

# 基于深度学习的超像素分割和图像分割

## Superpixel Segmentation and Image Segmentation Based on Deep Learning

工程领域: 软件工程  
作者姓名: 王凯  
指导教师: 李亮  
企业导师: \*\*\*

天津大学智能与计算学部  
二零二零年十月



## 独创性声明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作和取得的研究成果，除了文中特别加以标注和致谢之处外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得 天津大学 或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

学位论文作者签名: 签字日期: 年 月 日

## 学位论文版权使用授权书

本学位论文作者完全了解 天津大学 有关保留、使用学位论文的规定。特授权 天津大学 可以将学位论文的全部或部分内容编入有关数据库进行检索，并采用影印、缩印或扫描等复制手段保存、汇编以供查阅和借阅。同意学校向国家有关部门或机构送交论文的复印件和磁盘。

(保密的学位论文在解密后适用本授权说明)

学位论文作者签名: 导师签名:

签字日期: 年 月 日 签字日期: 年 月 日



# 摘 要

图像分割和超像素分割已经被研究很多年，但仍然是计算机视觉中一个重要的课题，对于一些高级的图像处理领域具有重要意义，例如人脸识别，指纹识别，场景识别，行人检测，医学影像等。基于像素的传统图像分割方法取得了不错的成果，但是随着数码产品的拍照功能迅速发展，图像的构成越来越复杂，越来越清晰，像素数量也是成指数级增长。在这样的背景下，基于像素的传统图像分割方法处理分辨率高的图像，将花费更多的时间。超像素作为一种图像预处理技术解决了这个问题。超像素不仅有效减少了局部的冗余信息，后续处理过程中的计算量和复杂度大幅度降低，而且更利于局部特征的提取与表达，更有利帮助定位区域边界。本文将超像素分割和图像分割相结合，提出了两个新的算法。本文的主要工作如下：

(1) 本文提出了基于深度学习的超像素分割和图像分割神经网络，可以同时产生超像素和进行图像分割。其网络架构如图3-1所示。首先使用完全卷积网络和迭代可微聚类算法来获得超像素。接下来，采用超像素池层来获得超像素特征，并以此计算相邻超像素之间的相似度。如果相似度大于预先设定的阈值，则通过简单的步骤将其合并，得到目标片段。由于整个网络是端到端可训练的，因此可以很容易地组装到其他神经网络结构中，以备后续应用。我们的网络可以产生超像素并得到分割结果，与现有算法相比，该算法具有优良的性能和更高的精度。

(2) 本文提出了基于Boruvka算法和快速模糊C均值聚类的图像分割方法，其算法框架如图4-1所示。该方法首先使用一种基于Boruvka算法来产生超像素图像。在Boruvka算法获得的超像素图像的基础上，通过计算超像素图像的颜色直方图来实现快速模糊C均值聚类算法，实现彩色图像的快速分割。Boruvka算法具有线性时间解，可并行化。由于超像素图像中不同颜色的数目远小于原始彩色图像，相对于基于像素的分割方法更具高效性。

**关键词：** 图像分割，超像素，深度学习，Boruvka算法



# ABSTRACT

Image segmentation and superpixel generation have been studied for many years, but they are still important topics in computer vision, which is of great significance to some advanced image processing fields, such as face recognition, fingerprint recognition, traffic control systems, scene recognition, pedestrians Testing, medical imaging, etc. Traditional pixel-based image segmentation methods have achieved good results. However, with the rapid development of the camera function of digital products, the composition of images is becoming more and more complex, the resolution is constantly increasing, and the number of pixels is also increasing exponentially. In this context, traditional pixel-based image segmentation methods will take more time to process images with high resolution. Superpixel as a kind of image preprocessing technology solves this problem. Superpixels not only effectively reduce the local redundant information, and the amount of calculation and complexity in the subsequent processing are greatly reduced, but it is also more conducive to the extraction and expression of local features, and it is more conducive to help locate the boundary of the region. This paper combines the existing super pixel segmentation and image segmentation, and proposes two new algorithms. The main work of this paper is as follows:

(1)This paper proposes an end-to-end trainable network that can simultaneously generate superpixels and perform image segmentation. The network architecture is shown in Figure 3-1. First, a fully convolutional network and an iterative differentiable clustering algorithm are used to obtain superpixels. Next, the super pixel pool layer is used to obtain the super pixel characteristics, and the similarity between adjacent super pixels is calculated based on this. If the similarity is greater than the preset threshold, it is merged through simple steps to obtain the target segment. Since the entire network is end-to-end trainable, it can be easily assembled into other deep network structures for subsequent applications.

(2)This paper proposes an image segmentation method based on Boruvka algorithm and fast fuzzy C-means clustering. The algorithm framework is shown in Figure 4-1. The method first uses a Boruvka algorithm to generate super-pixel images. On the basis of the super pixel image obtained by the Boruvka algorithm, the fast fuzzy C-means clustering algorithm is realized by calculating the histogram of the superpixel image,

and the fast segmentation of the color image is realized. The Boruvka algorithm has a linear time solution and can be parallelized. Since the number of different colors in the superpixel image is much smaller than that of the original color image, it is more efficient than the pixel-based segmentation method.

**KEY WORDS:** Image segmentation, Superpixel, Deep learning, Boruvka algorithm

# 目 录

<b>摘要</b> .....	<b>I</b>
<b>ABSTRACT</b> .....	<b>III</b>
<b>第1章 绪论</b> .....	<b>1</b>
1.1 课题的研究背景 .....	1
1.2 国内外研究现状 .....	2
1.2.1 超像素分割算法研究现状 .....	2
1.2.2 图像分割研究现状 .....	4
1.2.3 利用超像素进行图像分割研究现状 .....	5
1.3 论文研究内容和结构安排 .....	6
1.3.1 论文研究内容 .....	6
1.3.2 论文结构安排 .....	6
<b>第2章 超像素分割和图像分割理论基础</b> .....	<b>9</b>
2.1 深度学习 .....	9
2.2 卷积神经网络 .....	11
2.3 基于卷积神经网络的图像分割 .....	13
2.4 超像素分割算法以及超像素在图像分割中的意义 .....	16
2.4.1 SLIC超像素分割 .....	16
2.4.2 基于图的超像素分割 .....	17
2.4.3 Kruskal算法VS Boruvka算法 .....	18
2.4.4 超像素在图像分割中的意义 .....	19
2.5 超像素池化层 .....	19
2.6 本章小结 .....	20
<b>第3章 基于深度学习的超像素分割和图像分割</b> .....	<b>21</b>
3.1 超像素生成 .....	21
3.2 超像素相似度 .....	22
3.3 损失函数 .....	23
3.3.1 像素和超像素表示之间的映射 .....	23
3.3.2 重建损失 .....	23
3.3.3 紧凑性损失 .....	24
3.3.4 相似性损失 .....	24

3.4 超像素融合 .....	24
3.5 网络结构 .....	25
3.6 实施细节 .....	26
3.7 本章小结 .....	26
<b>第4章 基于Boruvka算法和快速模糊C均值聚类的图像分割 .....</b>	<b>29</b>
4.1 基于Boruvka算法生成超像素 .....	29
4.2 基于超像素的快速模糊C均值聚类 .....	31
4.3 本章小结 .....	32
<b>第5章 实验结果和分析 .....</b>	<b>33</b>
5.1 实验数据集以及评估标准 .....	33
5.1.1 实验数据集 .....	33
5.1.2 评估标准 .....	33
5.2 消融实验 .....	37
5.2.1 基于深度学习的超像素分割和图像分割 .....	37
5.2.2 基于Boruvka算法和快速模糊C均值聚类的图像分割 .....	38
5.3 对比实验 .....	39
5.3.1 对比算法 .....	39
5.3.2 实验结果以及数据分析 .....	39
5.4 本章小结 .....	41
<b>第6章 总结与展望 .....</b>	<b>43</b>
6.1 总结 .....	43
6.2 展望 .....	44
<b>参考文献 .....</b>	<b>45</b>
<b>发表论文和参加科研情况说明 .....</b>	<b>51</b>
<b>致    谢 .....</b>	<b>53</b>

# 第1章 绪论

## 1.1 课题的研究背景

人每天接收到的信息里，百分之八十来自于视觉，而这些信息都是以图像的形式进入人的大脑，从而进行处理。可见，图像和人们的密切联系，它是人们认识世界和感受世界的主要媒介。

随着视觉计算和图形学的迅速发展，数字图像处理技术逐渐形成一个独立且完整的理论体系和实现架构。虽然作为一门跨学科的新领域，其发展时间不是很长，但相关技术和应用已经十分完善，尤其是与机器学习和深度学习的结合之后，越来越引起人们的注意。

计算机视觉研究中有不少经典难题，图像分割作为许多计算机视觉任务的关键组成部分便是其中一个。图像分割应用范围十分广泛，例如人脸识别<sup>[1-3]</sup>，指纹识别<sup>[4,5]</sup>，场景识别<sup>[6,7]</sup>，行人检测<sup>[8]</sup>，医学影像<sup>[9,10]</sup>等。

图像中每个元素都有各自的特征，如彩色、灰度、纹理等。图像分割就是根据像素特征的相似性，将图像分割成不同的区域。将图像分割成较大的感知区域，每个区域内的像素通常属于同一视觉对象，具有细微的特征差异。它已广泛应用于对象建议生成<sup>[11,12]</sup>、目标检测/识别<sup>[13,14]</sup>等领域。

与图像分割产生的大的感知区域不同，如图1-1所示，超像素分割对输入图像进行过度分割。它将图像分割成小的、规则的、紧凑的区域，这些区域通常由具有相似空间位置、纹理、颜色、亮度等的像素组成，同时保留了基于像素表示的显著特征。与图像分割相比，超像素通常具有很强的边界相干性，并且产生的分割易于控制。因为超像素在表示和计算方面的高效性，其已普遍作为输入或者辅助数据应用于相关领域。如语义分割<sup>[17-20]</sup>、物体检测<sup>[15,16]</sup>、目标跟踪<sup>[21,22]</sup>以及显著性估计<sup>[23,24]</sup>。

图像分割已被研究很多年，主要分为无监督图像分割和有监督图像分割两大类。无监督分割算法发展较早，已有许多成熟的算法，如聚类<sup>[25,26]</sup>、图割<sup>[27,28]</sup>、分水岭变换<sup>[29]</sup>、隐马尔可夫随机场<sup>[30]</sup>等。这种算法只需要很少的训练样本和标签图像。另一方面，FCN<sup>[31]</sup>、U-NET<sup>[32]</sup>、HFS<sup>[33]</sup>等有监督算法，利用CNN网络<sup>[34]</sup>进行特征学习，实现高效准确的图像分割。有监督学习对数据集的特征进行了更好的编码，使得分割更加精确。

尽管近年来深度学习在计算机视觉中应用更加广泛，但是现有超像

素算法是不可微的，如simple linear iterative clustering（SLIC）<sup>[35]</sup>, watershed-superpixel<sup>[36]</sup>。并且标准卷积运算通常在规则网格上定义，当在不规则超像素点阵上操作时变得低效。因此神经网络很少使用超像素。

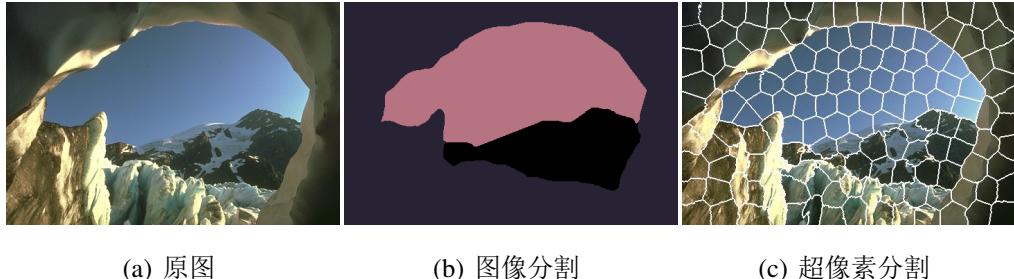


图 1-1 图像分割与超像素分割示例

## 1.2 国内外研究现状

### 1.2.1 超像素分割算法研究现状

自从Ren和Malik在2003年提出超像素<sup>[37]</sup>的概念以来，研究人员对这一领域做出了很多贡献，并取得了丰富的成果。本小结对于超像素算法进行简单整理和总结。

#### 1. 基于图论法

基于图论法的基本思想是将图像中的像素映射成加权无向图，无向图中的一个顶点代表了图像中的一个像素，点与点之间边的权重由像素之间的特征差异性计算得来。基于图论法就是将加权无向图根据一定的准则划分为数个连通子树，从而得到超像素分割结果。

设 $G = (V, E)$ 表示由 $n$ 个顶点 $v \in V$ 和 $m$ 个边 $e \in E \subseteq V \times V$ 组成的无向图，每个像素与一个顶点相关联。每个边 $e_{ij} = (v_i, v_j)$ 都被分配了一个权重（通常为非负实值），该权重度量两个顶点之间的差异。在超像素分割任务中， $k$ 表示要提取的超像素个数。超像素分割是将图 $G$ 划分为 $k$ 个不相交的分量，每个分量对应一个连通子图 $G' = (V', E')$ ，其中 $V' \subseteq V$ 和 $E' \subseteq E$ 。

Ncut<sup>[38]</sup>方法自从提出之后，已经发展了多个版本。它将图像映射到加权无向图中，边的权重由纹理特征和轮廓特征差异计算而来。并定义了目标函数，通过最小化目标函数，使得分割出的超像素类间距离尽可能大，并且类内距离尽可能小，从而实现超像素分割。虽然它生成的超像素比较规则，但边界的保持效果并不理想，且计算量较大，速度较慢。

2017年，Chen等提出的实现了线性谱聚类( Linear Spectral Clustering, LSC)<sup>[39]</sup>，通过计算像素之间的颜色相似性和空间接近性，利用内核相似度

度量函数, 将图像像素巧妙的映射到高维特征空间中, 通过特征空间的加权K-means<sup>[40]</sup> 度量优化目标函数, 实现低计算量且致密均匀的超像素, 大大提升了超像素分割的效率。

2018年, Xing等人提出Superpixel Hierarchy<sup>[41]</sup>, 利用Boruvka<sup>[42]</sup>算法实现无向图的区域合并。把一个图看成有 $n$ 个树的森林 ( $n$ 为图像中像素数目), 每个顶点看成一棵树。对于每棵树来说, 找到它最近的邻点, 即边的权重最小的那个点, 并把他们合并在一起。Boruvka算法重复融合树直到只有一棵树留下来。用Boruvka算法生成一个最小生成树 (MST), 同时记录每个边缘的顺序添加到MST。一旦边缘被添加到MST, 森林中的树木数量减少一个。假设需要提取 $k$ 个超像素, 通过前 $n - k$ 个边连接顶点, 并且具有 $k$ 个连接的组件, 这些组件恰好是超像素。

其优点是具有线性时间解, 可并行化。此外, 该方法在聚类的过程中融入了局部信息, 比SLIC和LSC等只利用每像素特征来确定聚类隶属关系的方法更为健壮。

## 2. 基于梯度下降法

简单线性迭代聚类 (simple linear iterative clustering, SLIC) 作为最经典的超像素分割算法, 具有很多优点, 计算简单且速度快, 可以生成规则的超像素, 此外可以控制超像素数量。但是其生成的超像素边界的保持效果并不理想, 而且抗噪性较差。SLIC将彩色图像转换为CIELAB颜色空间和XY坐标系下的5维特征向量, 然后构造5维特征向量的距离度量来对像素进行局部聚类。这个想法简单易行。与其他超像素分割方法相比, SLIC 算法在速度、紧凑性等方面都是理想的。

zhao等人在SLIC的基础上进行了改进, 提出了像素快速线性迭代聚类(Fast Linear Iterative Clustering, FLIC)<sup>[43]</sup>算法。FLIC从一个新的角度重新考虑图像的超像素分割问题, 利用先验信息, 提出了一种名为“Active Search”的新型搜索算法, 它明确地考虑了超像素的连通性。基于这种搜索方法, 设计了一个back-and-forth的遍历策略并且合并了SLIC中的分配和更新步骤。与SLIC相比, FLIC减少了收敛所需的迭代次数, 并且提高了超像素分割的边界灵敏度, 具有更好的边缘重合度。

## 3. 基于深度学习的超像素分割算法

近年来, 对于广泛的计算机视觉问题采用深度学习的情况急剧增加, 但超像素几乎不与深度学习和神经网络结合使用。这有两个主要原因。首先, 标准卷积运算通常定义在规则网格上, 当在不规则超像素网格上操作时变得不合适。其次, 现有的超像素算法是不可微的, 因此如果在神经网络中使用超像素, 就会在其他端到端可训练网络架构中引入了不可微的模块。

2018年，Varun Jampani等人提出了一种基于深度学习的聚类算法Superpixel Sampling Networks（SSN）<sup>[44]</sup>，通过放宽SLIC中存在的最近邻约束将其转化为可微算法。这种新的可微算法允许端到端训练，从而能够利用强大的深层网络来学习超像素。其优点包括：端到端可训练，可以轻松集成到其他深层网络架构中；速度快，可以生成边界保持度高的超像素。

### 1.2.2 图像分割研究现状

#### (1) 特征空间聚类法

图像分割根据灰度、彩色、空间纹理、几何形状等特征把图像划分成若干个互不相交的区域，使得这些特征在同一区域内表现出一致性或相似性，而在不同区域间表现出明显的不同。因此，可以将图像分割问题以像素的分类问题来解决。同类的像素在灰度、彩色、空间纹理等方面具有较高的相似性，而不同类中的像素间差异比较大。

特征空间聚类法作为无监督的图像分割算法，可以减少人为干预，自动完成分割。但一般需要提前确定分类数，然后通过迭代地来提取各类的特征值，来执行分类算法，更新聚类中心，例如K-均值<sup>[45]</sup>，Fuzzy C-means (FCM)<sup>[46-48]</sup>聚类算法等。

特征空间聚类作为已经很成熟的算法，有很多优点，例如：不需要训练样本，方法简单，方便执行。也有相应的缺点：分类数一般很难确定，且初始参数对最终结果有较大的影响。此外，特征空间聚类没有充分利用像素之间的局部信息，一般只是采用空间或颜色特征，从而对噪声比较敏感。

#### (2) 边缘检测法

所谓的图像边缘，即在一张图像中，两个不同区域的交界处，图像中相邻区域交界处的像素结果组成了图像的边缘。根据经验，沿着边缘走向，像素值变化较为平缓；而垂直于边缘，像素值变化较大。因此，可以将图像中灰度发生突变的像素看为边缘。

边缘检测的基本思路一般为：先根据一定的算法来确定图像中的边缘像素集合，目前边缘检测方法来得到边缘像素集合的方法，最实用也最简单方法就是构造边缘算子。采用何种边缘算子来提取图像边缘是边缘检测方法的核心问题。常见的边缘算子有Roberts，Sobel，PreWitt和Carry等。边缘检测得到的结果还需要进一步处理，如边缘跟踪，边缘松弛法，将边缘像素连接成图像轮廓，得到图像分割结果。

对于噪声比较小的图像，即图像中的不同区域差别明显，边缘检测方法可以取得较好的结果。若图像比较复杂，且不同区域间的差别不是特别明显，则会产生很多噪声。其难点在于确定边缘时，抗噪性和精确度的矛盾。若抗噪性提高，

就会产生位置偏差或轮廓漏检。若提高边缘精确度，那么噪声就会产生不合理的伪边缘。

### (3) 基于区域的方法

类似于边缘检测法，区域分割法利用了图像中局部区域的一致性，直接按照一定的一致性判断来寻找分区，分区内的像素具有相似的性质，最终得到图像分割结果。其中的一致性判断包括：灰度、色彩、纹理、形状等。基于区域的图像分割大致可以分为两大类：区域生长法，区域分裂与合并。

区域生长法的关键核心在于选取或制定合理的生长准则，按照一定的生长准则将像素或子区域合并成更大的区域。生长准则可以按照不用的判断标准来制定，基本使用图像的局部信息来制定，如基于灰度级类似准则，基于颜色相似准则，基于纹理相似准则等。不同的生长准则会影响分割过程，从而对最终的分割结果造成影响。区域生长法最主要的优点就是计算简单，适合于分割小的结构。但其对噪声敏感，抗噪性弱，导致分区中有空洞。

区域分裂与合并算法的基本思路类似于微分，即无穷分割，然后将分割后满足相似度准则的区域进行合并。四叉树分解法作为典型的区域分裂合并，应用广泛。我们以灰度级作为分裂合并准则，则基本的分裂与合并算法为：首先对于图像中灰度级不同的区域，均分为4个子区域；若相邻的子区域所有像素的灰度级相同，则将其合并；重复前面两个步骤，直到不再有新的分裂与合并为止，从而得到最终的分割结果。

区域生长算法和区域分裂合并算法作为基于区域的分割方法，在实际应用中经常结合使用，以取得更好的结果。该类算法对某些复杂物体定义的复杂场景的分割或者对某些自然景物的分割等类似先验知识不足的图像分割，效果较为理想。

### 1.2.3 利用超像素进行图像分割研究现状

通过学习像素或超像素的相似性进行分割也是一种趋势<sup>[49]</sup>。Ahn和Kwak<sup>[50]</sup>提出了仿射网来预测一对相邻图像坐标之间的语义一致性。利用仿射网预测的邻接词，通过随机游动实现语义传播。基于超像素描述子向量之间的距离度量来计算超像素相似度，Chaibou等人<sup>[51]</sup>引入了一种新的超像素上下文描述器来增强学习特征，以更好地进行相似性预测。然后通过迭代合并使用相似性加权目标函数选择的最相似的超像素对来实现图像分割。

超像素池网络（Superpixel Pooling Network, SPN）<sup>[52]</sup>提出的超像素池化操作为超像素特征提取提供了新思路。SPN利用输入图像的超像素分割作为一个池化布局来反映底层图像结构，用于学习和推断语义分割。

deep embedding learning (DEL)<sup>[53]</sup>算法利用超像素池运算提取超像素特征

进行图像分割，取得了良好的效果。在提取的超像素的基础上，利用超像素池运算提取超像素的特征来计算超像素的相似度。根据相似度来判断超像素是否进行合并，从而得到最终的分割结果。

2018年，Tao Lei等人提出了superpixel-based fast FCM（SFFCM）<sup>[53]</sup>，其算法通过多尺度形态梯度重建（multiscale morphological gradient reconstruction, MMGR）和分水岭变换（watershed transform, WT）获得超像素，然后利用基于超像素的SFFCM进行图像分割。与其他聚类算法相比，该算法耗时更少。然而，聚类中心的初始值难以确定，不能得到广泛的应用。

### 1.3 论文研究内容和结构安排

#### 1.3.1 论文研究内容

本文的主要研究内容是计算机视觉中两个主要的方向-超像素分割，图像分割。虽然现在有很多优秀且成熟的算法来实现超像素分割或者图像分割，但也存在一些问题：首先现在的神经网络没有将超像素分割和图像分割两个任务有效的结合起来，只有部分算法将现有的超像素作为图像分割的输入进行计算。其次，现有的超像素算法基本还是以传统算法为主，并没有真正与神经网络结合起来，发挥神经网络的优势。此外，图像分割算法还是基本以像素为单位进行运算，并没有引入超像素进入有效的计算。

针对此问题，本文提出了端到端的可训练网络，可以同时产生超像素和进行图像分割。使用完全卷积网络和迭代可微聚类算法来获得超像素。接下来，采用超像素池层来获得超像素特征，并以此计算相邻超像素之间的相似度。如果相似度大于预先设定的阈值，则通过简单的步骤将其合并，得到目标片段。本文在使用BSDS500数据集<sup>[54]</sup>上，最先进的结果进行多次使用比较，结果验证了本文方法的有效性和可行性，实现了精细的超像素分割和图像分割。

此外，本文还提出了基于Boruvka算法和快速模糊C均值聚类的图像分割方法。该方法首先使用一种基于Boruvka算法来产生超像素图像。在Boruvka算法获得的超像素图像的基础上，通过计算超像素图像的颜色直方图来实现快速模糊C均值聚类算法，实现彩色图像的快速分割。

#### 1.3.2 论文结构安排

本文结构安排如下：

第1章为绪论，介绍了超像素和图像分割的基本概念，并分类介绍了超像素分割和图像分割的经典方法，最后简要介绍了本文。

第2章首先简介了深度学习和神经网络理论，然后详细介绍了卷积神经网络基本理论，此外详细介绍了像素分割和图像分割的经典方法，最后介绍了超像素池化方法。

第3章首先回顾了经典的SLIC方法的核心步骤，然后介绍了基于深度学习的超像素分割和图像分割方法。

第4章基于Boruvka算法和快速模糊C均值聚类的图像分割方法。

第5章首先说明本文实验中的公开数据集及评定标准，然后基于第3章和第4章的方法进行消融实验，最后对本文介绍的算法与已有的先进算法进行对比实验并分析结果。

第6章对全文进行总结，并在本文工作的基础上，对进一步的工作进行了展望。



## 第2章 超像素分割和图像分割理论基础

本文第1章介绍了超像素和图像分割的基本概念，并分类介绍了超像素分割和图像分割的研究背景及研究现状。在本章中将对相关的基本理论进行简要的介绍。首先简要叙述深度学习和神经网络的相关理论研究，接着介绍经典的基于神经网络的图像分割方法。然后简述超像素的两个经典算法以及超像素在图像分割的意义。此外，介绍了将超像素整合到神经网络中的理论基础-超像素池化层。

### 2.1 深度学习

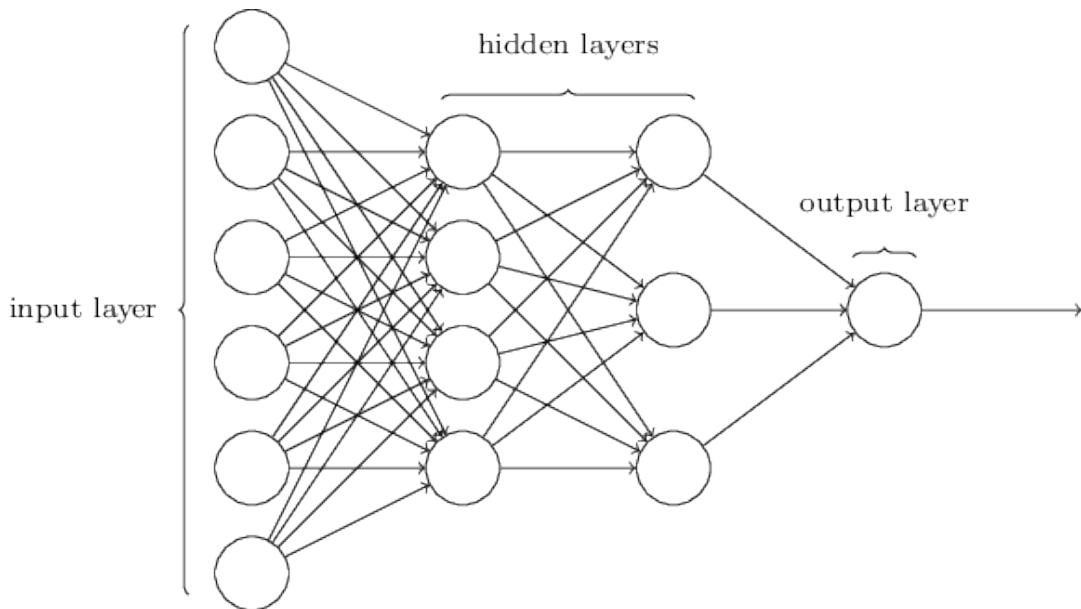


图 2-1 多层感知器示意图

2016年3月，在中央发表的”十三五”规划纲要中，“人工智能”一词格外引人注目。BAT等各大互联网公司对人工智能领域的投入更是将人工智能这一领域的发展推向一个新的高潮。目前深度学习<sup>[55]</sup>无疑是人工智能的重点研究领域之一，设计到人工智能众多领域，如计算机视觉、自然语言处理等。

深度学习其实是机器学习的一种，从最初的浅层机器学习发展到目前深度学习。与浅层机器学习相比，深度学习加深了模型结构深度，一般有5层，6层，甚至更多的隐层节点。一般而言，随着深度的增加，模型的学习能力也在增加。此外，深度学习是一种特征学习，能够利用大数据来学习特征，从而能够获取数

据更高层次的抽象表示，来描述数据的内在信息。

深度学习的概念源于人工神经网络的研究，深度学习网络最基本的结构是多层感知器(multilayer perception, MLP)。多层感知器，也就是我们所说的前向传播网络，一般有多层构成，每一层由若干神经元组成，如图2-1所示。

如图2-2所示，对于第*k*层，多层感知器的前向传播公式如下：

$$y_i^{k+1} = \sum_j W_k^{ij} y_j^k + b_i^k \quad (2-1)$$

其中， $y_j^k$ 表示第*k*层的第*j*个神经元的输出， $W_k^{ij}$ 表示第*k*层中第*j*个神经元与第*k+1*层的第*i*个神经元之间的权重， $b_i^k$ 是偏置值。通常进行完这一步之后，会在此基础上使用非线性激活函数进行激活运算，常见的激活函数有ReLU、Sigmoid等。

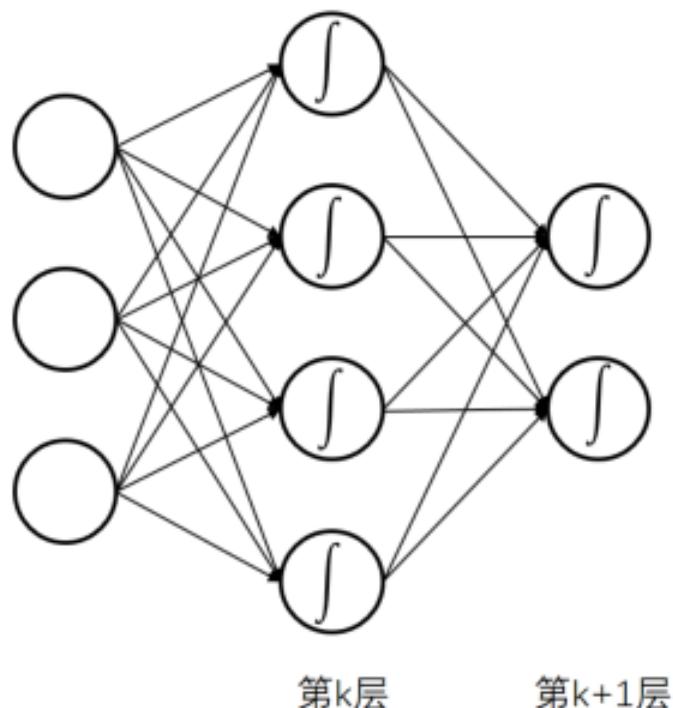


图 2-2 前向传播示意图

由于理论和计算力的原因，深度学习从提出到本世纪出，深度学习一度被边缘化，成果泛善可陈，直到2012年ImageNet比赛，多伦多大学团队开发的AlexNet<sup>[56]</sup>模型的出现，以卷积神经网络(Convolutional Neural Network, CNN)<sup>[34]</sup>为基础，识别效果超过了所有浅层的方法。从此开启了深度学习进入快速发展的新时期，已经被广泛应用到计算机视觉、语言识别、自然语言处理等领域。

2015年的世界大规模人脸识别竞赛LFW中，来自香港中文大学多媒体实验室的中国团队使用深度学习模型，打败FaceBook团队获得冠军。使得在人脸识

别领域，深度学习的识别能力穿越真人。此外在计算机视觉领域中的行人检测、多人姿态估计、物体跟踪、场景识别、图像分类、图像分割、物体跟踪等都有深度学习的身影。近年来，科研人员也将深度学习应用到很多有意思的方向，例如：去除马赛克、风格迁移、黑白照片自动上色、图片补全等。

深度学习虽然很早应用于计算机视觉，但在语音处理领域最先取得突破性进展。语音处理主要分为语音识别和语音合成两大方向。2016年，微软利用深度学习开发的语言识别模型，在日常对话识别准确度上首先达到了人类水平，真正让大家大吃一惊。此外，各大公司也都在研究利用深度学习来进行语言合成，并已有成熟的系统，例如谷歌的WaveNet模型，百度的Deep Voice3。

2013年，Tomas Mikolov等人发表论文《Efficient Estimation of Word Representations in Vector Space》<sup>[57]</sup>，提出了word2vector模型，这也是目前自然语言处理通常使用的模型，与传统的词袋模型（bag of words）相比，word2vector能够更好地表达语法信息。目前深度学习在自然语言处理领域的应用主要包括：问答系统、情感分析、机器翻译、句子成分分析等。

## 2.2 卷积神经网络

简单来说，卷积神经网络（Convolutional Neural Network，CNN）是深度学习模型中的其中一类，类似于人工神经网络的多层感知器。Yann LeCun提出的LeNet<sup>[58]</sup>模型，最早将CNN应用于数字识别。2012年，多伦多大学Alex Krizhevsky等人开发的AlexNet<sup>[56]</sup>模型第一次让大家注意到了CNN的强大之处。

LeNet模型只有卷积层和池化层，全连接层，后来AlexNet模型在此基础上，加入了Relu激励层，提高了效率。接下来会分别介绍相关理论和计算。

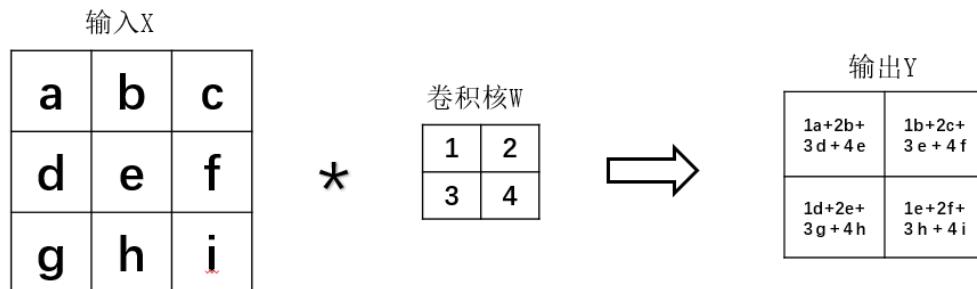


图 2-3 卷积计算过程示意图

卷积层作为卷积神经网络最重要的一层，其作用主要是提取图像的局部特

征。如图2-3所示，卷积层的卷积操作为卷积核W与输入矩阵X进行从左到右从上到下，步长为1的相乘相加操作，得出输出Y。卷积核W的大小为 $M_w \times N_w$ ，输入矩阵的大小为 $M_x \times N_x$ ，那么输出矩阵的大小等于 $(M_x - M_w + 1) \times (N_x - N_w + 1)$ 。以图2-4为例，一个 $3 \times 3$ 的卷积核在一个 $5 \times 5$ 的输入图像上以步长为1做卷积计算，得到 $3 \times 3$ 的输出矩阵，这个输出矩阵就是特征矩阵。

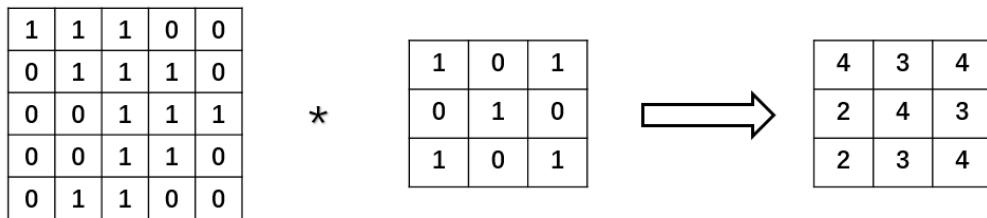


图 2-4 卷积操作示意图

最原始的感知机（perceptron）并没有激活层，无论有多少层神经网络，输出层的结都是输入数据的线性组合，不能满足解决神经网络复杂任务的需求。因而加入了激活层，使用非线性函数作为激活函数，使得深层神经网络有了学习的能力，可以逼近任何函数。

用sigmoid函数和tanh函数作为最初的激活函数，输出有界，很容易充当下一层的输入。随着神经网络的深度增加，卷积神经网络在反向传播过程中存在梯度消失的问题，Sigmoid等激活函数的导数很小，而连续多个很小的数相乘，结果几乎为0，因此梯度无法从输出层传到输入层。随着神经网络的深度增加会造成梯度消失的问题，从而导致训练难度大，效果不佳。此外，反向传播求梯度时，Sigmoid函数计算量较大。

2012年提出的AlexNet引入了一种新的激活函数- ReLU函数。该函数的提出很大程度的解决了深层神经网络的梯度耗散问题，而且减少了计算量。此外，ReLU会使一部分神经元的输出为0，这样就造成了网络的稀疏性，并且减少了参数的相互依存关系，缓解了过拟合问题的发生。

现在也有一些对ReLU的改进，比如prelu，random ReLU等，对于不同的任务上，准确率或速度上有一定的改进。此外，现在卷积神经网络，一般会在ReLU激活层之后会多做一步归一化操作，尽可能保证每一层网络的输入具有相同的分布，减少计算量。

在连续的卷积层之间一般会放入池化层，其目的是压缩数据，降低维度，减少计算。池化层用的方法包括最大池化（Max Pooling）和平均池化（Average Pooling）。以最大池化为例，如图2-5所示，在一个 $4 \times 4$ 的矩阵上，选

用 $2 \times 2$ 的filters，步长为2，对于每个 $2 \times 2$ 的窗口选出最大的数作为输出矩阵的相应元素的值，比如输入矩阵第一个 $2 \times 2$ 窗口中最大的数是6，那么输出矩阵的第一个元素就是6，如此类推。平均池化的原理类似，只不过输出矩阵中的元素是输入矩阵对应窗口中元素的平均值。

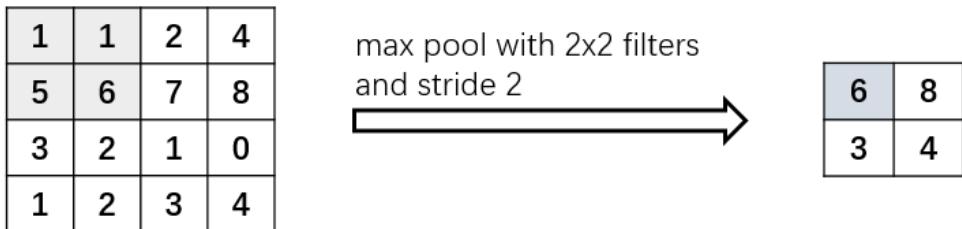


图 2-5 池化操作示意图

接下来具体介绍池化层的具体作用。如果输入数据是一张图片，那么池化层的作用就是对图像进行压缩。压缩过程中，去掉一些无关紧要的特征信息，留下最能体现图像特征的信息。我们知道，一张图像中含有的很多的信息，也具有很多的特征，但是有些特征信息，对于需要完成的任务则显得无关紧要，过于冗余。池化操作就是要去除这些冗余信息，留下最重要的特征信息。

池化操作具有特征不变性。所谓的特征不变性就是一张狗的图像被缩小了一倍，但还能认出这是一张狗的照片。这说明在压缩过程中只是去除了无关紧要的信息，但对狗的重要特征信息仍有保留，因此我们依然可以判断出这是一只狗。

批归一化（Batch Normalization, BN）<sup>[59]</sup>是深度学习发展中的一个里程碑式的技术，使各种网络能够进行训练。在卷积神经网络训练过程中，由于参数是不断更新的，随着网络层数的加深，对于深层网络，其输入数值不稳定，导致模型很难收敛。通过归一化，对中间层的输出结果进行标准化处理，使得其更加稳定，分布更加固定，有利于算法的稳定和收敛。

然而，沿着批处理维度进行规范化会带来问题——当批处理大小变小时，BN的错误会迅速增加，这是由于批次统计估计不准确造成的。2018年，Yuxin Wu提出了组归一化（Group Normalization, GN）<sup>[60]</sup>，GN将通道分成组，并计算每组中的均值和方差进行归一化。GN的计算与批量大小无关，更适合于批次较小的任务。

### 2.3 基于卷积神经网络的图像分割

上文介绍了卷积神经网络的概念以及相关理论基础，CNN相对传统算法而

言，其最大的优点在于通过构造神经网络，可以自动学习到不同层次的特征。浅层网络具有较小的感受野，可以学习到包含更多细节信息的局部特征。深层网络具有更大的感受野，学习到的特征更加抽象，包含更多的全局特征。这些全局特性相对来说更加笼统，对物体的颜色、大小、位置等信息敏感性更低，对于分类更有帮助，可能更加方便的辨别出物体的类别。但是由于缺失了细节信息，不能精确的给出图像中的像素属于哪个物体，因而给不出物体的精确轮廓，做不到准备的分割效果。

传统基于卷积神经网络的方法为了确定某个像素所属的类别，从而实现图像分割，一般将该像素周围的像素块作为卷积神经网络的输入来进行训练。这样造成了几个缺点：一是增加了存储开销，例如对于每个像素分类任务，采用的像素块为 $15 \times 15$ ，其需要原来的225倍的存储空间。二是对在相邻像素块进行运算，由于相邻像素块的重叠，其大部分运算是重复的，无用的，导致计算效率降低。三是像素块相对于整个图像而言小很多，在卷积神经网络训练中，对像素块进行卷积提取特征，只能提取到局部信息，导致分类的性能有所限制。

Long等人提出了全卷积网络（Fully Convolutional Network，FCN）<sup>[31]</sup>解决了以上问题，成功应用于图像分割。与经典的CNN在卷积层使用全连接层得到固定长度的特征向量进行分类不同，FCN可以接受任意尺寸的输入图像，采用反卷积层对最后一个卷积层的特征图进行上采样，使它恢复到与输入图像相同的尺寸，从而可以对每一个像素都产生一个预测，同时保留了原始输入图像中的空间信息，最后基于在上采样的特征图进行像素的分类。

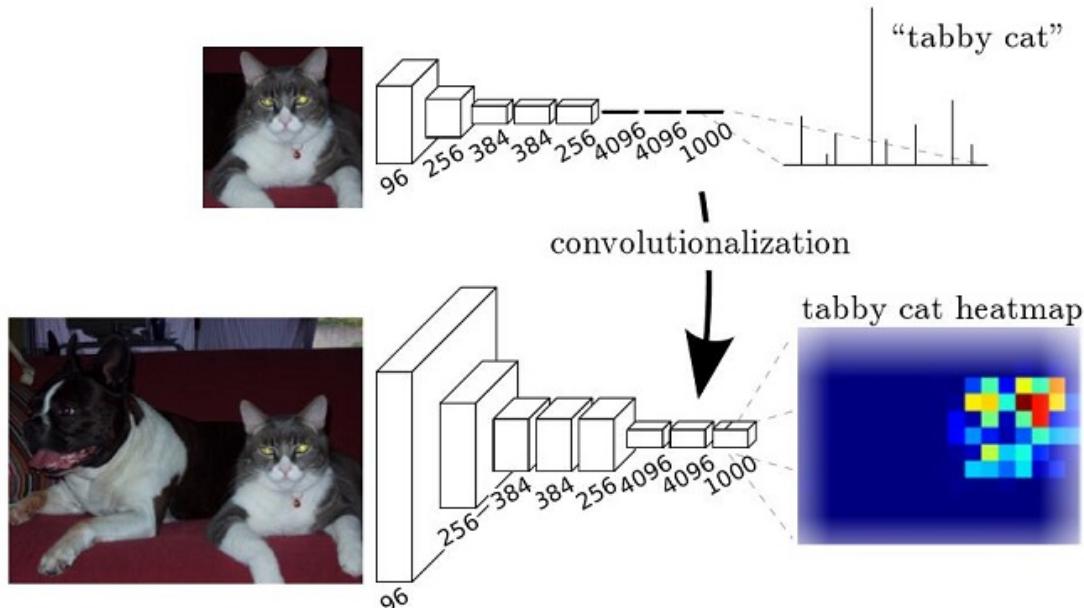


图 2-6 FCN网络流程图

如图2-6所示，FCN将传统卷积神经网络中最后三层的全连接层全都换成卷积层。传统卷积神经网络中，经过最后三层的全连接层分别会得到长度为4096、4096、1000的一维向量。FCN使用卷积核大小分别为(4096, 1, 1)、(4096, 1, 1)、(1000, 1, 1)的卷积层替换全连接层。

经过前五层的卷积和池化操作，图像大小变成了原图像的1/2、1/4、1/8、1/16、1/32，分辨率逐渐降低。为了恢复图像的分辨率，FCN采用了反卷积进行上采样操作，例如对第五层的结果进行反卷积，放大32倍，恢复到原图大小。

但由于深层特征缺失细节信息，得到的结果并不准确。FCN创造性地采用了跳层连接操作。如图2-7所示，第二层卷积池化之后，图像变为原图大小的1/4，从第三层卷积池化之后，保存池化后的输出。经过五层的卷积池化，分别得到原图的1/8、1/16、1/32大小的特征图。然后对第五层输出(即1/32尺寸的特征图)进行反卷积，上采样之后结果和第四层输出进入相加求和，从而补充细节信息，得到16倍上采样结果。同样，其结果继续和第三层输出进行求和相加，继续补充细节信息，得到8倍上采样结果。最终使得结果越来越精确。

FCN可以输入任意大小的图像，不再像以往的卷积神经网络对训练图像和测试图像的大小有限制。虽然经过跳层连接，进行了补充细节信息，但是得到的结果不是很精确。此外，没有充分考虑像素与像素之间的关系，忽略了局部信息的关联性。

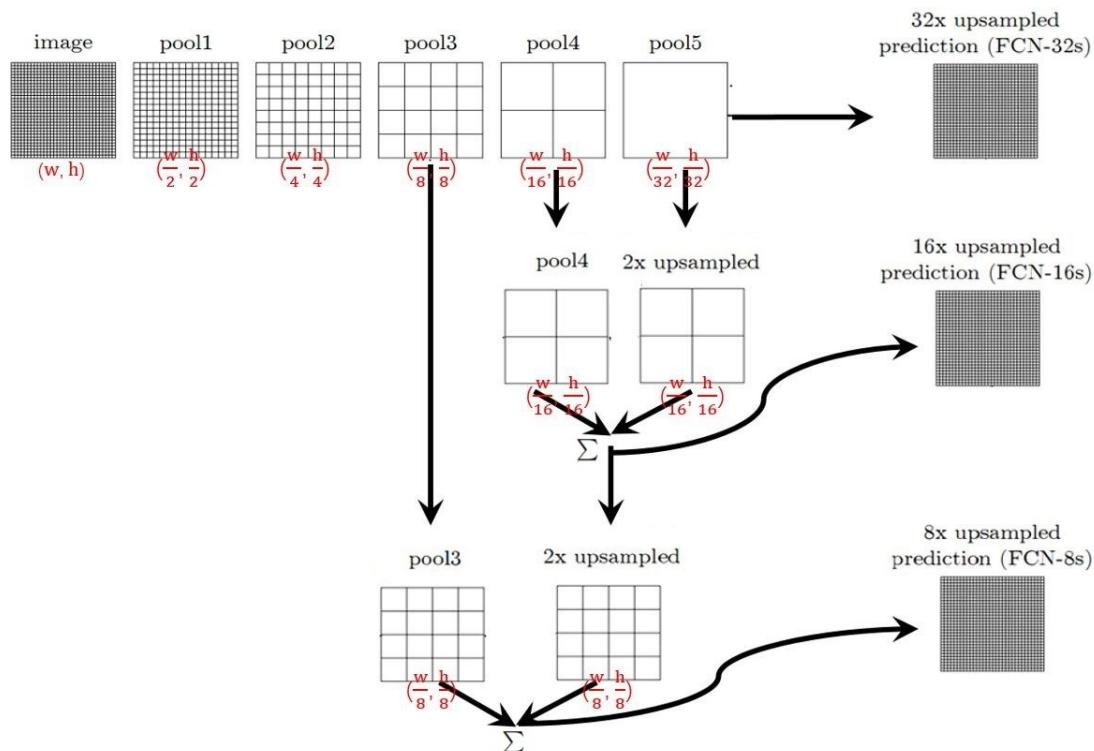


图 2-7 跳层连接结构示意图

## 2.4 超像素分割算法以及超像素在图像分割中的意义

### 2.4.1 SLIC超像素分割

SLIC作为最经典的超像素分割算法，具有很多优点，计算简单且速度快，可以生成规则的超像素，此外可以控制超像素数量。但是其生成的超像素边界的保持效果并不理想，而且抗噪性较差。SLIC将彩色图像转换为CIELAB颜色空间和XY坐标系下的5维特征向量，然后构造5维特征向量的距离度量来对像素进行局部聚类。这个想法简单易行。与其他超像素分割方法相比，SLIC算法在速度、紧凑性等方面都是理想的。

其具体实现步骤包括：

1. 初始化种子中心：通过设置超像素的个数，实现在图像内均匀分配种子点。例如一张图像具有 $N$ 个像素，划分为 $M$ 个超像素，则每个超像素中像素数目为 $N/M$ ，相邻种子点的距离为 $S = \sqrt{N/M}$ 。

2. 局部区域重新确定种子中心：为了避免初始种子中心落在区域边缘上，从而影响聚类结果。需要在初始种子点的 $n \times n$ 邻域（一般设为 $n=3$ ），计算所有像素点的梯度，选择梯度值最小的点作为种子点。

3. 为像素点分配种子标签：对于每个像素点，在 $2S \times 2S$ 的区域内搜索种子点，进行距离度量，选取最小值对应的种子点作为该像素的聚类中心，从而来分配种子标签，得到 $K$ 个超像素。距离度量由颜色距离和空间距离构成。

颜色距离计算公式为：

$$d_c = \sqrt{(l_j - l_i)^2 + (a_j - a_i)^2 + (b_j - b_i)^2} \quad (2-2)$$

空间距离计算公式为：

$$d_s = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2} \quad (2-3)$$

距离度量公式为：

$$D = \sqrt{\left(\frac{d_c}{N_c}\right)^2 + \left(\frac{d_s}{N_s}\right)^2} \quad (2-4)$$

其中 $N_c$ 代表最大颜色距离，一般设为常数。 $N_s = S = \sqrt{N/M}$ 。

4. 更新聚类中心：计算这 $K$ 个聚簇里所有像素点的平均向量值，重新得到 $K$ 个聚类中心。然后以这 $K$ 个中心，再计算与其周围 $2S \times 2S$ 区域内的像素距离度量，从而分配新的种子标签。更新聚类中心，再次迭代，如此反复直到收敛。

5. 增加连通性：避免出现超像素过小、孤立点的情况。

### 2.4.2 基于图的超像素分割

基于图像法的基本思想是将图像中的像素映射成加权无向图，无向图中的一个顶点代表了图像中的一个像素，点与点之间边的权重由像素之间的特性差异性计算得来。基于图论法就是将加权无向图根据一定的准则划分为数个连通子树，从而得到超像素分割结果。

Felzenszwalb和Huttenlocher提出了一种基于图的快速高效分割方法FH<sup>[61]</sup>，将像素作为无向图的顶点，利用边缘权重用于测量顶点之间的不相似性。类似于其他区域合并方法，它使用Kruskal<sup>[42]</sup>算法来构建最小生成树，其中每棵树都是一个超像素区域。每个顶点最初都放置在其自己的组件中，并且FH方法通过这样的准则合并区域，即所产生的分割既不会太粗糙也不会太精细。FH方法基于图像相邻区域中的可变性程度来自适应地调整分割标准，从而即使做出贪婪的决策，它也服从某些全局属性。

其思路是将图像映射到无向图 $G = (V, E)$ 。 $G = (V, E)$ 表示由 $n$ 个顶点 $v \in V$ 和 $m$ 个边 $e \in E \subseteq V \times V$ 。每个像素与一个顶点相关联，并与它的4个邻域局部连接。每个边 $e_{ij} = (v_i, v_j)$ 都被分配了一个权重（通常为非负实值），该权重度量两个顶点 $v_i$ 和 $v_j$ 之间的差异。分割区域 $S$ 是将图 $G$ 划分为数个不相交的分量，每个小区域 $C \in S$ 对应一个连通子图 $G' = (V', E')$ ，其中 $V' \subseteq V$ 和 $E' \subseteq E$ 。

该方法将区域 $C$ 对应的最小生成树MST( $C, E$ )中的最大权重作为区域 $C$ 的内部差异，其计算公式为：

$$Int(C) = \max_{e \in MST(C, E)} W(e) \quad (2-5)$$

将连接两个区域的最小权重边作为它们之间的区域差异性，其表示为：

$$Dif(C_1, C_2) = \min_{v_i \in C_1, v_j \in C_2, (v_i, v_j) \in E} W((v_i, v_j)) \quad (2-6)$$

通过比较两个区域的区域差异性和它们的内部差异来判断是否进行合并。若区域差异性大于它们之间任意一个的内部差异，则不能进行合并，可以用下面公式来表明：

$$D(C_1, C_2) = \begin{cases} \text{false} & \text{if } Dif(C_1, C_2) > MInt(C_1, C_2) \\ \text{true} & \text{otherwise} \end{cases} \quad (2-7)$$

其中， $MInt(C_1, C_2) = \min(Int(C_1) + \tau(C_1), Int(C_2) + \tau(C_2))$ 。通过 $\tau(C) = \frac{k}{|C|}$ ，来控制两个区域的差异性，保证区域差异性一定大于两个区域的内部差异性。 $k$ 为设定的常数， $|C|$ 为区域 $C$ 中顶点的个数，即像素的个数。

2018年，Xing Wei等人提出Superpixel Hierarchy，利用Boruvka算法实现无向图的区域合并。本文在第4章详细介绍了过程。其优点是具有线性时间解，可并行化。此外，该方法在聚类的过程中融入了局部信息，比SLIC和LSC等只利用每像素特征来确定聚类隶属关系的方法更为健壮。

接下来，本文详细比较了Boruvka算法和Kruskal算法各自的特性以及优缺点。

### 2.4.3 Kruskal算法VS Boruvka算法

基于图论的分割算法，其基本思路是将图像映射到加权无向图中，利用最小生成树算法对无向图进行分割。最经典也最常用的算法有Kruskal算法和Boruvka算法。

Kruskal算法构造最小生成树的主要思想是：假设无向图 $G = (V, E)$ 由 $n$ 个顶点 $v \in V$ 和 $m$ 个边 $e \in E \subseteq V \times V$ 组成。初始状态下，每个顶点可以看成一个连通分量。选取 $E$ 中权值最小的边 $e$ ，若边 $e$ 的两个顶点属于同一个连通分量，则舍去继续寻找并判断下一条权重最小的边。若 $e$ 的两个顶点属于不同的连通分量，则利用边 $e$ 将两个连通分量合并在一起。以此类推，经过数次迭代，直达所有顶点在同一连通分量，这个连通分量便是无向图 $G$ 的最小生成子树。

Kruskal算法最重要的是在构建连通分量过程中考虑是否会出现环的情况。当选取了权重最小的边，若这条边连接了不同的连通分量，说明这两个连通分量加入这条边之后，不会构成环路，从而可以将这个边连接的连通分量合并在一起。如这条边的两个顶点在同一个连通分量中，加入这条边之后必然产生环路，故而应该舍去这条边。

其算法步骤为：

- 步骤1：初始状态下无向图 $G = (V, E)$ 由 $n$ 个顶点 $v \in V$ 和 $m$ 个边 $e \in E \subseteq V \times V$ 组成，每个顶点看成一个连通分量。
- 步骤2：计算所有边的权重，并将所有边根据权重进行从小到大的排序。
- 步骤3：选择权重最小的边，判断其两端的顶点是否在同一连通分量中。两个顶点若属于不同的连通分量，则利用此边将两个连通分量合并。若两个顶点属于同一个的连通分量，则舍弃此边。
- 步骤4：重复步骤3，直到所有顶点在同一个连通分量为止。

Boruvka算法基本思想从当前所有的连通分量向外拓展，取最小边向其他连通块进行连接，直到只剩一个连通块。它是基于Kruskal算法和Prim算法。取最小边的贪心思想是Kruskal的算法主体；向外扩展，连接其他连通分量又是Prim的思想。

Prim算法即从图中的某一顶点出发，计算与其连接的各个边的权重，选取权重最小的边，连接那个顶点。继而计算这两个顶点相邻的所有顶点，同样与权重最小的边所连接的顶点进行连接。以此类推，直到连接所有的顶点。在拓展连接其他连通分量时，同Boruvka算法一样，也需要进行是否会形成环的判断。Boruvka算法相对于Boruvka算法最大的优点是可以并行化，具有线性时间解。

其算法步骤为：

- 步骤1：初始状态下无向图 $G = (V, E)$ 由 $n$ 个顶点 $v \in V$ 和 $m$ 个边 $e \in E \subseteq V \times V$ 组成，每个顶点看成一个连通分量。
- 步骤2：对于每个连通分量，计算并寻找与其相邻的最小权重边。
- 步骤3：对找到的边与对应的连通分量，进行是否构成环路的判断。若不构成环路，则此边连接的顶点加入到对应的连通分量中。否则舍弃此边。
- 步骤4：重复步骤2和步骤3，直到所有顶点在同一个连通分量为止。

#### 2.4.4 超像素在图像分割中的意义

对于图像分割任务，基于像素的传统处理方法取得了不错的成果。但是随着数码产品的拍照功能迅速发展，图像的构成越来越复杂，分辨率不断增大，越来越清晰，包括的像素数量也是成指数级增长。在这样的背景下，基于像素的传统图像分割方法处理分辨率高的图像，将花费更多的时间。

如何减少图像分割的计算量显得尤为重要。超像素作为一种图像预处理技术解决了这个问题。所谓超像素，就是由局部的许多像素构成的区域，这些区域内的像素通常由具有相似纹理、颜色、亮度等特征。相对于像素而言，超像素不仅有效减少了局部的冗余信息，后续处理过程中的计算量和复杂度大幅度降低，而且更利于局部特征的提取与表达，更有利帮助定位区域的边界。

### 2.5 超像素池化层

2017年，Suha Kwak等人提出了Superpixel Pooling Network (SPN)<sup>[52]</sup>网络，将超像素分割结果作为低阶结构的表征，利用提出的超像素池化层对输入网络的超像素进行提取特征，辅助语义分割的推断。此外，SPN网络验证了使用了超像素对于提高图像分割的准确率，确实能起到一定的效果。

在这里简单介绍一下SPN网络中超像素池化层的具体操作，每个超像素的特征向量生成如公式2-8所示：假设 $P_i = \{p_i^k\} k = 1, 2, \dots, K_i$ 。 $P_i$ 表示图像中第*i*个超像素， $K_i$ 表示第*i*个超像素中像素的个数，对于每个超像素 $P_i$ ，超像素特征的计算公式为：

$$v_i = \frac{1}{K_i} \sum_j \sum_k I(p_i^k \in r^j) z^j \quad (2-8)$$

其中， $r^j$ 表示感受野， $z^j$ 表示经过上采样得到的特征图中的第*j*个位置的值。 $I(p_i^k \in r^j)$ 是一个indicator函数，如果括号中的值为true，则返回1，否则返回0。

固定当前某一个超像素，将之池化。式中的 $r$ 存在是因为存在尺度差异，是感受野大小。作者的池化方式是固定超像素，遍历特征图和超像素内的元素，计算当前超像素元素在位置j感受野所占的比例，然后加权上去。通过式2-8便可以得到每个超像素的特征向量，从而可进行后续的运算。

## 2.6 本章小结

本章主要介绍了超像素分割和图像分割理论基础以及相关算法。首先分别深度学习的相关知识，包括深度学习的概念与发展，多层感受器，深度学习在计算机视觉与自然语言处理中的应用。然后详细介绍了卷积神经网络的各个基础组件，这也是本文构建的神经网络的基础。以LeNet模型和AlexNet模型为例，介绍了卷积神经网络的卷积层，激活层，池化层以及激活层。在第2.3小结，以FCN为例，详细介绍了卷积神经网络在图像分割领域的应用。2.4小结中，首先介绍了超像素分割的经典算法-SLIC算法和基于图论的相关算法。然后介绍了对比了基于图论的两个基础算法-Kruskal算法和Boruvka算法。最后说明了超像素在图像分割中的意义。最后一节中，介绍了超像素池化层，将超像素引入神经网络提供了理论基础以及实现的可能。

## 第3章 基于深度学习的超像素分割和图像分割

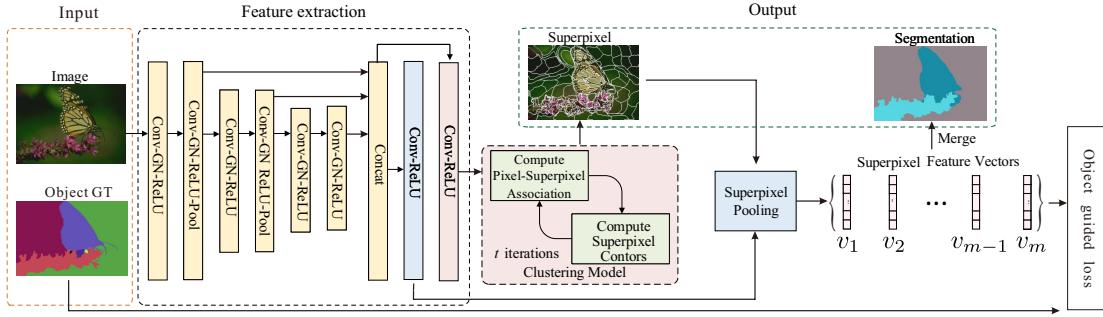


图 3-1 算法流程图。对于给定的图像，我们的算法同时生成超像素和图像分割。输入图像首先被输送到一个特征提取网络，该网络由一系列卷积层、归一化(GN)和ReLU操作组成。然后将提取的特征输入可微聚类模块，生成超像素。超像素池用于获取超像素特征向量。最后通过合并相似度高的超像素实现图像分割

如图3-1所示，本文提出的方法首先从CNN网络中学习图像特征，然后我们使用迭代可微分聚类算法模块来获取超像素。接下来，我们通过超像素池化层计算超像素特征向量，并计算相邻超像素之间的相似性。最后，根据相似度判断相邻的两个超像素是否合并。在本节中，将详细介绍提出的方法。

### 3.1 超像素生成

超像素将相似像素分组为同性区域，从而可以提高分割质量和效率。例如，在DEL<sup>[53]</sup>算法中，作者使用SLIC作为图像分割的开始。在本文中，与采用现有的超像素算法进行图像分割的方法不同，我们将超像素生成作为图像分割网络的一部分。为此，我们采用SSN中提出的可微聚类算法模块，以取代SLIC算法中的硬像素-超像素关联。

通常，对于 $n$ 个像素的图像 $I \in \mathbb{R}^{n \times 5}$ ，在CIELAB空间的特征为 $I_p = [x, y, l, a, b]$ ，我们希望将其划分为 $m$ 个小区域，即将图片分成 $m$ 个超像素。在介绍软关联之前，我们简要介绍SLIC算法中如何计算像素-超像素硬关联 $H = \{1, 2, \dots, m\}^{n \times 1}$ 。给定统一采样的超像素中心 $C^0$ 作为初始值，SLIC算法在每次迭代 $t$ 中计算每个像素 $p$ 处的新超像素分配，

$$H_p^t = \operatorname{argmin}_{i \in \{1, 2, \dots, m\}} \|I_p - C_i^{t-1}\|_2, \quad (3-1)$$

其中 $\|\cdot\|_2$ 表示输入向量的 $\ell_2$ 范数， $C_i^{t-1}$ 表示超像素中心 $i$ 的特征，该特征通过对第 $t$ 次迭代后，计算属于该超像素中心的像素的特征平均值来获取。

由于公式3-1获取像素-超像素硬关联 $H$ 的操作是不可微的，因此SLIC无法直接集成到神经网络中。在我们算法中的用到的可微分聚类算法模块，其将硬像素-超像素硬关联 $H$ 替换为软关联 $Q$ 。与原始SLIC相似，它在每次迭代中具有以下两个核心步骤：

1. 像素-超像素关联计算。第 $t$ 次迭代中像素 $p$ 及其相邻超像素 $i$ 之间的关联计算如下：

$$Q_{pi}^t = e^{-\|F_p - C_i^{t-1}\|_2^2}, \quad (3-2)$$

其中 $F_p$ 是像素 $p$ 的深层特征，来自网络的特征提取模块。 $Q_{pi}^t$ 是 $t$ 次迭代后像素 $p$ 与超像素中心 $i$ 之间的距离。

2. 超像素中心更新。新的超像素聚类中心是根据像素特征的加权总和得出的，

$$C_i^t = \frac{1}{Z_i^t} \sum_p Q_{pi}^t F_p, \quad (3-3)$$

其中 $Z_i^t$ 表示归一化过程，即 $Z_i^t = \sum_p Q_{pi}^t$ 。

将这两个步骤迭代数次(在本文中，设定迭代次数为10次)，最终得到像素-超像素软关联 $Q \in \mathbb{R}^{n \times m}$ 。与公式3-1相似，我们需要计算硬关联映射 $H' \in \mathbb{R}^{n \times 1}$ ，最终得到像素 $p$ 的超像素标签，

$$H'_p = \operatorname{argmax}_{i \in \{1, \dots, m\}} Q_{pi}. \quad (3-4)$$

值得注意的是，这种硬关联的计算是不可微的。因此在我们的算法中，这一步不参与反向传播。在实验中，我们发现计算像素和超像素聚类中心之间的软关联非常耗时。与SLIC相似，我们只是计算每个像素到周围超像素聚类中心的距离，大大减少了计算时间。

## 3.2 超像素相似度

在获得超像素之后，我们需要测量它们之间的相似性。超像素的特征可以用超像素池来计算，它实际上是计算属于某个超像素的像素特征的平均值。在我们的算法中，当执行超像素池时，我们使用了不同于在超像素生成中使用的像素特征，如图3-1所示，使用浅蓝色矩形所表示的图像特征。

获取超像素后，我们假设超像素的数量为 $m$ ，将超像素集合表示为 $\mathcal{S} = \{S_1, S_2, \dots, S_m\}$ ，根据图3-1所示，我们所得到的超像素进行超像素池化操作，得

到对应超像素的特征向量 $\{v_1, v_2, \dots, v_m\}$ 。超像素池化操作表示如下：

$$v_i = \frac{1}{|S_i|} \sum_{p \in S_i} F'_p \quad (3-5)$$

其中， $F'_p$  表示在超像素池中使用的像素 $p$ 的特征向量， $|S_i|$ 表示第*i*个超像素中含有像素的数目。

相邻超像素*i*和超像素*j*的相似度可以由如下公式获得：

$$s_{ij} = \frac{2}{1 + \exp(\|v_i - v_j\|_1)} \quad (3-6)$$

其中 $s_{ij}$ 的范围是[0, 1]。 $s_{ij}$ 值越大，超像素*i*和*j*的相似性越高。当 $v_i$ 和 $v_j$ 非常相似时， $s_{ij}$ 接近1，相反，当 $v_i$ 和 $v_j$ 极为不同时，它接近于0。我们根据相似度 $s_{ij}$ 决定是否合并超像素*i*和*j*。

### 3.3 损失函数

#### 3.3.1 像素和超像素表示之间的映射

利用提供硬聚类的传统超像素算法，这种从像素到超像素表示的映射是通过在每个聚类内部进行平均来完成的。从超像素到像素表示的逆映射是通过将相同的超像素特征分配给属于该超像素的所有像素来完成的。然而，由于这种硬关联的计算是不可微的，因此在集成到端到端可训练系统时可能不希望使用硬簇。值得注意的是，由网络生成的像素-超像素软连接也可以容易地用于像素和超像素表示之间的映射。公式3-3已经描述了从像素到超像素表示的映射，其是与列标准化Q矩阵的转置的简单矩阵乘法： $S = \hat{Q}^T F$ ，其中F和S分别表示像素和超像素。从超像素到像素表示的逆映射是通过将行标准化Q（表示为 $\tilde{Q}$ ）。与超像素表示（ $F = \tilde{Q}S$ ）相乘。之后，我们将利用这些映射来设计损失函数。

对于超像素分割，本文设计使用了重建损失和紧凑性损失。

#### 3.3.2 重建损失

本文把像素属性表示为 $R \in \mathbb{R}^{n \times l}$ 。如前所述，我们可以使用列标准化关联矩阵 $\hat{Q}$ ， $\check{R} = \hat{Q}^T R$ 将像素属性映射到超像素上，其中 $\check{R} \in \mathbb{R}^{m \times l}$ 。然后使用行标准化关联矩阵 $\tilde{Q}$ ，将得到的超像素表示 $\check{R}$ 映射回像素表示 $R^*$ ， $R^* = \tilde{Q}\check{R}$  其中 $R^* \in \mathbb{R}^{n \times l}$ 。然后重建损失给出为：

$$L = \mathcal{L}(R, R^*) = \mathcal{L}(R, \tilde{Q}\hat{Q}^T R) \quad (3-7)$$

本文使用 $\ell_1$ 交叉熵损失来学习的超像素。这里 $Q$ 表示在可微聚类模块的最终

迭代之后的关联矩阵 $Q^v$ 。为方便起见，我们省略了 $v$ 。

### 3.3.3 紧凑性损失

除了上述损失之外，本文还使用紧凑性损失来鼓励超像素在空间上紧凑，即在每个超像素簇内具有较低的空间变化。让 $I^{xy}$ 表示位置像素特征。我们首先将这些位置特征映射到我们的超像素表示中， $S^{xy} = \hat{Q}^T I^{xy}$ 。然后，通过将相同的超像素位置特征分配给属于该超像素的所有像素， $\bar{I}_p^{xy} = S_i^{xy} | H_p = i$ ，使用硬关联H而不是软关联Q对像素表示进行逆映射。紧凑性损失定义为以下 $\ell_2$ 规范：

$$L = \|I^{xy} - \bar{I}^{xy}\|_2 \quad (3-8)$$

这种损失促使超像素具有较低的空间方差。

### 3.3.4 相似性损失

我们假设同一分割区域内的超像素对之间的相似性大于不同分割区域中的超像素对的相似性。基于公式3-6中定义的相似性度量，我们定义的损失函数如下：

$$L = - \sum_{S_i \in S} \sum_{S_j \in \mathcal{R}_i} \left[ (1 - \alpha) \cdot l_{ij} \cdot \log(s_{ij}) + \alpha \cdot (1 - l_{ij}) \cdot \log(1 - s_{ij}) \right], \quad (3-9)$$

其中 $\mathcal{R}_i$ 是超像素 $S_i$ 的相邻的超像素集合， $l_{ij}$ 表示 $S_i$ 和 $S_j$ 是否属于同一个分割区域。在实际应用中， $l_{ij}$ 是由所获得的超像素集和数据集提供的真值来计算的。在 $S_i$ 和 $S_j$ 属于同一分割区域的情况下， $l_{ij} = 1$ ；否则， $l_{ij} = 0$ 。

注意，对于不同的输入图像， $l_{ij}$ 的矩阵是不同的。因此，训练阶段的小批量大小必须设置为1，即一次只向网络中输入一张图像。参数 $\alpha$ 表示在真值中属于同一区域的超像素对的比例，用于平衡正样本和负样品。通过将 $|Y_+|$ 表示为属于同一区域的超像素对的数目，将 $|Y|$ 表示为超级像素对的总数，通过 $\alpha = |Y_+| / |Y|$ 来计算。

## 3.4 超像素融合

通过合并相似的超像素得到最终的图像分割。我们利用相邻超像素之间的相似性和一个预先设定的阈值 $T$ 来确定两个相邻超像素是否合并。算法1概述了超像素合并的计算步骤。

**Algorithm 1** Superpixel merging algorithm

---

**Input:**  $s$  : similarity;       $T$  : similarity threshold;
 $\mathcal{S} = \{S_1, S_2, \dots, S_m\}$  : superpixels.
**Output:** Segmentation  $\mathcal{S}$ .

---

```

1: for each  $S_i \in \mathcal{S}$  do
2:   Construct adjacent superpixel set  $\mathcal{R}_i \subset \mathcal{S}$  of  $S_i$ ;
3:   for each  $S_j \in \mathcal{R}_i$  do
4:     if  $s_{ij} > T$  then
5:        $S_i \leftarrow S_i \cup S_j$ ,  $\mathcal{S} \leftarrow \mathcal{S} \setminus S_j$  ;
6:       Update  $\mathcal{R}_i$  ;
7:     end if
8:   end for
9: end for
```

---

### 3.5 网络结构

图3-1显示了我们的网络结构。用于特征提取的CNN网络由卷积层、组归一化(GN)和ReLU激活函数组成。我们将GN中的组数设置为8。在第2层和第4层卷积后，我们使用最大池来增加感受野。对第4层和第6层的输出进行上采样，然后与第2层的输出连接，以丰富提取的特征。我们使用 $3 \times 3$ 卷积滤波器将输出通道设置为每层64个。

注意，考虑到网络中每个小批量的大小必须为1，我们用GN替换了广泛使用的批归一化(BN)层。BN操作是深度学习发展中的一个里程碑式的技术，使各种网络能够进行训练。然而，沿着批处理维度进行规范化会带来问题——当批处理大小变小时，BN的错误会迅速增加，这是由于批次统计估计不准确造成的。相反，GN将通道分成组，并计算每组中的均值和方差进行归一化。GN的计算是独立于批量大小的，其精度在较小的批量范围内是稳定的。在实验中，我们还比较了用BN和GN作为归一化的结果。

在多任务学习中，不同层次的任务需要不同的图像特征，如UPerNet<sup>[7]</sup>。对于超像素生成和图像分割这两个不同层次的任务，我们进一步对上一步得到的图像特征进行卷积运算，得到不同的特征向量，以满足不同任务的需要。具体来说，对于超像素生成任务，我们使用核大小为 $3 \times 3$ 的卷积层来获得30个通道的特征向量。对于图像分割任务，我们首先对256个输出通道进行 $3 \times 3$ 卷积运算，然后使用 $1 \times 1$ 卷积核得到64个通道的特征向量。如图3-1所示，我们将得到的特征向量分别输入到后续的可微聚类算法模块和超像素池层，然后利用所提出的相应

的损失函数对网络进行训练。

### 3.6 实施细节

该方法基于Caffe框架来搭建神经网络。Caffe<sup>[62]</sup>框架由加州大学研究团队开发，是一个非常有效的深度学习框架，广泛应用于学术界和工业界。它的优点包括很多，首先Caffe框架学习成本低，以配置文件的形式来构建网络模型，过程简单。其次，Caffe相较于其他深度学习框架，训练速度更快。代码底层基于C++语言，代码质量高，运行稳定。然后，由于Caffe将各个组件模块化，拥有很好的拓展性。此外，已有大量训练好的成熟网络模型，方便直接使用。

对于超级像素的生成，就像在原始的SLIC算法中一样，我们在公式3-4之后的每个超级像素簇内的像素之间加强空间连通性。通过将小于某个阈值的超像素与周围的超级像素合并，然后为每个空间连接的组件分配一个唯一的聚类ID来实现的。对于图像分割，我们在合并后进行空间连通性增强操作。需要注意的是，加强空间连通性的运算是不可微的，我们只把它当作后处理，而不加入神经网络。

我们使用BSDS500数据集作为我们的训练和测试数据，该数据集已在图像分割领域得到广泛应用。由于BSDS500中训练样本数量较少，训练时需要进行数据扩充。我们将每一个真值作为一个单独的样本，即对于每一对图像和真值，我们将其作为训练样本提供给网络。通过这种方式，我们得到了1633对训练/验证对和1063对测试对。另外，我们采用了两种常用的数据扩充策略，即翻转和裁剪。具体地说，在训练阶段，我们将图像左右翻转，随机地将图像裁剪成 $201 \times 201$ 大小的图像块，进行数据增强。采用Adam<sup>[63]</sup>优化我们的网络。基本学习率设置为 $1e-5$ ，生成的超级像素数设置为100。动量设置为0.99，以在相对较小规模的数据上实现稳定优化，如FCN中所建议的那样。每个小批量的大小设置为1。

我们进行500K次迭代来训练深度学习模型，并根据验证的准确性选择最终的训练模型。

### 3.7 本章小结

本章主要介绍了一种基于深度学习的超像素分割和图像分割方法。该方法首先使用CNN深度网络来学习图像特征，然后使用迭代可微分聚类算法模块来获取超像素。接下来通过超像素池化层计算超像素特征向量，并计算相邻超像素之间的相似性。最后，根据相似度判断相邻的两个超像素是否合并。本章介绍了

超像素生成，计算超像素相似度，超像素融合以及详细说明了所使用的损失函数。此外，本章详细介绍了该方法的网络结构以及实施细节。



## 第4章 基于Boruvka算法和快速模糊C均值聚类的图像分割

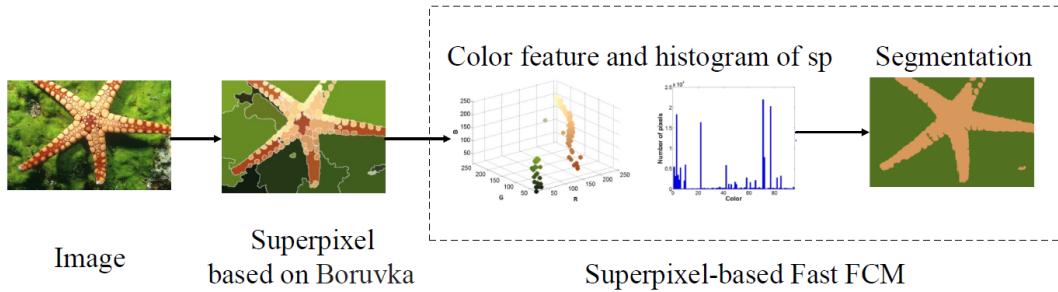


图 4-1 算法流程图

在这项工作中，我们使用一种基于Boruvka算法来产生超像素图像。Boruvka算法具有线性时间解，可并行化。此外，通过Boruvka 算法，得到连通分量。在每次迭代之后，元素特征在新形成的集群中进行聚合。这种方法比SLIC和LSC等只利用每个像素特征来确定聚类隶属关系的方法更为健壮。

在Boruvka算法获得的超像素图像的基础上，通过计算超像素图像的颜色直方图来实现快速模糊C 均值聚类。由于超像素图像中不同颜色的数目远小于原始彩色图像，因此计算超像素图像的颜色直方图非常容易。最后，以超像素颜色直方图作为目标函数的参数，实现彩色图像的快速分割。我们提出的算法的框架如图4-1所示。

### 4.1 基于Boruvka算法生成超像素

把一个图看成有 $n$ 个树的森林，每个顶点看成一棵树。对于每棵树来说，我们找到它最近的邻点，即边的权重最小的那个点，并把他们合并在一起。假设 $C_2$ 是 $C_1$ 的最小的邻点( $C_1$ 也许并不是 $C_2$ 最小的邻点)，我们定义两个树之间的距离为：

$$D(C_1, C_2) = \min_{v_i \in C_1, v_j \in C_2, (v_i, v_j) \in \epsilon} \omega((v_i, v_j)) \quad (4-1)$$

最近邻点选择之后，构建辅助图，其中每个顶点表示一个簇，每个边对应一个选定的权重最小的边。如果这里有多个最近的邻点，我们在辅助图上用重复的边来表示他们。另一方面，辅助图上的边是不同的。用深度优先搜索找到

连通图。Boruvka算法重复融合树直到只有一棵树留下来。我们用Boruvka算法生成一个最小生成树（MST），同时记录每个边缘的顺序添加到MST。一旦边缘被添加到MST，森林中的树木数量减少一个。假设需要提取 $k$ 个超像素，我们通过前 $n - k$ 个边连接顶点，并且具有 $k$ 个连接的组件，这些组件恰好是超像素。

Boruvka和Kruskal算法之间的主要区别在于前者是局部和并发搜索边缘，而后者在全局范围内对边缘进行排序并按顺序执行。因此，Boruvka算法可以并行处理。此外，Boruvka算法假设簇是均匀分布的，并缓解了Kruskal算法倾向于生成严重不平衡簇的缺点。在所提出的超像素层次方法中，我们采用Boruvka算法来构造MST。同时，记录每条边加入MST的顺序。一旦将一条边添加到MST中，森林中的树数将减少一棵。假设需要提取 $k$ 个超像素，我们通过第一个 $n-k$ 条边连接顶点，并且有 $k$ 个连通的分量，这些分量就是超像素。

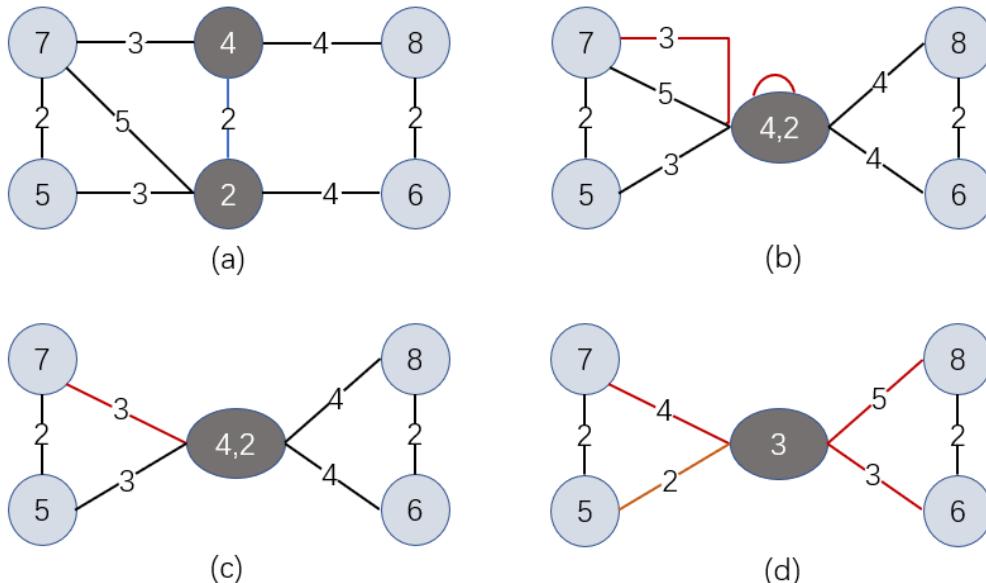


图 4-2 边缘收缩和特征聚合。每个顶点中的数字代表其特征值。边缘权重通过其两端的绝对距离来计算。(a) - (c) 在顶点4 和2之间执行边缘收缩。收缩边缘后，用自环替代收缩的边缘，并用平行边缘来保存原边的权重值。展平操作之后，移除自环并用权重较小的一条（红线）替换平行边缘。(d) 在每次迭代之后，通过从新形成的聚类中收集特征，然后更新边缘权重（红线）来进行特征聚合。

Boruvka算法可以直接应用于超像素分割。然而，直接应用该算法并不能产生令人满意的结果。这些问题源于贪婪的局部算法设计，在计算最小边权重量时，因为对于每棵树，只使用一个顶点的属性，所以对异常值很敏感。为了弥补这个缺陷，本方法对Boruvka 算法进行了改进，在同一个连通分量内合并邻域信息，称为边缘收缩。

边缘收缩如图4-2所示。在图4-2 (a) 的每个顶点，一个数字表示属性（例如，像素强度）。每个边权重由属性在两端的绝对距离计算。在顶点4和顶点2之

间执行边收缩，形成一个超顶点。在图 (c) 中，保留较小的平行边。由于我们在每次迭代后都会得到新的簇，即新的连通分量，所以很自然地会聚集每个簇的属性并更新连接到其他簇的权重。图4-2 (d) 说明了这一过程。合并顶点4 和顶点2 后，将形成一个平均值为3 的超顶点。将删除自循环。同时，更新连接到超顶点的所有边的权重。特征聚合利用了“群体智慧”，而不是只利用两个顶点，这样可以获得更好的性能。

## 4.2 基于超像素的快速模糊C均值聚类

由于Boruvka算法依赖于图像的局部特征，而模糊C均值聚类依赖于全局特征，因此Boruvka算法和模糊C均值聚类的结合能够提高图像分割的效果。在这一部分中，我们提出了一种基于超像素的模糊C均值聚类算法，它将自适应的局部空间信息引入模糊C均值聚类的目标函数中。

$$J_m = \sum_{l=1}^q \sum_{k=1}^c S_l u_{kl}^m \left\| \left( \frac{1}{S_l} \sum_{p \in R_l} x_p \right) - v_k \right\|^2 \quad (4-2)$$

其中， $l$ 是颜色级别， $1 \leq l \leq q$ ， $q$ 是超像素图像的区域数量，即超像素数量， $S_l$ 是第 $l$ 区域的像素数， $u_{kl}$ 表示颜色级别 $l$ 相对于第 $k$ 个聚类中心 $v_k$ 的模糊隶属度， $m$ 为加权指数。 $x_p$ 是超像素图像中第 $l$ 区域内的颜色像素。由于原始图像中的每个颜色像素被超像素图像对应区域内的颜色像素的平均值所代替，所以颜色级别的数量相当于超级像素图像中的区域数量。因此， $l \ll N$ ，计算复杂度被有效地降低。

利用拉格朗日乘数法，将上述优化问题转化为一个无约束优化问题，该优化问题的目标函数为：

$$\widetilde{J}_m = \sum_{l=1}^q \sum_{k=1}^c S_l u_{kl}^m \left\| \left( \frac{1}{S_l} \sum_{p \in R_l} x_p \right) - v_k \right\|^2 - \lambda \left( \sum_{k=1}^c u_{kl} - 1 \right) \quad (4-3)$$

其中 $\lambda$ 是拉格朗日乘数。我们分别计算了 $\widetilde{J}_m$ 关于 $u_{kl}$ 和 $v_k$ 的偏微分方程。

$$\begin{aligned} \frac{\partial \widetilde{J}_m}{\partial u_{kl}} &= \sum_{l=1}^q \sum_{k=1}^c \frac{\partial \sum_{l=1}^q \sum_{k=1}^c S_l u_{kl}^m \left\| \left( \frac{1}{S_l} \sum_{p \in R_l} x_p \right) - v_k \right\|^2}{\partial u_{kl}} - \lambda \\ &= \sum_{l=1}^q \sum_{k=1}^c m S_l u_{m-1}^{kl} \left\| \left( \frac{1}{S_l} \sum_{p \in R_l} x_p \right) - v_k \right\|^2 \\ &= 0 \end{aligned} \quad (4-4)$$

$$\begin{aligned}
\frac{\partial \widetilde{J}_m}{\partial v_k} &= \sum_{l=1}^q \sum_{k=1}^c \frac{\partial \sum_{l=1}^q \sum_{k=1}^c S_l u_{kl}^m \left\| \left( \frac{1}{S_l} \sum_{p \in R_l} x_p \right) - v_k \right\|^2}{\partial v_k} \\
&= \sum_{l=1}^q \sum_{k=1}^c S_l u_{kl}^m \frac{\partial \left\| \left( \frac{1}{S_l} \sum_{p \in R_l} x_p \right) - v_k \right\|^2}{\partial v_k} \\
&= \sum_{l=1}^q S_l u_{kl}^m \frac{\partial \left\| \left( \frac{1}{S_l} \sum_{p \in R_l} x_p \right) - v_k \right\|^2}{\partial v_k} \\
&= -2 \sum_{l=1}^q S_l u_{kl}^m \left\| \left( \frac{1}{S_l} \sum_{p \in R_l} x_p \right) - v_k \right\| \\
&= 0
\end{aligned} \tag{4-5}$$

将公式4-4和公式4-5结合起来，得到了 $u_{kl}$ 和 $v_k$ 的相应解：

$$v_k = \frac{\sum_{l=1}^q u_{kl}^m \sum_{p \in R_l} x_p}{\sum_{l=1}^q S_l u_{kl}^m} \tag{4-6}$$

$$u_{kl} = \frac{\left\| \left( \frac{1}{S_l} \sum_{p \in R_l} x_p \right) - v_k \right\|^{-2/(m-1)}}{\sum_{j=1}^c \left\| \left( \frac{1}{S_l} \sum_{p \in R_l} x_p \right) - v_j \right\|^{-2/(m-1)}} \tag{4-7}$$

我们提出的算法可以总结如下：

- Step 1: 设置 $c, m, \eta$ 的初始值, 其中 $\eta$ 是我们算法的收敛条件。
- Step 2: 根据超像素图像, 随机初始化隶属度划分矩阵 $U^{(0)}$ 。
- Step 3: 设置循环计数器 $b = 0$ 。
- Step 4: 使用公式4-6更新聚类中心。
- Step 5: 使用公式4-7更新隶属度划分矩阵 $U^{(t)}$ 。
- Step 6: 当 $\max U^{(b)} - U^{(b+1)} < \eta$ , 则停止; 否则, 设置 $b = b + 1$ , 跳转到步骤4。

### 4.3 本章小结

本章主要介绍了一种基于Boruvka算法和快速模糊C均值聚类的图像分割方法。该方法首先使用基于Boruvka的算法来产生超像素。在Boruvka算法获得的超像素图像的基础上, 通过计算超像素图像的颜色直方图来实现快速模糊C均值聚类, 得到图像分割结果。

## 第5章 实验结果和分析

### 5.1 实验数据集以及评估标准

#### 5.1.1 实验数据集

公开数据集berkeley segmentation dataset(BSDS500)<sup>[54]</sup>是伯利克里大学computer vision group课题组提供的数据集，已广泛应用于图像分割和物体边缘检测。该数据集中包含500张图像，其中200张用于训练，100张用于验证以及200张用于测试。所有的真值用.mat文件保存，包含segmentation和boundaries两种数据信息，每张图片对应真值有五个，为五个人标注的真值。BSDS500已经成为图像分割，超像素分割和边缘检测的标准基准。本文使用BSDS500数据集中的200张测试图片来进行实验及评估。

#### 5.1.2 评估标准

图像分割和超像素分割是计算机视觉领域重要的两个方向，并且已经有许多公开可用的评估基准。本小结将介绍本文所使用的图像分割和超像素分割的评估标准。

为了对实验结果进行评估，本文使用Boundary F-measure(BF), Probabilistic Rand Index (PRI)<sup>[51]</sup> 和Global Consistency Error (GCE)<sup>[64]</sup>作为图像分割的主要指标。用BF, Boundary Recall (BR), under segmentation error (UE) 和compactness(CO)作为超像素分割的主要指标。我们基于整个数据集的整体性能选择最佳参数。BF, BR, PRI和CO的分值越高，结果越好。GCE 分值和UE分值越低，结果越好。

对于评估标准BF和BP的计算，本文使用了以下四个数值来进行定义：

- True Positive, TP: 称为真阳性。以边界像素为例，某像素点被预测为边界，在groundtruth中真实分类也为边界；
- False Positive, FP: 称为假阳性，某像素点被预测为边界，但在groundtruth中真实分类不是边界；
- False Negative, FN: 称为假阴性，某像素点被模型不是边界，在groundtruth中真实分类却是边界

(1) 边界召回率BR可以理解为预测结果中，真正属于边界的像素数目占所

有像素数目比例，其计算公式为：

$$BR = \frac{TP}{TP + FN} \quad (5-1)$$

(2) 在介绍Boundary F-measure(BF)之前，首先介绍与召回率recall对应的评估指标精确率precision，以边界精确率为例，其含义为预测结果中，真正属于边界的像素数目占预测结果中所有边界像素的比例，计算公式为：

$$BP = \frac{TP}{TP + FP} \quad (5-2)$$

可见，精确率和召回率是相互影响的，理想情况下两者都高，但是一般情况下准确率高，召回率就低；召回率高，准确率就低。为了平衡这个两个指标，综合衡量P和R，更好的来评估结果，应该使用F值，其计算公式为：

$$F = \frac{(\alpha^2 + 1) \cdot P \cdot R}{\alpha^2 \cdot (P + R)} \quad (5-3)$$

$\alpha$ 为1时，就是常见的F1值（F1 score）：

$$F1 = \frac{2 \cdot P \cdot R}{P + R} \quad (5-4)$$

一般多个模型假设进行比较时，F1 score越高，说明它越好。本文使用的BF就是边界的F1值。

(4) PR曲线（precision-recall curve）：PR曲线的横坐标和纵坐标分别是召回率和准确率。PR曲线上点代表不同阈值下召回率和准确率。由上述公式可知，准确率和召回率成负相关，即准确率越高，召回率越小。对于图像分割的PR曲线，若曲线与X轴和Y轴包含的面积越大，表示分割效果越好，即得到的曲线形状越靠近右上方，则表示法性越好。

(5) Global Consistency Error (GCE)：计算两个区域相互一致的程度，定义为：

假设图像分割结果表示为  $S^t = \{C_1^t, C_2^t, \dots, C_{R^t}^t\}$ ，groundtruth分割图表示为  $S^g = \{C_1^g, C_2^g, \dots, C_{R^g}^g\}$ 。其中， $R^t$  是  $S^t$  中的区域  $C$  的数量， $R^g$  是  $S^g$  中的区域数。

对于特定的像素  $p_i$ ，我们考虑  $S^t$  和  $S^g$  中包括该像素的段。我们分别用  $C_{\langle p_i \rangle}^t$  和  $C_{\langle p_i \rangle}^g$  表示这些段。如果一个段是另一段的子集，则像素实际上包含在细化区域中，并且局部误差应等于零。如果没有子集关系，则这两个区域将以不一致的方式重叠，并且局部误差应不同于零。因此，局部细化误差 (LRE) 在像素  $p_i$  处表示为：

$$LER(S^t, S^g, p_i) = \frac{|C_{\langle p_i \rangle}^t \setminus C_{\langle p_i \rangle}^g|}{|C_{\langle p_i \rangle}^t|} \quad (5-5)$$

其中 \ 表示集合的差运算，而  $|C|$  代表像素集  $C$  的基数。所谓的全局一致性误

差 (GCE)，就是将每个像素处的LRE组合成整个图像的度量，公式如下：

$$GCE(S^t, S^g) = \frac{1}{n} \min \left\{ \sum_{i=1}^n LRE(S^t, S^g, p_i), \sum_{i=1}^n LRE(S^g, S^t, p_i) \right\} \quad (5-6)$$

其中n是图像中像素 $p_i$ 的数量。这种基于GCE的分割误差度量，其值属于区间[0,1]。0表示两个分段之间完全匹配，误差为1表示要比较的两个分段之间的最大差值。

(6) PRI是一种相似性度量，它计算分割图像与真实分割之间的一致性。

同样我们假设图像分割结果表示为 $S^t = \{C_1^t, C_2^t, \dots, C_{R^t}^t\}$ ，groundtruth 分割图表示为 $S^g = \{C_1^g, C_2^g, \dots, C_{R^g}^g\}$ 。其中， $R^t$ 是 $S^t$ 中的区域C的数量， $R^g$ 是 $S^g$ 中的区域数。

the Rand index (RI) 定义为：

$$RI = \frac{n_{11} + n_{00}}{n_{00} + n_{01} + n_{10} + n_{11}} \quad (5-7)$$

其中， $n_{11}$ 表示 $S^t$ 和 $S^g$ 中位于同一区域中的像素对数。 $n_{00}$ 表示 $S^t$ 和 $S^g$ 中位于不同区域中的像素对数。 $n_{10}$ 表示在 $S^t$ 中位于同一区域，而 $S^g$ 中位于不同区域中的对象对数。 $n_{01}$ 表示在 $S^t$ 中位于不同区域，而 $S^g$ 中位于同一区域的对象对数。

RI方法中，所有参数的权重是相同的，为了更好的平衡各项，PRI对各项进行了加权。

假设每个像素随机分配给一个区域，两个像素在 $S^t$ 和 $S^g$ 中位于同一个区域的概率为：

$$p_{11} = \frac{1}{R^t} \cdot \frac{1}{R^g} \quad (5-8)$$

两个像素在 $S^t$ 和 $S^g$ 中的不同区域中的概率是：

$$p_{00} = \frac{R^t - 1}{R^t} \cdot \frac{R^g - 1}{R^g} \quad (5-9)$$

两个像素在 $S^t$ 中位于同一个区域中，而在 $S^g$ 中位于不同区域的概率为：

$$p_{10} = \frac{1}{R^t} \cdot \frac{R^g - 1}{R^g} \quad (5-10)$$

两个像素在 $S^t$ 中位于不同区域，而在 $S^g$ 中位于同一个区域的概率为：

$$p_{01} = \frac{R^t - 1}{R^t} \cdot \frac{1}{R^g} \quad (5-11)$$

其中 $\sum_{h=00}^{11} p_h = 1$ (h以二进制表示)。概率越低，则事件确实发生的权重就越高。权重计算公式为：

$$w_h = -\log_2(p_h) \quad (5-12)$$

这些权重可用于定义Probabilistic Rand Index:

$$PRI = \frac{w_{11} \cdot n_{11} + w_{00} \cdot n_{00}}{w_{00} \cdot n_{00} + w_{01} \cdot n_{01} + w_{10} \cdot n_{10} + w_{11} \cdot n_{11}} \quad (5-13)$$

与RI相比，使用PRI的优点在于，对各项进行了权衡。

(7) under segmentation error (UE): 欠分割率, UE作为一种超像素分割评估方法, 该方法惩罚像素跟真实分割不重合的情况。其本质是衡量算法在根据真值对图像进行分割时所犯的错误。其计算公式为:

$$UE = \frac{1}{N} \left[ \sum_{i=1}^{R^g} \left( \sum_{S_j | S_j \cap g_i > B} |S_j| \right) - N \right] \quad (5-14)$$

其中,  $| \cdot |$ 表示此区域内像素的数量,  $N$ 表示图像的大小, (以像素为单位)。B是需要重叠的最小像素数, 默认将B设为 $|S_j|$ 的5%, 以解决真值分割数据中的小错误。表达式 $S_j \cap g_i$ 是超像素 $S_j$ 相对于地面真实部分 $g_i$ 的相交或重叠误差。

(8) compactness(CO): 紧凑度, 指标衡量了超像素是否“紧实”。

在数学中, 测量形状的紧凑度常用的量度是isoperimetric quotient。其规定圆形的isoperimetric quotient为1, 并且形状变得越不紧凑, 值越小。假设一个超像素的面积为 $A_s$ , 周长为 $L_s$ , 那么具有与超像素相同的周长的圆的半径为 $r = \frac{L_s}{2\pi}$ 。令 $A_c$ 为半径为 $r$ 的圆的面积, 即 $A_c = \pi r^2$ 。isoperimetric quotient的计算如下:

$$Q_s = \frac{A_s}{A_c} = \frac{4\pi A_s}{L_s^2} \quad (5-15)$$

基于isoperimetric quotient, Alexander Schick等人提出了一种衡量超像素分割的紧凑度(CO)的度量。对于给定的图像I, 其超像素分割结果为 $S = \{S_1, S_2, \dots, S_m\}$ , 分割结果的紧凑度为:

$$CO = \sum_{i=1}^m Q_{S_i} \cdot \frac{|S_i|}{|I|} \quad (5-16)$$

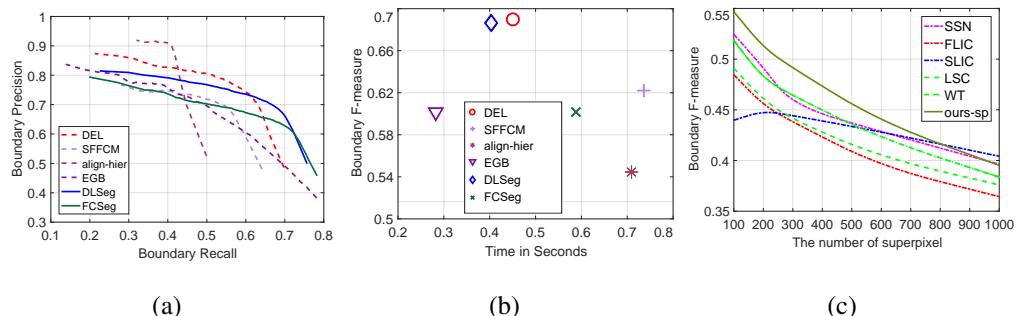


图 5-1 BSDS500数据集上的结果对比。左: 图像分割算法的边界P-R曲线图, 中: 图像分割算法的BF和时间比较, 右: 超像素分割算法的BF值比较

## 5.2 消融实验

### 5.2.1 基于深度学习的超像素分割和图像分割

#### 5.2.1.1 实验介绍

我们使用图3-1中的网络结构作为我们的基本结构，将其命名为DLSeg-GN8。我们用BSDS500 数据集来测试网络中每个组件的不同选择。除了ours-GN8模型，从超像素分割和图像分割两方面，我们将ours-GN8分别和其他4种变形进行了比较。

1. DLSeg-BN: 用batch normalization(BN)操作代替group normalization(GN)操作。

2. DLSeg-GN32: 在GN操作中，将group number参数设为32，而不是ours-GN8模型中的参数8。

3. DLSeg-conv7: 对于超像素分割和图像分割，我们使用同一个特征，即从conv7中获取到的特征。

- 4.DLSeg-w/o-concat:舍弃conv2→ concat和conv4→concat两个过程，即不包含从conv2和conv4 获取到的特征。

在下一节中，我们将评估图像分割和超像素分割的结果。

#### 5.2.1.2 实验结果以及数据分析

表 5-1 The performance of superpixel generation of 4 variants

Methods	BF( $\uparrow$ )	BR( $\uparrow$ )	UE( $\downarrow$ )	CO( $\uparrow$ )
DLSeg-BN	0.547	0.884	0.068	0.373
DLSeg-GN32	0.546	0.897	0.066	0.376
DLSeg-conv7	0.521	0.812	0.094	0.413
DLSeg-w/o-concat	0.521	0.895	0.071	0.340
DLSeg-GN8	0.547	0.918	0.065	0.316

表 5-2 The performance of image segmentation of 4 variants

Methods	BF( $\uparrow$ )	PRI( $\uparrow$ )	GCE( $\downarrow$ )
DLSeg-BN	0.661	0.807	0.146
DLSeg-GN32	0.685	0.820	0.171
DLSeg-conv7	0.492	0.714	0.088
DLSeg-w/o-concat	0.560	0.817	0.182
DLSeg-GN8	0.686	0.822	0.170

表5-1展示了五种模型在超像素分割方面的对比结果，表5-2展示了五种模型在图像分割方面的对比结果。从表5-1和表5-2可知，相对于其他4种变体而言，DLSeg-GN8模型性能表现最好，验证了我们选择组件的合理性。

DLSeg-GN8模型和DLSeg-GN32模型相对于DLSeg-BN模型而言，在超像素分割和图像分割的边界保持方面更有优势。GN操作解决了BN操作对批次大小依赖性的影响。对于小批次，GN操作可以取得更好的效果。但是DLSeg-GN32模型的效果较DLSeg-GN8模型的效果有所降低。当分组数量过大的时候，效果有所下降。

在卷积神经网络中，浅层网络包含更详细的信息，而深层网络包含更多的全局信息。因为DLSeg-GN8模型比DLSeg-w/o-concat模型包含更多提取到的特征，因此分割效果更好。DLSeg-conv7 的分割效果明显降低，DLSeg-GN8远远好于DLSeg-conv7，从而证明在多任务学习中，不同级别的任务需要不同的图像特征。

### 5.2.2 基于Boruvka算法和快速模糊C均值聚类的图像分割

为了确定不同超像素数目对最终图像分割结果的影响，本文将基于Boruvka算法产生的超像素数目N分别设置为8, 10, 20, 50, 100, 300, 来进行实验。此外为了验证快速模糊C均值聚类方法的有效性，设置了对照组—FCSeg-Boruvka，该模型只使用Boruvka进行分割来进行分割图像，不使用快速模糊C 均值聚类。实验数据如表5-3 所示。为了方便表示本文将该方法表示为FCSeg。

表 5-3 The performance of image segmentation of 4 variants

Methods	BF( $\uparrow$ )	PRI( $\uparrow$ )	GCE( $\downarrow$ )
FCSeg-Boruvka	0.460	0.696	0.339
FCSeg-8	0.531	0.711	0.372
FCSeg-10	0.537	0.717	0.352
FCSeg-20	0.571	0.724	0.306
FCSeg-50	0.580	0.735	0.268
FCSeg-100	0.594	0.736	0.241
FCSeg-300	0.576	0.720	0.280

从表中可以看出，FCSeg-Boruvka模型效果不佳，证明了基于Boruvka算法和快速模糊C均值聚类方法的合理性以及有效性。当超像素小于100时，分割效果随着超像素数量的增加而变好，但是当超像素数量超过100个时，分割效果就会越来越差。由经验可知，但超像素数量过高，每个超像素内包含的像素数量极少，超像素对边界的保持性降低，分割效果将会降低。故而，选择 $N_{sp} = 100$ 作为超像素的数量。

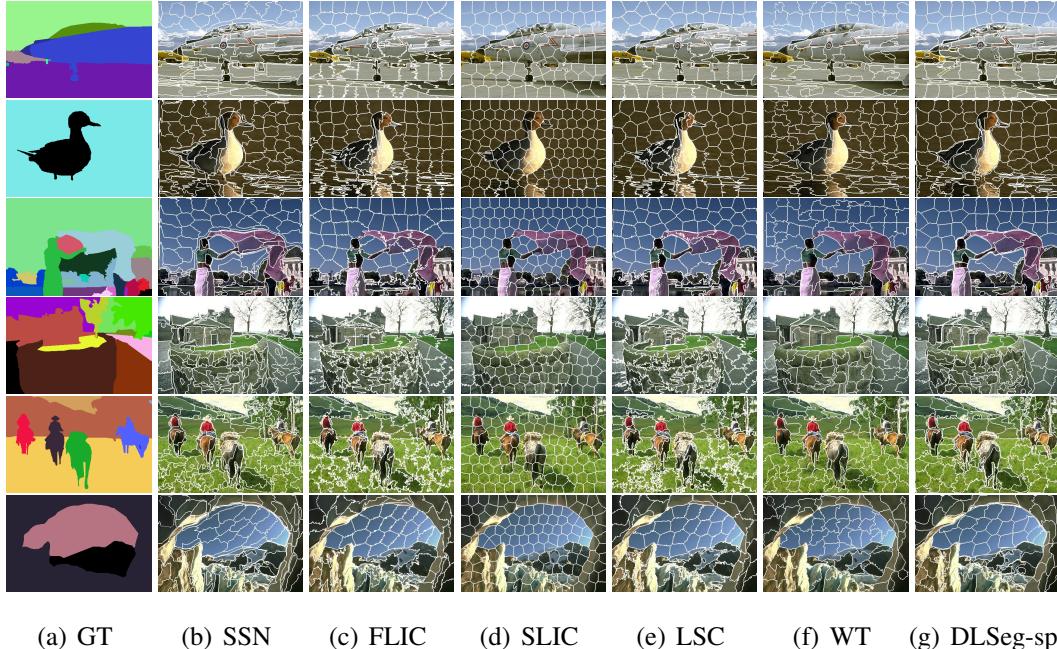


图 5-2 超像素分割。第一列显示来自BSDS500数据集的groundtruth。最后六列分别展示了由SSN、FLIC、SLIC、LSC、WT 和我们的方法生成的结果。

## 5.3 对比实验

### 5.3.1 对比算法

为了评估我们算法的有效性，我们将从超像素和图像分割两方面与一些最先进的分割算法进行比较，例如SSN<sup>[44]</sup>, FLIC<sup>[43]</sup>, SLIC<sup>[35]</sup>, LSC<sup>[39]</sup>, WT<sup>[36]</sup>, DEL<sup>[53]</sup>, SFFCM<sup>[65]</sup>, align-hier<sup>[66]</sup>, EGB<sup>[61]</sup>。其中，SSN, FLIC, SLIC, LSC和WT 是超像素算法。DEL, SFFCM, align-hier和EGB是图像分割算法。

### 5.3.2 实验结果以及数据分析

表 5-4 对比实验：超像素分割性能对比

Methods	BF( $\uparrow$ )	BR( $\uparrow$ )	UE( $\downarrow$ )	compactness( $\uparrow$ )
SSN	0.524	0.911	<b>0.060</b>	0.340
FLIC	0.485	0.845	0.141	0.249
SLIC	0.440	0.552	0.145	<b>0.661</b>
LSC	0.491	0.873	0.095	0.288
WT	0.518	0.837	0.124	0.438
DLSeg-sp	<b>0.547</b>	<b>0.918</b>	<u>0.065</u>	0.316

对于超像素分割和图像分割，我们将这两个任务与经典算法在BSDS500数据集进行了比较。图5-1显示了我们的算法与其他算法的比较。左图和中间图显示了我们算法的变体与其他图像分割算法之间的结果比较。可以看出，DLSeg算

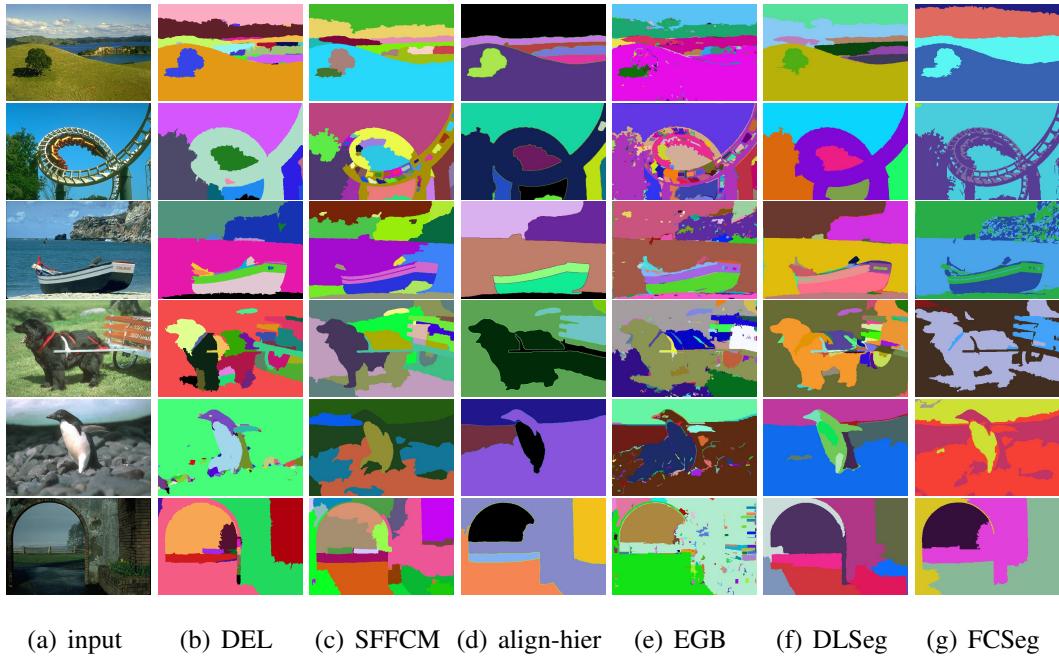


图 5-3 图像分割。第一列显示来自BSDS500数据集的原始图像和背景真相。最后五列分别显示了DEL、SFFCM、align-hier、EGB 和我们的方法生成的结果。

表 5-5 The performance of image segmentation of 4 variants

Methods	BF( $\uparrow$ )	PRI( $\uparrow$ )	GCE( $\downarrow$ )
DEL	<b>0.689</b>	0.809	0.161
SFFCM	0.622	0.776	0.259
align-hier	0.545	0.738	<b>0.141</b>
EGB	0.602	0.763	0.244
DLSeg	0.686	<b>0.821</b>	0.170
FCSSeg	0.594	0.720	0.241

法在图像分割中表现良好，并且在BF值和时间方面优于其他算法。尽管EGB算法花费的时间最少，却效果不佳。DEL算法在BF值上达到最佳，但在时间方面，较我们的算法有所欠缺。图5-1的最后一个子图显示了DLSeg方法中产生的超像素和其他超像素算法之间的比较。可以看出，DLSeg方法获得的超像素在BF值上实现了最佳性能。总而言之，在超像素分割和图像分割两方面，DLSeg算法实现了时间与效果之间的平衡。

表5-4和表5-5更详细的说明在超像素分割和图像分割两方面的定量比较（前两名分别以粗体和下划线突出显示）。与其他超像素分割算法相比，尽管DLSeg方法生成的超像素并不完全紧凑，但在BF和BR 指标上表现最佳，在UE上表现良好。此外，DLSeg生成的图像分割结果在BF以及PRI和GCE上均实现了良好的性能。从表5-4和表5-5可以看出，DLSeg算法在超像素生成和图像分割方面表现出色，并且可以生成与最先进算法相当的结果。由于DLSeg算法网络

是端到端可训练的，因此更适合于许多高级视觉任务。同时可以看到FCSeg算法的分割结果在边缘保持方面还有一定的欠缺。

我们在图5-2和图5-3中显示了一些定性比较。如图5-2所示，对于超像素分割，FLIC和LSC结果的边界是不规则的。SLIC生成的超像素虽然规则，但缺乏边界依从性。DLSeg的超像素结果可以获得更规则的边界和更好的边界依从性。如图5-3所示，对于图像分割，DEL，SFFCM和EGB的结果更加分散，并且align-hier方法的分割结果缺少一些细节。DLSeg的结果不那么分散，而且在视觉上更接近真值。FCSeg算法的分割结果在区域边界表现不是很好，需要进一步改进。

## 5.4 本章小结

本章根据第三章介绍的基于深度学习的超像素分割和图像分割方法和第四章中介绍的基于Boruvka算法和快速模糊C均值聚类的图像分割方法，进行实验分析。首先介绍了本文实验所使用的测试数据集BSDS500数据集；其次介绍了实验所使用的评估标准，包括超像素分割评估标准和图像分割评估标准；然后分别介绍第三章和第四章两个方法的消融实验，以找到最佳的模型结构和参数设置；最后与其他算法进行对比实验并进行分析，证明本文提出算法的有效性。



## 第6章 总结与展望

### 6.1 总结

社会快速发展，科技日新月异，人们的生活每天都有新的变化，不知不觉中人工智能产品已经进入人们的生活，成为生活中不可缺少的一部分。人工智能在改善人们生活的同时，也给未来产业的发展提供了无限的开拓空间。无论是政府，还是各大互联网企业，都十分重视人工智能技术的研究与发展，在不久的将来，人工智能领域会和人们的生活联系愈加紧密。

计算机视觉作为人工智能研究的一个重要方法，一直以来都受到众多研究人员的关注。而图像分割是计算机视觉重要的基础研究，对于后续的场景理解任务显得尤为重要。虽然已经被研究很多年，但仍然是计算机视觉中一个重要课题。

基于像素图像分割处理方法取得了不错的成果。但是随着拍照设备的不断升级，人们的需求不断增加，需要处理的图像的分辨率不断增大，越来越清晰。基于像素的传统图像分割方法处理分辨率高的图像，将花费更多的时间。超像素作为一种图像预处理技术极大减少了处理过程中的计算量和复杂度，而且更利于局部特征的提取与表达，更有利帮助定位区域的边界。

本文提出了两种基于超像素的图像分割算法，主要工作体现如下：

1) 提出了一种可以生成超像素和图像分割的端到端可训练网络。使用全卷积网络提取图像特征，然后使用可微分聚类算法模块生成精确的超像素。通过使用超像素池化操作获得超像素特征，并且计算两个相邻的超像素的相似度以确定是否合并以获得感测区域。该算法在BSDS500数据集上进行训练与测试。然后从超像素分割结果和图像分割，与最先进的已有算法进行对比实验，证明本文提出算法的高效性。

2) 提出了一种基于Boruvka算法和快速模糊C均值聚类的图像分割。在这项工作中，我们使用一种基于Boruvka算法来产生超像素图像。由于Boruvka算法可并行化的特性，可以快速高效的产生超像素。此外，在Boruvka算法计算连通分量过程中整合了局部信息，比SLIC和LSC等只利用每像素特征来确定聚类隶属关系的方法得到的超像素更加精确。在Boruvka算法获得的超像素图像的基础上，通过计算超像素图像的颜色直方图来实现快速模糊C均值聚类。由于超像素图像中不同颜色的数目远小于原始彩色图像，因此计算超像素图像的直方图非常容易。最后，以直方图作为目标函数的参数，实现彩色图像的快速分割。

## 6.2 展望

本文的主要研究内容是基于超像素的图像分割，图像分割研究是计算机视觉中一个重要的课题。本文所提出的两种图像分割算法的性能和效率等方面相比于其他算法有了一定的提升，但仍有不足。今后的研究工作以及需要进一步解决的问题主要包含：

- 1) 第三章提出的基于深度学习的超像素分割和图像分割虽然可以将超像素加入到深度学习中，但在学习超像素相似性和超像素融合过程中，方式过于简单，仅考虑局部相似性。所以今后考虑采用更加有效超像素融合方法，采用全局视角的规范化切割，应该会产生更一致的结果。此外，我们还计划探索算法在其他任务中的应用。
- 2) 第四章提出的基于Boruvka算法和快速模糊C均值聚类的图像分割算法，利用无监督的方法实现图像分割，省去了训练过程，但其性能方面仍存在很多不足。此外，在使用过程中需要预先初始化聚类中心数量，且初始参数对最终结果有较大的影响。所以在今后的研究可以考虑通过算法，自动学习计算聚类中心数量，减少初始化参数对算法结果的影响。

## 参考文献

- [1] Wu B, Ai H, Huang C, et al. Fast Rotation Invariant Multi-View Face Detection Based on Real Adaboost [C]. In IEEE International Conference on Automatic Face and Gesture Recognition, 2004: 79–84.
- [2] Chen D, Cao X, Wen F, et al. Blessing of Dimensionality: High-Dimensional Feature and Its Efficient Compression for Face Verification [C]. In IEEE Computer Vision and Pattern Recognition, 2013: 3025–3032.
- [3] Cao X, Wei Y, Wen F, et al. Face alignment by Explicit Shape Regression [C]. In IEEE Computer Vision and Pattern Recognition, 2012: 2887–2894.
- [4] Al-Shawabka A, Restuccia F, D’Oro S, et al. Exposing the Fingerprint: Dissecting the Impact of the Wireless Channel on Radio Fingerprinting [C]. In IEEE Computer Communications, 2020: 646–655.
- [5] Gupta R, Khari M, Gupta D, et al. Fingerprint image enhancement and reconstruction using the orientation and phase reconstruction [J]. Inf. Sci., 2020, 530: 201–218.
- [6] Farabet C, Couprie C, Najman L, et al. Learning hierarchical features for scene labeling [J]. IEEE Trans. Pattern Anal. Mach. Intell., 2012, 35 (8): 1915–1929.
- [7] Xiao T, Liu Y, Zhou B, et al. Unified Perceptual Parsing for Scene Understanding [C]. In Computer Vision - ECCV, 2018: 432–448.
- [8] Pont-Tuset J, Arbelaez P, Barron J T, et al. Multiscale combinatorial grouping for image segmentation and object proposal generation [J]. IEEE Trans. Pattern Anal. Mach. Intell., 2016, 39 (1): 128–140.
- [9] Yan P, Xu S, Rastinehad A R, et al. Adversarial Image Registration with Application for MR and TRUS Image Fusion [C]. In Machine Learning in Medical Imaging, 2018: 197–204.
- [10] Fan J, Cao X, Yap P, et al. BIRNet: Brain image registration using dual-supervised fully convolutional networks [J]. Medical Image Anal., 2019, 54: 193–206.
- [11] Yan J, Yu Y, Zhu X, et al. Object detection by labeling superpixels [C]. In IEEE Conference on Computer Vision and Pattern Recognition, 2015: 5107–5116.
- [12] Mayank Juneja C V J A Z, Andrea Vedaldi. Blocks That Shoot: Distinctive Parts for Scene Classification [C]. In IEEE Conference on Computer Vision and Pattern Recognition, 2013: 923–930.
- [13] Conze P-H, Tilquin F, Lamard M, et al. Unsupervised learning-based long-term superpixel tracking [J]. Image Vis. Comput., 2019, 89: 289–301.

- [14] Zhu W, Liang S, Wei Y, et al. Saliency Optimization from Robust Background Detection [C]. In IEEE Computer Vision and Pattern Recognition, 2014.
- [15] Kohli P, Torr P H, et al. Robust higher order potentials for enforcing label consistency [J]. Int. J. Comput. Vis., 2009, 82 (3): 302–324.
- [16] Shu G, Dehghan A, Shah M. Improving an Object Detector and Extracting Regions Using Superpixels [C]. In IEEE Conference on Computer Vision and Pattern Recognition, 2013: 3721–3727.
- [17] Gould S, Rodgers J, Cohen D, et al. Multi-class segmentation with relative location prior [J]. Int. J. Comput. Vis., 2008, 80 (3): 300–316.
- [18] Gadde R, Jampani V, Kiefel M, et al. Superpixel Convolutional Networks Using Bilateral Inceptions [C]. In Computer Vision - ECCV, 2016: 597–613.
- [19] Sharma A, Tuzel O, Liu M. Recursive Context Propagation Network for Semantic Scene Labeling [C]. In Advances in Neural Information Processing Systems, 2014: 2447–2455.
- [20] Xie X, Xie G, Xu X, et al. Automatic image segmentation with superpixels and image-level labels [J]. IEEE Access, 2019, 7: 10999–11009.
- [21] Wang S, Lu H, Yang F, et al. Superpixel tracking [C]. In IEEE International Conference on Computer Vision, 2011: 1323–1330.
- [22] Yang F, Lu H, Yang M-H. Robust superpixel tracking [J]. IEEE Trans. Image Process., 2014, 23 (4): 1639–1651.
- [23] He S, Lau R W, Liu W, et al. Supercnn: A superpixelwise convolutional neural network for salient object detection [J]. Int. J. Comput. Vis., 2015, 115 (3): 330–344.
- [24] Yang C, Zhang L, Lu H, et al. Saliency Detection via Graph-Based Manifold Ranking [C]. In IEEE Computer Vision and Pattern Recognition, 2013: 3166–3173.
- [25] Abu-Jamous B, Fa R, Nandi A K. Integrative cluster analysis in bioinformatics [M]. John Wiley & Sons, 2015.
- [26] Zeng S, Wang X, Cui H, et al. A unified collaborative multikernel fuzzy clustering for Multiview data [J]. IEEE Trans. Fuzzy Syst., 2017, 26 (3): 1671–1687.
- [27] Ma J, Li S, Qin H, et al. Unsupervised multi-class co-segmentation via joint-cut over  $L_{\{1\}}$ -manifold hyper-graph of discriminative image regions [J]. IEEE Trans. Image Process., 2016, 26 (3): 1216–1230.
- [28] Uijlings J R, Van De Sande K E, Gevers T, et al. Selective search for object recognition [J]. Int. J. Comput. Vis., 2013, 104 (2): 154–171.
- [29] Bai M, Urtasun R. Deep Watershed Transform for Instance Segmentation [C]. In IEEE Conference on Computer Vision and Pattern Recognition, 2017: 2858–2866.

- [30] Pereyra M, McLaughlin S. Fast unsupervised Bayesian image segmentation with adaptive spatial regularisation [J]. *IEEE Trans. Image Process.*, 2017, 26 (6): 2577–2587.
- [31] Long J, Shelhamer E, Darrell T. Fully convolutional networks for semantic segmentation [C]. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015: 3431–3440.
- [32] Ronneberger O, Fischer P, Brox T. U-Net: Convolutional Networks for Biomedical Image Segmentation [C]. In *Medical Image Computing and Computer-Assisted Intervention*, 2015: 234–241.
- [33] Cheng M, Liu Y, Hou Q, et al. HFS: Hierarchical Feature Selection for Efficient Image Segmentation [C]. In *Computer Vision - ECCV*, 2016: 867–882.
- [34] Krizhevsky A, Sutskever I, Hinton G E. ImageNet Classification with Deep Convolutional Neural Networks [C]. In *Advances in Neural Information Processing Systems*, 2012: 1106–1114.
- [35] Achanta R, Shaji A, Smith K, et al. SLIC superpixels compared to state-of-the-art superpixel methods [J]. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2012, 34 (11): 2274–2282.
- [36] Hu Z, Zou Q, Li Q. Watershed superpixel [C]. In *IEEE International Conference on Image Processing*, 2015: 349–353.
- [37] Ren X, Malik J. Learning a Classification Model for Segmentation [C]. In *IEEE International Conference on Computer Vision*, 2003: 10–17.
- [38] Shi J, Malik J. Normalized Cuts and Image Segmentation [J]. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2000, 22 (8): 888–905.
- [39] Li Z, Chen J. Superpixel segmentation using Linear Spectral Clustering [C]. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2015: 1356–1363.
- [40] Macqueen J. Some methods for classification and analysis of multivariate observations [J]. *Proc. Symp. Math. Statist. and Probability*, 1967, 1.
- [41] Wei X, Yang Q, Gong Y, et al. Superpixel Hierarchy [J]. *IEEE Trans. Image Process.*, 2018, 27 (10): 4838–4849.
- [42] Gross J L, Yellen J, Beineke L W, et al. Introduction to Graphs [M] // Gross J L, Yellen J, Beineke L W, et al. *Handbook of Graph Theory, Discrete Mathematics and Its Applications*. Chapman & Hall / Taylor & Francis, 2003: 2003: 1–55.
- [43] Zhao J, Ren B, Hou Q, et al. FLIC: Fast Linear Iterative Clustering With Active Search [C]. In *AAAI Conference on Artificial Intelligence*, 2018: 7574–7581.
- [44] Jampani V, Sun D, Liu M, et al. Superpixel Sampling Networks [C]. In *Computer Vision - ECCV*, 2018: 363–380.
- [45] Comaniciu D, Meer P. Mean shift: A robust approach toward feature space analysis [J]. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2002, 24 (5): 603–619.

- [46] Zaixin Z, Lizhi C, Guangquan C. Neighbourhood weighted fuzzy c-means clustering algorithm for image segmentation [J]. IET Image Process, 2013, 8 (3): 150–161.
- [47] Gong M, Liang Y, Shi J, et al. Fuzzy c-means clustering with local information and kernel metric for image segmentation [J]. IEEE Trans. Image Process., 2012, 22 (2): 573–584.
- [48] Zhang H, Wang Q, Shi W, et al. A novel adaptive fuzzy local information  $c$ -means clustering algorithm for remotely sensed imagery classification [J]. IEEE Trans. Geoscience and Remote Sensing, 2017, 55 (9): 5057–5068.
- [49] Chaibou M S, Conze P-H, Kalti K, et al. Learning contextual superpixel similarity for consistent image segmentation [J]. Multimed. Tools. Appl., 2020, 79 (3-4): 2601–2627.
- [50] Ahn J, Kwak S. Learning Pixel-Level Semantic Affinity With Image-Level Supervision for Weakly Supervised Semantic Segmentation [C]. In IEEE Conference on Computer Vision and Pattern Recognition, 2018: 4981–4990.
- [51] Carpineto C, Romano G. Consensus clustering based on a new probabilistic rand index with application to subtopic retrieval [J]. IEEE Trans. Pattern Anal. Mach. Intell., 2012, 34 (12): 2315–2326.
- [52] Kwak S, Hong S, Han B. Weakly Supervised Semantic Segmentation Using Superpixel Pooling Network [C]. In AAAI, 2017: 4111–4117.
- [53] Liu Y, Jiang P, Petrosyan V, et al. DEL: Deep Embedding Learning for Efficient Image Segmentation [C]. In International Joint Conference on Artificial Intelligence, 2018: 864–870.
- [54] Arbelaez P, Maire M, Fowlkes C, et al. Contour detection and hierarchical image segmentation [J]. IEEE Trans. Pattern Anal. Mach. Intell., 2010, 33 (5): 898–916.
- [55] Hinton G E, Osindero S, Teh Y W. A Fast Learning Algorithm for Deep Belief Nets [J]. Neural Comput., 2006, 18 (7): 1527–1554.
- [56] Krizhevsky A, Sutskever I, Hinton G E. ImageNet Classification with Deep Convolutional Neural Networks [C]. In Advances in Neural Information Processing Systems, 2012: 1106–1114.
- [57] Mikolov T, Chen K, Corrado G, et al. Efficient Estimation of Word Representations in Vector Space [C]. In International Conference on Learning Representations, 2013.
- [58] Lecun Y, Bottou L. Gradient-based learning applied to document recognition [J]. Proceedings of the IEEE, 1998, 86 (11): 2278–2324.
- [59] Ioffe S, Szegedy C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift [C]. In Proceedings of the 32nd International Conference on Machine Learning, ICML, 2015: 448–456.
- [60] Wu Y, He K. Group Normalization [C]. In Computer Vision - ECCV, 2018: 3–19.

- [61] Felzenszwalb P F, Huttenlocher D P. Efficient graph-based image segmentation [J]. *Int. J. Comput. Vis.*, 2004, 59 (2): 167–181.
- [62] Jia Y, Shelhamer E, Donahue J, et al. Caffe: Convolutional Architecture for Fast Feature Embedding [C]. In ACM International Conference on Multimedia, MM, 2014: 675–678.
- [63] Kingma D P, Ba J. Adam: A Method for Stochastic Optimization [C]. In International Conference on Learning Representations, 2015.
- [64] Khelifi L, Mignotte M. GCE-based model for the fusion of multiples color image segmentations [C]. In IEEE International Conference on Image Processing, 2016: 2574–2578.
- [65] Lei T, Jia X, Zhang Y, et al. Superpixel-based Fast fuzzy C-means clustering for color image segmentation [J]. *IEEE Trans. Fuzzy Systems*, 2019, 27 (9): 1753–1766.
- [66] Chen Y, Dai D, Pont-Tuset J, et al. Scale-Aware Alignment of Hierarchical Image Segmentation [C]. In IEEE Conference on Computer Vision and Pattern Recognition, 2016: 364–372.
- [67] Wang K, Li L, Zhang J. End-to-end trainable network for superpixel and image segmentation [J]. *Pattern Recognition Letters*, 2020, 140: 135–142.



## 发表论文和参加科研情况说明

### (一) 发表的学术论文

- [1] Wang K , Li L , Zhang J . End-to-end trainable network for superpixel and image segmentation[J]. Pattern Recognition Letters, 2020.140:135-142.

### (二) 申请及已获得的专利

- [1] 李亮, 王凯, 李亚军, 彭俊杰. 基于深度学习进行超像素生成和图像分割的方法: 中国, 2020110118787 [P].2020-09-23 .



## 致 谢

本论文的工作是在我的导师李亮老师的悉心指导下完成的，李亮老师严谨的治学态度和科学的工作方法给了我极大的帮助和影响。在此衷心感谢三年来李亮老师对我的关心和指导。

李亮老师悉心指导我们完成了实验室的科研工作，在学习上和生活上都给予了我很大的关心和帮助，在此向李亮老师表示衷心的谢意。

李亮老师对于我的科研工作和论文都提出了许多的宝贵意见，在此表示衷心的感谢。另外也感谢家人，他们的理解和支持使我能够在学校专心完成我的学业。