为分割方法时，需要根据具体的数据集特点和应用场景进行适当的调整和优化。

此处，我们观察数据集特点，判断我们的目标大多位于图像的中心区域，对模型加入了额外的空间信息并在返回掩膜之前修正发生逆置的掩膜，方法如下：如果该掩膜的中心点为背景标签，我们则视为该掩膜的标签是前景和背景是相反的，将掩膜的标签进行倒换，即将背景区域和目标区域进行置换；如果中心点为目标区域，则不进行任何操作。此外，我们尝试使用 K-means 算法来



图 3.3.23 GMM 正确划分

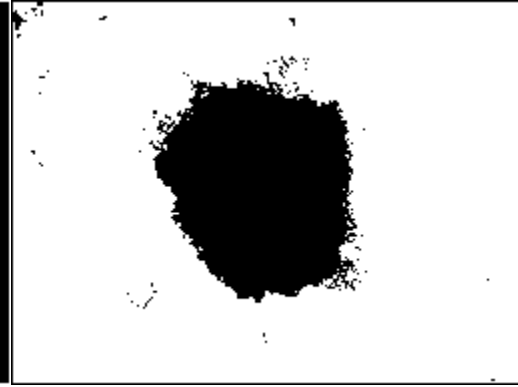


图 3.3.24 GMM 错误划分

初始化 GMM 的参数进而帮助克服随机初始化带来的不稳定性问题，但效果并不理想。我们对此做出的消融实验结果如表 3.3 所示。

表 3.3 GMM 算法的消融实验

方法	Dice	IOU
GMM	0.4875	0.3839
GMM + kmeans 初始化	0.4855	0.3824
<b>标签修正+GMM</b>	<b>0.6027</b>	<b>0.4909</b>

**马尔科夫随机场（MRF）**：在图像分割领域中也展现出了独特的优势，尤其是在处理具有空间一致性的图像时。MRF 通过建模像素之间的空间关系，能够有效地增强图像的局部平滑性和连贯性，从而提高分割效果。它利用邻域系统来描述像素间的相互作用，通过最大化后验概率（MAP）来实现最优分割。MRF 在处理边界清晰、目标和背景具有明显区别的图像时表现尤为出色。然而，MRF 也存在一定的局限性，例如在处理纹理复杂、噪声较多或边界模糊的图像时，可能会导致过度平滑或细节丢失（图 3.3.25、图 3.3.26）。此外，MRF 的计算复杂度较高，求解过程中需要使用迭代优化算法，如图割（Graph Cut）或贝叶斯采样（Gibbs Sampling），这使得其计算时间较长。

与 GMM 相似的是，MRF 的分割结果也具有目标区域与背景区域逆置的现象，对 MRF 算法进行了标签修正的操作后，分割准确率大幅增长（表 3.4）。

在实际应用中，MRF 在医学图像、卫星影像等对分割精度要求较高的领域表现良好，特别是那些目标和背景区域具有一定空间连续性的场景。而在颜色或亮度变化剧烈、目标和背景难以区分的场景中，MRF 的表现可能不尽如人意。为了提高 MRF 的分割效果，可以结合预处理步骤如降噪、初始分割以及后处理步骤如形态学操作来优化结果。此外，参数选择在 MRF 中至关重要，不当

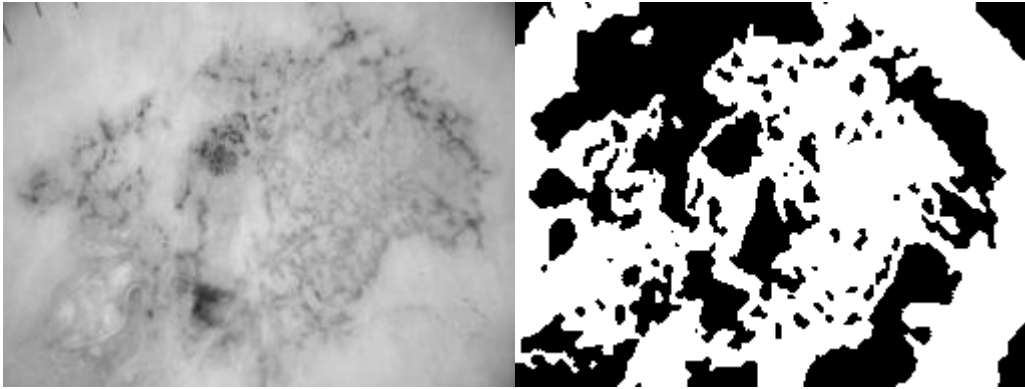


图 3.3.25 边缘不明显的原始图像      图 3.3.26 边缘不明显的原始图像的分割效果

表 3.4 MRF 算法的消融实验

方法	Dice	IOU
MRF	0.4875	0.3839
标签修正+MRF	<b>0.5844</b>	<b>0.4689</b>

的参数设置可能会导致分割结果的不稳定。在我们的实验中，通过对邻域系统的合理选择和参数的调优，我们能够显著改善分割效果。因此，在选择 MRF 作为分割方法时，需要根据具体的图像特性和应用需求进行仔细的参数调整和优化。

**U-Net:** 在图像分割领域中表现出强大的性能，尤其是在医学图像和遥感图像等高精度分割任务中。U-Net 是一种基于卷积神经网络（CNN）的深度学习模型，通过对图像的编码和解码过程实现高效的分割。其架构包括对称的编码路径和解码路径，编码路径提取图像的高层次特征，解码路径逐步恢复空间分辨率，并结合编码路径的特征图以增强分割效果。U-Net 通过跳跃连接（skip connections）将编码层的特征与解码层的特征融合，从而保留更多的图像细节信息，提高分割精度（图 3.3.27-3.3.30）。

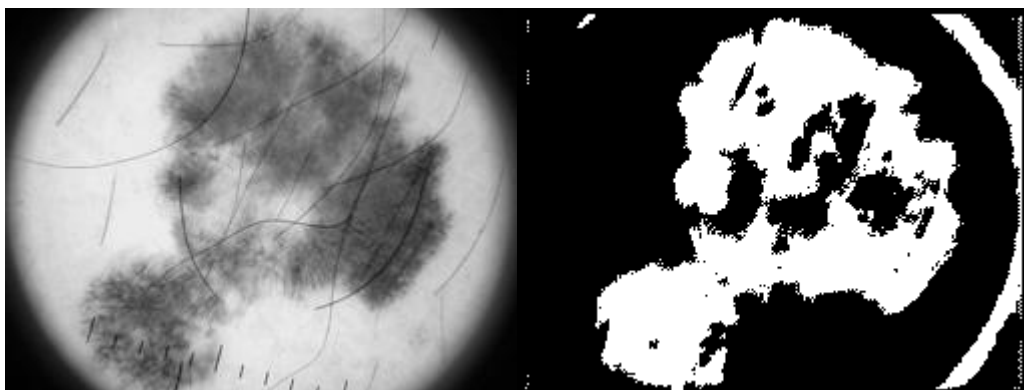


图 3.3.27 原始图像

图 3.3.28 分割掩膜

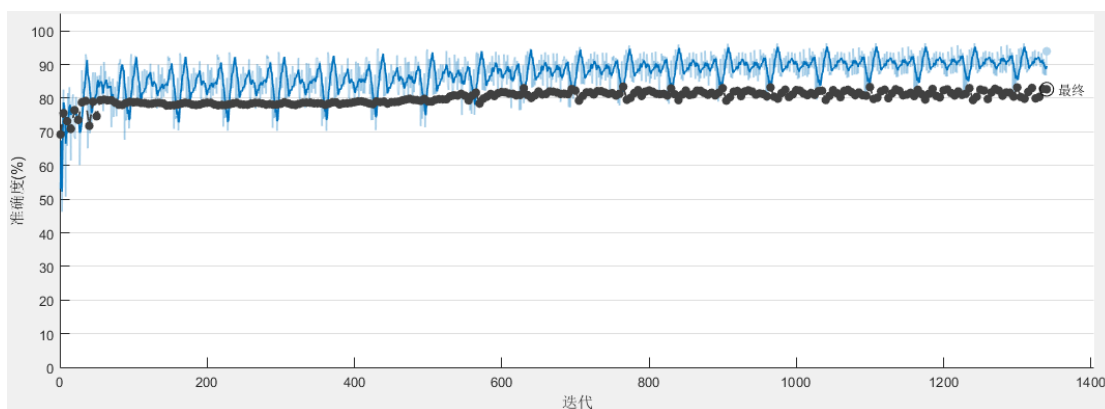


图 3.3.29 U-Net 预测准确率与迭代次数的关系

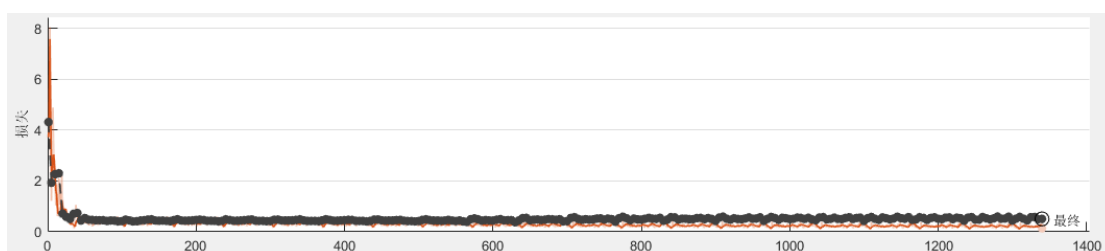


图 3.3.30 U-Net 预测损失与迭代次数的关系

然而，U-Net 也存在一些局限性，特别是与传统的图像分割方法相比。首先，U-Net 对大量标注数据的需求较高，深度学习模型需要大量的带标签数据进行训练，而标注过程通常耗时且昂贵。其次，U-Net 的训练过程需要大量的计算资源，包括高性能 GPU 和充足的存储空间，这在资源有限的情况下可能成为瓶颈。此外，U-Net 在处理复杂背景或弱对比度图像时可能会出现误分割，因为模型在这些情况下可能无法提取到足够的区分性特征。相比之下，传统的图像分割方法，如阈值分割和边缘检测等，通常对计算资源和数据量的需求较低，可以在较小的数据集上实现快速分割，适用于计算资源有限的场景。

### 3.4 分割性能的对比

在本研究中，我们系统地对比了多种图像分割算法的性能，包括传统分割算法和基于深度学习的 U-Net 模型。表格 3.2 详细列出了各算法在 Dice 系数、IOU（交并比）和运行时间上的表现。通过分析这些性能指标，我们可以深入了解各算法的优劣，并进一步探讨传统分割算法与深度学习分割算法之间的关系。

从表 3.5 和图 3.3.31 中的数据可以看出，传统分割算法在 Dice 系数和 IOU 方面的表现各有千秋。阈值分割的 Dice 系数为 0.5399，IOU 为 0.4163，运行时间仅为 4 秒，这表明其在简单场景下具有快速且一定的分割能力。然而，由于其依赖于图像的灰度值分布，对复杂图像的处理效果有限。区域生长算法

的 Dice 系数和 IOU 分别为 0.3071 和 0.2162，运行时间为 22 秒，效果不佳，反映出该算法在处理含有噪声和不规则区域的图像时表现不稳定。区域分裂与合并算法在 Dice 系数和 IOU 上分别达到 0.5139 和 0.3745，但运行时间高达 528 秒，计算成本显著增加，尽管效果有所提升，但计算效率低下。

边缘检测算法的表现相对较好，Dice 系数为 0.5641，IOU 为 0.4539，运行时间为 13 秒，说明其在该数据集的图像中具有优势。分水岭分割算法在 Dice 系数和 IOU 上分别为 0.5831 和 0.4552，运行时间为 35 秒，其性能较为均衡，但在复杂场景中可能出现过分割现象。活动轮廓模型的 Dice 系数为 0.5572，IOU 为 0.4362，运行时间为 690 秒，尽管在处理复杂图像结构上有一定优势，但计算时间过长。

图割算法表现出色，Dice 系数为 0.6258，IOU 为 0.5098，运行时间为 96 秒，说明其在图像分割中具有较高的准确性和效率。K 均值聚类的 Dice 系数和 IOU 分别为 0.5782 和 0.4498，运行时间为 11 秒，其在处理高维数据时表现良好，但对于复杂的图像场景仍有不足。高斯混合模型（GMM）的 Dice 系数为 0.6027，IOU 为 0.4909，运行时间为 52 秒，表现出较好的分割性能，尤其是在多模态图像分割中。

马尔科夫随机场（MRF）算法的 Dice 系数为 0.5844，IOU 为 0.4689，运行时间为 113 秒，其在考虑空间依赖性方面表现良好，但计算复杂度较高。GMM 与图割结合的算法（GMM+图割）在 Dice 系数和 IOU 上分别达到 0.6540 和 0.5538，运行时间为 250 秒，表明这种结合方法能在增加的时间成本可接受的情况下，显著提升分割性能。

基于深度学习的 U-Net 模型在 Dice 系数和 IOU 上分别达到 0.6885 和 0.5716，明显优于所有传统方法。U-Net 通过其多层卷积和池化操作，能够有效捕捉图像中的多尺度特征，其优越性尤其在处理复杂场景和大规模数据时得以体现。虽然其计算时间较长，但在现代高性能计算设备的支持下，这一劣势可以得到有效缓解。但值得注意的是，虽然它的训练时间高达 33 分钟，但该模型在预测与其他传统算法数量一致（900 张）的图像时，大概只需要 20s。对于深度学习模型，在实际应用中我们仅需要训练一次，这可能需要耗费一定时间，但一旦训练完成，之后的预测过程中我们能得到相较于传统算法更高的精度和可接受的预测时间，这也是为什么深度学习模型的应用广泛的原因。

综合以上分析，传统分割算法在处理简单场景和快速应用时具有一定优势，但在复杂场景和大规模数据处理中显得力不从心。相较之下，深度学习分割算法，特别是 U-Net，凭借其强大的学习能力和对多尺度特征的捕捉，在分割性能上表现优异，其精度和稳定性使其逐渐成为图像

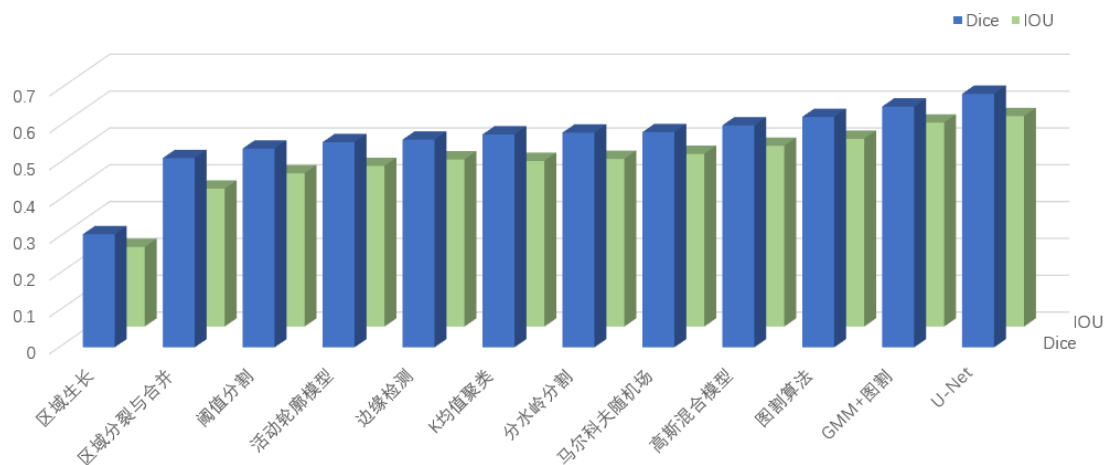


图 3.3.31 分割性能对比

表 3.5 分割性能对比

算法	Dice	IOU	RunTime（秒）
阈值分割	0. 5399	0. 4163	<b>4</b>
区域生长	0. 3071	0. 2162	22
区域分裂与合并	0. 5139	0. 3745	528
边缘检测	0. 5641	0. 4539	13
分水岭分割	0. 5831	0. 4552	35
活动轮廓模型	0. 5572	0. 4362	690
图割算法	0. 6258	0. 5098	96
K 均值聚类	0. 5782	0. 4498	11
高斯混合模型	0. 6027	0. 4909	52
马尔科夫随机场	0. 5844	0. 4689	113
GMM+图割	0. 6540	0. 5538	250
<b>U-Net</b>	<b>0. 6885</b>	<b>0. 5716</b>	20

分割领域的主流选择。但不得否认的是，相较于传统模型，对于大多数深度学习模型其训练仍然较高，且需要大量已标记过得数据集来完成训练。对于一个新的分割场景，他们可能由于训练数据集的不足而无法发挥原有的作用。如今传统分割模型在图像分割领域仍有一席之地，主要归功于他们都是无监督的分割模型，能够直接在多种不同的分割场景中应用。

## 结 论

本文综述了图像分割领域的主要方法，包括传统的基于特征的分割方法和近年来快速发展的基于深度学习的方法。传统方法如阈值分割、区域生长、边缘检测和分水岭分割等，在特定场景下具有较好的效果，但在处理复杂图像时往往表现出局限性。相比之下，深度学习方法如 U-Net 通过大规模数据训练，能够自动学习图像特征，显著提升了分割精度和泛化能力。

实验结果表明，深度学习方法在复杂场景下的表现优于传统方法，特别是在细节处理和边界识别方面具有明显优势。然而，深度学习方法对计算资源要求较高，且依赖于大量标注数据，这在某些应用中可能成为限制因素。因此，未来的研究可以考虑传统方法与深度学习方法的结合，充分利用两者的优点，以进一步提升图像分割的性能和适应性。

总之，图像分割技术在不断发展，随着深度学习技术的进步，未来必将涌现出更多高效且精确的分割方法，为各个领域的应用提供有力支持。本文的综述和实验分析为后续研究提供了参考方向和实践依据。

## 参考文献

- [1] 何俊,葛红,王玉峰.图像分割算法研究综述[J].计算机工程与科学,2009,31(12):58-61.
- [2] 卢宏涛,张秦川.深度卷积神经网络在计算机视觉中的应用研究综述[J].数据采集与处理,2016,31(01):1-17.DOI:10.16337/j.1004-9037.2016.01.001.
- [3] Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2015: 3431-3440.
- [4] Ronneberger O, Fischer P, Brox T. U-net: Convolutional networks for biomedical image segmentation[C]//Medical image computing and computer-assisted intervention—MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18. Springer International Publishing, 2015: 234-241.
- [5] 吴一全,孟天亮,吴诗姝.图像阈值分割方法研究进展 20 年(1994—2014)[J].数据采集与处理,2015,30(01):1-23.DOI:10.16337/j.1004-9037.2015.01.001.

- [6] 林通,石青云.一种基于边缘生长的灰度和彩色图象分割方法[J].中国图象图形学报,2000(11):33-37.
- [7] 魏伟波,芮筱亭.图像边缘检测方法研究[J].计算机工程与应用,2006(30):88-91.
- [8] 陈波,赖剑煌.用于图像分割的活动轮廓模型综述[J].中国图象图形学报,2007(01):11-20.
- [9] Tang M, Gorelick L, Veksler O, et al. Grabcut in one cut[C]//Proceedings of the IEEE international conference on computer vision. 2013: 1769-1776.
- [10] Kodinariya T M, Makwana P R. Review on determining number of Cluster in K-Means Clustering[J]. International Journal, 2013, 1(6): 90-95.
- [11] Reynolds D A. Gaussian mixture models[J]. Encyclopedia of biometrics, 2009, 741(659-663).
- [12] Cross G R, Jain A K. Markov random field texture models[J]. IEEE Transactions on pattern analysis and machine intelligence, 1983 (1): 25-39.