基于深度学习的超像素分割和图像分割

A Study on Indoor Localization Based on fingerprinting and inertial measurements

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

天津大学智能与计算学部
二零一六年十月

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

图像分割和超像素生成已经被研究很多年，但仍然是计算机视觉中一个重要的课题。虽然很多先进的计算机视觉的算法已经被用于图像分割和超像素生成方面，但是没有端到端可训练的算法来实现同时产生超像素和图像分割。对此，我们提出了一个端到端的可训练网络，它可以同时产生超像素和进行图像分割。我们的网络架构如图3-1所示。我们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，我们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。我们使用BSDS500数据集训练网络，并将我们的结果与最先进的结果进行比较，并得到了很好的结果。

本文提出的网络具有以下特点：

- 我们的网络是端到端可训练的，可以很容易地组装成其他深层网络结构，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，这是更有效的。与现有算法相比，该算法具有优良的性能和更高的精度。

关键词： 图像分割，超像素，深度学习

ABSTRACT

Image segmentation and superpixel generation have been studied for many years, and they are still active research topics in computer vision. Although many advanced computer vision algorithms have been used for image segmentation and superpixel generation, there is no end-to-end trainable algorithm that generates superpixels and segment images simultaneously. Specifically, we propose an end-to-end trainable network which can generate superpixels and perform image segmentation simultaneously. The architecture of our network is shown in Fig. 3-1. We use a fully convolutional network and the iterative differentiable clustering algorithm to obtain superpixels. Next, we adopt the superpixel pooling layer to get the superpixel features, with which the similarity between adjacent superpixels can be calculated. If the similarity is greater than the preset threshold, we merge them according to a simple procedure to get object segments. We train the network using BSDS500 segmentation benchmark dataset, compare our result with state-of-the-art ones and get good results.

The proposed network has the following characteristics:

- Our network is end-to-end trainable and can be easily as sembled into other deep network structures for subsequent applications.
- Our network can generate superpixels and get segmentation results, which is more efficient. The proposed algorithm has excellent performance and has higher precision than existing algorithms.

KEY WORDS: Image segmentation, Superpixel, Deep learning

目　录

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，图像分割作为许多不同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，指纹识别，交通控制系统，在卫星图像中定位物体（道路、森林等），行人检测，医学影像等。

所谓的图像分割根据灰度、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。简单的说就是在一副图像中，把目标从背景中分离出来。就是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、目标检测/识别等领域。

与图像分割产生的大的感知区域不同，超像素分割对输入图像进行过分割。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。由于超像素在表示和计算方面的高效性，超像素已广泛应用于计算机视觉算法，如对象检测、语义分割、目标跟踪以及显著性估计。

图像分割已有多年的研究历史，分割算法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、图割、分水岭变换、隐马尔可夫随机场等。这种算法只需要很少的训练样本和标签图像。随着深度学习的发展，有监督的方法，如FCN、U-NET、HFS等，利用CNN网络进行特征学习，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如SLIC,WT，是不可微的。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。除了一些方法，超像素几乎不与深度网络结合使用。

在本文中，我们提出了一个端到端的可训练网络，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，我们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，得到分割结果。

(a) 原图　　(b) 图像分割　　(c) 超像素分割

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介绍。

1. 基于图论法

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。

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，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，并定义了目标函数，包括纹理特征和轮廓特征，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(LSC)，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度量目标函数, 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 Wei等人提出Superpixel Hierarchy，利用Boruvka 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，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。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。

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（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(FLIC)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，SLIC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

素几乎不与现代深度网络结合使用。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，因此如果在深度网络中使用超像素，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，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，可以生成边界重合度高的超像素。

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，我们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，Fuzzy C-means (FCM)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，一般只是采用距离或颜色特征，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，我们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。在（Ahn and Kwak，2018）中，作者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 et al.，2020）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（SPN）提出的超像素池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布局来反映底层图像结构，用于学习和推断语义分割。由SPN生成的初始注释被用来学习另一个神经网络来估计像素语义标签。DEL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SFFCM，其算法通过多尺度形态梯度重建（MMGR）和分水岭变换（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。

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，并简单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，然后介绍了对SLIC 的改进，并利用提出的损失函数进行超像素分割。

第4章介绍了在第3章产生超像素的基础上，进行计算超像素相似度以及超像素融合步骤。

第5章首先说明本文使用的实验环境及对比介绍本文使用的深度学习框架，其次介绍本文实验中的公开数据集及评定标准；最后对本文介绍的网络模型以及改进的网络模型进行实验及对比分析。

第6章对全文进行总结，并对未来基于卷积神经网络的图像分割和超像素分割方法的发展进行了展望。

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

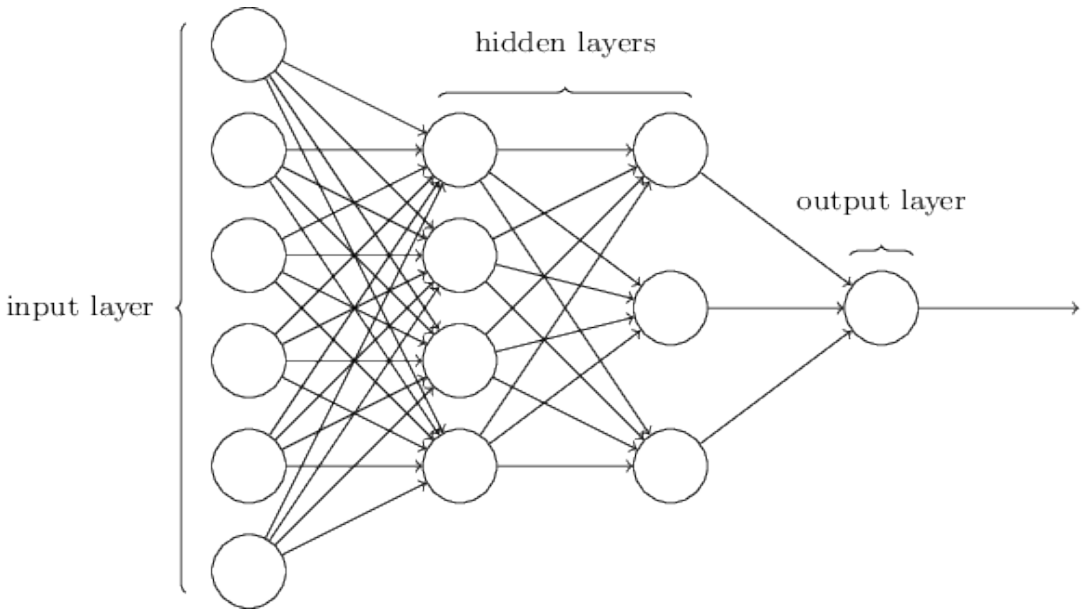


图 2-1　　多层感知器示意图

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*l*层，多层感知器的前向传播公式如下：

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

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。

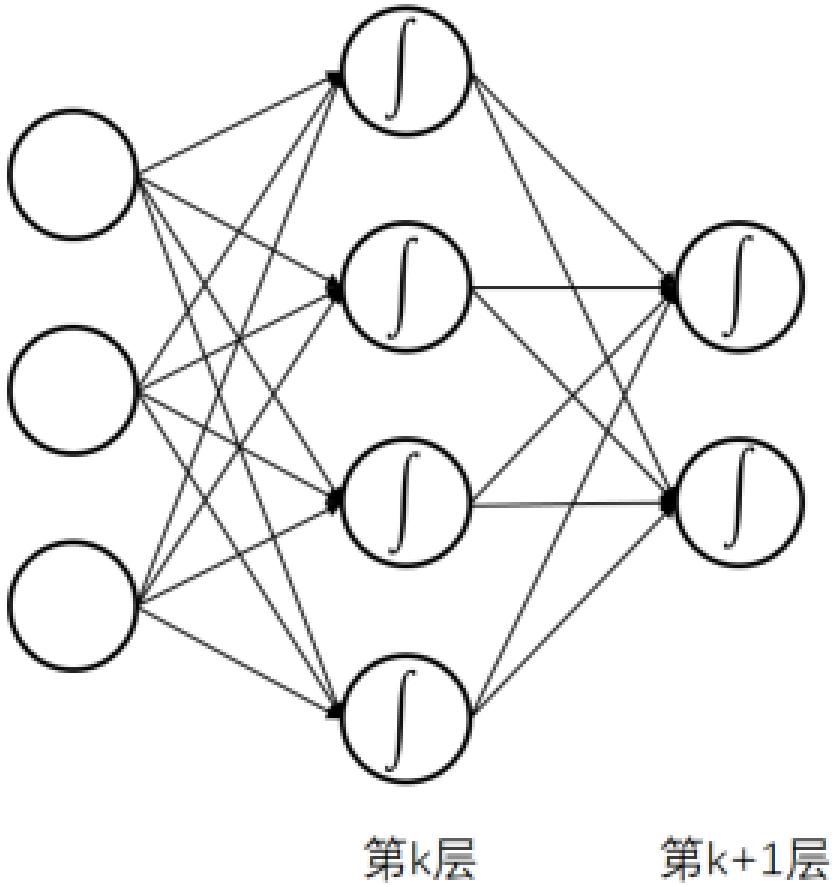


图 2-2　　前向传播示意图

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模型的出现，以卷积神经网络(convolutional neural network)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，来着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。是的在人脸识别领域，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，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》，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 Networks）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在1994年提出的LeNet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。

2.2.1 卷积层

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2-2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$M_w x N_w$，输入矩阵的大小为$M_x x N_x$,那么输出矩阵的大小等于$(M_x - M_w + 1)x(N_x - N_w + 1)$。以图2-3为例，一个3x3的卷积核在一个5x5的输入图像上以步长为1做卷积计算，得到3x3的输出矩阵，这个输出矩阵就是特征矩阵。

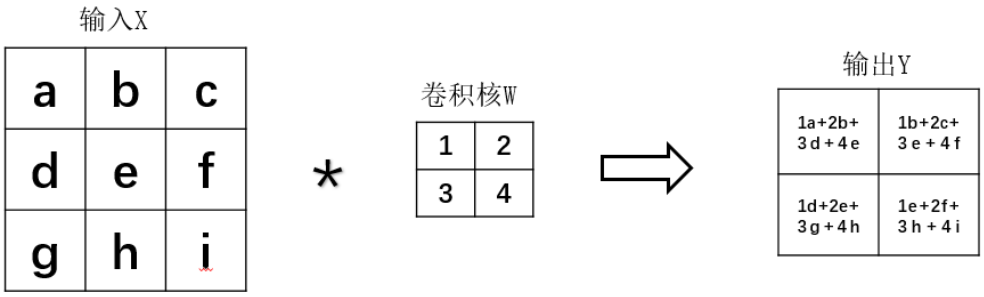


图 2-3　　卷积计算过程示意图

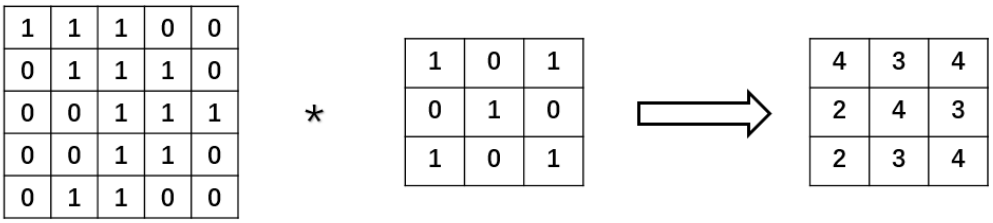


图 2-4　　卷积操作示意图

2.2.2　激活层

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，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BP算法在优化深层神经网络时的梯度耗散问题，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。

2.2.3 池化层

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2-3所示，在一个4*4的矩阵上，选用2*2的filters，步长为2，对于每个2 * 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 * 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。

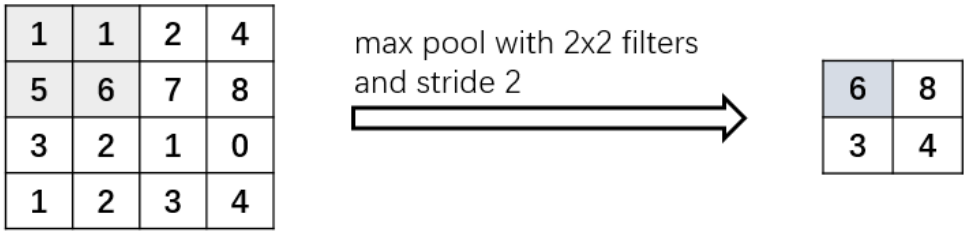


图 2-5　　池化操作示意图

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，因此我们依然可以能判断出这是一只狗。批归一化（Batch Normalization,BN）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Group Normalization，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 基于神经网络的图像分割

《基于深度学习的RGB-D图像语义分割》《FCN》http://www.sjocr.com/news/915.html

2.4 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，更有利于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)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Superpixel可以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2-2所示：假设$P_i = \left\{p_i^k\right\} k = 1, 2, \cdots, K_i$ 表示图像中第i个Superpixel,K_i表示第i个Superpixel 中像素的个数，对于每个Superpixel i，feature vector 的生成公式:

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

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2-2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。

第3章 超像素分割

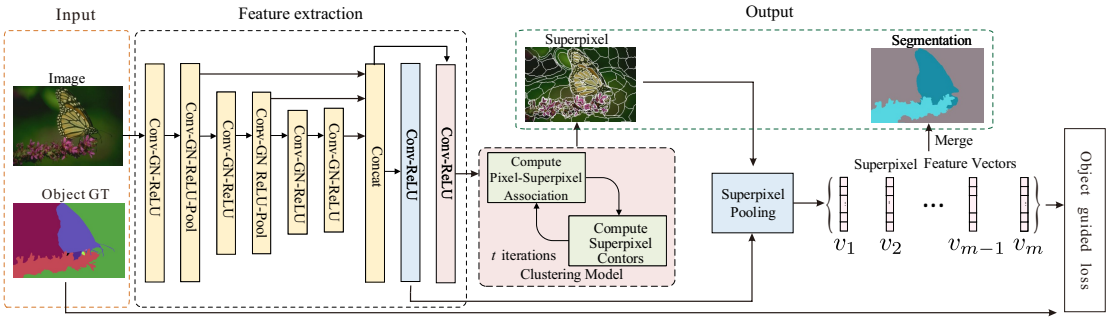


图 3-1 算法流程图。对于给定的图像，我们的算法同时生成超像素和图像分割。输入图像首先被输送到一个特征提取网络，该网络由一系列卷积层、归一化(GN)和ReLU操作组成。然后将提取的特征输入可微聚类模块，生成超像素。超像素池用于获取超像素特征向量。最后通过合并相似度高的超像素实现图像分割

如图3-1所示，我们的方法首先从CNN深度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，我们将详细介绍我们的方法。

3.1 超像素生成

超像素将相似像素分组为各向同性区域，从而可以提高分割质量和效率。例如，在DEL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。在介绍在描述软关联之前，我们简要介绍SLIC算法中如何计算像素-超像素硬关联$H = \{1, 2, \cdots, m\}^{n\times 1}$。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，

$$H_p^t = \underset{i\in\{1,2,\cdots,m\}}{\operatorname{argmin}} \left\| I_p - C_i^{t-1} \right\|_2, \tag{3-1}$$

其中$\|\cdot\|_2$表示输入向量的ℓ_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(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：

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

其中F_p是像素p的深层特征。在我们的情况下，它来自我们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，

$$C_i^{t} = \frac{1}{Z_i^{t}} \sum_{p} Q_{pi}^{t} F_p, \tag{3-3}$$

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(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，

$$H_p' = \underset{i \in \{1,\cdots,m\}}{\operatorname{argmax}} Q_{pi}. \tag{3-4}$$

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。

第4章　图像分割

4.1　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。其背后的原因是，超像素不包含语义信息，而图像分割却包含语义信息，因此这两种任务的特征是不同的。我们在实验中通过比较超像素和图像分割的结果以及该算法的一个变体（ours-conv7）来验证这一思想。参考第4.1了解更多细节。

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$\mathcal{S} = \{S_1, S_2, \cdots, S_m\}$，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$\{v_1, v_2, \cdots, v_m\}$。超像素池化操作表示如下：

$$v_i = \frac{1}{|S_i|} \sum_{p \in S_i} F'_p, \tag{4-1}$$

其中，F'_p 表示在超像素池中使用的像素p的特征向量。

相邻超像素i和超像素j的相似度可以由如下公式获得：

$$s_{ij} = \frac{2}{1 + \exp\left(\left\|v_i - v_j\right\|_1\right)}, \tag{4-2}$$

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。

4.2　损失函数

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（3-6）中定义的相似性度量，我们定义的损失函数如下：

$$\begin{aligned} L = -\sum_{S_i \in S} \sum_{S_j \in \mathcal{R}_i} \Big[& (1-\alpha) \cdot l_{ij} \cdot \log\left(s_{ij}\right) \\ & + \alpha \cdot \left(1 - l_{ij}\right) \cdot \log\left(1 - s_{ij}\right) \Big], \end{aligned} \tag{4-3}$$

其中$\mathcal{R}_i$是超像素S_i的相邻的超像素集合，S_i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_i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α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。通过反向传播，图像分割也指导了超像素分割。

4.3 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;

$\mathcal{S}=\{S_1, S_2, \cdots, S_m\}$: superpixels.

Output: Segmentation $\mathcal{S}$.

for each $S_i \in \mathcal{S}$ **do**
  Construct adjacent superpixel set $\mathcal{R}_i \subset \mathcal{S}$ of S_i;
  for each $S_j \in R_i$ **do**
    if $s_{ij} > T$ **then**
      $S_i \leftarrow S_i \cup S_j, \mathcal{S} \leftarrow \mathcal{S} \setminus S_j$;
      Update $\mathcal{R}_i$;
    end if
  end for
end for

4.4 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、组规范化(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3×3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处理规范化(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，其精度在较大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supernet。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3×3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3×3卷积运算，然后使用1×1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。

4.5 实施细节

我们基于Caffe框架来搭建神经网络，这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。所有代码都是用C++和Python包装编写的。

对于超级像素的生成，就像在原始的SLIC算法中一样，我们在(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201×201大小的图像块，进行数据增强。采用Adam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。如第3.5，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。

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

公开数据数据集berkeley segmentation data set(BSDS500)是伯利克里computer vision group 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，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，过度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。在本小结介绍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，本文使用了Boundary F-measure(BF), Probabilistic Rand Index (PRI) 和Global Consistency Error (GCE) 作为图像分割的主要指标。用BF, Boundary Recall (BR), under segmentation error (UE) 和compactness 作为超像素分割的主要指标。我们基于整个数据集规模的整体性能选择最佳参数。BF，BR，PRI和compactness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BP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：

$$BR = \frac{TP}{TP + FN} \tag{5-1}$$

（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：

$$BP = \frac{TP}{TP + FP} \tag{5-2}$$

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：

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

α 为1时，就是常见的F1值（F1 score）：

$$F = \frac{2 \cdot P \cdot R}{P + R} \tag{5-4}$$

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。

（3）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$S^t = \left\{C_1^t, C_2^t, \cdots, C_{R^t}^t\right\}$,groundtruth分割图表示为$S^g = \left\{C_1^g, C_2^g, \cdots, C_{R^g}^g\right\}$。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pi，我们考虑St和Sg中包括该像素的段。我们分别用Ct ¡pi¿和Cg ¡pi¿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pi处表示为：

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

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：

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

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。

（4）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$S^t = \left\{C_1^t, C_2^t, \cdots, C_{R^t}^t\right\}$，groundtruth 分割图

表示为$S^g = \left\{C_1^g, C_2^g, \cdots, C_{R^g}^g\right\}$。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：

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

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：

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

两个像素在S^t和S^g中的不同区域中的概率是：

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

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：

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

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：

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

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：

$$w_h = -\log_2(p_h) \tag{5-12}$$

这些权重可用于定义Probabilistic Rand Index：

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

与RI相比，使用PRI的优点在于，对各项进行了权衡。

（4）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实值对图像进行分割时所犯的错误。其计算公式为：

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

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$\left|S_j\right|$的5%,以解决真值分割数据中的小错

误。表达式$S_j \bigcap g_i$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。

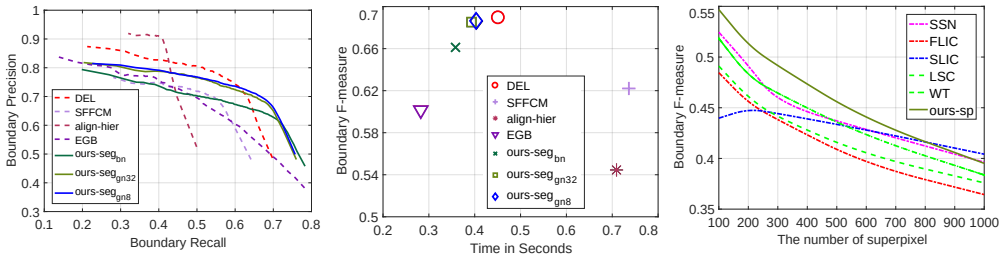
（5）compactness：紧凑度，指标衡量了超像素是否“紧实”。

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：

$$Q_s = \frac{A_s}{A_c} = \frac{4\pi As}{L_s^2} \tag{5-15}$$

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$S = \{S_1, S_2, \cdots, S_m\}$，分割结果的紧凑度为：

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



(a) (b) (c)

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

2. ours-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. ours-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.ours-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。

在下面一节中，我们将评估图像分割和超像素分割的结果。

5.2.2 实验结果以及数据分析

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

Methods	BF(↑)	BR(↑)	UE(↓)	compactness(↑)
ours-BN	0.547	0.884	0.068	0.373
ours-GN32	0.546	0.897	0.066	0.376
ours-conv7	0.521	0.812	0.094	0.413
ours-w/o-concat	0.521	0.895	0.071	0.340
ours-GN8	0.547	0.918	0.065	0.316

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

Methods	BF(↑)	PRI(↑)	GCE(↓)
ours-BN	0.661	0.807	0.146
ours-GN32	0.685	0.820	0.171
ours-conv7	0.492	0.714	0.088
ours-w/o-concat	0.560	0.817	0.182
ours-GN8	0.686	0.822	0.170

从表1和表2可知，相对于其他4种变体而言，ours-GN8模型性能表现最好，验证了我们选择组件的合理性。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。但是当group number过大的时候，效果有所下降。根据我们的经验，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ours-GN8模型比ours-w/o-concat 模型包含更多提取到的特征，因此分割效果更好。ours-conv7的分割效果明显降低，ours-GN8远远好于ours-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，例如UPerNet。

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和效率，我们将与一些最先进的分割算法进行比较，例如SSN，FLIC，SLIC，LSC，WT，DEL，SFFCM，align-hier，EGB。

(a) GT (b) SSN (c) FLIC (d) SLIC (e) LSC (f) WT (g) ours

图 5-2 超像素生成。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。

其中，SSN，FLIC，SLIC，LSC和WT是超像素算法。DEL，SFFCM，alignhier和EGB是图像分割算法。

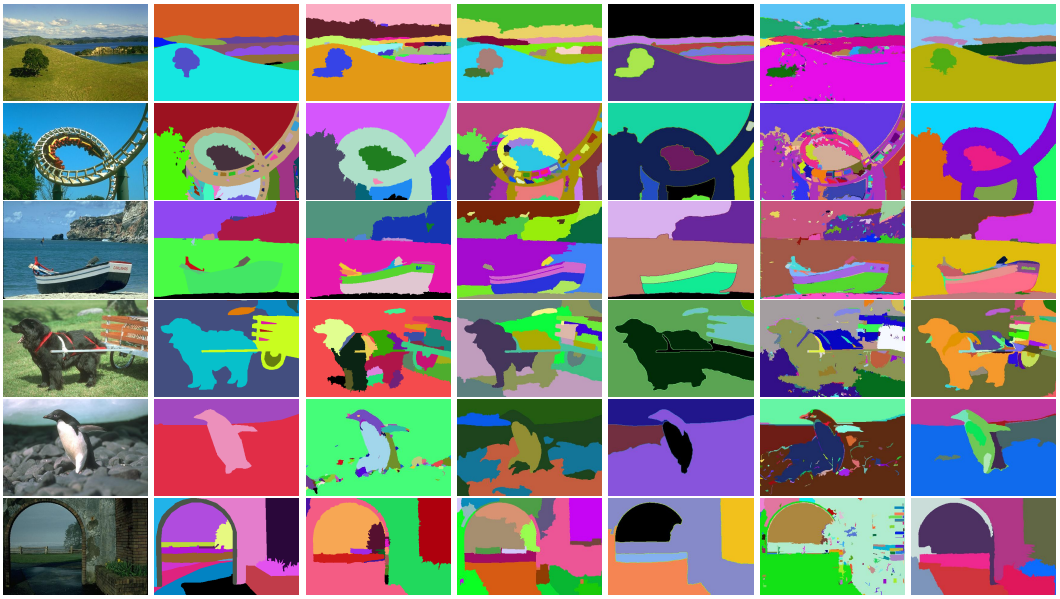
5.3.2 实验结果以及数据分析

表 5-3 对比实验：超像素分割性能对比

Methods	BF(↑)	BR(↑)	UE(↓)	compactness(↑)
SSN	0.524	0.911	**0.060**	0.340
FLIC	0.485	0.845	0.141	0.249
SLIC	0.440	0.552	0.145	**0.661**
LSC	0.491	0.873	0.095	0.288
WT	0.518	0.837	0.124	0.438
ours-sp	**0.547**	**0.918**	0.065	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2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，我们的算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，但比我们的算法慢。图2的最后一个子图显示了我们的超像素和其他超像素算法之间的比较。可以看出，我们获得的超像素在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，我们的算法实现了时间与效果之间的平衡。

表3和表4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



(a) input (b) groundtruth (c) DEL (d) SFFCM (e) align-hier (f) EGB (g) ours-seg

图 5-3 图像分割。前两列显示来自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 5-4 The performance of image segmentation of 4 variants

Methods	BF(↑)	PRI(↑)	GCE(↓)
DEL	**0.689**	0.809	0.161
SFFCM	0.622	0.776	0.259
align-hier	0.545	0.738	**0.141**
EGB	0.602	0.763	0.244
ours-seg	0.686	**0.821**	0.170

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，尽管我们生成的超像素并不完全紧凑，但我们的算法在BF和BR指标上表现最佳，在UE上表现良好。此外，我们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3和表4可以看出，我们的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。此外由于我们的网络是端到端可训练的，因此这使我们的算法更适合于许多高级视觉任务。

我们在图3和图4中显示了一些定性比较。如图3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。我们的结果可以获得更规则的边界和更好的边界依从性。如图4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。我们的结果不那么分散，而且在视觉上更接近真值。综上所述，可以看出我们的算法在超像素分割和图像分割方面都可以达到良好的水

平。

第6章 总结与展望

在本文中，我们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。具体来说，我们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。

我们提出的算法在超像素生成和图像分割方面都取得了良好的性能。此外，由于该算法是端到端可训练的，因此可以很容易地集成到其他深度网络结构中。它使我们的算法有潜力应用于许多其他视觉任务。

在将来的工作中，我们计划尝试其他合并算法以获取分割结果。本文中使用的合并过程相对简单，仅考虑局部相似性。其他一些程序，例如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。

发表论文和参加科研情况说明

（一）发表的学术论文

[1] XXX，XXX. Density and Non-Grid based Subspace Clustering via Kernel Density Estimation[C]. ECML-PKDD 2012, Bristol, UK.(Submitted, Under review)

[2] XXX，XXX. A tree parent storage based on hashtable for XML construction[C]. Communication Systems, Networks and Applications, Hongkong, 2010: 325-328. (EI DOI: 10.1109/ICCSNA.2010.5588732)

（二）申请及已获得的专利（无专利时此项不必列出）

[1] XXX，XXX. XXXXXXXXX：中国，1234567.8[P]. 2012-04-25.

（三）参与的科研项目

[1] XXX，XXX. XX 信息管理与信息系统, 国家自然科学基金项目.课题编号：XXXX.

致 谢

天津大学学位论文 LaTeX 模板主要参考以下内容：

- 哈尔滨工业大学 PlutoThesis 硕博士学位论文模板
- 武汉理工大学学位论文 WHUTThesis 模板
- 中科院 CASthesis 模板

感谢 ChinaTeX 大神的无私帮助。

谨将此论文模板，献给我们最爱的母校：天津大学。

本论文的工作是在我的导师[XXXX...] 教授的悉心指导下完成的，[XXXX...] 教授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[XXXX...] 老师对我的关心和指导。

[XXXX...] 教授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[XXXX...] 老师表示衷心的谢意。

[XXXX...] 教授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。

在实验室工作及撰写论文期间，[XXXX...] 、[XXXX...] 等同学对我论文中的[XXXX...] 研究工作给予了热情帮助，在此向他们表达我的感激之情。

另外也感谢家人[XXXX...] ，他们的理解和支持使我能够在学校专心完成我的学业。