中图分类号: 学校代码: 10055
UDC: 密级: 公开

南開大學

博 士 学 位 论 文

知识引导的自适应图像理解

Knowledge-guided Adaptive Image Understanding

论文作者 刘云 指导教师 程明明教授

申请学位 工学博士 培养单位 计算机学院

学科专业 计算机科学与技术 研究方向 计算机视觉

答辩委员会主席 胡清华 评 阅 人 匿名评审

南开大学研究生院

二〇二〇年十月

南开大学学位论文使用授权书

本人完全了解《南开大学关于研究生学位论文收藏和利用管理办法》关于南开大学 (简称“学校”) 研究生学位论文收藏和利用的管理规定，同意向南开大学提交本人的学位论文电子版及相应的纸质本。

本人了解南开大学拥有在《中华人民共和国著作权法》规定范围内的学位论文使用权，同意在以下几方面向学校授权。即：

1. 学校将学位论文编入《南开大学博硕士学位论文全文数据库》，并作为资料在学校图书馆等场所提供阅览，在校园网上提供论文目录检索、文摘及前 16 页的浏览等信息服务；

2. 学校可以采用影印、缩印或其他复制手段保存学位论文；学校根据规定向教育部指定的收藏和存档单位提交学位论文；

3. 非公开学位论文在解密后的使用权同公开论文。

本人承诺：本人的学位论文是在南开大学学习期间创作完成的作品，并已通过论文答辩；提交的学位论文电子版与纸质本论文的内容一致，如因不同造成不良后果由本人自负。

本人签署本授权书一份（此授权书为论文中一页），交图书馆留存。

学位论文作者暨授权人 (亲笔) 签字：_____

20　　年　　月　　日

南开大学研究生学位论文作者信息

论 文 题 目	知识引导的自适应图像理解				
姓　　名	刘云	学号	1120160127	答辩日期	2020-11-29
论 文 类 别	博士 ☑　学历硕士 ☐　专业学位硕士 ☐　同等学力硕士 ☐　划☑选择				
学院（单位）	计算机学院		学科/专业 (专业学位) 名称		计算机科学与技术
联 系 电 话	13512863965		电子邮箱		vagrantly@foxmail.com
通讯地址 (邮编)：安徽省蒙城县板桥集镇双鹿村小李庄 20 号（邮编 233529）					
非公开论文编号			备注		

注：本授权书适用我校授予的所有博士、硕士的学位论文。如已批准为非公开学位论文，须向图书馆提供批准通过的《南开大学研究生申请非公开学位论文审批表》复印件和“非公开学位论文标注说明”页原件。

南开大学学位论文原创性声明

本人郑重声明：所呈交的学位论文，是本人在导师指导下进行研究工作所取得的研究成果。除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人创作的、已公开发表或者没有公开发表的作品的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。本学位论文原创性声明的法律责任由本人承担。

学位论文作者签名：_____ 年 月 日

非公开学位论文标注说明

(本页表中填写内容须打印)

根据南开大学有关规定，非公开学位论文须经指导教师同意、作者本人申请和相关部门批准方能标注。未经批准的均为公开学位论文，公开学位论文本说明为空白。

论文题目			
申请密级	☐ 限制 (⩽2 年) ☐ 秘密 (⩽10 年) ☐ 机密 (⩽20 年)		
保密期限	20 年 月 日至 20 年 月 日		
审批表编号		**批准日期**	20 年 月 日

南开大学学位评定委员会办公室盖章 (有效)

注：限制 ★2 年 (可少于 2 年); 秘密 ★10 年 (可少于 10 年); 机密 ★20 年 (可少于 20 年)

摘要

如何使机器系统具有像人一样强大的视觉信息处理和理解能力是计算机视觉的主要研究目标。近年来，深度学习推动计算机视觉向该目标迅速前进，但基于深度学习的方法通常依赖大量的带有标注的训练数据并且具有较大的计算量。与人类的视觉系统相比，计算机视觉目前面临着三个主要挑战：

1. 数据有限：人类从出生开始，日积月累地接受了海量的训练数据，而人们无法为机器学习系统收集如此海量的数据；
2. 计算资源有限：人类大脑可以快速地处理视觉信息，计算能力比最强的超算还强，而现实中的机器往往计算能力有限，尤其是移动设备；
3. 标注有限：标注数据是非常昂贵且耗时的，尤其是对于像素级别的图像理解来说。

因此，如何在数据有限、资源受限、标注有限的条件下使机器能够理解无限复杂的真实世界成为了一个亟待解决的问题。

为此，本文提出了知识引导的自适应图像理解。具体来说，受人类视觉系统的启发，本文提出用图像边缘、图像过分割、图像显著性、以及似物性采样等通用的图像属性知识来辅助机器对图像进行理解，以解决数据有限的问题。本文通过研究基于轻量级卷积神经网络的图像理解来降低深度学习的计算量，以自适应计算资源有限的环境。以通用的图像属性知识为基础，本文通过研究基于弱监督学习的图像理解来解决标注数据有限的问题。由于像素级别的图像理解最具有代表性和广阔的应用场景，本文以实例分割和语义分割作为应用验证。

围绕上述分析，本文的主要研究内容和创新点如下：

1. 设计了基于多层次多粒度深度网络的图像通用属性提取方法，克服了目标任务数据有限的难题:
   (a) 充分利用来自卷积神经网络所有卷积层的卷积特征，提出了基于更丰富卷积特征的边缘检测技术，是第一个在著名的 BSDS500 数据集上以实时的速度超越人类标注的边缘检测算法。相关研究成果发表于 IEEE CVPR 2017、IEEE TPAMI、IJCV。
   (b) 利用超像素包含比单独的像素点更丰富信息的特点，提出了基于分层

区域合并的实时图像过分割算法；并进一步提出了基于深度特征嵌入学习的图像过分割算法，进一步提升了性能，且保持了较快的速度。相关研究成果发表于 ECCV 2016 和 IJCAI 2018。

(c) 通过理论和实验证明显著性检测中广泛使用的基于深监督的线性融合不是最优的，并进而提出了基于深监督的非线性融合技术，从而提高了显著性检测的性能。相关研究成果发表于 IEEE ICCV 2017、AAAI 2020、IEEE TCYB。

(d) 利用传统似物性采样算法的密集采样和深度学习的强大表征能力，提出了一种通过精炼传统方法的采样结果来生成少量且高质量的物体推荐的方法。相关研究成果发表于 IEEE TPAMI、CVM、IEEE CVPR 2020、Neurocomputing。

2. 基于语义分割需要丰富的多尺度信息以识别自然图像中多变的物体的特点，提出了一个基于多尺度学习的高效的轻量级卷积神经网络模型，从而自适应资源受限的环境。相关研究成果发表于 IEEE TPAMI。
3. 利用通用的图像属性知识，提出了一种基于多实例学习和多路割的弱监督实例/语义分割方法，所提出的方法同时在弱监督实例分割和弱监督语义分割两个任务上达到了目前最优的性能。相关研究成果发表于 IEEE TPAMI。

关键词： 图像理解；知识引导；轻量级卷积神经网络；弱监督学习

Abstract

How to make machine systems have the same powerful visual information processing and understanding capabilities as humans is the main research goal of computer vision. In recent years, while convolutional neural networks (CNNs) promote computer vision to advance rapidly towards this goal, it also relies on a large amount of labeled training data and has high computational cost. Compared with the human visual system, computer vision currently faces three main challenges:

1. Limited data: Human beings have been trained with massive amounts of data over and over from birth, but people cannot collect such massive amounts of data for machine learning;
2. Limited computing resources: The human brain has about one hundred billion neurons, which can quickly process visual information, and it is even stronger than the strongest supercomputer;
3. Limited annotation: Data annotation is very expensive and time-consuming, especially for pixel-level image understanding.

Therefore, how to enable machines to understand the complex real world under the conditions of limited data, limited resources, and limited annotations has become an important problem.

To this end, this thesis proposes knowledge-guided adaptive image understanding. Specifically, inspired by the human visual system, this thesis proposes to use general image attribute knowledge such as image edges, image over-segmentation, image saliency, and object proposals to assist machine systems in understanding images to solve the problem of limited data. This thesis studies lightweight CNNs for image understanding for reducing the computational cost to adapt to resource-constrained environments. Based on general image attribute knowledge, this thesis solves the problem of limited annotation data by studying image understanding with weakly supervised learning. As pixel-level image understanding is the most representative, this thesis uses instance segmentation and semantic segmentation for verification.

The main research contents and contributions of this thesis are as follows:

1. This thesis designs multi-level, multi-granular CNN-based general image attribute extraction methods to overcome the problem of limited data:
   (a) Leveraging convolutional features from all CNN layers, an edge detection method based on richer convolutional features is proposed, surpassing human annotators at real-time speed on the well-known BSDS500 dataset. Related research is published in IEEE CVPR 2017, IEEE TPAMI, and IJCV.
   (b) Since superpixels contain richer information than individual pixels, a real-time image over-segmentation algorithm based on superpixel merging is proposed; Deep feature embedding learning is further proposed to improve the accuracy while maintaining a fast speed. Related research is published in ECCV 2016 and IJCAI 2018.
   (c) It is verified by theory and experiment that widely-used deeply-supervised linear aggregation is suboptimal for saliency detection, and a nonlinear aggregation method is thus proposed to improve saliency detection. Related research is published in IEEE ICCV 2017, AAAI 2020, and IEEE TCYB.
   (d) Using the dense sampling of traditional object proposal generation methods and the powerful representation ability of CNNs, this thesis proposes to generate a small number of high-quality object proposals by refining the proposals generated by traditional methods. Related research is published in IEEE TPAMI, CVM, IEEE CVPR 2020, and Neurocomputing.
2. Since semantic segmentation requires rich multi-scale information to recognize objects with various scales, an efficient lightweight CNN model based on multi-scale learning is proposed to adapt to resource-constrained environments. Related research is published in IEEE TPAMI.
3. Using image attribute knowledge, a weakly supervised instance/semantic segmentation method is proposed, achieving state-of-the-art performance on both weakly supervised instance segmentation and semantic segmentation. Related research is published in IEEE TPAMI.

Key Words: Image understanding; knowledge guidance; lightweight convolutional neural network; weakly supervised learning

目录

第一章 绪论

第一节 论文背景与意义

如何使各种机器系统具有像人一样强大的视觉信息识别和理解能力是计算机视觉的主要研究目标。随着深度学习的迅猛发展，深度卷积神经网络已经成为计算机视觉领域最重要的基础技术之一[1–4]。一方面，深度卷积神经网络以其可自动学习的、有效的特征表示使人们可以轻松地对图像进行语义理解，从而推动了计算机视觉领域近年来的蓬勃发展。另一方面，深度卷积神经网络依赖大量提前标注好的训练数据，并且需要强大的计算能力来承载其巨大的计算复杂度。虽然一系列大规模数据集的提出[5–9] 以及 GPU 计算能力的提高暂时缓解了这种局面，但是本质的问题并没有被解决。具体来说，尽管在典型的实验室环境下，如标准的测试数据集上，很多基于卷积神经网络的方法在核心指标上已经达到了相当高的水平[10–13]，但是这距离将现有技术推向现实中的开放式环境仍然非常遥远，因为现实环境往往更加复杂、包含更多的对象类别，且计算资源通常是受限的。总体而言，与人类视觉系统相比，机器系统的视觉识别主要面临三点挑战：

1. 有限的数据：人类大脑获得的信息中有 80% 以上来自于视觉通道[14]，且人类从出生开始，每天都能看到大量的视觉信息，这些海量信息日复一日地不断提升人类的认知能力，而人们无法为机器学习搜集如此海量的数据进行模型训练；
2. 有限的计算资源：人类大脑有上千亿个神经元，其中大约 55% 的神经元是和视觉信息的处理相关的[15]，因而大脑可以快速地处理各种视觉信息，这种计算能力比最强大的超级计算机还要强，而现实中的机器往往计算能力有限，尤其是移动设备；
3. 有限的人工标注：著名的牛津大学教授 Andrew Zisserman 曾经说过，“当机器需要识别大量类别的物体时，需要人工操作的步骤（如数据搜集和标注）应当被尽量减少和简化，这是因为标注海量的数据是昂贵、耗时甚至不切实际的”[16]，但是在现实中，模型通常依赖提前标注好的训练数据。

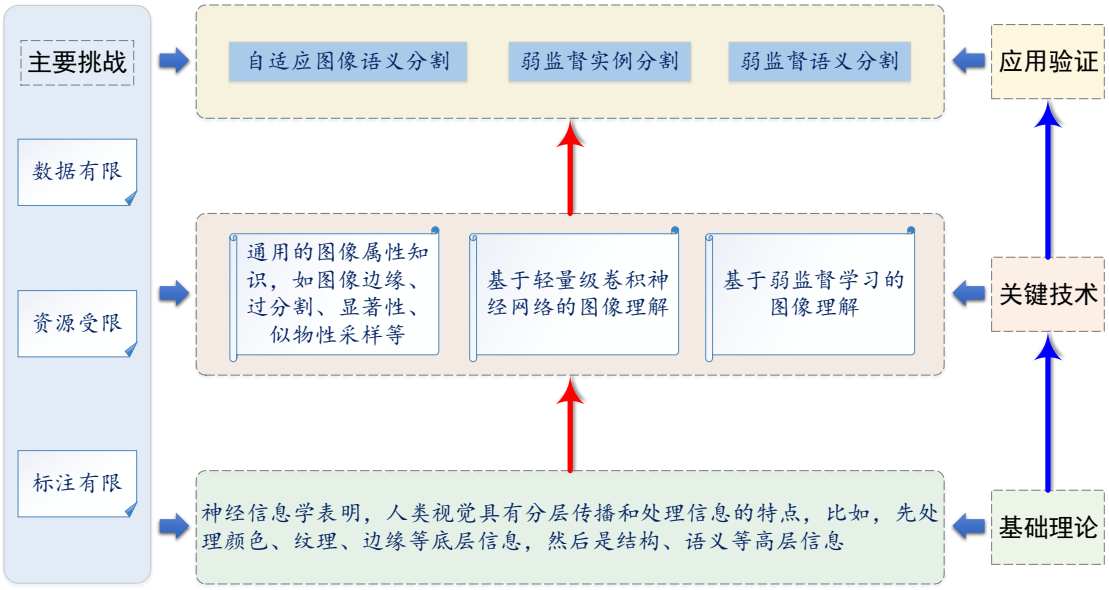


图 1.1 问题分析图。

因此，如何用有限的数据、有限的计算资源、有限的人工标注使机器能够理解无限复杂的真实世界变成了一个亟待解决的问题。

神经信息学的研究表明，人类视觉系统具有分层传播和处理信息的特点，比如大脑先处理颜色、纹理、边缘等底层信息，然后是结构、语义等高层信息[17–19]，还有研究表明人类视觉系统首先捕获视野中显著的物体或区域并进而对其进行识别[20]。这启发本文在数据有限的条件下通过使用通用的图像属性知识，如图像边缘、图像过分割、图像显著性、似物性采样等，以帮助机器进行图像理解，达到“数据不够，知识来凑”的目的。这种知识引导的图像理解既符合神经信息学的相关研究，也符合人们的直观感受。对于一个幼年的孩子，大人只需教他某个物体叫什么，而不需要像机器学习中那样，对物体进行精确的像素级分割、勾画出物体的轮廓等，这是因为人类本身便具有识别什么是物体的先验知识。因此，通过利用通用的图像属性知识来弥补机器学习中数据的不足是一个自然的选择。其次，深度卷积神经网络的计算复杂度很高，比如，著名的用于语义分割的卷积神经网络 PSPNet[13] 在强大的 TITAN Xp GPU 上需要几秒钟的时间才能处理一张普通图像。这使得深度学习模型在现实场景中很难部署，尤其是在机器人、智能手机、自动驾驶汽车这样的移动设备上。针对这个问题，本文通过探索基于轻量级卷积神经网络的图像理解技术，来提高深度学习处理图像的速度，并尽可能地提高精度。此外，基于深度学习的模型的性能很大程

度上依赖于大量带有标注的数据，尽管目前已经存在一些大规模的数据集[5–9]，但是由于数据的标注成本非常昂贵，即便是著名的物体检测数据集 MS-COCO[6] 也仅包含 80 个物体类别的多边形标记。像素级别标注的代价更加昂贵且耗时，比如，逐像素标注一张 Cityscapes 数据集中的图像耗时“平均超过 1.5 小时”[7]。可以想象，如果研究人员需要设计一个对 10000 个类别的物体进行像素级识别的系统（现实生活场景中的物体类别是海量的），那么对如此多的类别进行像素级别标注的代价和所需的时间是难以想象的[16]。

图1.1更清晰地展示上述对问题的分析。神经信息学的研究[17–19] 可以作为本文的理论基础；通用的图像属性知识、基于轻量级卷积神经网络的图像理解和基于弱监督学习的图像理解是关键技术，分别用来解决数据有限、计算资源受限、标注有限这三个计算机视觉的主要挑战；为了对以上关键技术进行验证，本文选择自适应图像语义分割、弱监督实例分割、以及弱监督语义分割等高层视觉任务作为关键技术的应用场景。以上的关键技术和应用验证部分是本文的主要工作内容。因此，本文的研究内容包括三个方面：1) 为了提取通用的图像属性知识，对诸如图像边缘、图像过分割、图像显著性、似物性采样等通用的图像属性知识的相关技术进行研究；2) 为了适应计算资源受限的情况，对以图像语义分割为例的基于轻量级卷积神经网络的像素级图像理解的探索；3) 为了解决标注数据昂贵的问题，对以通用的图像属性知识为基础的弱监督实例分割和弱监督语义分割的研究。

1.1.1 关键技术简介

本小节对本文研究的关键技术进行简要的介绍。首先介绍通用的图像属性知识，包括图像边缘、图像过分割、图像显著性、似物性采样等，这些技术都是图像处理和计算机视觉中的基本问题。它们都是与物体类别无关的，且只需训练一次，便可在各种数据上一直使用，因而是通用的。它们提供了对图像的基本理解，可以作为已有知识辅助机器系统进行进一步的理解，因而被称作通用的图像属性知识。下面来分别介绍它们。图像边缘检测旨在识别出图像中属性显著变化的像素点，一般是物体或者物体的某一部分的边界，以及不同属性的背景之间的边界[21–23]。图像边缘为图像理解提供了最基本的轮廓信息，因而是通用的图像属性知识之一。图像过分割，也被称作通用的图像分割，这是为了与图像语义分割进行区别。图像语义分割旨在将指定语义类别的物体分割出来，本质上是一个逐像素的分类问题；而图像过分割旨在把图像分成若干个特定的、

具有不同性质的区域，没有语义类别的限制，因而是通用的[24–26]。一方面，由于图像过分割所得到的每个区域都有相似的属性，从而可以使得后续的图像分析和理解变得更加简单；另一方面，由于过分割所得到的区域的数量将远远少于原始图像的像素数，从而使后续分析时的表示更加方便，并大大降低计算复杂度。图像超像素生成是一个与图像过分割相似的任务，但是超像素一般是很小的、具有相似颜色的图像块，不考虑区域的性质；相比之下，图像过分割生成的区域较大，往往表示物体或物体的一部分。本文所提出的两种新的图像过分割方法均是以超像素作为输入，并通过合并超像素来得到图像过分割结果。图像显著性检测指通过计算机算法模拟人类的视觉特点，提取出图像中显著的物体（即人类感兴趣的区域）[27–29]。如同人类视觉先提取重点区域再进行详细的分析一样[20]，由于所提取的显著性区域往往包含了需要重点分析的对象，这使得后续的图像分析和理解变得更加简单。似物性采样和显著性检测的目标有相似之处，但是似物性采样不考虑物体显著与否，而是旨在将图像中所有的物体都提取出来，以供后续高层应用将其用于进一步的分析和理解[26, 30, 31]。因此，似物性采样关注通用的物体属性，比如相似的区域属性和闭合的轮廓等。由于这四种技术都是通用的图像属性知识，因而本文将它们作为一个整体进行介绍。

接下来介绍基于轻量级卷积神经网络的图像理解。由于现实场景中的计算资源往往是受限的，尤其是在移动设备上以及当需要识别的任务更加复杂时（如更多的类别、更复杂的场景等），为了适应这种情况，降低卷积神经网络的计算量变成一件必然要面对的事情。一些研究工作通过将标准的卷积操作分解为多个子操作来降低计算量[32–34]，但是这些工作都是针对图像分类设计的，若直接将其应用于现实中更常见的像素级理解任务（如语义分割）上时，并不能取得令人满意的效果[34]。基于这些工作，已有一些专门设计用来进行语义分割的轻量级卷积神经网络[35–37]，但效果仍然无法满足现实应用的需求。

最后来介绍基于弱监督学习的实例分割和语义分割。实例分割致力于分割出图像中的每个物体并识别其类别，它区分同类别的不同个体[11]；而语义分割不考虑个体的信息，即只对每个像素进行语义类别的识别[38]。由于实例分割和语义分割是像素级密集预测任务的典型代表，本文选择以其为例，研究以通用的图像属性知识作为输入的弱监督学习。因为分割所需的像素级别的数据标注非常昂贵而图像级别的标注却容易得多，本文探索仅使用图像级标签的弱监督实例分割和语义分割[39–43]。

第二节 研究目标与主要贡献

根据上文对计算机视觉面临的三个主要挑战（即数据有限、资源受限、标注有限）的分析，本文的研究内容包括三个方面，即通用的图像属性知识提取技术、基于轻量级卷积神经网络的语义分割、利用通用的图像属性知识的弱监督实例分割和语义分割。可以利用通用的图像属性知识作为先验知识来辅助机器对图像进行理解，解决数据有限的问题，并可作为弱监督学习的输入，来解决标注有限的问题，从而实现知识引导的图像理解；基于轻量级卷积神经网络的语义分割可以解决资源受限的问题，从而实现自适应的图像理解。因此，通过这三个方面的研究，本文实现了知识引导的自适应图像理解。依据层次关系，本文提出了解决计算机视觉主要挑战的一系列方法：

1. 设计了基于多层次多粒度深度网络的任务类别无关的图像通用属性提取方法，具体包括图像边缘检测、图像过分割、图像显著性检测、似物性采样等，克服了目标任务数据有限的难题：
   (a) 利用被以往工作忽略的卷积神经网络的中间层所包含的多尺度信息，提出了基于更丰富卷积特征的边缘检测技术，以充分利用来自卷积神经网络所有卷积层的多尺度卷积特征，所提出的方法是第一个在著名的 BSDS500 数据集[25] 上以实时的速度超越人类标注的边缘检测算法，所提出的利用所有卷积层特征的思想也被广泛应用于其他领域；
   (b) 利用超像素生成方法产生的超像素来加速分割过程，提出了基于分层区域合并的实时图像过分割算法；并进一步将所提出算法中的手工设计的特征替换为深度卷积神经网络的多尺度卷积特征，提出了基于特征嵌入学习的图像过分割算法，在进一步提升性能的情况下，仍保持了较快的速度；
   (c) 通过对现有的基于深度学习的显著性检测技术的理论和实验分析，发现一直被广泛使用的基于深监督的多尺度线性融合对于显著性检测不是最优的，并进而提出了基于深监督的多尺度非线性融合技术，从而提高了显著性检测的性能；
   (d) 利用传统似物性采样算法生成的密集采样和深度学习的强大表示能力，提出了通过精炼传统方法的采样结果来生成少量且高质量的物体推荐，所提出的方法通过与其他高级应用联合训练从而具有很少的额外计算量。

2. 基于语义分割需要丰富的多尺度信息以识别自然图像中多变的物体的特点，提出了一个基于多尺度学习的轻量级卷积神经网络模型，可以以很快的速度进行较准确的语义分割，从而自适应资源受限的环境，克服了传统大规模网络难以在移动设备上部署的问题。
3. 利用通用的图像属性知识作为输入，提出了一种基于多实例学习和多路割的弱监督实例/语义分割方法，充分利用了图像属性知识，所提出的系统同时在弱监督实例分割和弱监督语义分割两个任务上达到了目前最优的性能，解决了标注有限的问题。

第三节　本文的组织结构

这里介绍本文的各部分内容。第一章介绍本文的课题背景、研究意义，并简单介绍本文的研究内容和主要贡献；第二章介绍了国内外在相关领域的研究现状，并简述了目前面临的问题；第三章介绍本文在通用的图像属性知识提取方面的工作，具体来说，包括针对图像边缘检测、图像过分割、图像显著性检测、似物性采样等技术提出的新方法，以解决数据有限的问题；第四章介绍本文在基于轻量级卷积神经网络的语义分割方面的工作，以解决计算资源受限的问题；第五章介绍本文在弱监督实例/语义分割方面的工作，以解决标注有限的问题；第六章总结本文并进一步讨论下一步的研究方向。

第二章 国内外研究现状

本章介绍国内外在相关领域的研究现状。先介绍现有的关于通用的图像属性知识提取方面的工作，包括边缘检测、图像过分割、显著性检测、似物性采样等。如第一章所述，通用的图像属性知识具有只需训练一次，即可通用于所有的图像理解的特点，因而可以作为通用知识来解决计算机视觉中数据不足的问题。然后，本章介绍基于轻量级卷积神经网络的图像理解的相关工作，轻量级卷积神经网络是解决计算机视觉中计算资源受限的一个可行途径，使得模型可以自适应计算资源受限的部署环境。最后，本章介绍基于弱监督学习的图像理解方面的相关工作，弱监督学习通过以通用的图像属性知识作为先验来降低对数据标注的要求，从而可以解决计算机视觉中标注有限的问题。

第一节 通用的图像属性知识

2.1.1 图像边缘检测

第一个被介绍的通用的图像属性知识是图像边缘检测。边缘检测是计算机视觉中最基本的问题之一[44–46]，国内外研究人员已经进行了近 50 年的相关研究，并取得了很多成就。早期的开拓性方法主要利用了图像的强度和颜色梯度。Robinson[44] 提出了通过选择颜色坐标来提取视觉上显著的边缘的定量度量方法。Sobel[45] 提出了著名的 Sobel 算子来计算图像的梯度图，然后通过对梯度图进行阈值化来得到图像边缘。Canny[46] 是 Sobel 的扩展，它添加了高斯平滑作为预处理步骤，并使用滞后阈值，使得其对噪声更加鲁棒。由于 Canny 的计算复杂度低、效率高，它现在在各种应用中仍然很受欢迎。但是，早期的这些方法由于只考虑了图像中的梯度信息，无法对图像进行高层的理解，因而准确性较低，难以适用于很多对准确性要求高的应用。

后来，研究人员开始使用一些底层信息（例如强度、梯度和纹理等）来手工设计特征，然后采用各种复杂的机器学习方法对提取的特征进行边缘和非边缘像素的分类[47, 48]。Konishi 等人[49] 通过学习两组边缘滤波器的响应的概率分布，提出了第一种数据驱动的方法。Martin 等人[50] 将亮度、颜色和纹理的变化规范化为 Pb 特征，并训练一个分类器以结合这些特征中的信息。Arbeláez 等

人[25] 通过使用标准化切割（NCuts）[51] 将上述局部信息整合到一个全局框架中，以获得全局特征，于是，Pb 便被改进为 gPb。Lim 等人[52] 提出了新的可表示中间层信息的 Sketch tokens 特征，并在边缘检测任务上取得了不错的效果。Dollár 等人[53] 使用随机决策森林来表示局部图像小块的结构，只要输入颜色和梯度特征，所提出的结构化森林就可以预测出高质量的图像边缘。然而，这些方法都是基于手工设计的特征的，而手工特征在表示高层语义信息时能力比较有限，所以这些方法对具有语义意义的边缘进行检测时效果并不理想。

随着最近深度学习的快速发展，很多基于深度学习的边缘检测方法已经被提出。Ganin 等人[54] 通过结合卷积神经网络和最近邻搜索提出了 N^4-Fields 方法。Shen 等人[21] 将边缘形状划分为若干个子类，并通过学习模型参数来拟合每个子类。Hwang 等人[22] 将边缘检测看成逐像素的分类问题，他们使用卷积神经网络为每个像素提取一个特征向量，然后使用 SVM 分类器将每个像素分为边缘或非边缘类别。Xie 等人[23] 提出了一个高效且准确的边缘检测器 HED（Holistically-nested Edge Detection），它可以实现“图像-图像”的训练和预测。HED 在 VGG16[3] 每一个卷积阶段的最后一个卷积层后连接一个侧输出层进行预测，这个侧输出层由一个 1×1 卷积层、一个反卷积层和一个 sigmoid 层组成。最近，Liu 等人[55] 使用由下而上的边缘所生成的松弛标签来指导 HED 的训练过程，并在性能上取得了一些提升。Li 等人[55] 提出了一个巧妙的无监督学习模型，但其性能比使用有限的 BSDS500 数据集训练的模型要差。

前面所提到的基于卷积神经网络的模型已经显著提高了边缘检测的性能，但当对像素点进行分类时，这些方法通常仅使用每个卷积阶段的最后一层的特征，因此他们都损失了部分有用的多尺度卷积特征。为了解决这个问题，本文提出了一个可以有效地利用每个卷积层特征的全卷积神经网络 RCF。将在第三章第一节详细介绍所提出的方法。

2.1.2 图像过分割

图像过分割也是计算机视觉中的一个基本问题[50]。在过去的几十年中，研究人员为该领域做出了许多有益的贡献。由于篇幅所限，本节只重点介绍一些与本文所提出的方法相关的代表性方法，推荐读者参考流行的 BSDS500 基准[25] 和其他的近期研究[26, 56–58] 来获得全面的关于图像过分割技术的讨论。

过去，人们对有效地计算基于归一化切割的聚类分割算法 NCuts[51] 给予了高度的关注。Maire 等人[57] 设计了一种多网格特征向量生成器，可以大大提高

NCuts 中特征向量的计算速度。Taylor 等人[58] 尝试使用分水岭过分割来减小特征向量的大小，从而可以在不到半秒的时间内计算特征向量。Pont-Tuset 等人[26] 提出先对特征向量进行下采样，以在减小的尺寸下对 NCuts 问题进行求解，然后对结果进行上采样以恢复图像结构。尽管可以通过上述方法获得令人满意的分割结果，但是这些基于聚类的方法的速度非常慢，严重制约了其实际应用。

与图像过分割相似但略有不同，超像素生成致力于将一张图像分割成较小的、规则的、紧凑的区域，即超像素。超像素不考虑区域的性质，只考虑颜色信息，一般通过较简单聚类算法来实现。沿着这个方向，SLIC[59] 成为最著名的超像素方法之一，它在准确性和速度之间取得了很好的平衡，并在许多应用中被广泛采纳[60–63]。SLIC 提出了一种 k 均值聚类方法，该方法通过以规则的网格步长采样像素来初始化聚类中心，然后执行标记过程，其中每个像素都用聚类中心的索引来标记，该聚类中心的搜索区域与它的位置重合。

Felzenszwalb 和 Huttenlocher[24] 提出的基于图的聚类方法也已被广泛使用。该方法将像素点看作图的顶点并且用边来测量相邻像素之间的差异，从而构造无向图。然后通过执行图割操作，使每个区域都是所涉及像素的最小生成树。由于它直接从具有简单颜色信息的单个像素开始合并过程，所以该方法容易产生噪音。与该方法相比，本文所提出的方法从比单个像素包含更多信息的超像素开始聚类过程，因而更加鲁棒。

其他流行的图像分割方法包括基于特征学习[64] 的方法。这些方法通过使用各种分类器将诸如亮度、颜色和纹理属性之类的特征融合在一起，证明了其良好的表示能力。Ren 等人[64] 提出了一种分层的过分割方法，其中应用了一系列边界分类器，以从超像素开始递归地合并区域。在这个层面上，本文所提出的方法与该方法有一定的相似之处，都是分层地进行区域合并。本文所提出的 HFS 算法通过与以上基于图的过分割方法[24] 结合使用，并仔细研究适用于 GPU 实现的手工设计的特征，从而达到了实时过分割的目标。本文的 DEL 算法提出了一种基于深度嵌入学习的策略，将 HFS 中的手工特征替换为多尺度的卷积特征。作为通用的图像属性知识，本文所提出的 HFS 和 DEL 不仅对开放环境中的场景理解具有重要意义，也将有益于许多计算机视觉任务。

2.1.3 图像显著性检测

显著性物体检测，也称显著性检测，因其广泛的应用范围和具有挑战性的处理场景而成为一个非常活跃的研究领域。早期的启发式显著性物体检测方法先

提取手工设计的底层特征，然后使用机器学习模型对这些特征进行分类[65]，并利用一些启发式的先验来确保其准确性，例如颜色对比[27]、中心优先[28]、背景优先[66] 等。由于深度卷积神经网络在计算机视觉领域取得了巨大成功，基于卷积神经网络的方法也被用于显著性物体检测。基于区域的显著性物体检测[67–69]出现在早期的基于深度学习的方法中。这种方法将每个图像小块视为进行显著性检测的基本处理单元。近年来, 基于卷积神经网络的“图像-图像”的显著性物体检测方法[29, 70–79] 在该领域占据了主导地位，它们将显著性物体检测视为逐像素的回归任务并进行“图像-图像”的预测。因此，下面主要回顾一下基于卷积神经网络的“图像-图像”的显著性物体检测。

由于显著性物体检测需要高层的全局信息（存在于卷积神经网络的顶层）和底层的局部细节（存在于卷积神经网络的低层），因此如何有效地融合多层次和多尺度的卷积特征是主要的研究方向[29, 70–72, 74–81]。这方面的研究非常多，但是最近的神经网络设计整体倾向于越来越复杂。下文通过简单地将基于多尺度深度学习的方法分为四类来继续当前的讨论：超特征学习（Hyper Feature Learning）、U-Net 类型、HED 类型格、以及 U-Net+HED 类型。

超特征学习. 超特征学习[82] 是学习多尺度信息的最直观的一种方式。已有很多研究将超特征学习用于显著性物体检测[75, 77]。这些模型将来自于骨干网络的不同层[77] 或多流网络的不同分支[75] 的多尺度卷积特征进行拼接或相加，然后将所融合的超特征（也被称为 Hypercolumn）用于最终的显著性物体预测。

U-Net 类型. 众所周知，深度卷积神经网络的顶层往往包含高层语义信息，而低层则往往学习底层的细节特征。因此，对超特征学习的一种合理改进就是将深度特征从高层到低层逐步融合[83, 84]。以这种方式，高层的语义特征将会与低层的底层特征相结合来捕获细粒度的细节。特征融合可以是简单的特征图逐元素求和[83]、特征图拼接（U-Net）[84] 或基于它们的更复杂的设计。目前，许多显著性物体检测模型都是属于这种类型的[71, 72, 78, 85–88]。这里需要注意的是，超特征学习和 U-Net 类型并没有使用深监督，因此它们是没有侧输出这个概念的。

HED 类型. HED 类型的网络[23] 首先是被提出来用于图像边缘检测的，它是由卷积神经网络的深监督[89] 发展而来的。在此之后，类似的思想也被用来进行显著性物体检测[70, 76]。HED 类型的网络在中间层增加了深监督来获得侧输出预

测，最终结果是所有侧输出预测的线性组合。与多尺度特征融合不同，HED 使用了多尺度预测融合。

U-Net+HED 类型. U-Net+HED 类型的方法结合了 U-Net 和 HED 的优点，它是在 U-Net 解码器的每个卷积阶段都进行深监督。最近所提出的很多显著性物体检测模型都属于这一类型[29, 73, 74, 79, 90–93]，而它们之间的不同之处在于使用不同的融合策略。这些模型的一个明显的相似之处是，最终的预测都是将侧输出预测进行线性融合而产生的。此处，多尺度学习通过以下两个方面来实现：1) U-Net 以编码-解码的形式融合从高层到低层的多层次卷积特征；2) 多尺度侧输出预测被线性融合后作为最终预测。当前，该领域的研究主要集中在第一个方面，一些性能较高的模型已经为此设计了非常复杂的特征融合策略[29, 74]。

显著性物体检测的完整文献综述超出了本文的范围，更为详细的介绍请参考相关的综述文章[94, 95]。本文致力于上述 U-Net+HED 多尺度学习的第二个方面，即多尺度侧输出融合。本文发现，传统的线性侧输出预测融合的效果的上限仅局限于侧输出预测。为此，本文提出了一种以非线性方式融合侧输出特征的方法 DNA，以使所融合的混合特征可以更好地利用互补的多尺度深度特征，而将 DNA 结合于非常简单的 U-Net 就可以取得很好的性能。

2.1.4 似物性采样

似物性采样致力于为输入图像生成若干个物体推荐，以覆盖图像中所有的物体。可以将相关研究工作大致分为四类：基于图像过分割的似物性采样方法、基于图像边缘的方法、基于深度学习的方法、以及物体推荐的后处理方法。

基于图像过分割的似物性采样方法. 该类方法使用图像过分割作为输入，并尝试找到这些图像块的正确组合来覆盖图像中所有的物体。这些方法通常结合一些底层特征（例如显著性、颜色、SIFT[96] 等）来对边界框进行评分，然后选择分数较高的推荐框。Selective Search[30] 是最著名的似物性采样方法之一，它利用穷举搜索和过分割的优势，通过对超像素进行分层合并来获得高质量的推荐。MCG[26] 引入了一种可有效利用多尺度信息的高性能图像过分割算法，并通过探索组合空间，将所产生的具有多尺度的层次结构的区域组合为物体推荐。Manen 等人[97] 建立了图像超像素的连通图，并使用 Prim 算法的随机版本生成了具有较大边缘权重的期望总和的生成树。这些生成树的边界框就是最终的物

体推荐。Rantalankila 等人[98] 对超像素执行局部搜索以形成分割层次，并使用全局搜索来获得中间层次结构的基于图割的过分割结果。很多其他的似物性采样方法[99–101] 也属于此类。

基于边缘的方法. 该类方法利用了自然图像中的完整物体通常具有明确的封闭边界的特点[102]。近年来，已经有多种使用边缘特征的高效算法被提出。Zhang 等人[103] 设计了一种级联排序 SVM（即 CSVM）方法，使用梯度特征获得物体推荐。Cheng 等人[104] 提出了一种非常有效的算法 BING，该算法通过将 CSVM[103] 量化为一些二进制运算使其以 300fps 的速度运行。Lu 等人[105] 提出了一种新的基于封闭路径积分的封闭轮廓度量。Edge Boxes[31] 根据每个边界框中完全包含的轮廓数来计算似物性分数。

基于深度学习的方法. 该类方法用卷积神经网络生成物体推荐，例如 RPN[106]、DeepMask[107]、SharpMask[108] 等，这是受到卷积神经网络具有强大的特征表征能力的启发。RPN[106] 同时预测图像的卷积特征图的每个位置的物体推荐和似物性分数。DeepMask[107] 的训练目标有两个：给定一个图像小块，系统首先输出一个与类别无关的分割蒙版，然后输出该小块以整个物体为中心的概率。SharpMask[108] 使用一种新颖的自上而下的优化方法来增强前馈网络，以进行物体分割，这种自底向上/自上而下的体系结构能够高效地生成较准确的物体蒙版。但是，对于自然图像，这些基于卷积神经网络的方法生成的物体推荐的数量仍然太多，例如通常为几百个，远多于物体的实际数量。

物体推荐的后处理. 该类方法致力于改进物体推荐，以便在图像中准确定位物体。Kuo 等人[109] 提出了一个名为 DeepBox 的小型卷积神经网络，用于重新计算已存在的物体推荐框的似物性分数，然后根据新的似物性分数对这些推荐框重新排序。Chen 等人[110] 尝试将物体推荐框与超像素对齐。Zhang 等人[111] 进一步讨论了似物性采样的优化。他们首先使用边缘，然后使用超像素来优化物体推荐框。他们基于过分割的优化加速了 MTSE[110] 中超像素的生成，因此最终的系统能够以非常快的速度运行。He 等人[112] 提出了具有不同方向的物体推荐框，而不仅仅是常规方法中使用的垂直框。本文建立了一个精炼网络来精炼已有的边界框，所提出的 RefinedBox 生成的优化框在似物性采样的评测和物体检测的评测中均达到了最佳性能。

第二节 轻量级卷积神经网络

本节关于轻量级卷积神经网络的介绍以语义分割为背景，因为语义分割作为像素级的图像理解任务，可以代表基本的图像理解要求。自从全卷积网络（Fully Convolutional Network, FCN）[83] 发明以来，基于 FCN 的方法就支配了语义分割的发展。本节首先简要回顾经典的高精度分割模型和技术，然后介绍轻量级模型。

自然场景中的物体呈现出很大的尺度变化，因此多尺度学习对于语义分割至关重要。大多数方法旨在设计网络以从彩色图像中学习有效的多尺度特征表示。例如，FCN[83]、U-Net[84]、DeconvNet[113] 和 SegNet[114] 建立了编码-解码网络，从而以一种从顶层到低层的方式来融合深度特征。一些方法[82, 115–117] 聚合了来自多层的多尺度深度特征，以进行最终的密集预测。DeepLab[38] 及其变体[118–120] 通过使用具有不同扩张率的扩张卷积来设计 ASPP 模块，以学习多尺度特征。基于 ASPP 模块，DenseASPP[121] 以密集的方式连接一组扩张卷积层，从而生成密集覆盖了更大尺度范围的多尺度特征。

除了多尺度学习之外，一些研究还致力于通过上下文编码[122]、金字塔池化[13] 和非近邻操作[123, 124] 来利用全局信息。此外，Wu 等人[125] 试图找到网络深度和宽度之间的良好折衷，以提高分割精度。DFN[126] 通过设计网络来处理类内不一致问题，并引入了一个边界网络来使边界两侧的特征可区分。一些方法[118, 127, 128] 使用条件随机场（Conditional Random Field, CRF）或马尔可夫随机场（Markov Random Field, MRF）来建模语义分割中的空间关系。上述模型旨在不考虑模型大小和推理速度的情况下提高分割精度，因此对于移动设备来说并不适用。本文的目标是设计一种网络规模小、速度快、准确性高的轻量级模型。

ENet[2] 打开了轻量级语义分割的大门，它减少了 ResNet[4] 的卷积通道，以少量参数实现了实时分割。ERFNet[129] 将标准的二维卷积分解为两个非对称的一维卷积。ContextNet[35] 结合了小分辨率的深层网络和全分辨率的浅层网络。ESPNet[36] 将标准卷积分解为一个逐点卷积和由扩张卷积构成的空间金字塔。ESPNetv2[37] 在 ESPNet[36] 的基础上进行了扩展，它使用分组的逐点卷积和深度可分离的扩张卷积。ICNet[130]、BiSeNet[131]、SQNet[132] 和 FRRN[133] 试图在分割精度和推理速度之间取得良好的平衡。最近的一些技术报告[134] 也为轻量级语义分割提供了新的设计。本文的目标是在不牺牲速度和增加参数量的前提下提高轻量级语义分割的准确性。

第三节 基于弱监督学习的图像理解

和上节的原因相同，本节关于弱监督学习的图像理解的介绍以像素级的任务，即实例分割和语义分割，作为背景。本文所提出的基于弱监督学习的实例分割和语义分割可以代表图像理解的基本要求。

实例分割. 实例分割是用于场景理解的一个活跃的研究领域，但长期的努力都集中在全监督的条件下。大多数效果好的方法都是基于物体检测网络来输出排序好的物体分割，而不是输出边界框[11, 135–138]。在这些方法中，Mask R-CNN[11]及其派生方法[137, 138]主导了最新技术。一些研究人员还基于初始的语义分割网络提供了一些方法来生成实例蒙版[139–141]。尽管全监督的方法可以实现高精度，但它们通常需要具有昂贵的逐像素标注的大规模训练数据，而现实中的很多应用都面临着标注有限的问题。

弱监督实例分割. 对于弱监督实例分割，Khoreva 等人[142]首先提出使用边界框标注作为监督，而不是像素级的蒙版。具体来说，他们使用改进版的 GrabCut[143]从已有的边界框估计物体的分割。通过 MCG[26]生成的基于分割的似物性采样（Segment-based Object Proposal, SOP）进一步改善所获得的物体分割。Li 等人[144]通过迭代改善伪真值来扩展了 Khoreva 等人[142]的工作，他们使用训练集上的网络输出作为新的伪真值。Hsu 等人[145]通过基于每个边界框的扫掠线生成正负袋来将此问题表示为多实例学习（Multiple Instance Learning, MIL）[146]任务，该 MIL 表示可以集成到端到端的网络中，以训练实例分割模型。Hu 等人[147]引入了使用迁移学习的半监督实例分割模型，训练数据中的一些类具有逐像素标注，而其他类则仅有边界框标注。

Zhou 等人[39]提出了一个更具有挑战性的问题，即在图像级的弱监督下训练神经网络进行实例分割。他们引入了一个非常新颖的类峰值响应的概念，该响应反映了驻留在每个语义实例内部的强视觉信息，所学习的类峰值响应图可用于查询和排序 SOP。他们的方法明显优于各种基准方法。在该工作[40]之后，Zhu 等人[40]提出了一种实例范围填充的方法，以选择性地从有噪声的 SOP 中收集伪监督。伪监督用于学习一个可区分的填充模块，该模块可为每个实例预测一张与类无关的激活图。Cholakkal 等人[41]通过构造物体类别密度图，引入了一种图像级别的监督方法，以用于普通物体计数和图像级监督的实例分割。Ahn

等人[42]扩展了图像分类模型的类激活图（Class Activation Map, CAM）[148]，以发现被视为伪真值的整个实例区域，用其训练一个全监督的模型。Ge 等人[43]提出的 Label-PEnet 通过交替训练四个顺序级联的模块（包括多标签分类、物体检测、实例细化和实例分割）来逐步将图像级标签转换为逐像素标签。本文遵循这些研究[39–43]仅将图像级监督用于实例分割，并尝试同时使用每个 SOP 的固有属性和整个训练数据库的总体数据分布来确定每个 SOP 的语义类别，而不是使用基于 CAM 的模型。

弱监督语义分割. 语义分割与实例分割高度相关，语义分割仅识别每个像素的类别，而不会区分不同的物体实例。通过简单地消除对物体实例的区分，可以将弱监督实例分割应用于语义分割。本文还提供了语义分割的评测结果，因此这里概括地综述了弱监督语义分割的相关工作。

使用提供位置信息的标注，例如点[149]、涂鸦[150]或边界框[151]，最近的方法已经取得了良好的性能。使用图像级标注的弱监督语义分割仍然是一个具有挑战性的问题。给定图像级标注，CAM[148]是发现粗略物体位置的一个很好的起点。但是，CAM[148]倾向于将注意力集中在目标物体的小的具有区分性的区域上，这不适合于训练语义分割网络。当前大多数方法旨在改进 CAM 以使用图像标签提取完整的物体。这些方法要么采用图像遮挡和擦除操作，以防止分类器仅关注物体的具有区分性的部分[152–154]，要么使用特征层面的处理[155–160]和区域增长技术[161–164]。这些方法经常使用各种辅助信息，例如显著性图[165]、图像边缘[23]、似物性采样[26]等，以提高准确性[154, 160, 163, 164, 166–170]。

除上述方法外，Saleh 等人[171]和 Pinheiro 等人[172]提出了用于弱监督语义分割的 MIL 方法，但是它们的方法仅限于按像素分类，无法学习实例感知信息。相反，本文介绍的 MIL 框架着重于学习用于区分物体实例的实例感知信息。最近，Fan 等人[173]提出了一种基于图的弱监督语义分割模型，该模型与本文所提出的 LIID 模型（详见第五章）相关。尽管 LIID 专注于与 Fan 等人[173]不同的任务，但这里分析了 LIID 与该方法[173]之间的差异，可以将其总结如下：

1) LIID 在对 SOP 的信息提取方面不同于 Fan 等人[173]。对于每个 SOP，Fan 等人[173]直接使用 CAM[148]来估计概率分布，然后采用预先训练的 ImageNet[5]模型来提取语义特征。与此不同，LIID 提出了一个端到端的 MIL 框架，以便从给定图像中同时学习概率分布和语义特征。

2) LIID 在图建模方面不同于 Fan 等人[173]。Fan 等人[173] 通过将所有 SOP 视为图结点将类别标签分配表示为一个普通的图割问题，并且初始概率仅在优化公式中用作平衡项。与此不同，LIID 通过将所有 SOP 视为图的普通结点（非终端结点），将目标类别标签视为终端结点来构建无向图。SOP 的概率分布和语义特征用于计算不同类型边缘的权重。LIID 将类别分配表示为一个多路割问题，然后提出一种有效的近似优化算法来解决该问题。

尽管 LIID 和 Fan 等人[173] 都使用图来利用数据集级别的信息，但是 LIID 在模型训练、概率预测、特征提取、图构造和图割方面更合理、更直观，这导致了所提出的方法的性能明显更好。

第三章 通用的图像属性知识

本章介绍本文在通用的图像属性知识提取方面的工作。由于现实中经常面临数据不足的问题，使用通用的图像属性知识为机器系统提供先验知识，达到“数据不够，知识来凑”的目的，是解决数据不足的一个理想方式。通用的图像属性知识包括图像边缘检测、图像过分割、图像显著性检测、似物性采样等，这些技术都是与物体类别无关的，因而只需要训练一次，便可以应用在各种场景下。由于这些任务都需要多尺度和多粒度的信息，本章针对各个任务分别提出了各种基于多层次多粒度的深度卷积神经网络。第一节介绍了本文在图像边缘检测方面的研究工作；第二节介绍了在图像过分割方面的工作；第三节介绍了在图像显著性检测方面的工作；第四节介绍了在似物性采样方面的工作；第五节对全章进行了总结。

第一节 通用的图像属性知识 - 图像边缘检测

3.1.1 引言

边缘检测旨在从自然图像中提取视觉上显著的物体边缘，几十年来一直是计算机视觉领域重要且极具挑战性的任务之一。它通常被认为是一种底层技术，很多高层任务都已经得益于边缘检测的发展，比如物体检测[174, 175]、似物性采样[30, 31, 104, 111]、以及图像过分割[26, 176] 等。

传统的边缘检测方法先提取亮度、颜色、梯度、纹理或其他手工设计的特征（如 Pb[50]、gPb[25]、以及 Sketch tokens[52] 等），然后使用各种复杂的机器学习方法[53, 177] 来对这些提取的特征进行分类，以预测边缘和非边缘像素。尽管这些年来使用底层特征进行边缘检测的方法有了很大的进步[178]，但其局限性也很明显。这是因为通常情况下所定义的边缘是具有语义意义的，比如表示物体、物体的一部分或者背景的边界，而使用底层特征很难表示出物体级别的语义信息。在这种情况下，gPb[25] 和 Structured Edges[53] 都尝试使用复杂的策略来尽可能地捕获全局特征。

在过去的几年中，卷积神经网络通过大幅推进各种任务的发展，如图像分类[1–3]、目标检测[106, 179] 和语义分割[83, 118] 等，而在计算机视觉社区变得很流

(a) 原始图像 (b) 真值 (c) conv3_1 (d) conv3_2

(e) conv3_3 (f) conv4_1 (g) conv4_2 (h) conv4_3

图 3.1 本节基于 VGG16[3] 设计了一个简单的神经网络来得到侧输出 conv3_1、conv3_2、conv3_3、conv4_1、conv4_2 和 conv4_3。可以清楚地看到，卷积特征逐渐变粗糙，中间层 conv3_1、conv3_2、conv4_1 和 conv4_2 包含许多其他层没有的细节信息。

行。由于卷积神经网络具有强大地自动学习自然图像的高层表征的能力，因此使用卷积神经网络进行边缘检测已成为最近的趋势。一些著名的基于卷积神经网络的方法已经显著地推动了该领域的发展，如 DeepEdge[180]、N^4-Fields[54]、CSCNN[22]、DeepContour[21]、和 HED[23] 等。本节所提出的 RCF 也属于此类。

如图3.1所示，为了观察边缘检测中不同卷积层所学到的信息，本节用具有 5 个卷积阶段的 VGG16[3] 构建了一个简单的网络来产生中间层的侧输出。可以发现，从神经网络的低层到顶层，卷积特征会逐渐变得粗糙，而且中间层包含许多有用的细节信息。此外，由于丰富的卷积特征对于许多视觉任务都非常有效，因此许多研究人员都致力于探索更深的网络结构[4]。但是，由于梯度消失/爆炸和训练数据不足（例如，用于边缘检测的数据）等原因，使网络架构变得很深

时会难以收敛。那么，为什么不充分利用现有的卷积特征呢？本章所提出的方法的灵感正是来源于这些观察。与以前的方法仅仅使用部分卷积层的卷积特征不同，本节所提出的方法通过精心设计来结合所有卷积层的卷积特征，从而可以获得更丰富的卷积特征（Richer Convolutional Features, RCF），因此能够对不同尺度、不同层次的物体或部分物体进行更准确的特征表示。RCF 是首个使用卷积神经网络所有卷积层的卷积特征的研究工作。此外，RCF 设计了一种“图像-图像”的方式进行精确的边缘预测。RCF 在边缘检测方面表现得非常出色，其设计思想也可以用作其他的计算机视觉任务。

当在 BSDS500 数据集[25] 上评测所提出的 RCF 时，RCF 在*最佳数据集尺度*（Optimal Dataset Scale, ODS）下 F-measure 为 0.811、速度为 8fps，取得了检测性能和效率之间的最佳平衡，它甚至超过了人类对边缘感知的结果（ODS F-measure 为 0.803）。此外，本章还介绍了一个快速版本的 RCF，该版本可实现在速度为 30fps 时，ODS F-measure 达到 0.806。

3.1.2 方法

由于自然图像中的物体具有各种不同的尺寸和比例，所以学习丰富的层次特征对边缘检测任务很重要。卷积神经网络已被证明对该任务很有效，并且，随着感受野的变大，卷积神经网络中的卷积特征也会逐渐变得粗糙。根据上述观察，本节尝试使用更丰富的卷积特征来提高边缘检测性能。所提出的 RCF 网络通过组合所有有意义的卷积特征来充分利用物体的多尺度和多层次信息，以此实现“图像-图像”的预测。下面将依次介绍 RCF 的网络架构、损失函数、多尺度策略、以及与 HED[23] 的对比。

3.1.2.1 网络架构

受之前深度学习相关研究[23, 83, 106, 179] 的启发，本章通过修改 VGG16 网络[3] 来设计 RCF 网络。由 13 个卷积层和 3 个全连接层组成的 VGG16 网络已经在包括图像分类[3] 和目标检测[106, 179] 在内的许多任务上实现了最佳性能。它的卷积层可以分为五个阶段，其中每个阶段之后都连接一个池化层。每个卷积层所学到的有用信息会随着其感受野的增加而越来越粗糙。在表3.1中可以看到不同层的感受野的具体大小。而 RCF 网络设计的初衷就在于假设使用这种丰富的多层信息会对边缘检测有所帮助。

RCF 所提出的网络架构如图3.2所示。与标准的 VGG16 相比，RCF 做出如

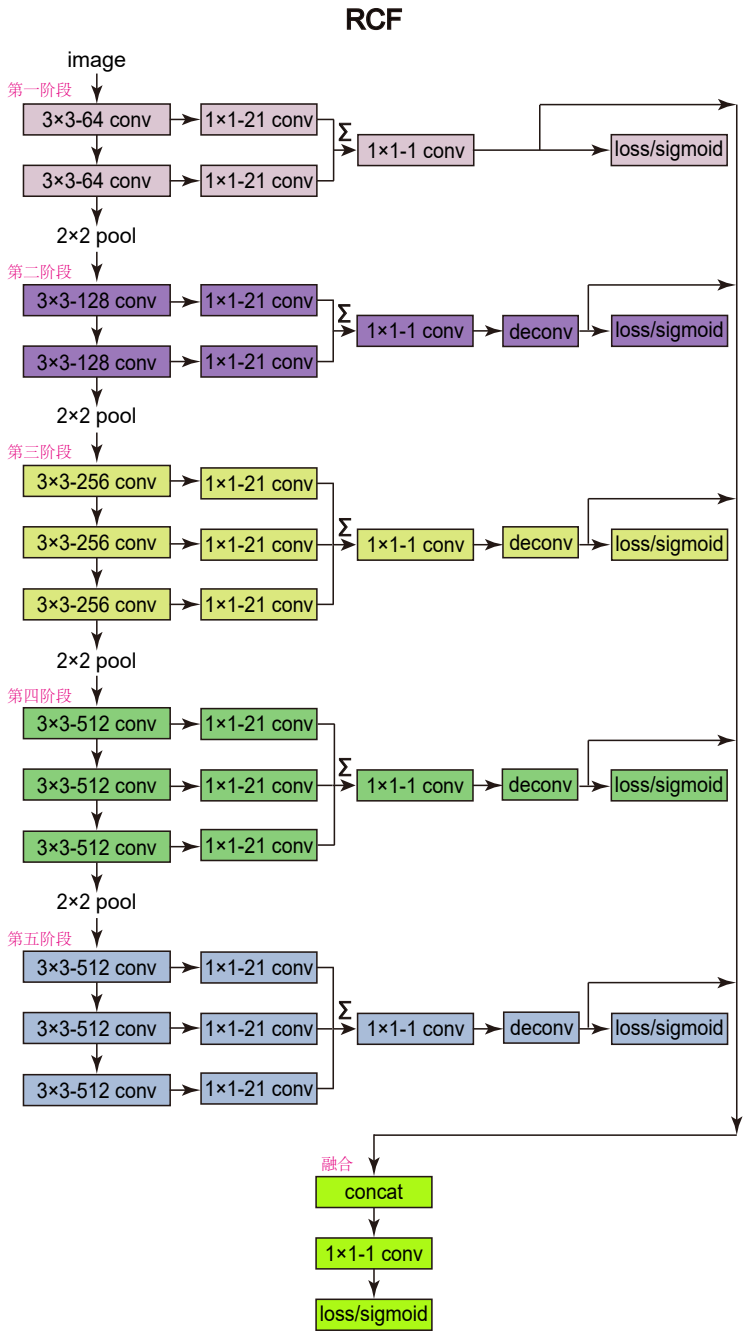


图 3.2 所提出的 RCF 网络架构。输入为任意大小的图像，输出相同大小的边缘概率图。

下修改：

- 由于全连接层不符合 RCF 的全卷积神经网络的设计思想，而添加 pool5 层将使步幅增加两倍，这对定位边缘位置是有害的。因此，RCF 移除了所有的全连接层和 pool5 层。

表 3.1 标准 VGG16 网络[3] 的感受野和步幅的具体大小

网络层	conv1_1	conv1_2	pool1	conv2_1	conv2_2	pool2
感受野	3	5	6	10	14	16
步长	1	1	2	2	2	4
网络层	conv3_1	conv3_2	conv3_3	pool3	conv4_1	conv4_2
感受野	24	32	40	44	60	76
步长	4	4	4	8	8	8
网络层	conv4_3	pool4	conv5_1	conv5_2	conv5_3	pool5
感受野	92	100	132	164	196	212
步长	8	16	16	16	16	32

- VGG16 中的每个卷积层后都再连接一个卷积核为 1×1、输出通道数为 21 的卷积层，然后将各个阶段内的卷积结果使用逐元素相加来生成混合特征。
- 在每个逐元素相加操作之后都再连接一个卷积核为 1×1、输出通道数为 1 卷积层，然后使用反卷积层对该特征图进行上采样。
- 在每个阶段的反卷积层之后连接一个交叉熵损失或者 sigmoid 层。
- 将所有反卷积层的输出结果都拼接到一起，然后使用一个 1×1 卷积层融合每个阶段的特征图。最后，再连接一个交叉熵损失或者 sigmoid 层来获取融合后的损失或者输出。

因此，RCF 将来自所有卷积层的分层的特征整合进一个整体框架内，且该框架的所有参数都是自动学习的。由于 VGG16 中不同卷积层的感受野大小互不相同，因此 RCF 网络可以学习多尺度信息，包括底层信息和高层信息，而这些信息均有助于边缘检测。图3.3展示了每个阶段的中间结果。可以看出，从上到下，边缘响应变得越来越粗糙，同时对较大的物体或部分物体的边缘响应会更强烈，这与预期的卷积层会随着感受野的变大来检测较大的物体是一致的。由于 RCF 结合了所有卷积层来获取更丰富的多尺度特征，因此有望提高边缘检测的准确性。

3.1.2.2 对标注鲁棒的损失函数

边缘检测的数据集通常是由多个标注者根据他们自身对物体或部分物体的认知来标记的。尽管每个人的认知略有不同，但是他们对同一张图像所标注的边缘却基本一致。对于每张图像，RCF 对所有真值图求平均来生成范围为 [0,1] 的边缘概率图。其中，0 表示没有标注者将此像素标注为边缘，而 1 表示所有标

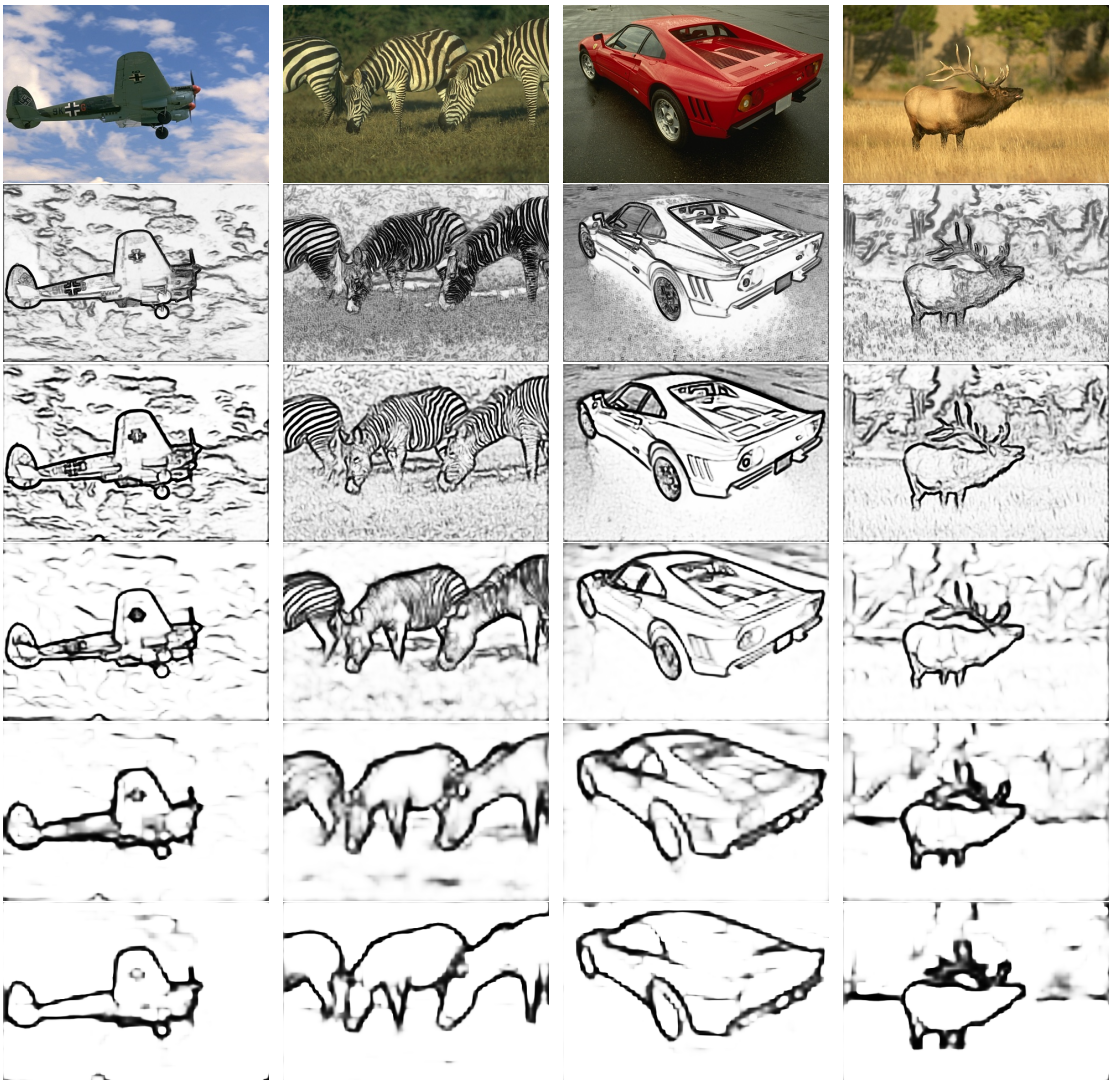


图 3.3 RCF 每个阶段的输出的示例。第一行是 BSDS500 数据集[25] 中的原始图像，从第二行到第六行分别是第一、二、三、四和五阶段的输出。

注者都将此像素标注为边缘。RCF 将边缘概率高于 η 的像素视为正样本，将边缘概率等于 0 的像素视为负样本。如果一个像素的边缘概率大于 0 且小于 η，那么该像素点将是一个有争议的边缘点，无论将其视为正样本还是负样本都将干扰神经网络的训练，因此，RCF 忽略这类像素点。

RCF 按下式计算每个像素相对于标签的损失

$$l(X_i;W) = \begin{cases} \alpha \cdot \log\left(1 - P(X_i;W)\right) & if\ y_i = 0, \\ 0 & if\ 0 < y_i \leq \eta, \\ \beta \cdot \log P(X_i;W) & otherwise, \end{cases} \tag{3.1}$$

其中，

$$\begin{aligned}\alpha &= \lambda \cdot \frac{|Y^+|}{|Y^+| + |Y^-|},\\ \beta &= \frac{|Y^-|}{|Y^+| + |Y^-|}.\end{aligned} \tag{3.2}$$

Y^+ 和 Y^- 分别表示正样本集和负样本集。超参数 λ 用于平衡正负样本。在像素 i 处的激活值（卷积特征向量）和真值边缘概率分别用 X_i 和 y_i 表示。$P(X)$ 是标准的 sigmoid 函数，W 表示神经网络中所有需要学习的参数。因此，所改进的损失函数可以表示为

$$L(W) = \sum_{i=1}^{|I|} \Big(\sum_{k=1}^{K} l(X_i^{(k)}; W) + l(X_i^{fuse}; W) \Big), \tag{3.3}$$

其中，$X_i^{(k)}$ 是来自阶段 k 的激活值，而 X_i^{fuse} 表示来自融合层的激活值。$|I|$ 是图像 I 中的像素数，K 是神经网络的阶段数（这里为 5）。

3.1.2.3 多尺度边缘检测

单尺度边缘检测直接将原图像输入到经过训练的 RCF 网络中，然后输出边缘概率图。为了进一步提高边缘的质量，本章在测试时使用了图像金字塔。具体来说，通过缩放图像来构建图像金字塔，再分别将金字塔中的每一张图像输入到 RCF 网络中。然后，使用双线性插值将得到的所有边缘概率图缩放到原图像大小。最后，通过求这些边缘图的平均值来获得最终的预测图。此外，作者还尝试了使用加权和的方式来获得最终的预测图，但最后发现使用简单的平均操作的效果要更好。考虑到精度和速度之间的权衡，本节使用 0.5、1.0 和 1.5 三个尺度。在 BSDS500 数据集[25] 上进行评测时，尽管使用这个简单的多尺度策略会让检测速度从 30fps 下降到 8fps，但却将 ODS F-measure 从 0.806 提高到 0.811。

3.1.2.4 与 HED 的比较

本节所提出的 RCF 与 HED[23] 相比有三个最明显的区别。首先，HED 仅考虑 VGG16 各个阶段中的最后一个卷积层，因此忽略了许多对边缘检测有用的信息。与之对比，RCF 使用来自所有卷积层的更丰富的特征，使得它可以准确地捕获更多的各种尺度下的物体或部分物体的边缘。其次，本节提出了一个新的损失函数来合理地处理训练样本。RCF 只考虑被大多数标注者标记为正样本的边缘像素，由于这些边缘像素是高度一致的，因此易于训练；RCF 忽略了一些由少量标注者标注的边缘像素，因为它们的边缘属性是模棱两可的。最后，RCF

使用多尺度的金字塔来增强边缘。RCF 的评测结果证明了这些选择的优势：与 HED 相比，ODS F-measure 提高了 2.3%（详情请参见3.1.3）。

3.1.3 实验

本节使用著名的开源框架 Caffe[181] 来实现所提出的网络，并使用在 ImageNet[5] 上预训练的 VGG16 模型[3] 来初始化该网络。RCF 将 pool4 的步幅改为 1，并使用扩张卷积[118] 来保持感受野大小。在 RCF 的训练中，与骨干网络相连的 1×1 卷积层的权重由标准偏差为 0.01 的零均值高斯分布初始化，卷积层的偏差初始化为 0。融合阶段的 1×1 卷积层的权重初始化为 0.2，而偏差同样初始化为 0。整个网络使用随机梯度下降（Stochastic Gradient Descent, SGD）进行训练，在每次迭代中为小批量随机采样 10 张图像。对于其他 SGD 的超参数，全局学习率设置为 1e-6，并且每 10k 次迭代后就将其除以 10。动量（Momentum）和权重衰减（Weight Decay）分别设置为 0.9 和 0.0002。训练总共进行 40k 次迭代。此外，损失函数中的参数 η 和 λ 是根据训练数据设置的。本节中所有实验均使用一块 NVIDIA TITAN X GPU 完成。

给定一张边缘概率图，需要设置一个阈值来得到相应的边缘图像。设置此阈值的方法有两种。第一种为最佳数据集尺度（Optimal Dataset Scale, ODS），它对数据集中的所有图像采用固定的阈值。第二种称为最佳图像尺度（Optimal Image Scale, OIS），它为每张图像选择相应的最佳阈值。本节在实验中同时使用了 ODS 和 OIS 两种方式下的 F-measure 来进行评测。

3.1.3.1 BSDS500 数据集上的评测

BSDS500[25] 是边缘检测中广泛使用的一个数据集。它由 200 张训练图像、100 张验证图像和 200 张测试图像组成，且每张图像都被 4 到 9 个标注者所标注。本节使用训练集和验证集对 RCF 网络进行训练，然后利用测试集进行评测。数据增强与 HED 方法[23] 中相同。受之前研究[182–184] 的启发，本节将进行数据增强后的 BSDS500 和经过翻转的 PASCAL VOC Context 数据集[185] 混合在一起作为训练数据。在训练时，损失函数中的参数 η 和 λ 分别设置为 0.5 和 1.1。在评测时，使用标准的非极大值抑制（Non-Maximum Suppression, NMS[53]）来使检测到的边缘变细。本节将所提出的 RCF 与一些非深度学习算法，如 Canny[46]、EGB[24]、gPb-UCM[25]、ISCRA[64]、MCG[26]、MShift[186]、NCuts[51]、SE[53] 和 OEF[187]，以及一些基于深度学习的方法，如 DeepContour[21]、DeepEdge[180]、

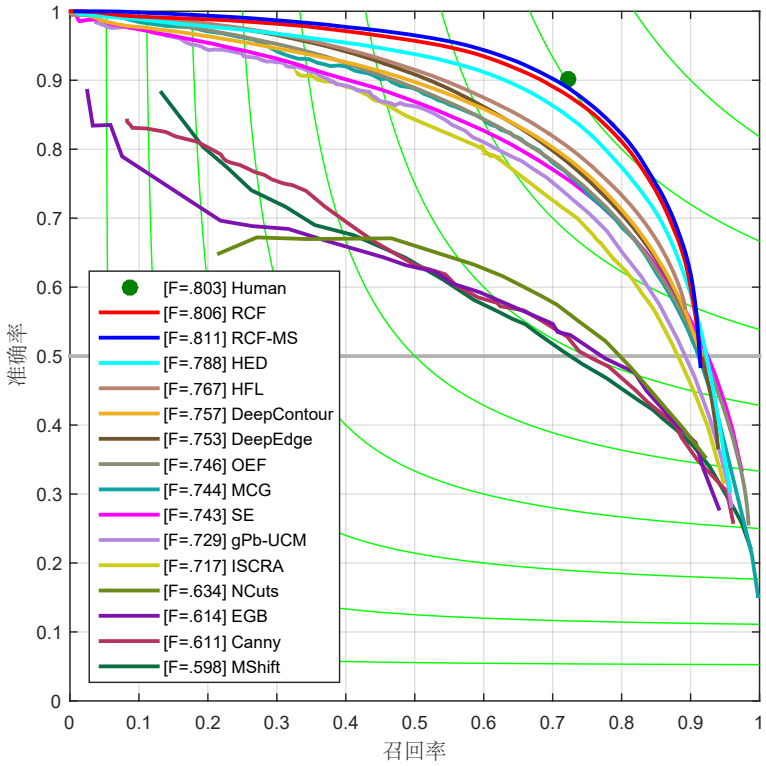


图 3.4 RCF 在标准的 BSDS500 数据集[25] 上的评测结果。单尺度和多尺度版本的 RCF 都取得了比人类标注更好的性能。

HED[23]、HFL[188] 和 MIL+G-DSN+MS+NCuts[184] 等进行了比较。

评测结果如图3.4所示。人类标注在边缘检测中的准确性被认为是 0.803 ODS F-measure，而单尺度和多尺度（RCF-MS）版本的 RCF 都实现了比人类更好的性能。在 ODS F-measures 指标上，所提出的 RCF-MS 和 RCF 分别比 HED[23] 高 2.3% 和 1.8%，而且 RCF 的准确率-召回率曲线也高于 HED。这些显著的提升证明了本章所提出的更丰富的卷积特征的有效性，也就是说，每个卷积阶段中，不只最后一层而是所有的卷积层都包含有用的层次信息。

定量比较结果如表3.2所示。从 RCF 到 RCF-MS，虽然检测速度从 30fps 下降到 8fps，但 ODS F-measure 从 0.806 增加到 0.811，这证明了所提出的多尺度策略的有效性。此外，当在 BSDS500 基准上使用默认参数进行评测时，RCF 曲线的长度不如其他方法长，这一现象表明 RCF 更倾向于检测置信度更高的边缘。与最近的边缘检测器（例如 RDS[182] 和 CEDN[183]）相比，RCF 也取得了更好的检测结果。RDS 使用松弛标签和额外的训练数据来重新训练 HED 网络，与 HED 相比，在 ODS F-measure 上提高了 0.4%。而所提出的 RCF 在 ODS F-measure 上

表 3.2 RCF 在 BSDS500 数据集[25] 上与其他方法的比较。† 表示 GPU 时间。性能最好的三个结果分别以红色、绿色和蓝色突出显示。

方法	ODS	OIS	帧/秒
Canny[46]	.611	.676	**28**
EGB[24]	.614	.658	**10**
MShift[186]	.598	.645	1/5
gPb-UCM[25]	.729	.755	1/240
Sketch Tokens[52]	.727	.746	1
MCG[26]	.744	.777	1/18
SE[53]	.743	.763	2.5
OEF[187]	.746	.770	2/3
DeepContour[21]	.757	.776	1/30†
DeepEdge[180]	.753	.772	1/1000†
HFL[188]	.767	.788	5/6†
N^4-Fields[54]	.753	.769	1/6†
HED[23]	.788	.808	**30†**
RDS[182]	.792	.810	**30†**
CEDN[183]	.788	.804	**10†**
MIL+G-DSN+MS+NCuts[184]	**.813**	**.831**	1
RCF	**.806**	**.823**	**30†**
RCF-MS	**.811**	**.830**	8†

比 RDS 还要高 1.4%，这表明本节的改进不是微不足道的。

从以上可以看出，RCF 实现了检测性能和检测效率之间的最佳平衡。尽管 MIL+G-DSN+MS+NCuts[184] 的精度比 RCF 要好一些，但是 RCF 和 RCF-MS 的速度要比它快很多。单尺度的 RCF 的速度能达到 30fps，RCF-MS 也可以达到 8fps。值得注意的是，RCF 网络仅向 HED 添加了一些 1×1 卷积层，因此时间消耗与 HED 几乎相同。而 Iasonas 等人[184] 往 HED 中添加了一些有用的组件，例如多实例学习（Multiple Instance Learning, MIL）[146]、G-DSN[89]、多尺度、使用 PASCAL Context 数据集[185] 作为额外训练数据、以及标准化切割（NCuts）[25] 等。相比之下，RCF 比 MIL+G-DSN+MS+NCuts[184] 要简单得多。由于 RCF 边缘检测器简单而高效，因此可以很容易地将它们应用到其他高层计算机视觉任务中去。

3.1.3.2 NYUD 数据集上的评测

NYUD 数据集[189] 由 1449 个密集标记的成对的 RGB 图像和深度图组成，主要包含了各种室内场景，对室内应用具有重要意义。最近，许多研究[53, 177] 都

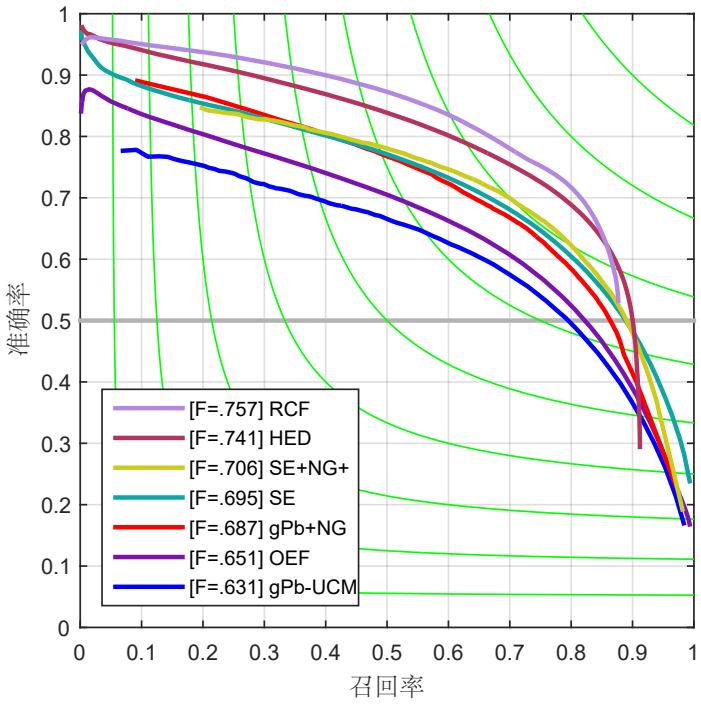


图 3.5 RCF 在 NYUD 数据集[189] 上的评测结果，这里的 RCF 为单尺度版本。

在此数据集上对边缘检测进行评测。Gupta 等人[190] 将 NYUD 数据集划分为 381 张训练图像、414 张验证图像和 654 张测试图像。本节保持和他们相同的设置，并像 HED[23] 中一样，使用全分辨率的训练集和验证集来训练 RCF 网络。本节通过使用 HHA[191] 来利用深度信息，其中，深度信息被编码为三个通道：水平视差、地上高度和重力角。因此，HHA 特征可以看作一张彩色图像。然后，分别使用 RGB 图像和 HHA 特征图像来训练两个模型。每张图像和相应的标注被旋转到四个不同的角度（即 0、90、180 和 270 度），并在每个角度都对它们进行水平翻转，因而将训练数据增强了八倍。在训练过程中，将 RGB 和 HHA 的 λ 都设置为 1.2。由于 NYUD 的每张图像都只有一个相应的真值标注图，因此 η 在这里是无效的。其他的网络设置与在 BSDS500 上的设置相同。在测试时，通过将 RGB 模型和 HHA 模型的输出简单地取平均来得到最终的边缘预测。在对预测结果进行评测时，由于 NYUD 数据集中的图像大于 BSDS500 数据集中的图像，因此本节遵循之前的研究[23, 53] 将定位公差（该值控制了在预测的边缘点与真值图的边缘点之间进行匹配的过程中所允许的最大偏移距离）从 0.0075 提高到 0.011。

本节仅将单尺度版本的 RCF 与一些著名的边缘检测方法进行比较。其中，

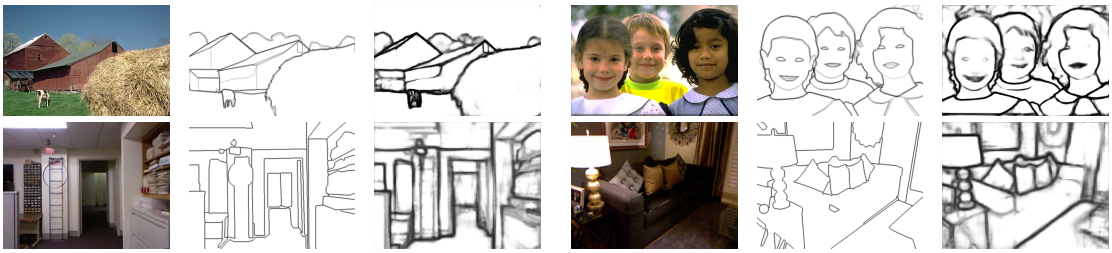


图 3.6 RCF 的一些示例图。第一行和第二行分别选自于 BSDS500[25] 和 NYUD[189] 数据集。从左到右依次是：原图、真值图、RCF 边缘图、原图、真值图、RCF 边缘图。

表 3.3 RCF 在 NYUD 数据集[189] 上与一些方法的比较。† 表示 GPU 时间。

方法	ODS	OIS	帧/秒
OEF[187]	.651	.667	1/2
gPb-UCM[25]	.631	.661	1/360
gPb+NG[190]	.687	.716	1/375
SE[53]	.695	.708	5
SE+NG+[191]	.706	.734	1/15
HED-HHA[23]	.681	.695	**20**†
HED-RGB[23]	.717	.732	**20**†
HED-RGB-HHA[23]	.741	.757	10†
RCF-HHA	.705	.715	**20**†
RCF-RGB	.729	.742	**20**†
RCF-HHA-RGB	**.757**	**.771**	10†

OEF[187] 和 gPb-UCM[187] 仅使用了 RGB 图像而没有使用深度信息，而其他方法则同时使用了深度和 RGB 信息。图3.5展示了准确率-召回率曲线。从图中可知，RCF 在 NYUD 数据集上获得了最佳性能，其次是 HED[23]。表3.3展示了定量比较的结果。可以看出，RCF 不仅在单独的 HHA 或 RGB 数据上，在合并的 HHA-RGB 数据上也取得了比 HED 更好的结果。对于 HHA 和 RGB 数据，RCF 在 ODS F-measure 指标上分别比 HED 高 2.4% 和 1.2%。对于合并的 HHA-RGB 数据，RCF 比 HED 要高 1.6%。此外，只用 HHA 的边缘的性能要比只用 RGB 的差，但将 HHA 和 RGB 边缘进行平均后就可取得更高的结果。这说明组合不同类型的信息对于边缘检测非常有用，这可能也是 OEF 和 gPb-UCM 比其他方法表现更差的原因。

3.1.3.3 RCF 网络讨论

为了进一步探究所提出的 RCF 网络结构的有效性，本小节使用 VGG16[3] 实现了一些混合网络，方法是将本节提出的基于更丰富的卷积特征的侧输出连接

表 3.4 一些混合网络的结果

使用 RCF 侧输出的阶段	使用 HED 侧输出的阶段	ODS	OIS
1, 2	3, 4, 5	.792	.810
2, 4	1, 3, 5	.795	.812
4, 5	1, 2, 3	.790	.810
1, 3, 5	2, 4	.794	.810
3, 4, 5	1, 2	.796	.812
–	1, 2, 3, 4, 5	.788	.808
1, 2, 3, 4, 5	–	.798	.815

到一些卷积阶段，而将 HED[23] 的侧输出连接到其他卷积阶段。当仅在 BSDS500 数据集[25] 上进行训练并使用单尺度进行测试时，这些混合网络的评测结果如表3.4所示。该表的最后两行分别对应 HED 和 RCF。可以观察到，所有混合网络的性能都比 HED 好，但比完全连接了 RCF 侧输出的 RCF 要差，这清楚地证明了所提出的更丰富的卷积特征对于边缘检测的重要性。

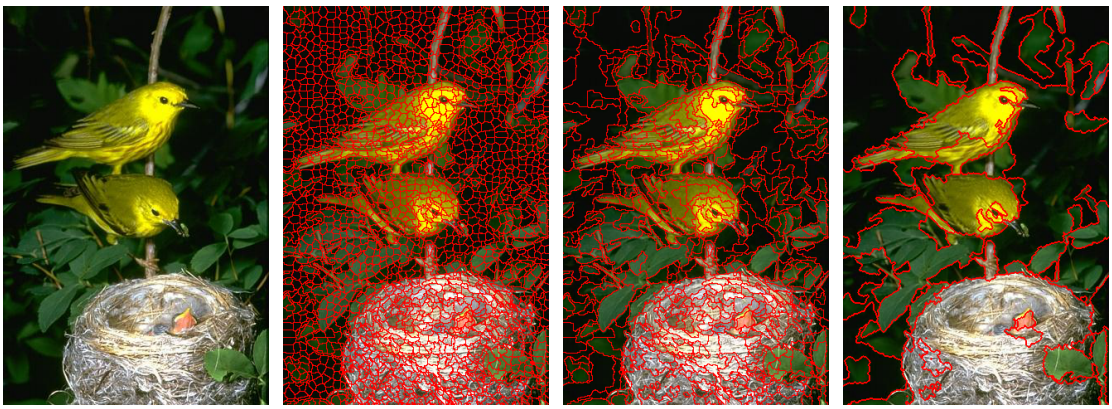
为了研究添加更多的非线性激活是否有帮助，本节在每个阶段的 $1 \times 1 - 21$ 或 $1 \times 1 - 1$ 卷积层后连接 ReLU 层，却发现网络性能变得更差。特别是将非线性层添加到 $1 \times 1 - 1$ 卷积层后时，发现网络无法正常收敛。

第二节 通用的图像属性知识 - 图像过分割

3.2.1 引言

图像过分割被认为是计算机视觉的主要挑战之一，过去已经对其进行了广泛的研究。经过数十年的研究，本领域的研究人员之间达成了共识，即准确的片段（无论是大区域还是小超像素）可作为中层和高层的视觉任务的有效输入，一些典型任务包括：物体检测/识别[192, 193]、跟踪[194]、显著性估算[27, 28]、似物性采样[30, 102, 110]、语义分割[195]、以及 3D 推理[196] 等。发生这种情况的原因有三点：1) 提取的分割是有意义的单元，具有诸如形状、纹理等信息特征[197–199]；2) 分割的数量通常大大低于原始图像中的像素数量，从而产生了更紧凑的表示形式，并带来了极大的速度优势[27]；3) 与原始像素相比，超像素表示通常具有更好的相干性和鲁棒性[200]。

过去，已经出现了几项具有开创性的工作，它们是在本领域中被广泛采用的最新系统：基于频谱聚类的归一化切割（NCuts）方法[51]；有效的特征（颜色）空间模式搜索方法，均值漂移（Mean Shift）算法[186]；基于图的高效图像



(a) 输入 (b) 第一步的 SLIC (c) 第二步的合并 (d) 第三步的合并

图 3.7 所提出的过分割方法 HFS 在不同步骤的示例结果。原始图片（481 × 321）来自 BSDS500 数据集[25]。这些图像过分割结果在 GPU 上以 50fps 生成。

过分割方法（EGB）[24]；基于轮廓转换的分层区域树（gPb-UCM）[25]；以及多尺度归一化切割算法（MCG）[26]。在这些选择中，基于图的高效图像过分割[24]和 SLIC[59] 方法由于其巨大的速度优势而在计算机视觉和计算机图形学中特别受欢迎[27, 28, 30, 102, 110, 192–194, 196, 201]。但是它们的准确性特别差，尤其是在基于区域的评测指标上，因此不能满足当今的视觉任务的要求。而尽管 MCG[26] 和 gPb[25] 可以生成高质量的分割，但它们速度太慢了，从而无法应用于对时间敏感的任务。因此，现有的方法难以平衡分割精度和计算时间的关系。

与图像过分割相似但略有不同，超像素生成算法将输入图像分割成小的、规则的、紧凑的区域，这些区域被称作超像素，这与通过普通图像过分割技术生成的大的感知区域不同，典型的超像素生成算法如 SLIC[59]。超像素通常具有很强的边界一致性，并且产生的超像素的数量易于控制。由于超像素方法通常被设计为生成小区域，因此不宜直接使用它们来生成大区域。但是，超像素算法为图像过分割提供了一个良好的开端。本章旨在设计一种图像过分割算法，该算法可以在效率和效果之间取得良好的平衡。考虑到效率，本章的工作从快速的超像素生成方法开始，即 SLIC 的 GPU 版本[59, 202]，通过合并超像素来生成图像的过分割。由于超像素的数量远少于图像的像素数，因而提高了速度；由于超像素具有比单个像素更加丰富的特征信息，因而还取得了可观的性能。

因为图像过分割没有固定类别的标签，所以无法像语义分割那样直接为其学习深度特征。因此，本节首先提出了一种基于手工设计特征的分层特征选择的快速图像过分割方法（Hierarchical Feature Selection, HFS），该方法可以为实

时的计算机视觉任务生成高质量的图像过分割。HFS 提出了一个分层的特征选择框架，该框架可学习分层结构各个阶段中的特征组合。HFS 始于 SLIC 方法的 GPU 版本[59, 202]，以通过执行超像素生成来快速获得初始种子区域（超像素）。然后从各个种子区域中提取图像特征，随后进行特征组合过程，并结合从训练数据中学习到的距离度量。请注意，为保持系统的效率，HFS 仅考虑适合并行计算（即通过 GPU）的那些手工设计的图像特征。然后基于学习的距离度量执行区域合并过程，以输出用于分层结构中下一级的一组新区域。然后，HFS 系统重复进行几次迭代，从而输出最终的图像过分割。

由于深度学习具有强大的特征表征能力，因此将上述 HFS 方法中的手工设计的特征替换为深度特征是一件值得探索的事。为此，本节还提出了一种基于深度嵌入学习的方法（Deep Embedding Learning, DEL），以解决图像过分割没有固定类别的标签而无法用神经网络直接学习的难题。DEL 训练了一个全卷积神经网络来学习深层特征嵌入空间，为每个超像素学习多尺度的深度特征。并且引入了深度嵌入度量，该度量将相邻超像素的特征嵌入向量转换为相似度值。每个相似度值表示两个相邻超像素属于同一区域的概率。通过这种方式，DEL 可以端到端地训练深度嵌入空间，以学习每对超像素之间的相似性。进而提出了一种用于嵌入学习的新的网络，该网络结合了底部的精细细节和顶部的高级信息的特征。如果学习到的相邻超像素之间的相似度大于阈值，则将其合并为大图像块。由于深度学习特征的强大表示能力，这种简单的合并操作可以比 HFS 的层次合并获得更好的性能。

本节在 BSDS500[25] 和 PASCAL Context[185] 数据集上进行了广泛的实验，以评测所提出的 HFS 和 DEL 算法。实验结果表明，HFS 通过以 50fps 的速度生成高质量的图像过分割（另请参见图3.7），对各种实时应用具有实际意义。与其他方法[24–26, 59, 203] 相比，HFS 在分割质量和计算效率之间取得了良好的平衡。此外，尽管速度略慢，但是 DEL 取得了比 HFS 更高的精度。与其他方法相比，DEL 在效率和有效性之间取得了良好的折衷。具体而言，DEL 可以达到与最新的方法可比的分割结果，但比它们快得多，例如 DEL 的 11.4fps 与 MCG[26] 的 0.07fps。这意味着 DEL 有潜力在许多实际应用中使用。

3.2.2 基于分层特征选择的过分割方法（HFS）

本小节提出了基于分层特征选择的图像过分割方法，该方法以每秒 50 帧的速度进行图像过分割。HFS 通过着重于两个方面来尝试改善以前的图像过分割

系统的性能：1) 在现代 GPU 上精心地实现系统以进行有效的特征计算；2) 一种有效的分层特征选择与学习融合策略。与经典的分割算法相比，HFS 展示了其在速度方面的特殊优势，并且在分割质量上具有可比的结果。在诸如显著性物体检测和似物性采样之类的应用程序中采用 HFS 可以显著提高性能，HFS 还可以用于基于图像过分割和超像素提取的其他计算机视觉任务。具体来说，本节首先介绍 HFS 的问题表述和分层合并算法，然后解释参数学习和特征提取过程，最后讨论关于该方法背后的设计选择。

3.2.2.1 问题表述

给定图像 I，HFS 将其划分为 L 级分割 $\mathcal{S}=\{\mathcal{S}_1,\mathcal{S}_2,\cdots,\mathcal{S}_L\}$。每个分割 $\mathcal{S}_l$ 是具有 K_l 个区域的图像 I 的分解，

$$\mathcal{S}_l=\{R_1^{(l)},R_2^{(l)},\ldots,R_{K_l}^{(l)}\}, \tag{3.4}$$

其中 l 表示分层结构中的层次索引。HFS 从由大量区域组成的最细分割 $\mathcal{S}_1$ 开始，逐步将区域从 $\mathcal{S}_l$ 合并到较粗略的 $\mathcal{S}_{l+1}$。因此，最粗糙层次的分割由最少的区域组成。

在每个步骤中，HFS 采用基于图的方法[24] 来实现区域合并过程 $\mathcal{S}_l\Rightarrow\mathcal{S}_{l+1}$。令

$$G_l=(\mathcal{S}_l,\mathcal{A}_l) \tag{3.5}$$

为无向图，顶点为如上定义的一组区域 $\mathcal{S}_l$，而边 $(R_i^{(l)},R_j^{(l)})\in\mathcal{A}_l$ 对应于成对的相邻顶点。每条边 $(R_i^{(l)},R_j^{(l)})\in\mathcal{A}_l$ 具有特征向量 $\mathbf{T}_{i,j}^{(l)}$（另请参见第3.2.2.4节）和相应的预测分数 $s_{i,j}^{(l)}$，这是区域 $R_i^{(l)}$ 和 $R_j^{(l)}$ 之间距离的非负度量。根据上述问题定义，HFS 的任务是快速合并区域，以产生与人类注释最匹配的连贯分割，例如 BSDS500 基准[25] 中的分割。

3.2.2.2 分层合并

为了获得高质量并保持高效率，HFS 提出：1) 在每个层次进行区域合并后，迭代地学习如何组合特征和更新图像特征；2) 在合并之前，使用快速的并行的超像素生成方法[59, 202] 将图像像素分组到初始区域。

HFS 的流程显示在图3.8中，示例结果显示在图3.7中，算法显示在算法1中。第一步，利用 GPU-SLIC 方法[59, 202] 将输入图像过度分割为超像素，这些超像素用作第一层次中的种子区域 $\mathcal{S}_1=\{R_1^{(1)},R_2^{(1)},\ldots,R_{K_l}^{(1)}\}$。在随后的步骤中，将

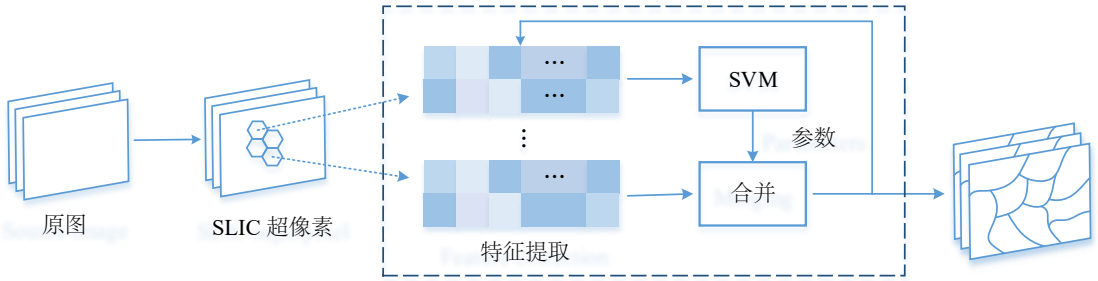


图 3.8 所提出 HFS 的算法流程

提取内部和边缘处的特征（另请参见第3.2.2.4节）。使用支持向量机（Support Vector Machine, SVM）回归器，HFS 从训练数据中学习（另请参见第3.2.2.3节）如何将特征向量 $\mathbf{T}_{i,j}^{(l)}$ 映射到区域 $R_i^{(l)}$ 和 $R_j^{(l)}$ 之间的合适的距离度量。遵循基于图的高效图像过分割（EGB）的框架[24]，HFS 用公式 (3.5) 中定义的图来逐步合并 $\mathcal{S}_l$ 中的区域，以获得较粗略的分割 $\mathcal{S}_{l+1}$。

HFS 的设计原则是受到最近使用判别式学习方法为各种视觉任务找到合适的特征组合的趋势所激发的[25, 28, 50]。许多心理物理学的研究[204] 表明，人类使用多种线索来分离自然场景中的物体。事实证明，与临时设计相比，提取图像特征并允许数据为自己说话是学习如何组合不同视觉线索的合适方法[25, 50]。在确定是否应合并两个区域时，图像特征在不同层次中扮演着不同的角色，这也促成了 HFS 的系统设计。在精细的层次，例如像素级别，颜色相似性和空间距离很重要，这在许多最新的图像过分割方法[24, 59] 中都可以看到。随着区域合并/分组为粗糙的层次，纹理相似度、区域之间的边缘、以及其他线索变得更加重要，成为判断是否应该合并两个区域的决定因素。

与其为所有层次学习线索/特征组合的单一规则[25, 50]，HFS 尝试了一种替代方法，该方法可以迭代地更新区域特征及其组合权重（另请参见图3.9），可以在实验中看到了该方法的有效性。基于这一点，HFS 设计了一个分层结构，涉及多个层次，递归地进行区域合并[24] 和特征更新。

3.2.2.3 参数学习

如上所述，给定一组初始区域，HFS 在每个区域对 $(R_i^{(l)}, R_j^{(l)}) \in \mathcal{A}_l$ 之间学习边的权重 $\mathbf{w}_{i,j}^{(l)} \in \mathbf{w}^{(l)}$。由于每个区域对都与一个特征向量 $\mathbf{T}_{i,j}^{(l)}$ 相关联，因此 HFS 的下一步是为层次 l 中的每个区域对提供一个标签。由于每个层次的初始区域可能具有不规则的形状，因此 HFS 使用 F-measure 来帮助确定 $\mathcal{A}_l$ 中每个区

算法 1 HFS 的区域合并

输入： 图像 I，权重 $\mathbf{w}$，迭代次数 L
输出： 分割 $\mathcal{S}_L$
初始化： $\mathcal{S}_l = \{R_1^{(l)}, R_2^{(l)}, \ldots, R_{K_l}^{(l)}\} \Leftarrow$ GPU-SLIC(I)[59, 202]
for $l = \{1, 2, \ldots, L-1\}$ **do**
  for each $(R_i^{(l)}, R_j^{(l)}) \in \mathcal{A}_l$ **do**
    $s_{i,j}^{(l)} \Leftarrow (\mathbf{T}_{i,j}^{(l)})^T \cdot \mathbf{w}^{(l)}$, 请分别参见第3.2.2.4节和3.2.2.3来了解 $\mathbf{T}_{i,j}^{(l)}$ 和 $\mathbf{w}^{(l)}$
  end for
  $\mathcal{S}_{l+1} =$ EGB$(\mathcal{A}_l, s_{i,j}^{(l)})$[24]
end for

域对的真值标签。

HFS 首先计算层次为 l 的初始分割的 F-measure，表示为 $F_{init}^{(l)}$。然后，对于 $\mathcal{A}_l$ 中的每个区域对 $(R_i^{(l)}, R_j^{(l)})$，HFS 计算在合并 $(R_i^{(l)}, R_j^{(l)})$ 之后的 F-measure。如果合并 $(R_i^{(l)}, R_j^{(l)})$ 之后的 F-measure 大于 $F_{init}^{(l)}$，则 $(R_i^{(l)}, R_j^{(l)})$ 的对应标签 $y_{i,j}^{(l)}$ 将被分配为 0。否则，$y_{i,j}^{(l)}$ 将分配为 1。HFS 采用支持向量机（SVM）回归器来学习特征权重 $\mathbf{w}^{(l)}$。

3.2.2.4 特征提取

HFS 探索了可以在现代 GPU 上高效计算的一组简单特征，这里把超像素内部和边缘处的特征都考虑了。表3.5列出了 HFS 所考虑到的特征。下面讨论系统中使用的这些特征的详细信息。

亮度与颜色. CIELAB 颜色空间中的亮度和颜色特征已被证明对图像过分割非常有效[25, 50]。本节使用一个区域的平均 L*a*b* 值来表示其颜色。为了容忍亮度和颜色的相对权重的变化，HFS 同时使用了两个相邻区域的欧拉距离（d_c）和每个通道的距离（d_l, d_a, d_b）。

沿边界平均的最大梯度. 先前的工作表明，梯度信息是边界检测中的重要线索。与其直接使用梯度，HFS 使用非极大值抑制之后的梯度。对于相邻的区域 R_i 和 R_j，计算开始于在像素 $p_k \in \Gamma$ 处放置一个小圆盘，其中 Γ 表示 R_i 和 R_j 之间的边界。然后计算圆盘中的最大梯度 $\delta'(p_k)$，最后计算边界上的平均 $\delta'(p_k)$ 作为梯度差异 $d_g(R_i, R_j)$。

表 3.5 HFS 使用的相邻区域的特征

特征	维度	符号
CIELAB 各通道的差异	3	d_l, d_a, d_b
CIELAB 值的欧拉距离	1	d_c
沿边界平均的最大梯度	1	d_g
RGB 直方图之间的 χ^2 距离	1	χ^2_h
梯度直方图之间的 χ^2 距离	1	χ^2_H
RGB 值的方差	3	s_r, s_g, s_b
CIELAB 值的方差	3	s'_l, s'_a, s'_b

RGB 直方图之间的 χ^2 距离. 为了利用颜色信息的详细信息，HFS 采用了颜色直方图，该颜色直方图在 RGB 颜色空间中具有 $8 \times 8 \times 8$ 个维度。对于属于相邻区域的直方图，HFS 使用 χ^2 距离来衡量它们的差异。

梯度直方图之间的 χ^2 距离. 在为每个区域计算方向梯度直方图时，两个区域的 χ^2 距离也是一个有吸引力的选择。

方差. 方差是衡量一组数据波动的好方法。HFS 将 RGB（s_r, s_g, s_b）和 CIELAB（s'_l, s'_a, s'_b）颜色空间的方差应用于 $R_i \bigcup R_j$，其中 R_i 和 R_j 是相邻的区域。方差的大小反映了两个区域之间的相似性。

以上特征在不同层次上扮演着不同的角色。第一层和第二层的特征的权重比较显示在图3.9中。考虑到计算复杂性，HFS 只选择一小部分易于计算的特征，而不是全部使用。前五个特征是 d_l、d_a、d_b、d_c 和 d_g。本章中报告的所有实验结果均基于这些特征。

3.2.2.5 实现细节

为了设计一个实用的系统，HFS 选择 $L = 3$ 作为默认值。作者使用配备有 Intel Xeon CPU E5-2676 v3 @ 2.40GHz 和 NVIDIA GeForce GTX 980 Ti 的计算机进行所有实验。报告的所有运行时间都没有数据并行。

3.2.3 基于深度嵌入学习的图像过分割（DEL）

本小节提出了一种基于深度嵌入学习的图像过分割方法，该方法仍然从 SLIC 超像素[59] 开始。DEL 首先训练一个深度卷积神经网络，以学习相邻超像素之间的相似性，然后使用学习到的相似性将它们直接合并。本节将详细描述

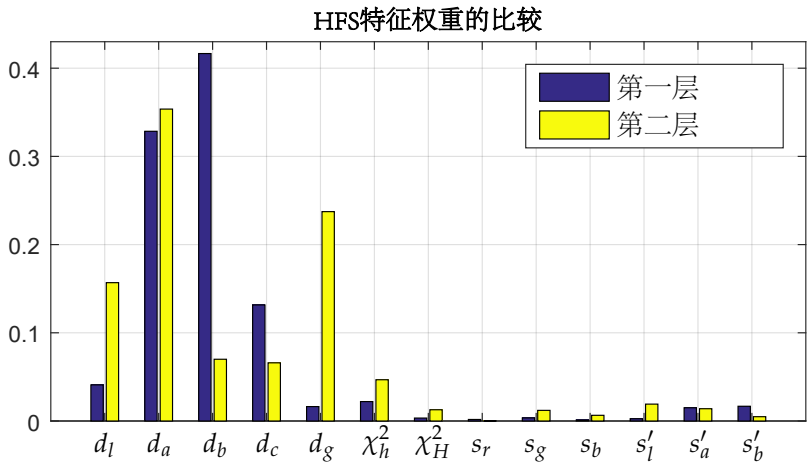


图 3.9 HFS 在第一层和第二层学习的特征的权重比较

DEL 算法的四个组成部分，依次是特征嵌入学习、网络架构、超像素合并、以及实现细节。

3.2.3.1 特征嵌入学习

与 HFS 算法一样，DEL 算法也是首先用 SLIC[59] 的 GPU 版本 gSLIC[202] 来为输入图像生成超像素。为了平衡运行时间和生成的超像素的边界一致性，DEL 控制每个超像素包含约 64 个像素。假设图像 I 生成了 M 个超像素，则生成的超像素集合表示为 $\mathcal{S}=\{S_1,S_2,\cdots,S_M\}, S_i=\{1,2,\cdots,|I|\}^{|S_i|}$。紧接着，DEL 训练一个深度卷积神经网络以学习特征嵌入空间。如图3.10中所示，DEL 在特征嵌入空间上执行池化操作以为每个超像素提取特征向量 $\mathbf{v}=\{\mathbf{v}_1,\mathbf{v}_2,\mathbf{v}_3,\cdots,\mathbf{v}_M\}$。每个特征向量是在超像素的相应区域中学习到的深度特征图的平均值。它可以表述为：

$$\mathbf{v}_i=\frac{1}{|S_i|}\sum_{k\in S_i}\mathbf{x}_k, \tag{3.6}$$

其中 $\mathbf{x}_k$ 表示第 i 个超像素对应区域内的特征向量。这种池化操作被称为*超像素池化*。在 DEL 的设计中，每个特征嵌入向量 $\mathbf{v}_i$ 是 64 维的。超像素池化层相对于输入 $\mathbf{x}_k$ 的反向传播函数可以写成

$$\frac{\partial L}{\partial \mathbf{x}_k}=\sum_{S_i\in\mathcal{S}}1_{\{k\in S_i\}}\cdot\frac{1}{|S_i|}\cdot\frac{\partial L}{\partial \mathbf{v}_i}, \tag{3.7}$$

其中 $1_{\{k\in S_i\}}$ 是一个指示函数。

DEL 设计了一个距离度量来测量相邻超像素之间的相似性。所提出的距离

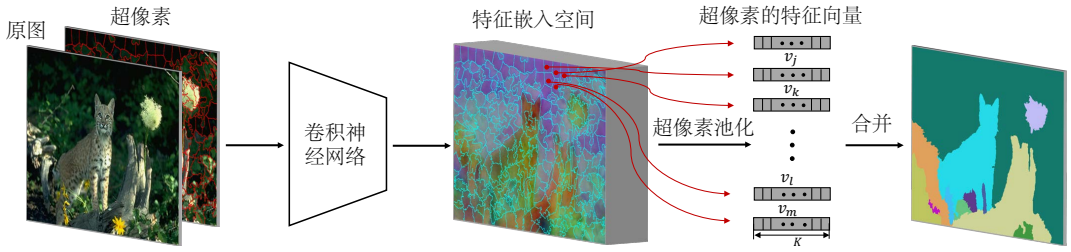


图 3.10 DEL 图像过分割算法的流程

度量可以表述为

$$d_{ij} = \frac{2}{1 + \exp(\|\mathbf{v}_i - \mathbf{v}_j\|_1)}. \tag{3.8}$$

相似度 $d_{i,j}$ 的取值范围为 $[0,1]$。当 v_i 和 v_j 相似时，其值接近 1；而当 v_i 和 v_j 极其不同时，其值接近 0。由于建立了距离度量，因此考虑相似度损失函数，如下所示：

$$\begin{aligned} L = -\sum_{S_i \in \mathcal{S}} \sum_{S_j \in R_i} & [(1-\alpha) \cdot l_{ij} \cdot \log(d_{ij}) \\ & + \alpha \cdot (1 - l_{ij}) \cdot \log(1 - d_{ij})], \end{aligned} \tag{3.9}$$

其中 $l_{ij} = 1$ 表示 v_i 和 v_j 属于同一区域，而 $l_{ij} = 0$ 则表示 v_i 和 v_j 属于不同的区域。R_i 是超像素 S_i 的相邻超像素集。$\alpha = |Y_+|/|Y|$，表示真值中属于相同区域的超像素对所占的比例，DEL 使用此参数来平衡训练中正负样本的比例。利用这种相似度损失函数，DEL 可以以一种端到端的方式学习特征嵌入空间。在相同的真值区域中的超像素对之间的相似性将被期望大于属于不同区域的超像素对之间的相似性。在下一步中，就可以使用学习到的相似性距离度量来合并这些超像素。

3.2.3.2 网络架构

本节介绍用于学习特征嵌入空间的网络架构。DEL 的网络基于 VGG16 网络[3] 构建。根据池化层，VGG16 中的卷积层可以分为五个卷积阶段。如图3.11中所示，DEL 剪切了 VGG16 网络中的 pool5 层和全连接层。由于 conv5 阶段的侧输出分辨率较低，因此 DEL 将 pool4 的步幅从 2 修改为 1，并使用扩张卷积[118] 来保持第五阶段的卷积的感受野大小与原始 VGG16 网络相同。一般认为，随着网络的加深，学习到的特征变得越来越粗糙。精细特征包含更多的详细信息，而粗糙特征则表示全局信息。五个阶段的特征被拼接起来，便可将粗糙的全局信息与精细的局部信息相结合。

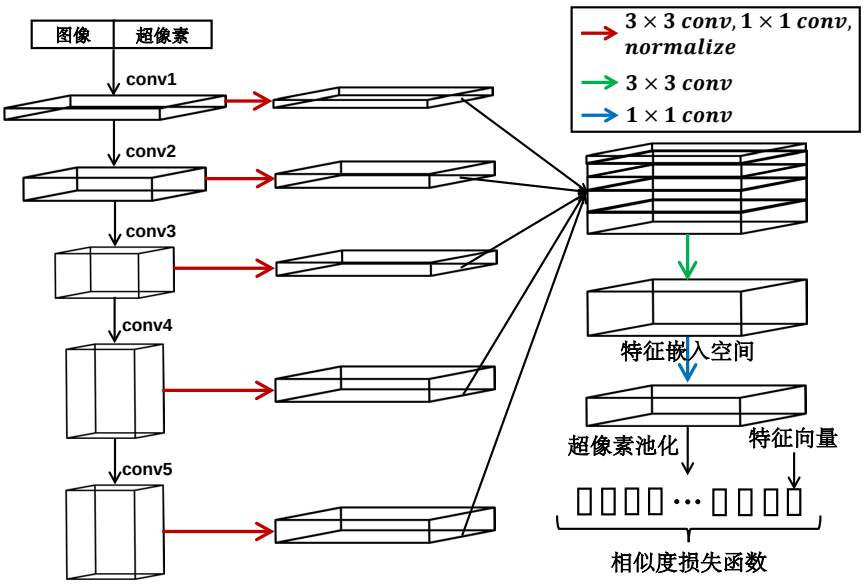


图 3.11 DEL 特征嵌入学习的网络架构

具体来说，DEL 在第 1-5 阶段分别连接一个具有 32、64、128、256、256 个输出通道的 3×3 卷积层。在这个 3×3 的卷积层之后，第 1-5 阶段再分别连接一个具有 32、64、64、128、128 个输出通道的 1×1 卷积层。由于不同卷积阶段的特征（数值）尺度不同，因此将多个阶段的特征直接拼接起来将使某些阶段的特征变得毫无意义。因此，DEL 使用 L2 归一化技术[205] 对不同阶段的响应进行归一化。归一化后，DEL 拼接各个阶段的特征图，然后跟随一个具有 256 个输出通道的 3×3 卷积层。最后，DEL 使用一个 1×1 卷积层来获得 64 维特征嵌入空间。如第3.2.3.1节所述，DEL 将特征嵌入合并到与超像素对应的特征向量，然后使用所提出的相似度损失函数来训练网络。

3.2.3.3 超像素合并

深度神经网络学习到的相邻超像素之间的差异度被用于将超像素合并为大的感知区域。DEL 设置一个阈值以确定是否应该合并两个相邻的超像素。算法2中展示了超像素合并的算法伪代码。为了提高合并效率，DEL 利用 EGB[24] 中提出的的数据结构 universe 来实现合并操作。与 HFS 中的分层合并策略不同，DEL 仅执行一次合并操作。HFS 使用一些底层特征的线性组合，并在每个合并阶段重新训练组合权重。由于深度特征更强的表示能力，DEL 的单阶段合并也可以显著优于 HFS。

算法 2 DEL 的超像素合并算法

输入： 图像 I, 差异度 $f=(1-d)$, 阈值 T, 超像素 $\mathcal{S}=\{S_1,S_2,\cdots,S_M\}$
构造 $\mathcal{R}=\{R_1,R2,\cdots,R_M\}$, 其中 R_i 是 S_i 的相邻超像素的集合
for each $S_i \in \mathcal{S}$ **do**
  for each $S_j \in R_i$ **do**
    if $f_{i,j} < T$: **then**
      $S_i \leftarrow S_i \cup S_j, \mathcal{S} \leftarrow \mathcal{S} \backslash S_j$
      更新 $\mathcal{R}$
    end if
  end for
end for
输出： 过分割 $\mathcal{S}$

3.2.3.4 实现细节

DEL 的网络用广泛使用的深度学习框架 Caffe[181] 来实现。一般来说，分割区域通常是指物体、部分物体或部分背景。因此，DEL 首先在 SBD 数据集[206] 上对网络进行语义分割任务的预训练，以获取网络的语义信息。预训练通过用语义分割任务的分类层替换特征嵌入空间来调整网络。然后，DEL 微调用于特征嵌入空间的预训练模型。使用随机梯度下降（SGD）技术来优化神经网络，基本学习率设置为 1e-5，权重衰减率为 0.0002，批处理大小为 5。使用 step 的学习率策略，并且针对 step size 为 8000 的 SGD 总共运行 10000 次迭代。如深度度量学习中建议的那样，特征嵌入层的学习率将被设置为大于其他卷积层。

数据增强对于深度学习非常重要。当使用 BSDS500 数据集[25] 中的 300 张训练和验证图像训练特征嵌入模型时，每张图像被旋转到 16 个角度，并且在每个角度水平翻转。然后，从转换后的图像中裁剪出最大的矩形，从而生成 9600 张训练图像。在具有 7605 张训练图像和 2498 张测试图像的 PASCAL Context 数据集[185] 上进行训练时，仅水平翻转每张图像，因为该数据集中的图像数量已经足够多。

3.2.4 实验

本小节在 BSDS500 数据集[25] 和 PASCAL Context 数据集[185] 上评测所提出的 HFS 和 DEL。为了评测图像过分割，本节使用了著名的基准 SEISM[207]，SEISM 既包含传统的基于边缘的度量，还包含新的基于区域的度量。因此，本小节在 ODS 和 OIS 上报告了边界 F-measure 和区域 F-measure。本小节将

表 3.6 在 BSDS500 数据集上对 DEL 的消融实验

方法	边缘		区域		时间（秒）
	ODS	OIS	ODS	OIS	
DEL-Max	0.703	0.738	0.323	0.389	0.088
DEL-*conv5*	0.667	0.695	0.278	0.343	0.070
DEL-EGB	0.662	0.686	0.305	0.325	0.091
DEL	0.704	0.738	0.326	0.397	0.088
DEL-C	0.715	0.745	0.333	0.402	0.165

HFS 和 DEL 与一些其他过分割算法进行了比较，包括 EGB[24]、Mean Shift[186]、NCuts[208]、gPb-UCM[25]、MCG[26]、SLIC[59]、以及 GPU-SLIC[202]。除了 SLIC 的 GPU 版本之外，本节还使用 SLIC 的 CPU 版本为 DEL 生成超像素，即 DEL-C。

3.2.4.1 消融实验

在与其他方法比较之前，先在 BSDS500 数据集[25] 上对 HFS 和 DEL 进行消融实验。图3.9展示了 HFS 所选特征的权重比较，可以清楚地看到权重的重要性在不同的层次上是不同的，这验证了 HFS 所使用的分层区域合并技术对手工设计特征的必要性。接下来评测每个 DEL 组件的不同选择。第一个 DEL 的变体表示为 DEL-Max，它用最大值操作替换超像素池化中的平均操作，其他 DEL 的组件保持不变。第二个变体 DEL-conv5 仅使用 VGG16 网络的最后一个卷积层（conv5）。第三种变体 DEL-EGB 通过将每个超像素视为图割问题中的一个结点来应用 EGB 的合并策略。评测结果汇总在表3.6中。与原始 DEL 相比，这些变体的性能变差。它表明 DEL 组件的原始的选择是合理的。例如，DEL 中提出的网络架构既可以捕获细节信息，也可以捕获粗糙信息，而 DEL-conv5 的简单设计仅使用了粗糙信息。因此，DEL 比 DEL-conv5 好得多。此外，EGB 对于超像素合并似乎毫无用处。最大值池化比平均池化稍差，这符合大家的直觉，即最大值池化和平均池化通常具有相似的效果。

3.2.4.2 BSDS500 数据集上的评测

由于 ODS F-measure 是最重要的分割指标，因此本小节在图3.12 中显示了 ODS F-measure 与运行时间的关系。所提出的 HFS 比 MCG[26] 和 gPb-UCM[25] 快数百倍，可达到 50fps。启用数据并行性后，速度最高可以达到 200+fps。因此，HFS 可以轻松地用于当今几乎所有的应用程序中，包括一些实时系统。与 EGB[24] 和 SLIC[59] 相比，HFS 要快得多，更重要的是，准确性也有了显著提

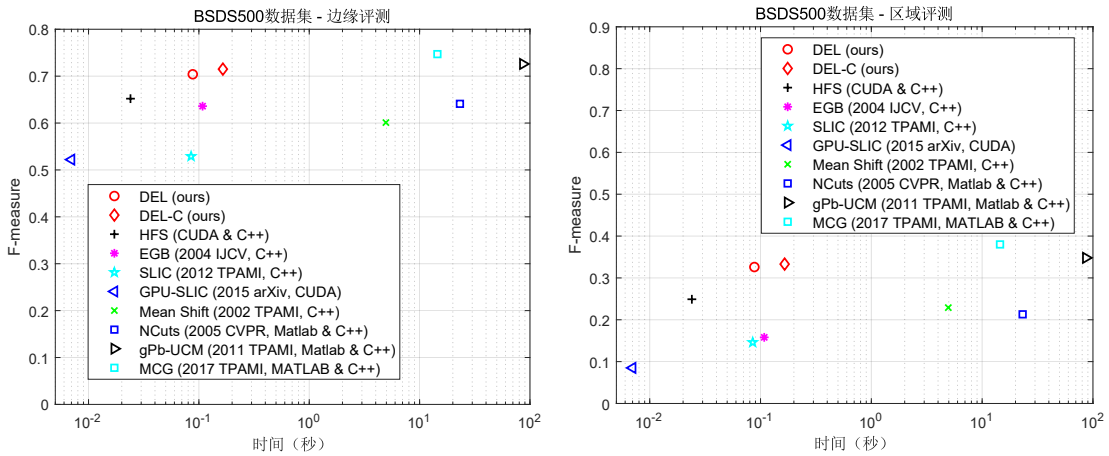


图 3.12 在 BSDS500 数据集上对 HFS 和 DEL 的评测结果。上图：边缘度量；下图：区域度量。

表 3.7 在 BSDS500 数据集上对 HFS 和 DEL 的评测结果

方法	边缘		区域		时间（秒）
	ODS	OIS	ODS	OIS	
EGB	0.636	0.674	0.158	0.240	0.108
SLIC	0.529	0.565	0.146	0.182	0.085
GPU-SLIC	0.522	0.547	0.085	0.132	0.007
MShift	0.601	0.644	0.229	0.292	4.95
NCuts	0.641	0.674	0.213	0.270	23.2
gPb-UCM	0.726	0.760	0.348	0.385	86.4
MCG	0.747	0.779	0.380	0.433	14.5
HFS	0.652	0.686	0.249	0.272	0.024
DEL	0.704	0.738	0.326	0.397	0.088
DEL-C	0.715	0.745	0.333	0.402	0.165

高。与 Mean Shift[186] 和 NCuts[208] 进行比较时，HFS 的速度优势显而易见，尽管 F-measure 仅比它们高一点。尽管所提出的 DEL 并没有达到最佳性能，但是它在效率和有效性之间取得了很好的权衡。从 GPU-SLIC/SLIC 到 DEL/DEL-C 的改进证明了 DEL 的深度嵌入特征学习方式的有效性。有趣的是，GPU-SLIC 的性能略低于 SLIC，而 DEL 的性能也略低于 DEL-C。由于 GPU-SLIC 的效率高（尽管效果略差），本文选择将其用作 DEL 的默认设置，用更准确的超像素生成方法替换 SLIC 可能会产生更好的性能。DEL 提供了从超像素到图像过分割的转换器，新的超像素技术将以这种方式有益于图像过分割。SLIC[59] 和 GPU-SLIC[202] 似乎在图像过分割方面遇到了困难。这符合一般的直觉，即超像素生成方法不适用于图像过分割。尽管 MCG[26] 获得了准确的结果，但它的低

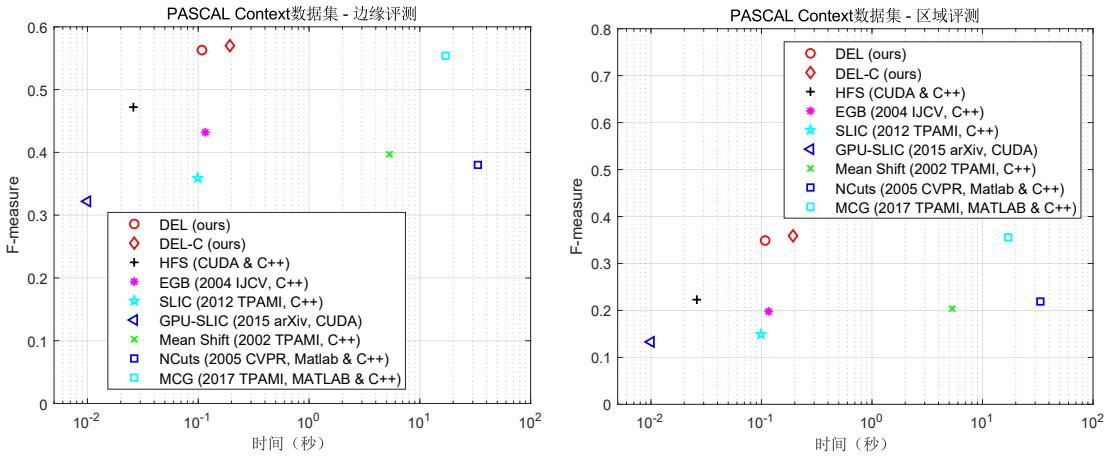


图 3.13 在 PASCAL Context 数据集上对 HFS 和 DEL 的评测结果。上图：边缘度量；下图：区域度量。

表 3.8 在 PASCAL Context 数据集上对 HFS 和 DEL 的评测结果

方法	边缘		区域		时间（秒）
	ODS	OIS	ODS	OIS	
EGB	0.432	0.454	0.198	0.203	0.116
SLIC	0.359	0.409	0.149	0.160	0.099
GPU-SLIC	0.322	0.340	0.133	0.157	0.010
MShift	0.397	0.406	0.204	0.214	5.32
NCuts	0.380	0.429	0.219	0.285	33.4
MCG	0.554	0.609	0.356	0.419	17.05
HFS	0.472	0.495	0.223	0.231	0.026
DEL	0.563	0.623	0.349	0.420	0.108
DEL-C	0.570	0.631	0.359	0.429	0.193

速限制了它在许多视觉任务中的应用。请注意，由于 MCG 不是可并行化的算法，因此不能直接实现 GPU 版本的 MCG。

数值比较总结在表3.7中。DEL 的 ODS 下的边缘 F-measure 和区域 F-measure 分别比 HFS 高 5.2% 和 7.7%，这归结于深度卷积神经网络强大的表示学习能力。在速度方面，HFS 达到 41.7fps，而 DEL 达到 11.4fps。尽管 DEL 的速度慢一些，但是从 HFS 到 DEL 的精度提高对于许多应用而言都很重要。与 EGB 相比，HFS 和 DEL 在准确性和速度上均具有更好的性能。DEL 可以产生与最新的性能相当的结果，但速度要快得多。因此，HFS 和 DEL 实现了有效性与效率之间的良好折衷，这使得它们可以适用于许多高级视觉任务。在图3.14中展示了一些定性比较。可以看到，DEL 可以适应复杂的场景并产生更准确和规则的分割区域。

图 3.14 一些定性比较。从左到右：BSDS500 中的原始图像、EGB、MCG、HFS 和 DEL。

3.2.4.3 PASCAL Context 数据集上的评测

PASCAL Context 数据集[185] 包含 540 个用于语义分割的类别，由于整个图像都按像素进行标记并且类别十分丰富，因此可以用于评测图像过分割方法。通过连通域标记将语义分割标注转换为图像过分割区域。将 DEL 在其训练集和验证集上进行训练，并在测试集上进行测试。由于测试图像更多，因此此数据集比 BSDS500 更具挑战性。

评测结果被汇总在图3.13中。HFS 在速度和精度上均优于除了 MCG 以外所有的之前的方法，MCG 虽然精度较高，但是速度很慢。DEL 和 DEL-C 比 MCG 具有更好的性能，而且，DEL 比 MCG 快 160 倍。可以看到，DEL 在精度和运行时间之间具有良好的权衡。在表3.8中列出了评测的数值结果。在边界度量和区域度量方面，DEL 分别比 HFS 高 9.1% 和 12.6%。这表明 DEL 学习到的深度特征比 HFS 中使用的手工设计特征更有效。因此，DEL 是采用基于深度学习的特征进行图像过分割的良好开端。

第三节 通用的图像属性知识 - 图像显著性检测

3.3.1 引言

显著性物体检测（也称为显著性检测）旨在模拟人类视觉系统来检测自然图像中最明显且最吸引眼球的物体或区域[27, 28]。显著性物体检测的发展对很多视觉应用都有所帮助，包括图像检索[209]、视觉跟踪[210]、场景分类[211]、内容感知的图像/视频处理[212]、缩略图生成[213]、视频物体分割[214] 和弱监督学习[154, 156] 等。尽管已经有许多模型[29, 70, 73, 88] 被提出并且取得了重大突破，但

是准确地检测静态图像中完整的显著性物体（尤其是在复杂场景中）仍然是一个有待解决的问题。

传统的显著性物体检测方法[27, 28, 215] 通常设计手工制定的底层特征和启发式先验，这些方式很难表征语义对象和场景。显著性物体检测的最新研究主要基于卷积神经网络。一方面，通过逐渐增大的感受野和下采样尺度，卷积神经网络可以自然地在不同深度的卷积层中学习多尺度和多层次的特征表示。另一方面，由于图像内及图像间的物体/场景的尺度各不相同，所以显著性物体检测需要多尺度学习[104]。因此，当前最先进的显著性物体检测器[29, 70–72, 74–76] 主要旨在设计复杂的卷积神经网络体系结构来利用多尺度的卷积特征，即高层语义信息和与之互补的底层空间细节信息。

由于 U-Net[84]（或 FCN[83]）和 HED[23] 在多尺度学习中的优越性，许多领先的显著性物体检测器均在 U-Net 网络的基础上添加了深监督[29, 73, 74, 79]。可以发现，这些网络首先使用侧输出来预测多尺度显著性图，然后通过逐像素卷积（即 1×1 卷积）等操作将生成的多尺度侧输出预测进行线性融合，以此来获得最终的显著性预测，从而可以有效地利用所有侧输出预测的优势。但是，本章在理论上和实验上均证明了这种线性融合侧输出预测的做法不是最优的，它对侧输出特征中的互补的多尺度信息的利用是有限的。这一点将在3.3.2中提供更详细的证明。

不同于线性融合侧输出预测，本节提出了一种非线性侧输出融合的方法。具体来说，本文将侧输出特征而不是侧输出预测进行拼接，然后使用非线性变换来预测显著性物体。与此同时，本文还对侧输出特征添加了深监督，使得卷积神经网络在训练阶段能被更好地优化。用这种方式，所拼接的特征可以更好地利用多尺度侧输出特征。所提出的这种方法被命名为深监督非线性融合（Deeply-supervised Nonlinear Aggregation, DNA）。将 DNA 用于经简单修改后的 U-Net，在不需要复杂的工程技巧的前提下，所提出的网络即可取得比当前先进的显著性物体检测方法更好的效果，并且具有更少的参数和更快的检测速度。

3.3.2 深监督线性融合的回顾

深监督和相应的线性侧输出预测融合在许多计算机视觉任务中都被证实非常有效[23, 29, 74]。本节从理论和实验两个角度分析了线性侧输出融合的局限性。

假设一个具有深监督的网络有 N 个侧输出预测图 $\{\mathcal{O}_1, \mathcal{O}_2, \cdots, \mathcal{O}_N\}$，它们均使用真值图进行监督。在不失一般性的前提下，假设线性侧输出融合是一个

逐像素卷积，即 1×1 卷积。因此，当前的线性侧输出融合方式可以写成

$$\widehat{\mathcal{O}}=\sum_{i=1}^{N}\boldsymbol{w}_i\cdot\mathcal{O}_i, \tag{3.10}$$

这里，逐像素卷积的权重 $\boldsymbol{w}_i$ 是可以学习的。注意，这里有 $\boldsymbol{w}_i\geq0$；否则，由于 $\mathcal{O}_i$ 对 $\widehat{\mathcal{O}}$ 有负面影响, 因此在融合时需要将其排除在外。要获得输出的显著性概率图，还必须在 $\widehat{\mathcal{O}}$ 上使用标准的 sigmoid 函数 $\sigma(x)=\frac{1}{1+e^{-x}}$。这样，融合的概率图就变为

$$\widehat{\mathcal{P}}=\sigma(\widehat{\mathcal{O}})=\sigma(\sum_{i=1}^{N}\boldsymbol{w}_i\cdot\mathcal{O}_i). \tag{3.11}$$

同理，可以计算侧输出概率图 $\{\mathcal{P}_1,\mathcal{P}_2,\cdots,\mathcal{P}_N\}$。

理论 1 *若 $\|\boldsymbol{w}\|_1=1$，融合输出 $\widehat{\mathcal{P}}$ 的平均绝对误差（Mean Absolute Error, MAE）受侧输出预测的限制。*

证明. 若 $\|\boldsymbol{w}\|_1=1$，由于 $\boldsymbol{w}_i\geq0$，所以

$$\min(\mathcal{O}_i)\leq\sum_{i=1}^{N}\boldsymbol{w}_i\cdot\mathcal{O}_i\leq\max(\mathcal{O}_i). \tag{3.12}$$

由于 sigmoid 函数 $\sigma(x)$ 单调递增，所以可以得到

$$\min(\mathcal{P}_i)\leq\widehat{\mathcal{P}}\leq\max(\mathcal{P}_i). \tag{3.13}$$

若某一像素 $\boldsymbol{p}$ 为正，$\mathrm{MAE}(\widehat{\mathcal{P}})_{\boldsymbol{p}}=|1-\widehat{\mathcal{P}}(\boldsymbol{p})|=1-\widehat{\mathcal{P}}(\boldsymbol{p})$ 且 $1-\max(\mathcal{P}_i)_{\boldsymbol{p}}\leq1-\widehat{\mathcal{P}}(\boldsymbol{p})\leq1-\min(\mathcal{P}_i)_{\boldsymbol{p}}$，所以 $\min(\mathrm{MAE}(\mathcal{P}_i)_{\boldsymbol{p}})\leq\mathrm{MAE}(\widehat{\mathcal{P}})_{\boldsymbol{p}}\leq\max(\mathrm{MAE}(\mathcal{P}_i)_{\boldsymbol{p}})$。若某一像素 $\boldsymbol{p}$ 为负，$\mathrm{MAE}(\widehat{\mathcal{P}})_{\boldsymbol{p}}=|0-\widehat{\mathcal{P}}(\boldsymbol{p})|=\widehat{\mathcal{P}}(\boldsymbol{p})$ 且 $\min(\mathcal{P}_i)_{\boldsymbol{p}}\leq\widehat{\mathcal{P}}(\boldsymbol{p})\leq\max(\mathcal{P}_i)_{\boldsymbol{p}}$, 所以 $\min(\mathrm{MAE}(\mathcal{P}_i)_{\boldsymbol{p}})\leq\mathrm{MAE}(\widehat{\mathcal{P}})_{\boldsymbol{p}}\leq\max(\mathrm{MAE}(\mathcal{P}_i)_{\boldsymbol{p}})$。需要注意的是，因为 $\boldsymbol{w}$ 通常有 N 维 (在 VGG16[3] 和 ResNet[4] 中 $N\leq6$)，所以很难保证上述左等式成立。因此，传统的线性融合在 MAE 度量上受到限制。然而，研究者们期望的却是通过充分利用多尺度信息来突破该限制。 □

引理 1 *若 $\|\boldsymbol{w}\|_1\neq1$，传统的线性融合（如公式 (3.10) 和 公式 (3.11) 所示）等效于先使用 $\|\tilde{\boldsymbol{w}}\|_1=1$ 进行融合，然后使用一个单调递增的映射。*

证明. 若 $\|\boldsymbol{w}\|_1\neq1$，可以设 $\boldsymbol{w}=\tilde{\boldsymbol{w}}\cdot\|\boldsymbol{w}\|_1$，并且假设 $\|\tilde{\boldsymbol{w}}\|_1=1$。$\widehat{\mathcal{P}}$ 的计算变为

$$\widehat{\mathcal{P}}=\sigma(\|\boldsymbol{w}\|_1\cdot\sum_{i=1}^{N}\tilde{\boldsymbol{w}}_i\cdot\mathcal{O}_i), \tag{3.14}$$

这里，$\sigma(\|\boldsymbol{w}\|_1\cdot x)$ $(\|\boldsymbol{w}\|_1>0)$ 是关于 x 的单调递增函数。 □

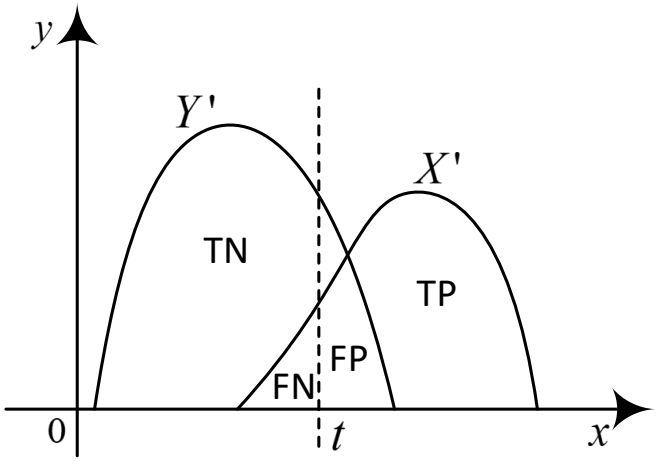


图 3.15 概率（x 轴）和 X' 及 Y' 的密度（y 轴）。TN：真阴性；FN：假阴性；TP：真阳性；FP：假阳性。

理论 2 *$\sigma(\|\boldsymbol{w}\|_1 \cdot x)$ $(\|\boldsymbol{w}\|_1 > 0)$ 的单调递增映射无法改变 ROC 曲线和 AUC 指标*[1]。

证明. 假设正样本的预测值服从 $X \sim F(x)$ 的分布，而负样本的预测值服从 $Y \sim G(x)$ 的分布。可以假设 F 和 G 是连续函数。$\varphi(x) = \sigma(k \cdot x)$ $(k > 0)$ 是 sigmoid 函数的一种变体，因此可以得到 $\varphi : \mathbb{R} \to (0,1)$ 且 φ 是单调递增函数。令 $X' = \varphi(X)$ 和 $Y' = \varphi(Y)$ 是两个变换分布，很容易得到

$$\begin{aligned} \mathbb{P}(X' \leq u) = \mathbb{P}(\varphi(X) \leq u) &= \mathbb{P}(X \leq \varphi^{-1}(u)) \\ &= F(\varphi^{-1}(u)), \end{aligned} \tag{3.15}$$

因此可以得到 $X' \sim F(\varphi^{-1}(x))$ 且 $Y' \sim G(\varphi^{-1}(x))$。

令 t 为阈值，如图3.15所示，真阳性率（True Positive Rate, TPR）和假阳性率（False Positive Rate, FPR）可以计算为

$$\begin{aligned} \mathrm{TPR} &= \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}} = \mathbb{P}(X' > t) = 1 - F(\varphi^{-1}(t)), \\ \mathrm{FPR} &= \frac{\mathrm{FP}}{\mathrm{FP}+\mathrm{TN}} = \mathbb{P}(Y' > t) = 1 - G(\varphi^{-1}(t)). \end{aligned} \tag{3.16}$$

因此，可以将 ROC 曲线表示为 $\{(1 - F(\varphi^{-1}(t)), 1 - G(\varphi^{-1}(t))) : t \in (0,1)\}$。很容易看出，随着 t 从 0 连续变化到 1，$(1 - F(\varphi^{-1}(t)), 1 - G(\varphi^{-1}(t)))$ 也将从 $(1,1)$ 连续单调变化到 $(0,0)$。显而易见，$\{(1 - F(\varphi^{-1}(t)), 1 - G(\varphi^{-1}(t)))\}$ 和

[1] AUC 是 ROC 曲线下的面积。

表 3.9 线性侧输出预测融合和非线性侧输出特征融合之间的比较。数据集和评测指标将在第3.3.4.1节中进行介绍。HED[23] 和 DSS[76] 的线性融合即为原论文中使用的融合方式，而它们的非线性融合就是将线性融合替换为本文所提出的 DNA。

方法	融合	DUTS-TE		ECSSD		HKU-IS		DUT-O		THUR15K	
		F_β	MAE	F_β	MAE	F_β	MAE	F_β	MAE	F_β	MAE
HED[23]	线性	0.796	0.079	0.892	0.065	0.893	0.052	0.726	0.100	0.757	0.099
	非线性	**0.827**	**0.057**	**0.911**	**0.053**	**0.912**	**0.039**	**0.752**	**0.078**	**0.775**	**0.083**
DSS[76]	线性	0.827	0.056	0.915	0.056	0.913	0.041	0.774	0.066	0.770	0.074
	非线性	**0.833**	**0.055**	**0.918**	**0.056**	**0.916**	**0.040**	**0.784**	**0.060**	**0.773**	**0.072**
DNA	线性	0.844	0.048	0.921	0.050	0.917	0.034	0.765	0.066	0.785	0.071
	非线性	**0.865**	**0.044**	**0.935**	**0.041**	**0.930**	**0.031**	**0.799**	**0.056**	**0.793**	**0.069**

$\{(F(\varphi^{-1}(t)),G(\varphi^{-1}(t)))\}$ 是关于点 $(\frac{1}{2},\frac{1}{2})$ 对称的。假设曲线 $\{(1-F(\varphi^{-1}(t)),1-G(\varphi^{-1}(t)))\}$ 下的面积是 S_1 以及曲线 $\{(F(\varphi^{-1}(t)),G(\varphi^{-1}(t)))\}$ 下的面积是 S_2。通过对称性，$S_1+S_2=1$ 成立。

根据以上结论，可以用

$$\begin{aligned} S_2 &= \int_0^1 G(\varphi^{-1}(t))dF(\varphi^{-1}(t)) \\ &= \int_{-\infty}^{+\infty} G(x)dF(x) \end{aligned} \tag{3.17}$$

计算出 S_2。因此，S_2 与函数 $\varphi(x)$ 的具体形式无关，同样的，$S_1=1-S_2$ 也与 $\varphi(x)$ 的具体形式无关。此外，由于 t 的范围是 $(0,1)$，因此 $\varphi^{-1}(t)$ 的范围是 $\mathbb{R}$。于是有

$$\begin{aligned} &\{(1-F(\varphi^{-1}(t)),1-G(\varphi^{-1}(t))) : t\in(0,1)\} \\ &=\{(1-F(x),1-G(x)) : x\in\mathbb{R}\}, \end{aligned} \tag{3.18}$$

其同样与函数 $\varphi(x)$ 的具体形式无关。当 $F(x)$ 和 $G(x)$ 是离散的，集合 $\{(1-F(\varphi^{-1}(t)),1-G(\varphi^{-1}(t))) : t\in(0,1)\}$ 也是离散的，但仍与 $\varphi(x)$ 无关。因此，可以得出结论，$\varphi(x)$ 无法更改 ROC 曲线和 AUC 度量。 □

类似于定理1的证明，也可以简单地证明引理1 的第一步，即使用 $\|\tilde{\boldsymbol{w}}\|_1=1$ 进行线性融合会限制 MAE 结果。从定理2，可以得出引理1的第二步，即一个单调递增的映射，无法改变 ROC 曲线和 AUC 值。因此，证明得出，传统的显著性检测方法使用 $\|\boldsymbol{w}\|_1\neq 1$ 对侧输出进行线性融合的提升有限。结合定理1，可以得出结论，对侧输出进行线性融合的提升效果是有限的。

除了理论上的证明，本文还进行了实验来比较显著性物体检测中的线性融合与非线性融合。为此，本文使用所提出的非线性侧输出特征融合（在第3.3.3.2节

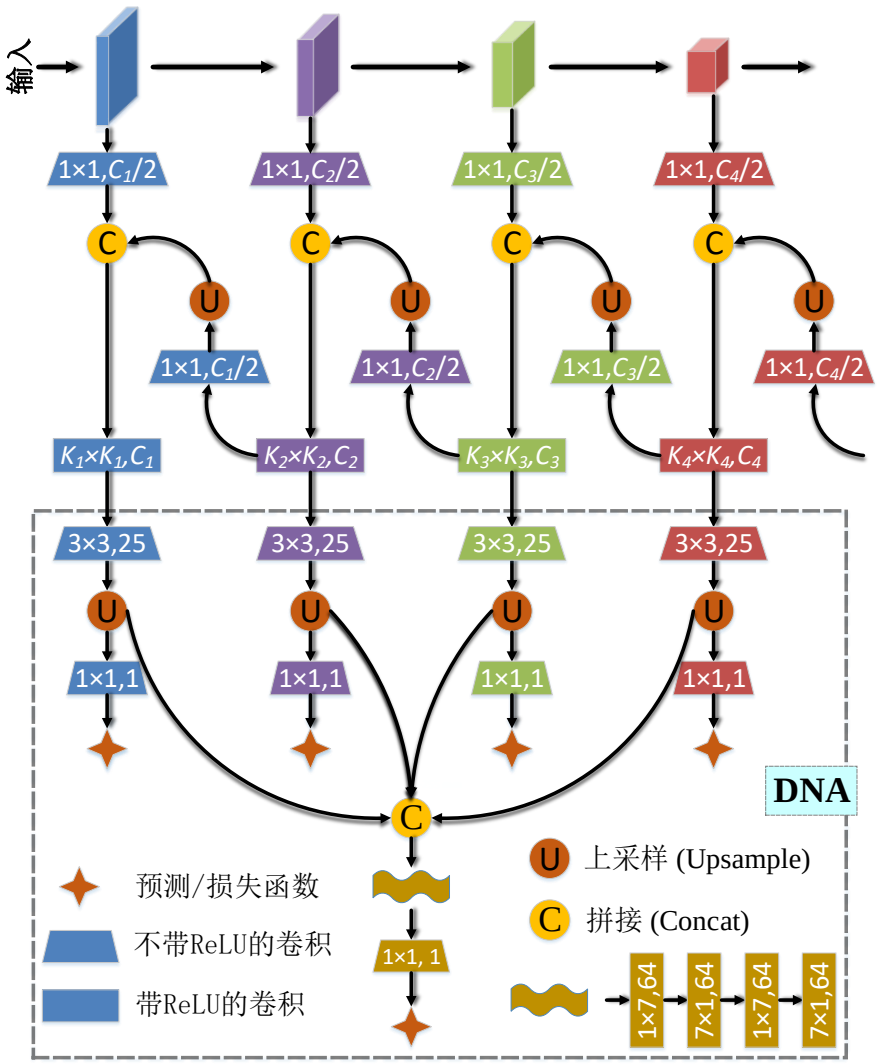


图 3.16 网络架构示意图。图中仅画出前四个网络阶段，而其他两个可以用相同的方式构造。虚线框即为所提出的 DNA 模块。参数 $K_i \times K_i$ 和 C_i 将在正文中进行介绍。

中）进行非线性回归，来评测两个著名的模型，即 HED[23] 和 DSS[76]，以及所提出的 DNA 模型。评测结果如表3.9中所示，可以看到，从线性回归到非线性回归结果有着显著提升。基于此，本文旨在设计一个简单的具有非线性侧输出融合的网络，来实现有效的显著性物体检测。

3.3.3 方法

本节将详细说明所提出的用于显著性物体检测的框架。首先，在第3.3.3.1节中介绍了所设计的基础网络。然后，在第3.3.3.2节中介绍所提出的深监督非线性融合。整个网络体系结构如图3.16所示。

3.3.3.1 基础网络

骨干网络. 与之前的研究[29, 70, 71] 类似，本节也使用全卷积网络进行显著性物体检测。具体来说，本节使用 VGG16 网络[3] 作为骨干网络，且为了实现“图像-图像”的转换，去掉了其最后的全连接层。物体检测通常需要全局信息来定位显著性物体的大致位置[27]，所以需要扩大网络的感受野。为此，正如之前的研究[76]一样，本文保留 VGG16 最后的池化层，并添加两个卷积层来代替最后的两个全连接层。其中，第一个卷积层的通道数是 $C_6^{(1)} = 192$，卷积核大小为 3×3，另一个卷积层通道数为 $C_6^{(2)} = 128$，卷积核大小为 7×7。这里，由于较大的卷积核（即 7×7）会产生更多的参数，因此先用 3×3 的卷积层来减少特征通道。

骨干网络中有五个池化层，它们将卷积层分为六个卷积块，从下到上分别表示为 $\{\mathcal{S}^1, \mathcal{S}^2, \mathcal{S}^3, \mathcal{S}^4, \mathcal{S}^5, \mathcal{S}^6\}$，将 $\mathcal{S}^6$ 作为高层的阀门来控制在网络中所传递的上下文信息。每个卷积块中的特征图的分辨率是前一个卷积块的一半。与之前研究相同[23, 76]，每个卷积块的侧输出是从其最后一个卷积层接出来的。

编码-解码网络. 如图3.16所示，在骨干网络的基础上，本文设计了一个编码-解码网络。具体来说，首先在每一个卷积块 $\mathcal{S}^6$ 和 $\mathcal{S}^5$ 后连接一个 1×1 卷积层来调整通道数（如表3.10所示）。然后，将从 $\mathcal{S}^6$ 得到的特征图上采样 2 倍。将上采样后的特征图和来自 $\mathcal{S}^5$ 的特征图进行拼接。为了融合拼接后的特征图，使用两个连续的卷积层来生成解码器侧端 $\widetilde{\mathcal{S}}^5$。解码器其他侧端 $\{\widetilde{\mathcal{S}}^4, \widetilde{\mathcal{S}}^3, \widetilde{\mathcal{S}}^2, \widetilde{\mathcal{S}}^1\}$ 均可以使用相同的方式获得。为清楚起见，可以将上述过程表示为：

$$
\begin{aligned}
\widetilde{\mathcal{S}}^i &= \varphi(\mathrm{Concat}(\phi_1(\mathcal{S}^i), \phi_2(\widetilde{\mathcal{S}}^{i+1}))), \\
\phi_1(\cdot) &= \mathrm{Conv}(\cdot), \\
\phi_2(\cdot) &= \mathrm{Upsample}(\mathrm{Conv}(\cdot)), \\
\varphi(\cdot) &= \mathrm{ReLU}(\mathrm{Conv}(\cdot)), \\
\forall i &\in \{1,2,3,4,5\}
\end{aligned}
\tag{3.19}
$$

注意，由于 $\mathcal{S}^6$ 是编码路径中的最后一个块，也是解码路径中的第一个块，因此有 $\widetilde{\mathcal{S}}^6 = \mathcal{S}^6$。通过这种方式，所提出的编码-解码网络可以把高层上下文信息传递给低层，因此较低层可以强调突出图像中显著性物体的细节。这里，解码器侧端 $\widetilde{\mathcal{S}}^i$ 的两个连续的卷积层（$\varphi(\cdot)$）的卷积核大小均为 $K_i \times K_i$，输出通道为 C_i。实验部分将详细讨论这些参数设置。

表 3.10 网络设置

侧端	C_i	$K_i \times K_i$	分辨率
第一侧端	64	3×3	1
第二侧端	128	3×3	1/2
第三侧端	128	5×5	1/4
第四侧端	128	5×5	1/8
第五侧端	128	5×5	1/16
第六侧端	-	-	1/32

3.3.3.2 深监督非线性融合

与以前的研究[29, 73, 74, 79]对多个侧输出预测使用线性融合不同，本节提出以非线性的方式融合侧输出特征。所提出的 DNA 模块如图3.16的虚线框内所示。具体地说，首先使用一个 3×3 卷积来为每个 $\widetilde{S}^i$ 调整通道数量。然后，将特征图上采样到与原图像相同大小，以此来生成侧输出特征，而侧输出特征可以使用一个简单的 1×1 卷积来预测显著性图。在训练阶段，对这些预测图进行深监督。

本文将所有侧输出特征都拼接起来，以构成包含丰富的多尺度和多层次信息的混合特征。非线性融合的关键思想之一是使用非对称卷积将标准二维卷积分解为两个一维卷积，即将一个 $n \times n$ 卷积分解为两个连续卷积，其卷积核大小分别为 $1 \times n$ 和 $n \times 1$。这里使用不对称卷积有两个原因。一方面，在实验中，作者发现由于混合特征图具有较大的分辨率（即和原图相同的分辨率），因此在 DNA 模块中使用大卷积核可以提高性能。另一方面，对于分辨率较大的特征图而言，大卷积核的卷积是非常耗时的。根据以上分析，本文将非对称卷积的卷积核设置为 $n = 7$，而不是小的卷积核尺寸。而将卷积核设置的更大后，虽然会使准确度得到很微量的提升，但却导致计算负荷增加很多。第3.3.4.3节中尝试使用了不同的 n 值以及不对称/标准卷积，而结果验证了所选择的参数设置的有效性。DNA 使用了两组非对称卷积，每组由一个 1×7 和一个 7×1 卷积组成。当输入图像为 300×300 时，这些非对称卷积的 FLOP 数量（Multiply-Adds）为 13.8G，而如果使用标准的二维 7×7 卷积，则 FLOP 数量为 60.4G。最后，在非对称卷积后连接一个 1×1 卷积来预测最终的输出的显著性图。

训练时，本文使用类别平衡的交叉熵损失函数[23]来监督侧输出预测和最终的融合预测。由于 DNA 模块中的卷积层之后均连接非线性激活函数（即 ReLU），因此多尺度侧输出特征的融合是非线性的。尽管非线性函数的选择性有很多，

例如 ReLU，PReLU 和 LeakyReLU 等，但本文仅使用最常见的 ReLU 函数来证明非线性侧输出融合的有效性。传统的线性侧输出预测融合只能线性地组合多尺度预测，而所提出的非线性侧输出特征融合可以利用互补的多尺度特征来进行最终预测，因此也更加有效。相对于以前的方法，当使用第3.3.3.1节中所描述的简单编码-解码网络时，DNA 即可实现比以前的方法更好的检测效果。值得注意的是，以前的方法[29, 73, 74, 92] 通常设计各种网络体系结构、模块和操作来提高性能，但是本文所提出的 DNA 仅将经过简单修改的 U-Net 作为基础网络。

3.3.4 实验

3.3.4.1 实验设置

实现细节. 关于 K_i 和 C_i 的详细设置详见表3.10。由于高层使用较大的卷积核有助于提高准确性，因此，当 $i=1,2$ 时，$K_i \times K_i$ 等于 3×3；当 $i=3,4,5$ 时，$K_i \times K_i$ 等于 5×5。对于 $i=1,\cdots,5$，C_i 的值分别为 64、128、128、128 和 128。在测试阶段，由于没有使用侧输出预测结果，因此本文删除了这些侧输出预测模块。而在训练阶段，深监督可以帮助训练并提高最终显著性预测的准确性，因此将其保留（将在第3.3.4.3节中进行证实）。

本文使用 Caffe 框架[181] 来实现所提出的网络，用 ImageNet[5] 预训练的模型初始化 VGG16[3] 骨干网络。上采样操作由具有双线性插值核的反卷积层实现，该双线性插值内核在训练过程中冻结。由于反卷积层不需要训练，因此在计算参数数量时可以直接将其忽略。整个网络使用 SGD 进行优化，学习率策略为 poly，即当前学习率等于初始学习率乘以 $(1-curr\_iter/max\_iter)^{power}$。其中，超参数 *power* 和 *max_iter* 分别设置为 0.9 和 20000，因此总共需要训练 20000 次迭代。初始学习率设置为 1e-7。动量和权重衰减分别被设为经典的 0.9 和 0.0005[1, 3]。本文的所有实验均在一块 TITAN Xp GPU 上实现。

数据集. 本文在六个最常用的数据集上评测了所提出的方法，包括 DUTS[216]、ECSSD[217]、SOD[218]、HKU-IS[68]、THUR15K[165] 和 DUT-O (即 DUT-OMRON)[66]。这六个数据集分别由 15572、1000、300、4447、6232 和 5168 张复杂的自然图像组成，并均带有相应的标记好的像素级真值图。其中，DUTS 数据集[216] 是由在非常复杂的场景中的 10553 张训练图像和 5019 张测试图像组成。为了公平比较，本文也像之前的研究一样[29, 72, 75]，将 DUTS 训练集用于模型训练，将 DUTS 测试集（DUTS-TE）和其他数据集用于测试。

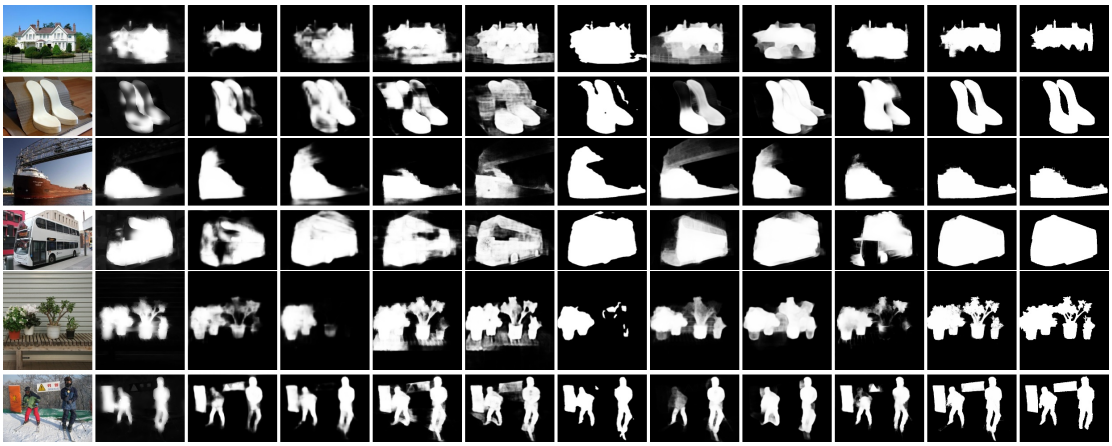


图 3.17 DNA 与最新的显著性检测模型的定性比较结果

评测指标. 本文使用最大 F_β 值（F_β-measure）和平均绝对误差（Mean Absolute Error, MAE）来评测各种模型。给定预测出的具有连续概率值的显著性图，可以通过选择一个阈值将其转换为二值图并计算相应的准确率/召回率（Precision/Recall）。通过取整个数据集所有图像的准确率/召回率的平均值，可以获得许多平均准确率/召回率对。而 F_β-measure 是一个总性能指标：

$$F_\beta = \frac{(1+\beta^2) \times \text{Precision} \times \text{Recall}}{\beta^2 \times \text{Precision} + \text{Recall}}, \tag{3.20}$$

其中，β^2 通常设置为 0.3 来强调精度。根据之前的研究[29, 70, 74, 76, 78, 86, 88]，本文计算不同阈值下的 F_β-measure 的最大值。将给定的显著性图 S 和相应的真值图 G 归一化为 [0，1] 后，MAE 可计算为

$$\text{MAE} = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} |S(i,j) - G(i,j)|, \tag{3.21}$$

其中，H 和 W 分别表示高度和宽度，$S(i,j)$ 表示在位置 (i,j) 的显著性值，$G(i,j)$ 的定义与 $S(i,j)$ 相同。

3.3.4.2 性能比较

本节将所提出的显著性物体检测器与最近的 15 种具有竞争力的显著性物体检测模型进行了比较，这 15 种模型分别为 MDF[68]、LEGS[67]、DCL[77]、DHS[79]、ELD[69]、RFCN[85]、NLDF[78]、DSS[76]、SRM[75]、Amulet[74]、UCF[86]、BRN[72]、PiCA[29]、C2S[88] 和 RAS[70]。其中，由于 MDF[68] 使用了 HKU-IS 数据集[68] 中

表 3.11 所提出的 DNA 与 15 个显著性物体检测模型在六个数据集上关于 F_β-measure 和 MAE 两个指标的检测结果比较。分别展示了以 VGG16[3] 和以 ResNet-50[4] 作为骨干网络的检测结果。每一列中，检测结果最好的前三个模型分别用 红色、绿色和蓝色突出显示。而对于基于 ResNet-50 的方法，仅突出显示达到最好性能的模型。

方法	DUTS-TE		ECSSD		HKU-IS		DUT-O		SOD		THUR15K	
	F_β	MAE	F_β	MAE	F_β	MAE	F_β	MAE	F_β	MAE	F_β	MAE
VGG16[3] 骨干网络												
MDF[68]	0.707	0.114	0.807	0.138	-	-	0.680	0.115	0.764	0.182	0.669	0.128
LEGS[67]	0.652	0.137	0.830	0.118	0.766	0.119	0.668	0.134	0.733	0.194	0.663	0.126
DCL[77]	0.785	0.082	0.895	0.080	0.892	0.063	0.733	0.095	0.831	0.131	0.747	0.096
DHS[79]	0.807	0.066	0.903	0.062	0.889	0.053	-	-	0.822	0.128	0.752	0.082
ELD[69]	0.727	0.092	0.866	0.081	0.837	0.074	0.700	0.092	0.758	0.154	0.726	0.095
RFCN[85]	0.782	0.089	0.896	0.097	0.892	0.080	0.738	0.095	0.802	0.161	0.754	0.100
NLDF[78]	0.806	0.065	0.902	0.066	0.902	0.048	0.753	0.080	0.837	0.123	0.762	0.080
DSS[76]	0.827	**0.056**	0.915	**0.056**	**0.913**	**0.041**	**0.774**	**0.066**	**0.842**	**0.122**	0.770	**0.074**
Amulet[74]	0.778	0.085	0.913	0.061	0.897	0.051	0.743	0.098	0.795	0.144	0.755	0.094
UCF[86]	0.772	0.112	0.901	0.071	0.888	0.062	0.730	0.120	0.805	0.148	0.758	0.112
PiCA[29]	**0.837**	**0.054**	**0.923**	**0.049**	**0.916**	**0.042**	0.766	0.068	0.836	**0.102**	**0.783**	0.083
C2S[88]	0.811	0.062	0.907	0.057	0.898	0.046	0.759	0.072	0.819	**0.122**	**0.775**	0.083
RAS[70]	**0.831**	0.059	**0.916**	0.058	**0.913**	0.045	**0.785**	**0.063**	**0.847**	0.123	0.772	**0.075**
DNA	**0.865**	**0.044**	**0.935**	**0.041**	**0.930**	**0.031**	**0.799**	**0.056**	**0.853**	**0.107**	**0.793**	**0.069**
ResNet-50[4] 骨干网络												
SRM[75]	0.826	0.059	0.914	0.056	0.906	0.046	0.769	0.069	0.840	0.126	0.778	0.077
BRN[72]	0.827	0.050	0.919	0.043	0.910	0.036	0.774	0.062	0.843	**0.103**	0.769	0.076
PiCA[29]	0.853	0.050	0.929	0.049	0.917	0.043	0.789	0.065	0.852	**0.103**	0.788	0.081
DNA	**0.873**	**0.040**	**0.938**	**0.040**	**0.934**	**0.029**	**0.805**	**0.056**	**0.855**	0.110	**0.796**	**0.068**

的部分数据进行训练，所以本文没有报告 MDF 在 HKU-IS 数据集上的结果。同样地，本文没有报告 DHS[79] 在 DUT-O 数据集[66] 上的结果。由于 SRM[75] 和 BRN[72] 是基于 ResNet-50[4] 骨干网络构建的，为了公平比较，本文还评测了基于 ResNet-50 版本 DNA 的和 PiCA[29] 模型。所有以前的方法都使用其公开代码和作者公布的预训练模型以及默认设置进行测试。

F_β-measure 和 MAE. 表3.11总结了 F_β-measure 和 MAE 在六个数据集上的评测结果。在大多数情况下，DNA 的性能都优于其他检测模型，因此可以证明其有效性。使用 VGG16[3] 作为骨干网络时，在 DUTS-TE、ECSSD、HKU-IS、DUT-O、SOD 和 THUR15K 六个数据集上，DNA 的 F_β-measure 比次优方法分别高 2.8%、1.2%、1.4%、1.4%、0.6% 和 1.0%。关于 MAE 指标，除了在 SOD 数据集上，DNA 比 PiCA[29] 性能稍差以外，DNA 也都达到最好结果。总体而言，PiCA[29] 是除 DNA 以外的最优模型。当使用 ResNet-50 作为骨干网络时，DNA 仍然比之前的

表 3.12 消融实验。U-Net 是指使用 VGG16 作为骨干网络的标准 U-Net[84]。如果删除 DNA 模块和深监督，那么所提出的网络（图3.16中）就变为一个编码-解码网络（Encoder-Decoder, ED）。如果进一步将编码-解码网络高层的所有卷积替换为 3×3 卷积，就得到 ED w/ K3。而 ED w/ lin 是将图3.16中的 DNA 模块替换为 HED[23] 中的传统线性融合。DS 表示深监督（Deep Supervision）。

方法	DUTS-TE		ECSSD		HKU-IS		DUT-O		SOD		THUR15K	
	F_β	MAE	F_β	MAE	F_β	MAE	F_β	MAE	F_β	MAE	F_β	MAE
U-Net	0.793	0.080	0.890	0.065	0.894	0.051	0.723	0.101	0.811	0.115	0.758	0.099
ED w/ K3	0.766	0.101	0.869	0.081	0.876	0.064	0.687	0.129	0.778	0.131	0.736	0.112
ED	0.831	0.053	0.911	0.052	0.916	0.037	0.754	0.073	0.830	0.117	0.780	0.077
ED w/ lin	0.844	0.048	0.921	0.050	0.917	0.034	0.765	0.066	0.839	0.120	0.785	0.071
DNA w/o DS	0.867	0.042	0.932	0.041	0.927	0.032	0.788	0.059	0.860	0.103	0.794	0.068
DNA	0.865	0.044	0.935	0.041	0.930	0.031	0.799	0.056	0.853	0.107	0.793	0.069

显著性物体检测模型的性能更好。这说明 DNA 对不同的网络体系结构都非常鲁棒。因此，建议未来的显著性物体检测模型均使用非线性侧输出融合来代替传统的线性融合。图3.17展示了一些例子，以进行 DNA 与其他方法之间的定性比较。可以看到，DNA 在各种不同的场景下均能取得最佳的性能。

参数数量和运行时间. DNA 具有较少的参数，具体而言，VGG16 版本 DNA 的参数约 20M，ResNet-50 版本 DNA 参数约 29M。并且，DNA 的运行速度也比其他方法快。对于 VGG16 版本，运行速度达到 25fps，ResNet-50 版本也可达到 12.8fps。

3.3.4.3 消融实验

非线性融合和线性融合. 为了证明非线性融合的有效性，通过用传统的线性侧输出预测融合[23] 替换所提出的网络中的 DNA 模块来获得一个新的具有深监督的编码-解码网络，即 ED w/ lin（Encoder-Decoder with linear aggregation）。结果如表3.12所示，可以清楚地看到，就 F_β-measure 和 MAE 而言，非线性融合在各个数据集上的效果都明显好于线性融合。

所提出的编码-解码架构和标准的 U-Net. 如果移除 DNA 模块和深监督，那么所提出的编码-解码（Encoder-Decoder, ED）架构就变成了简单修改版本的 U-Net[84]。首先，将位于高层的所有卷积的卷积核大小，即 $K_3 \times K_3$、$K_4 \times K_4$ 和 $K_5 \times K_5$，全部改为 3×3。如表3.12所示，得到的模型 ED w/ K3 的性能不如标准 U-Net[84]。这可能是因为所提出的编码-解码架构具有更少的特征通道数，因

此就具有更少的参数（U-Net 有 31.06M 参数）。接下来，在高层使用默认的卷积核大小 5×5。所得到的编码-解码网络 ED 的性能优于 U-Net。这说明在高层使用较大的卷积核对于提高性能很重要。

有/无深监督的编码-解码架构. 在表3.12中，ED w/ lin 性能要优于 ED。如果移除了 DNA 里的深监督，所得到的模型 DNA w/o DS（DNA without Deep Supervision）在大多数情况下都比原本的 DNA 性能差。因此，深监督可以明显地改善显著性预测性能。

第四节 通用的图像属性知识 - 似物性采样

3.4.1 引言

生成少量物体推荐的同时覆盖图像中尽可能多的物体，可以通过减少搜索空间和误报，而对于后续高级应用（如，物体检测[106, 179]、语义实例分割[11, 139]、多标签分类[219]、视频总结[220] 和深度多实例学习[221] 等）的效率和准确性至关重要。在过去的十年中，已经提出了许多自底向上的似物性采样方法，旨在生成密集的推荐来覆盖尽可能多的物体，例如 Selective Search[30]、Edge Boxes[31] 和 MCG[26]。由于使用传统的手工提取的特征很难表示高层语义信息，因此这些自底向上的方法通常 1) 无法对生成的物体推荐进行正确排序，并且 2) 必须使用大量的物体推荐才能确保检测召回率。虽然这些现有的自底向上的算法可以通过在每张图像上生成数千个推荐来实现较高的检测召回率，但是由于存在大量的误报且计算负载繁重，这些生成的大量的物体推荐使后续的分析变得非常困难[166, 219, 221, 222]。最近，一些基于深度学习的似物性采样方法在该领域引起了很多关注，包括 RPN[106]、DeepMask[107] 和 SharpMask[108]。RPN[106] 通过从下采样的卷积特征图（1/16 尺度）中采样锚点来生成物体推荐，而 DeepMask[107] 和 SharpMask[108] 通过扫描图像小块来发现物体。这些采样策略使他们难以充分利用卷积神经网络的强大能力，图像中真实物体的数量（通常少于十个）仍然比这些方法生成的推荐的数量（通常数百个）少得多。

能否在保持高召回率的同时大幅减少物体推荐的数量呢？这对于更广泛的应用至关重要，例如当从大量未标记/弱标记的数据中挖掘知识[219, 221] 时，大量误报的存在不仅对计算效率甚至是系统稳定，都构成了重大挑战。本节专注于减少推荐数量的同时获得较高的检测召回率。可以观察到，当候选框的数量足

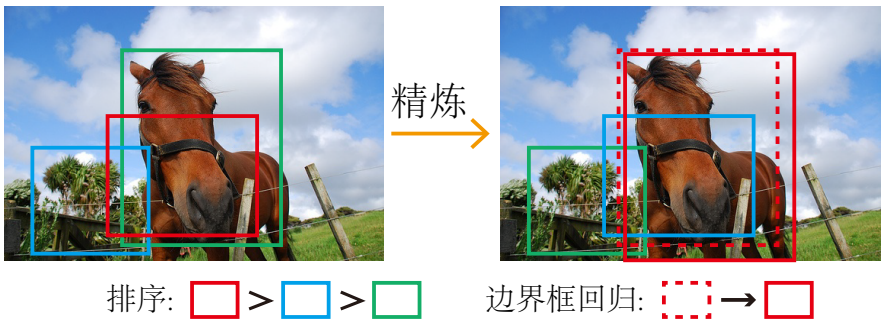


图 3.18 物体推荐精炼的概述。左图显示了初始的物体推荐，右图显示了精炼后的结果。首先通过计算新的似物性分数对物体推荐重新排序，然后对每个推荐框进行回归来实现准确定位。

够多时，某些传统的似物性采样方法能够实现较高的检测召回率，这是因为与深度学习中简单的推荐采样策略[106–108] 不同，传统方法通常设计巧妙的策略来搜索物体的所有可能位置。当然，大量的候选框会在后续的应用中引起许多误报，从而影响最终的性能。但是，如果可以从大量的候选框中选择优秀的候选框，那么这将有利于一系列的计算机视觉任务。最近，已经有几种算法被提出来改善物体推荐，包括 DeepBox[109] 和 MTSE[110]。DeepBox 建立了一个神经网络来重新计算初始框的似物性分数，然后对其进行重新排序。MTSE 试图使用超像素优化每个框，具体是通过使每个框紧密覆盖一些内部的超像素。然而，DeepBox 的物体推荐质量比 RPN[106] 还差，因此无法减少推荐数量。此外，MTSE 的性能取决于超像素的质量，并且 MTSE 中的图像过分割会导致计算负荷的显著增加。

为了结合传统似物性采样方法的优势和卷积神经网络强大的表征能力[106–108]，本节提出了一种新的方法来在神经网络的单次前向传播中，对现有推荐框进行重新排序和推荐框回归。所提出的方法的概述如图3.18所示。本节对候选框的精炼包括两个步骤：重新排序和推荐框回归。重新排序这一步尝试根据推荐框覆盖完整物体的紧密程度对推荐框进行重新排序。推荐框回归这一步则是尝试微调框的形状和位置，使其更紧密地覆盖真实物体。为了实现这个目标，精炼网络旨在学习新的似物性分数并同时进行框回归。为了简洁起见，在本文的其余部分中，将所提出的方法称为 RefinedBox。RefinedBox 可以通过与高级应用联合训练来共享卷积特征。为了展示一个联合训练的示例，本节通过在基础网络（例如 VGG16[3]）的最后一个卷积层之后连接 RefinedBox 模块，来将 RefinedBox 和著名检测框架 Fast R-CNN[179] 整合为一个统一的框架，然后引入一种交替微调策略进行训练。最终，RefinedBox 可以与后续的物体检测网络

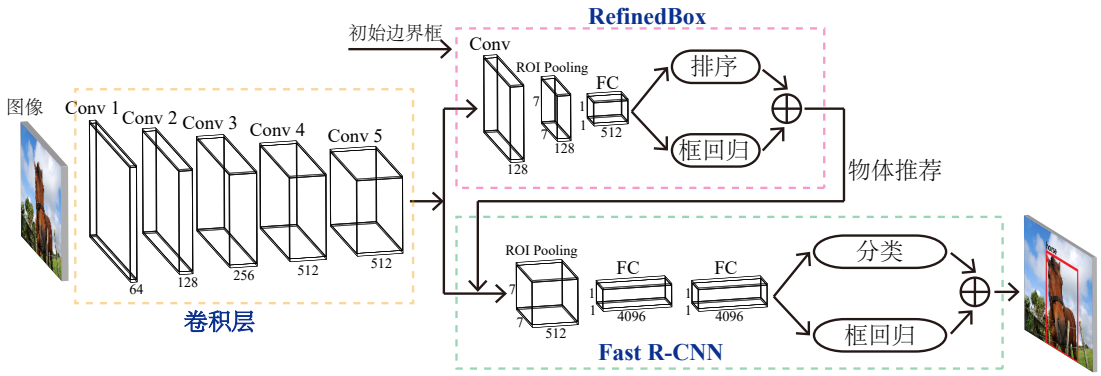


图 3.19 所提出的网络架构概述。这里以将 RefinedBox 与物体检测进行联合训练为例。所提出的网络将自然图像和由其他似物性采样方法（例如 Edge Boxes）生成的相应初始框作为输入。设计 RefinedBox 分支来精炼初始框，然后将精炼的推荐框输入到 Fast R-CNN 分支中进行分类。值得注意的是，推荐框的精炼和随后的物体检测可以共享卷积特征。

共享基本卷积层，从而使物体推荐的精炼过程非常高效。

用各种传统方法生成的推荐框作为输入，对于在 VOC2007 数据集[223] 上的似物性采样，RefinedBox 在重叠率（Intersection-over-Union, IoU）为 0.5 和 0.7 下的检测召回率分别为 80.4 % 和 67.9%，且每张图像仅使用 10 个精炼后的推荐框。当仅使用 10 个推荐框进行物体检测时，RefinedBox 的平均精度（mean Average Precision, mAP）为 65.4%，而 RPN[106] 的 mAP 是 54.1%。实验表明，所提出的 RefinedBox 方法可以在物体推荐数量有限的情况下生成高质量的物体推荐。

3.4.2 方法

3.4.2.1 网络架构

RefinedBox 以其他似物性采样方法生成的物体推荐作为输入，然后尝试对其进行精炼。精炼包括两步：重新排序和推荐框回归。为了对现有的物体推荐框重新排序，RefinedBox 使用深度神经网络中的语义信息重新计算每个推荐框的似物性分数。为了进行推荐框回归，RefinedBox 设计网络来学习每个物体推荐框的中心坐标、宽度和高度的回归。

VGG16[3] 是深度学习领域广泛使用的骨干网络体系结构。它由 13 个卷积层和 3 个全连接层组成。受到之前研究[106, 179] 的启发，本节基于 VGG16 构建网络来阐述所提出的精炼方法 RefinedBox。图3.19中显示了 RefinedBox 的网络体系结构。所提出的网络采用自然图像和相应的初始物体推荐框作为输入，初始框是由其他似物性采样方法生成的。本节以一些著名的似物性采样方法为例，用

RefinedBox 来精炼它们生成的物体推荐，如 Edge Boxes[31]、MCG[26]、Selective Search[30] 和 RPN[106]。输入图像首先经过一些卷积层的前馈，例如 VGG16 中的 13 个卷积层。为了减少推荐框精炼的时间消耗，本节设计了一个计算轻量级的神经网络。具体而言，RefinedBox 首先在第 13 个卷积层之后连接一个卷积核大小为 3×3 的卷积层来将通道数从 512 减少到 128。然后，连接一个 ROI 池化层（ROI Pooling[179]），将每个初始框区域下采样为固定的特征图大小，即 7×7。ROI 池化层将输入特征图划分为宽度和高度相同的网格，并在每个网格中进行最大池化操作。然后，将其连接到一个只有 512 个输出神经元的全连接层。在所添加的卷积层和全连接层之后分别连接 ReLU 层。最后，使用排序和推荐框回归两个分支来分别重新计算似物性分数并获得每个初始框的位置偏移。排序分支是一个具有两个输出神经元的全连接层，两个输出神经元分别表示该推荐框是否是一个物体的概率。推荐框回归分支预测推荐框的回归值，这将在下面进行描述。

在 RefinedBox 的训练中，为每个初始的物体推荐框分配一个是否为物体的二值的类标签。损失函数可以写成

$$L_{obj}(p,u) = -[1_{\{u=1\}} \log\ p_1 + 1_{\{u\neq 1\}} \log\ p_0], \tag{3.22}$$

其中，p 是在全连接层的两个输出上求 softmax 计算得到的，u 是此框的标签（1 或 0）。推荐框回归层是一个全连接层，旨在学习坐标偏移。本节按以下方式对四个坐标进行参数化：

$$\begin{aligned} t_x &= (x - x_{in})/w_{in}, \quad & t_y &= (y - y_{in})/h_{in}, \\ t_w &= log(w/w_{in}), & t_h &= log(h/h_{in}), \\ v_x &= (x^* - x_{in})/w_{in}, & v_y &= (y^* - y_{in})/h_{in}, \\ v_w &= log(w^*/w_{in}), & v_h &= log(h^*/h_{in}), \end{aligned} \tag{3.23}$$

其中，x、y、w 和 h 分别代表推荐框的中心的坐标、宽度和高度。变量 x、x_{in} 和 x^* 分别是对于预测框、输入框和真值框来说的；y、w 和 h 也使用类似的定义。变量 v 是回归目标，t 是预测的元组。推荐框回归的损失函数定义为：

$$\begin{aligned} L_{reg} &= \sum_{i\in\{x,y,w,h\}} \text{smooth}_{L_1}(t_i - v_i), \\ \text{smooth}_{L_1}(x) &= \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise,} \end{cases} \end{aligned} \tag{3.24}$$

算法 3 RefinedBox 的交替训练过程

Input: 所提出的神经网络的骨干网络 W_{VGG}、RefinedBox 模块 W_{RB}、物体检测模块 W_{Det}；初始的物体推荐 B_{in}；在 ImageNet 上预训练的骨干网络 W_{VGG}^{pre}
Output: 训练好的 W_{VGG}、W_{RB} 和 W_{Det}
Step 1: $W_{VGG} \leftarrow W_{VGG}^{pre}$; $W_{RB} \leftarrow random()$
Step 2: $W_{VGG}, W_{RB} \leftarrow finetune(W_{VGG}, W_{RB}; B_{in})$
Step 3: $B' \leftarrow rerank(B_{in}; W_{VGG}, W_{RB})$
Step 4: $W_{VGG} \leftarrow W_{VGG}^{pre}$; $W_{Det} \leftarrow random()$
Step 5: $W_{VGG}, W_{Det} \leftarrow finetune(W_{VGG}, W_{Det}; B')$
Step 6: $W_{RB} \leftarrow random()$
Step 7: $W_{RB} \leftarrow finetune(W_{RB}; W_{VGG}, B_{in})$
Step 8: $B' \leftarrow rerank(B_{in}; W_{VGG}, W_{RB})$
Step 9: $W_{Det} \leftarrow random()$
Step 10: $W_{Det} \leftarrow finetune(W_{Det}; W_{VGG}, B')$

其中，$\text{smooth}_{L_1}(x)$ 是一个著名的回归损失函数[179]。因此，联合损失函数可以写成

$$L(p,u,t,v) = L_{obj}(p,u) + \lambda \cdot 1_{\{u=1\}} L_{reg}(t,v), \tag{3.25}$$

其中，参数 λ 是一个平衡参数，在本节中将其设置为 1。

3.4.2.2 与物体检测联合训练

到目前为止，已经描述了如何训练物体推荐的精炼网络。由于所提出的网络是轻量级的，因此它有与高级应用共享卷积特征的潜力。这里以物体检测为例，阐述 RefinedBox 及其后续应用的联合训练过程。为了测试 RefineBox 生成少量且高质量物体推荐的能力，每张图像仅使用 RefinedBox 生成的前 10 个物体推荐进行物体检测。

如图3.19所示，在卷积层之后连接一个著名的检测框架 Fast R-CNN[179]，将其作为与 RefinedBox 并行的一个分支。将由 RefinedBox 分支生成的精炼后的物体推荐输入到 Fast R-CNN 中。为了使 RefinedBox 和 Fast R-CNN 共享相同的卷积特征，本节使用了一个交替进行的微调过程，如算法3所示。物体检测的训练取决于先前步骤所生成的重新排序的物体推荐。在步骤 6 之前，物体推荐和物体检测的网络是分别训练的。然后，固定骨干网络，并对用于 RefinedBox 和物体检测的特定层进行微调。经过交替训练，两个网络形成一个统一的网络。

对于其他高级应用，联合训练也以类似的方式进行。换句话说，通过将 W_{Det} 替换为其他任务的模块，算法3也适用于其他任务。算法3的关键是，通过在高

级任务和 RefinedBox 模块之间进行交替训练，使高层任务和 RefinedBox 共享相同的骨干网络，因此输入图像只需要通过骨干网络一次。

浮点运算（Floating-Point Operations, FLOPs）的数量通常用于衡量网络的计算消耗，其中浮点运算表示乘加运算（Multiply-Add Operations）。对于每个物体推荐框，Fast R-CNN 分支的全连接层有 120.0M（million）个 FLOPs，而 RefinedBox 分支的全连接层只有 3.2M 个 FLOPs。因此，RefinedBox 分支只会带来很少的额外的计算开销。

3.4.3 实验

3.4.3.1 实验设置

实现细节. 对于 RefinedBox 的训练，每个随机梯度下降（Stochastic Gradient Descent, SGD）的小批量都是从一张图像中选择 256 个推荐框作为训练样本构造而成的。在每一批中，所采样的推荐框一半为正样本，一半为负样本。重叠率（Intersection-over-Union, IoU）是指两个框的相交面积与并集面积的比率。正采样框与真值框的 IoU 重叠率至少为 0.7，而负采样框与真值的最大 IoU 重叠率在 $[0.1,0.5)$ 之间。初始学习率设置为 1e-3，并在 12 个纪元后除以 10。SGD 总共运行 16 个纪元。为了训练检测模块，每个小批量都有 256 个来自同一图像的物体推荐。与 Fast R-CNN[179] 中一样，这些物体推荐中有 25% 的推荐与真值的 IoU 重叠率至少为 0.5，它们被视为正样本。其余的负样本与真值的最大 IoU 重叠率在区间 $[0.1,0.5)$ 内。使用 RefinedBox 生成的前 1000 个推荐进行训练。对于前 12 个纪元，学习率为 1e-3，而对于另外 4 个纪元，学习率除以 10。对于测试，每张图像仅使用 RefinedBox 的前 10 个推荐。相比之下，传统的物体推荐方法（例如 Edge Boxes 和 Selective Search）通常需要数千个推荐。本节基于公开的代码[2] 实现了所提出的方法。训练和测试是在一块 GTX TITAN X GPU 上进行的。

数据集. 本节在广泛使用的物体检测数据集 PASCAL VOC2007[223] 上评测了所提出的方法以及已存在的方法。PASCAL VOC2007 数据集[223] 由 2501 张训练、2510 张验证和 4952 张测试图像组成，所有图像都带有相应的 20 个物体类别的标注。这里，使用 VOC2007 的 trainval 集（训练和验证集）来训练模型，并在 VOC2007 的测试集上进行测试。

[2]https://github.com/rbgirshick/py-faster-rcnn

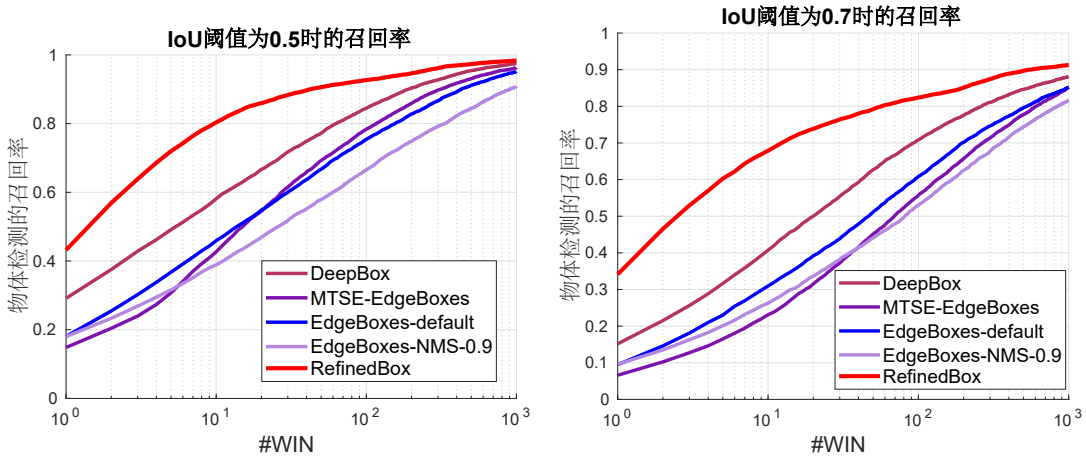


图 3.20 PASCAL VOC2007 数据集[223] 上不同精炼算法的评测。这两个子图分别显示在 IoU 阈值分别为 0.5（左）和 0.7（右）下的物体检测召回率 *vs.* 物体推荐数（#WIN）。EdgeBoxes-default 使用 Edge Boxes[31] 方法的默认参数，EdgeBoxes-NMS-0.9 将非极大值抑制（Non-Maximum Suppression, NMS）的参数改为 0.9。

对比的方法. 为了证明所提出的物体推荐的精炼方法的有效性，将本节所提出的 RefinedBox 与现有的主流的似物性采样方法进行了比较，包括基于非深度学习的方法，如 BING[104]、CSVM[103]、Edge Boxes[31]、Endres[100]、GoP[224]、LPO[101]、MCG[26]、Objectness[102]、Rahtu[99]、RandomPrim[97]、Rantalankila[98] 和 Selective Search[30]，以及最近的基于深度学习的方法，如 RPN[106]、DeepBox[109]、DeepMaskZoom[107] 和 SharpMaskZoom[108]。所使用的 DeepMaskZoom 和 SharpMaskZoom 分别是 DeepMask[107] 和 SharpMask[108] 的最好版本（详情请参考其论文）。本节首先与这些似物性采样方法进行比较。然后，对于 PASCAL VOC2007 数据集[223]，本节将这些方法生成的物体推荐输入到著名的基于区域的物体检测框架 Fast R-CNN[179] 中，以便于在物体检测中评测物体推荐的质量。本节的实验结果表明，所提出的 RefinedBox 可以为物体检测生成高质量的物体推荐，并且效率很高。

指标. 为了评测物体推荐，本节使用的指标有物体检测召回率（Detection Recall, DR）、平均最佳重叠率（Mean Average Best Overlap, MABO）和平均召回率（Average Recall, AR）。检测召回率（DR）认为当真值物体与一个物体推荐的 IoU 重叠率大于阈值时，那么就认为这个真值物体被找到了。为了计算特定类别的平均最佳重叠率（ABO），计算（属于此类的）每一真值标注与为相应图像生成的物体推荐之间的最佳 IoU 重叠率，并对该类别中的所有真值物体进行平均。

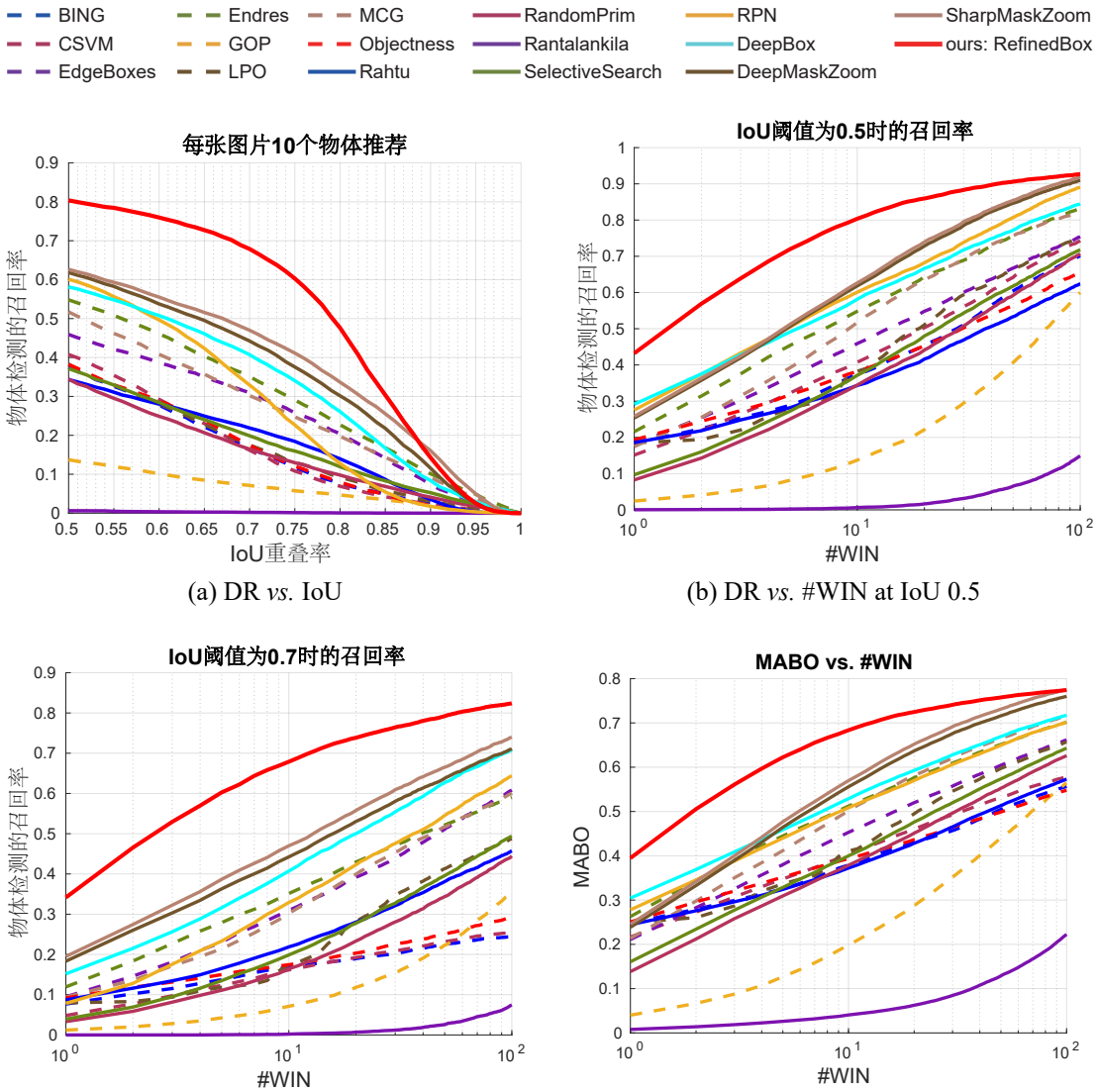
表 3.13 在 PASCAL VOC2007 测试集[223] 上关于 DR 的评测结果（%）。RefinedBox^1、RefinedBox^2、RefinedBox^3、和 RefinedBox^4 分别表示基于 Edge Boxes、MCG、Selective Search 和 RPN 的 RefinedBox。

#WIN	DR (IoU=0.5)				DR (IoU=0.7)				时间 (秒)
	10	30	50	100	10	30	50	100	
BING	37.5	51.0	60.4	70.1	16.9	20.2	22.5	24.4	**0.003**
CSVM	40.8	56.1	64.2	74.3	16.2	20.9	23.1	25.5	0.33
EdgeBoxes	45.9	60.0	66.7	75.4	31.0	43.8	51.1	60.8	0.25
Endres	54.8	68.9	75.6	83.3	35.1	47.1	52.2	59.0	19.94
GOP	13.7	29.5	40.7	60.0	0.7	15.6	22.3	35.6	0.29
LPO	38.2	59.4	66.4	75.3	17.5	34.8	41.3	48.8	0.46
MCG	51.7	69.3	75.8	82.1	30.2	45.4	51.7	60.1	17.46
Objectness	38.2	50.2	56.4	65.4	17.4	22.6	25.0	29.3	0.91
Rahtu	34.3	46.9	53.3	62.3	21.9	32.1	38.1	45.8	0.67
RandomPrim	34.4	50.7	59.2	70.7	16.4	28.1	34.4	44.5	0.12
Rantalankila	0.6	3.1	6.5	14.9	0.2	1.2	2.6	7.4	3.57
SelectiveSearch	37.1	54.3	61.8	71.8	19.9	32.7	39.6	49.4	1.60
RPN	60.1	73.8	80.7	89.0	32.9	47.6	54.5	64.4	0.10
DeepBox	58.1	71.8	77.2	84.5	40.7	55.4	62.7	70.9	0.45
DeepMaskZoom	61.8	78.5	84.7	91.0	44.2	58.1	63.8	71.1	1.20
SharpMaskZoom	62.6	79.5	85.4	91.9	47.0	60.9	66.5	74.0	0.57
RefinedBox^1	80.4	88.3	90.6	**92.7**	67.9	**76.4**	**79.2**	**82.4**	0.31
RefinedBox^2	**80.5**	87.6	88.8	89.6	68.2	75.2	76.4	77.1	17.52
RefinedBox^3	79.2	86.4	88.2	89.7	**68.6**	76.1	78.0	79.6	1.66
RefinedBox^4	79.5	**88.6**	**90.8**	92.4	65.3	75.2	77.6	79.5	0.16

MABO 被定义为所有类别的平均 ABO[30]。Hosang 等人[225] 引入了 AR，以计算一定数量的物体推荐在 IoU 阈值为 $[0.5 : 0.05 : 0.95]$ 下的平均召回率。

3.4.3.2 在 VOC2007 数据集上的似物性采样的评测

这里首先将所提出的 RefinedBox 与其他物体推荐精炼方法进行比较，包括 DeepBox[109] 和 MTSE[110]。图3.20展示了不同推荐精炼方法之间的比较结果。本节选择 Edge Boxes[31] 来生成输入到这些精炼算法的初始推荐，但是将其非极大值抑制的默认参数从 0.75 改为 0.9 来获得更多推荐框。可以发现，当 IoU 阈值为 0.5 和 0.7 时，所提出的 RefinedBox 均比其他方法实现了更高的物体检测召回率，且与其他方法之间的差距非常大。当每张图像仅使用一个推荐，RefinedBox 在 IoU 0.5 和 IoU 0.7 时的检测召回率分别为 43.2% 和 34.2%，而原始 Edge Box 的召回率分别为 29.1% 和 15.2%。此外，RefinedBox 可以与后续的物体检测共享



(a) DR *vs.* IoU

(b) DR *vs.* #WIN at IoU 0.5

(c) DR *vs.* #WIN at IoU 0.7

(d) MABO *vs.* #WIN

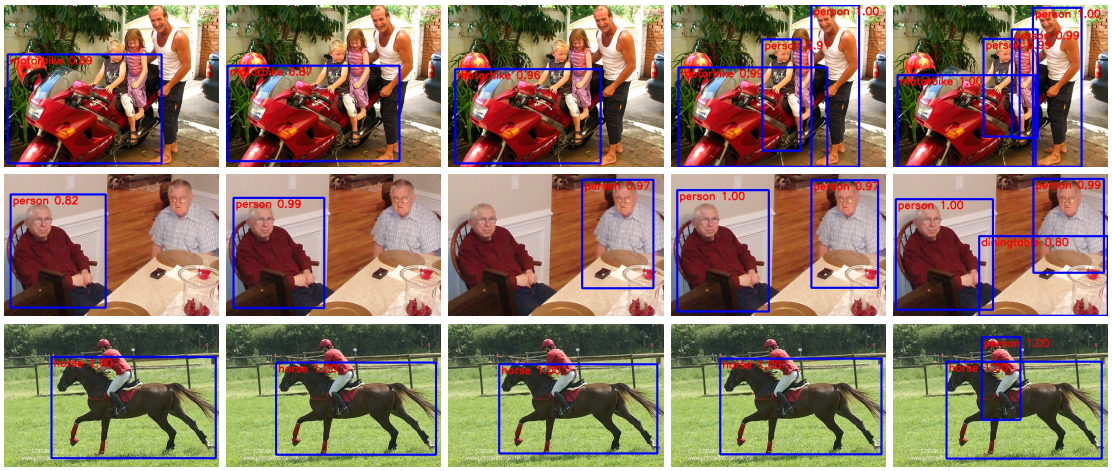
图 3.21 在 PASCAL VOC2007 测试集[223] 上的评测结果（%）。(a) 展示了物体检测召回率 *vs.* IoU 重叠率阈值，每张图像使用 10 个物体推荐。(b) 和 (c) 分别展示了在 IoU 阈值 0.5 和 0.7 下，物体检测召回率 *vs.* 物体推荐数（#WIN）。(d) 显示了 MABO *vs.* 物体推荐数，每张图片最多使用 100 个推荐。

卷积层，并且 RefinedBox 的其他层在计算上是轻量级的，因此 RefinedBox 是一个高效的检测框架。实际上，RefinedBox 和后续物体检测的总时间消耗类似于 Faster R-CNN[106]，每张图像大约需要 0.13 秒。DeepBox 建立了一个独立的网络来对推荐框进行重新排序。而 MTSE 首先对图像进行过分割，然后使用过分割结果来精炼推荐框，然而，图像过分割步骤是一个耗时的操作。因此，RefinedBox 更适合在许多应用中使用。

表 3.14 在 PASCAL VOC2007 测试集[223] 上关于 AR、MABO 和 mAP（每张图像使用 10 个物体推荐的物体检测性能）的评测结果（%）。RefinedBox^1、RefinedBox^2、RefinedBox^3 和 RefinedBox^4 分别表示基于 Edge Boxes、MCG、Selective Search 和 RPN 的 RefinedBox。

#WIN	AR				MABO				mAP
	10	30	50	100	10	30	50	100	
BING	16.5	21.3	24.6	27.9	37.9	45.7	50.5	55.7	34.4
CSVM	17.0	22.7	25.5	29.1	40.3	49.2	53.1	57.9	35.7
EdgeBoxes	26.3	36.3	41.3	48.0	45.3	55.7	60.4	66.2	39.1
Endres	31.1	40.5	44.8	50.6	51.2	60.9	65.1	70.2	42.8
GOP	6.8	14.6	20.7	31.9	19.8	35.2	44.0	56.4	13.3
LPO	17.2	31.1	36.7	43.2	41.1	54.6	59.7	65.7	34.5
MCG	27.6	40.5	45.9	52.9	50.1	62.1	66.5	71.6	41.2
Objectness	16.8	22.0	24.6	28.8	39.4	46.5	49.9	54.8	34.9
Rahtu	18.5	26.5	30.8	36.8	37.2	46.5	51.3	57.3	32.4
RandomPrim	16.1	25.8	31.3	39.6	37.9	49.6	55.3	62.6	31.9
Rantalankila	0.2	1.2	2.7	7.0	4.1	8.5	12.9	22.3	2.4
SelectiveSearch	18.6	29.8	35.5	43.6	40.0	52.0	57.4	64.3	34.1
RPN	28.4	38.1	42.7	48.9	50.8	60.6	65.0	70.1	54.1
DeepBox	33.9	44.5	49.2	54.9	52.9	62.8	66.9	71.8	50.9
DeepMaskZoom	37.1	48.5	53.2	59.1	55.6	67.6	71.6	76.0	52.7
SharpMaskZoom	39.7	51.5	56.1	62.0	57.0	69.2	73.1	77.3	53.5
RefinedBox^1	53.0	**58.7**	**60.6**	**62.4**	68.4	**74.1**	**75.8**	**77.4**	65.4
RefinedBox^2	**53.7**	58.4	59.3	59.8	**68.9**	73.8	74.7	75.3	65.2
RefinedBox^3	53.5	**58.7**	60.0	61.1	67.9	73.2	74.6	75.8	**65.5**
RefinedBox^4	49.8	56.1	57.7	59.0	66.6	72.9	74.3	75.4	65.0

接下来，如图3.21所示，将 RefinedBox 与最近的似物性采样方法进行比较。RefinedBox 依然使用 Edge Boxes 作为输入，而本节使用默认参数来对 Edge Boxes 进行评测。RefinedBox 在所有情况下都实现了最佳性能。当 IoU 为 0.7 时，对于物体检测召回率 *vs.* 物体推荐数量，RefinedBox 相对于其他方法的性能提升非常大。较高的检测召回率和较少的推荐将有利于后续的高层应用。最近，RPN 在物体检测中非常流行，但是所提出的 RefinedBox 比它准确得多。每张图像仅包含 10 个物体推荐的 RefinedBox 的物体检测召回率和每张图像使用 100 个推荐的 RPN 差不多。从 RPN 到 RefinedBox 的提升证明了 RefinedBox 的有效性。只需很少的物体推荐，RefinedBox 就可以取得比其他方法更好的性能，包括最近著名的基于深度学习的 DeepMask[107] 和 SharpMask[108]。仅使用 30 个物体推荐，当 IoU 重叠率分别为 0.5 和 0.7 时，RefinedBox 可以实现 88.3% 和 76.4% 的检测召回率。这将满足很多应用对于少量但高质量的物体推荐的要求。



Edge Boxes　　MCG　　Selective Search　　RPN　　RefinedBox

图 3.22　仅使用前 10 个物体推荐进行物体检测的定性比较。这里，RefinedBox 使用 Edge Boxes[31] 作为输入。所有图像均来自 VOC2007 测试集[223]。

为了量化这些图，在表3.13中列出了相应的数字。与各种初始输入方法相比，RefinedBox 实现了更好的性能。在使用 Edge Boxs 且 IoU 阈值为 0.5 时，当每张图像使用 10、30、50 和 100 个物体推荐时，RefinedBox 的检测召回率比次优方法（SharpMaskZoom[108]）分别高 17.8%、8.8%、5.2% 和 0.8%。在 IoU 阈值为 0.7 下，当每张图像分别使用 10、30、50 和 100 个物体推荐时，RefinedBox 的检测召回率比 SharpMaskZoom 分别高 20.9%、15.5%、12.7% 和 8.4%。本节的目标是大幅减少物体推荐的数量，而评测结果正表明 RefinedBox 已经实现了这一目标。还可以注意到，RPN[106] 比传统的基于非深度学习的方法要好得多，而这也是为什么 Faster R-CNN 可以实现比 Fast R-CNN 更好的性能。由于 RefinedBox 旨在从先前方法生成的所有推荐中选择和精炼好的推荐，因此，影响最大的因素是输入的物体推荐的上限，即当具有足够数量的物体推荐时，之前方法能够取得的最大的检测召回率，而不是在一定数量的物体推荐下的性能。在 VOC2007 数据集上，Edge Boxes 可以通过足够数量的推荐实现较高的检测召回率，这就是基于 Edge Boxes 的 RefinedBox 性能最佳的原因。每张图像的 RefinedBox 的运行时间约为 0.06 秒，与传统的方法相比，这是非常快的。表3.14中报告了各种对比方法的 AR 和 MABO，RefinedBox 再次达到了最佳性能。

3.4.3.3　在 VOC2007 数据集上的物体检测

因为物体检测是似物性采样的一个重要应用，所以可以根据在物体检测中的性能来评估不同似物性采样算法的质量。本节将上述那些方法产生的物体推

荐框输入到著名的基于区域的物体检测框架 Fast R-CNN[179] 中，并使用上述联合训练算法（算法3）来优化 RefinedBox。本节的实验使用与之前研究[111] 相同的设置。每张图像的前 1000 个物体推荐用于重新训练 Fast R-CNN 网络。所有这些方法都在 VOC2007 的 trainval 集[223] 上进行训练，并在其测试集上进行测试。需要注意的是，在测试时，每张图像仅使用前 10 个物体推荐来评估不同方法生成少量物体推荐的能力。

结果如表3.14所示。就 mAP 而言，RefinedBox（精炼后的结果）分别比原始的 Edge Boxes、MCG、Selective Search 和 RPN 高 23.63%、24.03%、31.43% 和 10.93%。与其他似物性采样方法相比，RefinedBox 也可以实现更高的检测性能。这些评测结果表明，RefinedBox 可以生成少量且高质量的物体推荐。有趣的是，RPN[106] 在物体检测方面的性能比 DeepBox[109]、DeepMask[107] 和 SharpMask[108] 稍好，而 RPN 在似物性采样方面表现较差。这可能是因为 RPN 是针对于 Faster R-CNN 框架[106] 中的物体检测精心设计的。图3.22中展示了 RefinedBox 和其他方法关于物体检测的定性比较。可以看到，RefinedBox 显著提高了其他方法的检测性能。

第五节　小结与讨论

本章设计了基于多层次多粒度深度网络的任务类别无关的图像通用属性提取方法，包括图像边缘检测、图像过分割、图像显著性检测、似物性采样等，克服了目标任务数据有限的难题。这些通用技术的模型一旦训练好，便可永久使用，而与具体应用场景无关。

第一节致力于提升通用的图像属性知识之边缘检测，提出了一种新的卷积神经网络架构 RCF，该架构充分利用了来自卷积神经网络的所有卷积层的卷积特征。所提出的 RCF 方法可以非常高效地生成高质量的边缘，RCF 是第一个在著名的 BSDS500 数据集[25] 上以实时的速度超越人类标注的边缘检测方法，这使其可以被应用于知识引导的自适应图像理解。另一方面，RCF 架构可以看作是全卷积神经网络（例如 FCN[83] 和 HED[23]）的未来的发展方向，已经被应用于很多其他的计算机视觉任务。

第二节研究了通用的图像属性知识之图像过分割，探索了基于超像素合并的过分割方式，以获得速度和精度之间的良好权衡。本章先提出了一种基于分层特征选择的过分割方法（HFS），HFS 设计了一个分层体系结构，以获得在不

同尺度层次上使用不同特征设置的好处。继而又提出了一种基于深度嵌入学习的方法（DEL），突破了过分割无法直接使用深度学习训练的瓶颈，将 HFS 中的手工设计特征替换为深度特征，以进一步提高过分割精度。所提出的 HFS 和 DEL 方法在效率和有效性之间取得了很好的权衡，作为通用的图像属性知识，这使得它们有潜力被应用于在其他高层视觉分析任务。

第三节研究了通用的图像属性知识之图像显著性检测，图像显著性可模拟人类视觉的聚焦过程，关注图像中重要的部分。因此，它在知识引导的自适应图像理解中起着重要作用。之前的深监督显著性物体检测网络使用线性侧输出预测融合，本章在理论上和实验上均证明了线性侧输出融合是次优的且不如非线性融合。基于此，本章提出了以非线性方式融合多层次的侧输出特征的 DNA 模块。与 15 个最近的显著性检测模型相比时，将 DNA 应用于经简单修改过的 U-Net，DNA 便可以在各种指标下达到最好的效果。所提出的网络还具有更少的参数和更快的运行速度，这更加证明了其有效性。在未来的研究中，作者计划将 DNA 用于进一步改进显著性物体检测，并将其用于需要多尺度和多层次信息的其他计算机视觉任务中。

第四节研究了通用的图像属性知识之似物性采样，提出了一种使用重新排序和框回归的物体推荐精炼方法 RefinedBox。因为添加的层被设计为计算轻量级的，因此 RefinedBox 是非常高效的。实验表明，RefinedBox 可以显著地减少以前算法生成的物体推荐的数量。由于所提出的精炼网络可以很简单的被优化，因此可以将其与后续应用一起进行联合训练。在物体检测上的评测证明了 RefinedBox 的有效性。

第四章　基于轻量级卷积神经网络的资源自适应的图像理解

根据第一章中的分析，计算机视觉经常面临着计算资源受限的问题，尤其是对移动设备上的应用来说，而传统的深度卷积神经网络的计算量却通常很大。本章通过研究基于轻量级的语义分割技术，来缓解计算机视觉的资源受限问题，从而达到资源自适应的图像理解。为此，本文基于以下观察提出了一种新的轻量级分割模型：1) 语义分割依赖于多尺度特征学习；2) 下采样是加速网络推理和扩大卷积感受野的最有效方法；3) 网络深度和卷积通道数之间的良好平衡对于轻量级模型至关重要。具体来说，本文所提出的轻量级卷积神经网络 MiniNet 采用 空间金字塔卷积（Spatial Pyramid Convolution, SPC）模块和 空间金字塔池化（Spatial Pyramid Pooling, SPP）模块作为多尺度特征学习的基本单元。此外，MiniNet 将大多数网络层和操作放在较小的尺度上，即原始图像分辨率的 1/16，而不是先前模型中常用的 1/8 尺度；MiniNet 还设法平衡网络深度和卷积通道数。这些有效的设计使得 MiniNet 能够以极少的参数和计算量、较快的速度达到较高的准确率。第一节介绍了本章的研究背景和研究动机；第二节提供了与本章相关的研究工作的概述；第三节介绍了所提出的基于轻量级网络的语义分割方法 MiniNet；第四节对所提出的方法进行实验验证；第五节对全章进行了总结。

第一节　引言

语义分割是计算机视觉中的一个基本问题，也是图像理解的常用手段。GPU 不断增长的计算能力加速了用于精确语义分割的 全卷积网络（Fully Convolutional Network, FCN）的发展。对于最新的模型[13, 116, 120, 122, 126]，通过引入更多的参数和各种复杂的操作来提高精度是很常见的。例如，PSPNet[13] 的参数约为 66M，它需要几秒钟的时间才能在 TITAN Xp GPU 上处理一张普通图像。但是，例如机器人、智能手机、自动驾驶汽车和增强现实智能眼镜这样的移动设备无法部署大型且耗电的强大 GPU（例如，TITAN Xp GPU 的功耗约为 250W），因此有限的计算资源阻止了最新的分割模型[36, 37] 的实际应用。此外，移动设备只有有限的存储空间。例如，智能手机不可能使用数百 MB 的内存来存储针对某个特定应用的预训练的深度模型。这启发本文开发在准确性、效率、

参数量和功耗之间取得良好平衡的语义分割模型。

为此，研究者们近来对轻量级语义分割的研究兴趣迅速增加，已经出现了许多轻量级的分割模型[35–37, 129–131, 134, 226]。这些模型通常采用深度可分离卷积[35, 37, 131]、非对称卷积[129, 134, 226] 和密集连接[134] 等技术，以减少网络参数和操作量。为了用浅层网络获得较大的感受野，在轻量级模型中还使用了扩张卷积[227]。尽管现有技术水平已得到一定程度的发展，但当前模型要么无法取得令人满意的准确率，要么参数量过多。例如，ESPNetv2[37] 具有 0.73M 参数，但在 Cityscapes 测试数据集[7] 上仅达到 62.1% 的 mIoU；而 ICNet[130] 达到了 69.5% 的 mIoU，却具有 6.68M 参数（即需要 6.68M×4=26.72M 的存储内存）。为了使模型能够灵活地应用于移动设备，本文认为参数数量应少于 0.5M，即少于 0.5M×4=2.0M 的存储内存。

在介绍所提出的模型之前，本文总结了一些对语义图像分割任务的观察。首先，语义分割高度依赖于多尺度学习来识别自然场景中的多尺度物体，这是以往的优秀方法成功的关键[82, 83]。现有的研究已经提出了各种技术来利用多尺度的深度学习特征，例如 *编码-解码*[83, 84] 结构、*空洞空间金字塔池化*（Atrous Spatial Pyramid Pooling, ASPP）[38]、*金字塔池化模块*（Pyramid Pooling Module）[13] 和*多路径优化*（Multi-path Refinement）[116]。其次，降低特征图的分辨率是提高推理速度和扩大感受野的最有效方法。例如，一半分辨率下的特征图所需的操作数是原始特征图的操作数的 1/4。因此，最近非常深的神经网络[4, 228] 通常会将输入图像降采样到非常小的分辨率。第三，在给定一定数量的参数的情况下，与具有更多卷积通道数的较浅的网络相比，具有较少卷积通道数的适当的较深的网络可以实现更高的精度，但速度更低。例如，假设一个具有 C 个输入通道和 C 个输出通道的 1×1 卷积，其参数的数量为 C^2。如果将输入和输出通道的数量更改为 $C/2$，则参数的数量变为 $C^2/4$。因此，好的轻量级分割模型应在网络深度和卷积通道数之间做出良好的权衡。

基于以上观察，本文提出了一种针对移动应用的轻量级、快速且高效节能的语义分割网络，即 MiniNet。本文引入了简单而有效的*空间金字塔卷积*（Spatial Pyramid Convolution, SPC）和*空间金字塔池化*（Spatial Pyramid Pooling, SPP）模块来作为 MiniNet 的基本单元，以从自然图像中学习多尺度的特征表示。为了加快计算速度，MiniNet 将特征图下采样到原始分辨率的 1/16，并在较小的尺度上放置大多数网络层和操作，这与以前的轻量级模型[36, 37, 129, 134, 226] 将大多数

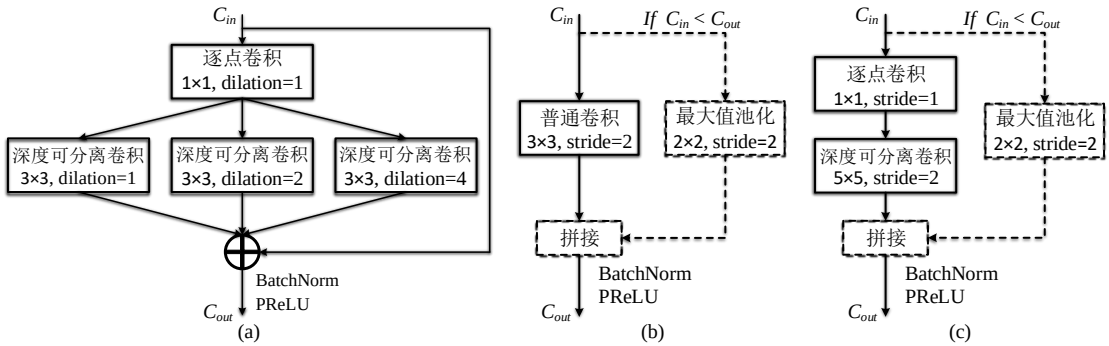


图 4.1 MiniNet 中基本模块的图示。(a) 带有三个分支的 SPC 模块；(b) 带有标准卷积的 SPP 模块；(c) 带有深度可分离卷积的 SPP 模块。

网络层置于 1/8 尺度不同。特征图的小尺度还有助于 MiniNet 扩大感受野，以便 MiniNet 能够以更少的网络层（意味着更少的参数）学习高层的抽象的语义信息。然后，本文提出了一种高效的解码器，以融合顶部的高层语义特征和底部的底层细粒度特征，以实现带有清晰的物体边界的准确分割。本文还设法在网络深度和卷积通道数之间取得良好的平衡，以实现更好的性能。由于模型较小，MiniNet 不需要 ImageNet[5] 预训练，因此可以灵活地适应新数据和新任务。

本文在三个数据集上进行了广泛的实验，包括 Cityscapes[7]、CamVid[229] 和 Mapillary Vistas[8]，以证明所提出的 MiniNet 的有效性和高效率。在 Cityscapes 测试数据集[7] 上，没有 ImageNet[5] 预训练的情况下，MiniNet 仅用 211K 参数和 2.4G FLOPs 达到了 66.3% 的 mIoU，速度达到了 94.3fps。较小版本的 MiniNet 以 95K 参数量，能够以 104.2fps 的速度达到 64.1% 的 mIoU。本文还进行了详细的消融实验，以评估各种设计选择的影响。

第二节 MiniNet

空间金字塔卷积. 众所周知，一个标准卷积可以分解为一个逐点卷积和一个深度可分离卷积[32]。逐点卷积实际上就是卷积核大小为 1×1 的卷积，深度可分离卷积是分组卷积，其分组数等于输出通道数。如上所述，多尺度学习对于语义分割至关重要。为了有效地进行多尺度学习，SPC 模块用一组扩张金字塔卷积代替了单个深度可分离卷积。假设有 r 个并行的扩张卷积，则其扩张率分别为 $1,2,\cdots,2^{r-1}$。那么 SPC 模块能够很自然地学习到多尺度信息，其感受野分别为 $3,5,\cdots,2^r+1$，其中最大的感受野为 2^r+1，远大于标准卷积。为了更好的训练优化，本文在将多个并行的扩张卷积的结果逐元素相加之后，添加了一个残差

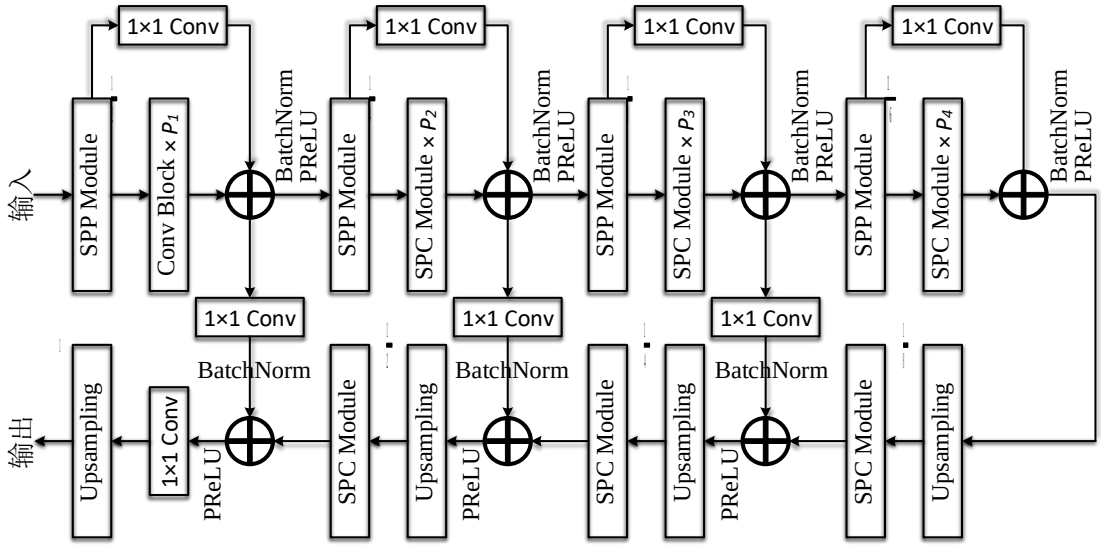


图 4.2 所提出的 MiniNet 的网络结构

连接[4]，其后使用批归一化[230] 和非线性激活 PReLU[10]。图4.1(a) 中显示了一个具有三个分支的 SPC 模块。

空间金字塔池化. SPP 模块用于对编码路径中的特征图进行下采样。它由两个分支组成，其中一个分支在不重叠的 2×2 窗口内进行最大池化，另一个分支中使用标准卷积或分解卷积，分别如图4.1(b) 和图4.1(c) 所示。在第一个卷积阶段中，SPP 使用标准卷积，其通道数较小，随后具有分解卷积的 SPP 在更深层被使用。假设有 C_{in} 个输入通道和 C_{out} 个输出通道。若 $C_{in} < C_{out}$，则最大池化分支会生成一个 C_{in} 个通道的特征图，卷积分支负责生成输出特征图中剩下的 $C_{out} - C_{in}$ 个通道。否则，最大池化分支会被省略，卷积分支会直接生成一个通道数为 C_{out} 的特征图。本文将在第4.3.2节中展示这种二分支设计优于单个分支的设计。

尺度下采样. 记 $F_{in} \in \mathbb{R}^{C_{in} \times Y \times X}$ 为一个卷积层的输入特征图，$F_{out} \in \mathbb{R}^{C_{out} \times Y' \times X'}$ 为输出特征图。那么，卷积核可以定义为 $T \in \mathbb{R}^{C_{out} \times C_{in} \times t \times t}$，其中 $t \times t$ 表示核大小。输出特征图的大小由卷积步长 s（即 $Y' = Y/s$ 和 $X' = X/s$）控制。因此，在这样的卷积层中，操作数大约与 $X'Y'C_{out}C_{in}t^2 = XYC_{out}C_{in}t^2/s^2$ 成正比。从这种表述中，可以发现卷积层中的运算数量大约与输入特征图、输入和输出通道数、以及卷积核大小成正比。如果将特征图下采样 2 倍，那么下采样后的特

征图的操作数将仅具有原始操作数的 1/4。此外，下采样还可以将感受野扩大两倍，从而可以减轻对网络深度的需求。因此，减小图像大小是加速卷积网络的最有效方法。之前的语义图像分割模型通常仅将图像降采样为原图的 1/8 大小[35, 36, 129, 134, 226]，但是本文将大部分卷积层放置在原图的 1/16 尺度下。

网络结构. 图4.2中展示了所提出的 MiniNet 的网络结构。MiniNet 是一个具有编码-解码结构的卷积神经网络。其中，编码部分由四个阶段组成。因为低层网络层的通道数通常较少，所以标准卷积比深度可分离卷积更高效，参数也只是略多[32, 33, 231]。因此，编码部分的第一个阶段使用标准卷积，而不是深度可分离卷积。对于编码的第一阶段，首先使用带有标准卷积的 SPP 模块（如图4.1(b) 所示）将输入图像降采样到 1/2 的尺度下。随后，将 P_1 个*残差卷积模块*顺序地连接，每个残差卷积模块可以用如下公式表示：

$$F_1^l = \text{PReLU}(\text{BatchNorm}(W_1^l * F_1^{l-1} + F_1^{l-1})), \tag{4.1}$$

其中，$*$ 表示卷积操作符，F_1^l 表示第一编码阶段的第 l ($l \in \{1,2,\cdots,P_1\}$) 个残差卷积模块的输出特征图。$W_1^l \in \mathbb{R}^{C_1 \times C_1 \times 3 \times 3}$ 是第一阶段的第 l 个模块的权重，其中，C_1 是特征的通道数。SPP 模块的输出特征图是 F_1^0。注意为了提高效率，本文遵循[226] 的建议忽略了卷积的偏差项。

对于第二个编码阶段，本文首先应用具有深度可分离卷积的 SPP 模块（图4.1(c)）将特征图进一步下采样为 1/4 尺度。然后，P_2 个 SPC 模块（图4.1(a)）紧随其后。编码部分的第三个和第四个阶段与第二个阶段相似，分别将特征图下采样到 1/8 和 1/16 尺度。与第一个阶段类似，对于后三个阶段，定义卷积通道数分别为 C_2、C_3、C_4，SPC 模块的数量分别为 P_2、P_3、P_4，输出特征图分别为 F_2^l ($l \in \{1,2,\ldots,P_2\}$)、F_3^l ($l \in \{1,2,\ldots,P_3\}$)、F_4^l ($l \in \{1,2,\ldots,P_4\}$)。后三个 SPP 模块的输出特征图分别为 F_2^0、F_3^0 和 F_4^0。对于第 k 个阶段的 SPP 模块，池化分支能够生成带有 C_{k-1} 个通道数的特征图，卷积分支（标准卷积或深度可分离卷积）负责生成剩下的 $(C_k - C_{k-1})$ 个通道。因此，第 k 个阶段的所有的残差卷积模块或 SPC 模块拥有 C_k 个输入特征通道和 C_k 个输出特征通道。为了便于优化，本文对于编码部分每个阶段的 SPP 模块和最后一个 SPC 模块设计了一个*长残差连接*，可用如下公式表示：

$$\begin{aligned} F_k^{P_k} = \text{PReLU}(\text{BatchNorm}(W_k^{long} * F_k^0 + F_k^{P_k}), \\ \text{s.t. } k \in \{1,2,3,4\}, \end{aligned} \tag{4.2}$$

表 4.1 两种具体的 MiniNet 配置

编号	(C_1, C_2, C_3, C_4)	(P_1, P_2, P_3, P_4)	FLOPs	参数量
#1	(16, 32, 64, 64)	(2, 2, 3, 4)	2.1G	95K
#2	(16, 32, 64, 96)	(2, 2, 3, 10)	2.4G	211K

其中，$W_k^{long} \in \mathbb{R}^{C_k \times C_k \times 1 \times 1}$ 是第 k 个编码阶段的 1×1 卷积的权重矩阵。除了长残差连接，本文将每个基础模块（即 SPC 模块或残差卷积模块）内部的残差连接称为*短残差连接*。

解码网络逐渐聚合顶部的粗糙的语义特征和底部的细粒度特征，以分割图像并使其具有清晰的边界。解码网络共有三个阶段，每个阶段可以写成如下形式：

$$\begin{aligned} \hat{F}_k' &= \text{BatchNorm}(\hat{W}_k * F_k^{P_k}), \\ \hat{F}_k'' &= \text{SPC}(\text{Upsampling}(\hat{F}_{k+1})), \\ \hat{F}_k &= \text{PReLU}(\hat{F}_k' + \hat{F}_k''), \\ &\text{s.t. } k \in \{1,2,3\} \end{aligned} \tag{4.3}$$

其中，$\hat{W}_k \in \mathbb{R}^{C_{k+1} \times C_k \times 1 \times 1}$ 表示 1×1 卷积层的权重矩阵。注意，$\hat{F}_4 = F_4^{P_4}$。本文省略了(4.3)中 SPC 模块的 PReLU 激活函数，因为 PReLU 随后已被用来激活 $\hat{F}_k'$ 和 $\hat{F}_k''$ 的和。最终，本文对 $\hat{F}_1$ 使用一个 1×1 的卷积来获取分割预测图，该预测图随后被上采样到与原图像一样的大小，以得到最终的语义分割结果。

深监督. 事实证明，深监督对很多计算机视觉任务都有帮助，例如图像分类[89]、物体检测[12]、视觉跟踪[232] 和边缘检测[23] 等。本文所提出的的 MiniNet 也采用了深监督来提高性能。在解码路径中，如 $\hat{F}_1$ 一样，本文将一个 1×1 卷积和一个上采样层分别连接在 $\hat{F}_2$、$\hat{F}_3$ 和 $\hat{F}_4$ 之后。训练过程中，所有的这些预测结果都使用真值和标准的 softmax 损失函数进行监督。和之前研究[13, 122, 130] 中一样，$\hat{F}_1$ 对应的损失函数的权重被设置为 1.0，而 $\hat{F}_2$、$\hat{F}_3$、$\hat{F}_4$ 对应的辅助的损失函数的权重被设置为 0.4。在测试阶段，$\hat{F}_2$,$\hat{F}_3$ 和 $\hat{F}_4$ 的预测结果被直接丢弃，并将 $\hat{F}_1$ 的预测结果作为最终输出的语义分割结果。

模型分析. 对于具有 r 个分支和 C 个输入、输出通道的 SPC 模块，其逐点卷积具有 C^2 个参数，而其金字塔的扩张的深度可分离卷积共有 rCt^2 个参数。由于卷积核大小 $t \times t$ 实际上是 3×3，并且为了效率，令 $r \leq 4$，所以逐点卷积占了

网络参数的绝大部分。因此，网络的参数量与通道数的平方大约成正比。具有 C 个通道的 SPC 模块的参数量大约等于 4 个具有 $C/2$ 通道的 SPC 模块的参数量。因此，在网络通道数和网络深度之间取得良好的平衡十分重要。此外，除了自然的多尺度学习之外，MiniNet 还容易获得大的感受野，因为 1) 每个 SPC 模块都具有大的感受野，例如一个具有四分支的 SPC 模块的感受野为 17；2) 大多数 MiniNet 的网络层在原始图像的 1/16 尺度上操作。考虑到以上几点，MiniNet 的目标是仅使用少量网络参数便能进行精确的语义图像分割。

一些网络设置. 本文在表4.1中设计了两种具体的网络配置。这两种 MiniNet 变体分别仅有 95K 和 211K 的网络参数量。对于编码网络中第二、第三和第四个阶段的 SPC 模块，本文设置 SPC 分支数（r）分别为 3、4、4。更多的分支能够获得更高的精度，但是速度会有所下降。注意，编码网络的第一阶段使用公式 (4.1) 所示的残差卷积模块而不是 SPC 模块。

第三节 实验

4.3.1 实验设置

数据集. 本文在三个著名的语义分割数据集上评估了所提出的方法，其中包括 Cityscapes 数据集[7]、CamVid 数据集[229] 和 Mapillary Vistas 数据集[8]。Cityscapes 数据集[7] 包含在 50 个不同城市的街道场景中记录的各种图像集，它由 2975 张训练图像、500 张验证图像和 1,525 张测试图像以及相应的像素级别的标注组成。所有图像均具有 1024×2048 的高分辨率，共 19 个类别，被分为 7 组。本文在验证集上进行消融实验，并在测试集上与其他方法进行比较。CamVid 数据集[229] 也是用于城市场景理解的，它包括 367 张训练图像、101 张验证图像和 233 张测试图像，共有 11 个类别。所有图像的分辨率均为 360×480。本文遵循先前的工作[36, 114, 130, 226] 采用训练集和验证集进行训练，并采用测试集进行测试。另外，本文使用 Mapillary Vistas 数据集[8] 来研究网络的泛化性。本文将其验证集中的 66 个类（2000 张图像）映射到 Cityscapes 数据集中的 19 个类。随后，使用在 Cityscapes 数据集上预先训练的模型在该数据集上评测。

实现细节. 本文使用流行的 PyTorch 框架[233] 来实现 MiniNet。采用 Adam 优化器[234] 来训练，权重衰减系数为 $1e$-4，初始学习率设置为 $2e$-3。本文使用“poly”

学习率策略，当前学习率等于基础学习率乘以 $(1-curr\_iter/max\_iter)^{power}$（其中 $power=0.9$，$curr\_iter$ 和 max_iter 分别表示当前和总共的迭代次数）。当与之前的模型进行比较时，本文遵循之前的研究[36, 37] 对 MiniNet 训练 300 个纪元。但是，当进行消融研究时，本文遵循之前的研究[36]，只训练 100 个纪元以节省时间。本文同样遵循之前的研究[36, 37] 使用标准的缩放、裁剪和翻转等操作对数据进行增广处理。对于 Cityscapes 数据集[7]，其图像分辨率为 1024×2048。本文遵循之前的研究[36, 37] 将其图像降采样为 512×1024 以进行训练；而为了进行准确的性能评估，本文使用双线性插值对网络的输出进行上采样，使其变为原始图像的大小，即 1024×2048。乘法-加法操作（Multiplication-Add Operations, FLOPs）的计算量和推理速度都是在 512×1024 的分辨率下进行的，该分辨率已经能够满足大多数实际应用的需求。Cityscapes 的训练仅使用该数据集中提供的精细标签。MiniNet 是从随机初始化开始训练的，而没有在 ImageNet 数据集[5] 上进行预训练。测试时，本文直接将网络的输出作为最终结果，不使用任何额外的后处理。所有实验都在一块 NVIDIA TITAN Xp GPU 上运行。

4.3.2 消融实验

在与以前的分割模型进行比较之前，本文首先评估 MiniNet 中各种设计选择的合理性。本文使用具有挑战性的 Cityscapes 验证集[7] 进行消融实验。所有的消融实验都将以 MiniNet 的 211K 版本（表4.1中的第二个变体）作为默认设置，在训练集上进行训练并在验证集上进行评测。

下采样大小. MiniNet 将大多数网络层置于 1/16 尺度下，而不是之前常用的 1/8 尺度下[35, 36, 129, 134, 226]。表4.2(a) 展示了具有不同下采样尺度的实验结果。从实验结果中可见，与 1/8 尺度相比，MiniNet 的尺度为 1/16 时，精度更高、速度更快且 FLOPs 更少。

解码器. 表4.2(b) 表明 MiniNet 中的解码器在将特征图从一个很小的分辨率（即 $1/16$）解码到原图大小中扮演了十分重要的角色。移除解码器，直接上采样 $1/16$ 尺度下的预测作为输出结果，将非常显著的降低网络性能。

长/短残差连接. 从表4.2(c) 中可见，短残差连接对于 MiniNet 是必不可少的。并且长残差连接能够进一步提高网络的效果，同时并不会显著增加计算量。

表 4.2 MiniNet 的消融实验的结果

(a) 不同下采样尺度的效果

尺度	参数量	FLOPs	速度	mIoU (val)
1/8	217K	3.3G	64.9fps	64.9
1/16	211K	2.4G	94.3fps	65.5

(b) MiniNet 解码器的影响

解码器	参数量	FLOPs	速度	mIoU (val)
w/o	182K	1.5G	156.3fps	61.4
w/	211K	2.4G	94.3fps	65.5

(c) 所提出的长/短残差连接的影响

残差连接	参数量	FLOPs	速度	mIoU (val)
w/o long	196K	2.3G	98.0fps	64.8
w/o short	211K	2.4G	96.2fps	62.7
w/ all	211K	2.4G	94.3fps	65.5

(d) SPP 模块中每个分支的影响

分支	参数量	FLOPs	速度	mIoU (val)
pooling	194K	2.2G	101.0fps	64.0
conv.	219K	2.5G	91.7fps	64.6
both	211K	2.4G	94.3fps	65.5

(e) 深监督的影响

深监督	参数量	FLOPs	速度	mIoU (val)
w/o	211K	2.4G	94.3fps	64.8
w/	211K	2.4G	94.3fps	65.5

(f) PReLU 激活函数的影响

激活函数	参数量	FLOPs	速度	mIoU (val)
ReLU	209K	2.4G	94.3fps	64.5
PReLU	211K	2.4G	94.3fps	65.5

(g) 不同卷积通道数量的效果

(C_1, C_2, C_3, C_4)	参数量	FLOPs	速度	mIoU (val)
(16, 32, 32, 64)	108K	1.7G	106.4fps	62.6
(16, 32, 64, 64)	135K	2.2G	98.0fps	63.6
(16, 32, 64, 96)	211K	2.4G	94.3fps	65.5
(16, 32, 64, 128)	318K	2.8G	88.5fps	65.7
(32, 64, 96, 128)	376K	5.6G	68.5fps	67.3
(16, 64, 128, 128)	415K	4.0G	67.6fps	67.6
(32, 64, 128, 128)	431K	6.1G	65.4fps	67.4

(h) 不同数量的 SPC 模块的效果

(P_1, P_2, P_3, P_4)	参数量	FLOPs	速度	mIoU (val)
(1, 1, 3, 10)	207K	2.0G	101.0fps	65.0
(2, 2, 3, 10)	211K	2.4G	94.3fps	65.5
(2, 2, 5, 10)	224K	2.5G	87.0fps	66.3
(3, 3, 5, 10)	229K	2.9G	80.0fps	65.8
(2, 2, 3, 5)	146K	2.3G	99.0fps	63.3
(2, 2, 3, 8)	185K	2.4G	96.2fps	64.5
(2, 2, 3, 12)	237K	2.5G	91.7fps	65.2

SPP 模块的两个分支. 在表4.2(d) 中评估了 SPP 模块的两个分支的影响。注意，仅有池化分支的 MiniNet 在编码器的第一阶段中使用带有步长的卷积来进行下采样，否则的话，将直接对图像用池化进行下采样。从表4.2(d) 中可见，带有两个分支的 SPP 模块效果最佳。

深监督. 如表4.2(e) 所示，采用深监督来提高效果是十分必要的。由于在网络推理时，深监督相关的计算会被丢掉，所以深监督可以在没有任何消耗的情况下将 mIoU 从 64.8% 提高到 65.5%。

PReLU 非线性激活函数. 本文遵循 ESPNet[36] 采用 PReLU[10] 作为非线性激活函数，而不是最常用的 ReLU[236] 激活函数。为了验证这一选择的有效性，本文使用 ReLU 非线性激活函数代替了 PReLU 非线性激活函数。结果显示在表4.2(f) 中。PReLU 可以将 ReLU 的 mIoU 从 64.5%提高到 65.5%。因此，PReLU 激活函数在轻量级分割任务上优于 ReLU 激活函数。

表 4.3 在 Cityscapes 数据集[7] 上 MiniNet 与其他分割模型的对比。“-”表明本文无法获取相应的结果。

方法	预训练	参数量	FLOPs	速度	mIoU (%)		
					class (val)	category (test)	class (test)
DANet[235]	ImageNet	68.50M	551.7G	< 1fps	81.5	-	81.5
DeepLabv3+[120]	ImageNet+COCO+JFT	54.61M	165.9G	< 1fps	79.6	-	82.1
BiSeNet[131]	ImageNet	5.80M	6.6G	42.0fps	69.0	-	68.4
DenseASPP[121]	ImageNet	28.64M	244.9G	< 1fps	78.9	90.7	80.6
DFN[126]	ImageNet+COCO	44.84M	165.9G	1.2fps	-	-	79.3
PSPNet[13]	ImageNet+COCO	65.58M	514.0G	< 1fps	-	90.6	80.2
FRRN[133]	ImageNet	17.71M	475.8G	5.0fps	-	88.9	71.8
SegNet[114]	ImageNet	29.45M	326.0G	8.0fps	-	79.1	57.0
DeepLabv1[118]	ImageNet	42.54M	362.9G	1.0fps	-	-	63.1
DeepLabv2[38]	ImageNet+COCO	43.90M	374.3G	< 1fps	71.4	86.4	70.4
FCN-8s[83]	ImageNet	134.46M	334.4G	6.8fps	-	85.7	65.3
ICNet[130]	ImageNet	6.70M	7.4G	62.5fps	67.7	-	69.5
ShuffleNetv2[34]	ImageNet	2.60M	3.5G	91.7fps	60.3	-	-
ENet[226]	No	364K	3.8G	34.7fps	-	80.4	58.3
ESPNet[36]	No	364K	4.5G	61.0fps	61.4	82.2	60.3
ESPNetv2[37]	ImageNet	99K	564M	142.0fps	54.1	-	54.7
ESPNetv2[37]	ImageNet	725K	3.4G	83.0fps	62.7	-	62.1
MiniNet	No	95K	2.1G	104.2fps	63.3	84.2	64.1
MiniNet	No	211K	2.4G	94.3fps	67.3	85.1	66.3

卷积通道数. 在表4.2(g) 中评估了不同卷积通道数的影响。更多的通道数能够产生更好的结果，但同时伴随着更多的参数、更多的 FLOPs、以及更低的速度。考虑到精度、速度、参数量等之间的权衡，对于 MiniNet 的两种变体，本文选择了两组 (C_1, C_2, C_3, C_4) 的设置，即 $(16, 32, 64, 64)$ 和 $(16, 32, 64, 96)$。

SPC 模块的数量. 表4.2(h) 展示了不同 SPC 模块数量的评估结果。添加更多的 SPC 模块能够产生更好的结果，对于编码过程的第三个阶段（即 P_3）尤其有效，因为当前 P_3 较小。考虑到精度、速度、参数量等之间的权衡，本文设置 $(2, 2, 3, 10)$ 作为 211K 参数版本的 MiniNet 的设置。

4.3.3 与最优模型的比较

Cityscapes 数据集. 本节将所提出的 MiniNet 在 Cityscapes 数据集上[7] 与高精度语义分割模型以及最近的轻量级模型进行比较，包括 DANet[235]、DeepLabv3+[120]、BiSeNet[131]、DenseASPP[121]、DFN[126]、PSPNet[13]、FRRN[133]、SegNet[114]、DeepLabv1[118]、DeepLabv2[38]、FCN-8s[83]、ICNet[130]、ShuffleNetv2[34]、ENet[226]、ESPNet[36] 和 ESPNetv2[37]。本文报告这些模型的参数数量、FLOPs 和速度。对于 Cityscapes 验证集的精度评测采用所有类别的 IoU 的均值，对于

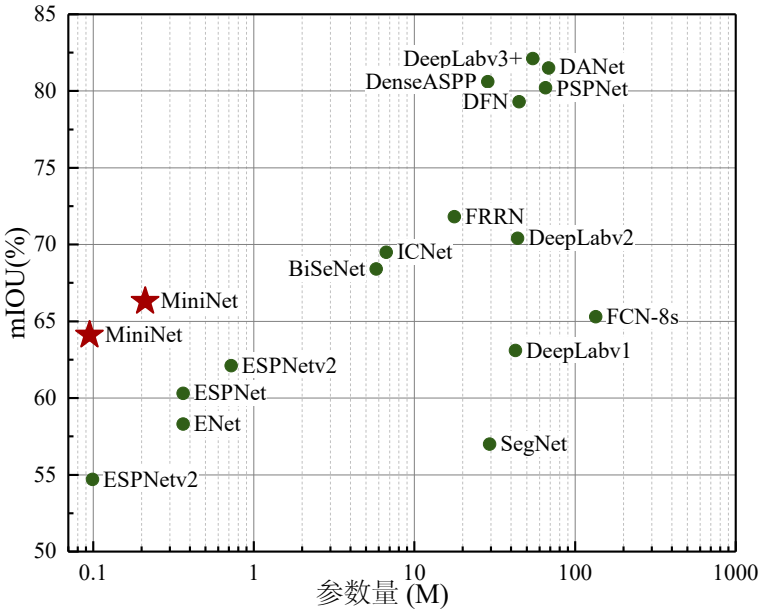


图 4.3 各种语义分割方法的网络参数量与在 Cityscapes 测试集[7] 上的逐类别 mIoU 的关系。左上的点表示模型的精度越高、参数越少。

测试集同时采用所有类别的 IoU 的均值和 7 大类别分组的 IoU 的均值。结果如表4.3所示。

首先来分析所提出的 MiniNet 与高精度模型的比较。MiniNet 具有最少的参数量，即便较精确版本的 MiniNet 也只有 211K 参数，比 DANet[235] 少约 320 倍，比 DeepLabv3+[120] 少约 250 倍。MiniNet 的少量参数使其可以灵活地被部署到各种移动设备上，精确版的 MiniNet 仅需要不到 1M 的存储空间，而 95K 参数版本的 MiniNet 仅需要 0.4M 的存储空间。此外，精确版的 MiniNet 仅有 2.4G FLOPs，比其他高精度模型少两个数量级。很少的 FLOPs 意味着 MiniNet 的能耗很小，使其适合在移动设备上的部署。精确版的 MiniNet 还实现了 94.3fps 的超实时速度，而大多数高精度模型甚至达不到 1fps 的速度。与 FCN-8s[83] 和 DeepLabv1[118] 相比，MiniNet 具有更快的速度、更少的参数、更少的 FLOPs、更高的准确性，且 MiniNet 的精度可与 DeepLabv2[38] 媲美。另外，MiniNet 不需要像很多大型网络一样在 ImageNet 数据集[5] 上预训练。这是因为从随机初始化开始训练小型网络比较容易，而没有在 ImageNet 数据集上预训练的大型网络则很难收敛。没有任何预训练使得开发新的网络结构十分灵活。例如，如第4.3.2节所示，用户可以轻松地通过堆叠适当更多的模块或适当增大卷积通道数来获得更好的性能，而无需进行预训练。

原图　　真值　　MiniNet

图 4.4　所提出的 MiniNet 在 Cityscapes 数据集[7] 上的一些分割结果

接着来分析所提出的 MiniNet 与高精度模型的比较。从表4.3中可见，ESPNetv2[37] 具有比 ESPNet[36] 更好的精度，但是 ESPNetv2 参数更多。然而，具有 95K 参数的 MiniNet 却比具有 725K 参数的 ESPNetv2[37] 具有更好的效果。具体来说，与 ESPNetv2 的 725K 参数的版本相比，MiniNet 的 95K 版本的参数减少了 7.6 倍，且 FLOPs 更少、速度更快、准确性更高。与 ESPNetv2 的 99K 版本相比，MiniNet 的 95K 版本的 mIoU 高出 9% 以上。与 ICNet[130] 相比，MiniNet 的参数减少了 30 倍，速度更快，且性能可与其媲美。MiniNet 也大大优于 ShuffleNetv2[34]。请注意，MiniNet 并未使用 ImageNet[5] 预训练，而大多数其他网络都已在 ImageNet 上进行了预训练。

图4.3中展示了各种语义分割方法的参数量与在 Cityscapes 测试集[7] 上的逐类别 mIoU 的关系图。从该图可见，本文的网络可以用最少的参数获得可观的性能。为了更好地查看 MiniNet 的分割结果，本文在图4.4中提供了一些示例。

CamVid 数据集.　接下来在另一个城市场景理解数据集（即 CamVid 数据集[229]）上评测所提出的网络。本文与可获取该数据集上结果的最新分割模型进行了比

表 4.4 所提出的 MiniNet 与其他分割模型在 CamVid 测试数据集[229] 上的效果比较

方法	参数量	FLOPs (G)	mIoU (%)
BiSeNet[131]	5.80M	2.2	65.6
PSPNet50[13]	46.58M	117.2	69.1
SegNet[114]	29.45M	104.3	55.6
Dilation8[227]	140.8M	-	65.3
DeepLabv1[118]	42.52M	121.4	61.6
FCN-8s[83]	134.35M	139.6	57.0
ICNet[130]	6.68M	2.6	67.1
ENet[226]	364K	1.3	51.3
ESPNet[36]	353K	1.3	55.6
MiniNet	95K	0.7	66.7
MiniNet	211K	0.8	67.5

表 4.5 Mapillary Vistas 验证集[8] 上的模型泛化性评估，该数据集中的类别已被映射到与 Cityscapes 数据集的类别相同。所有的模型均在 Cityscapes 训练集上训练并且没有再微调。

方法	参数量	Pixel Acc. (%)	mIoU (%)
PSPNet[13]	65.58M	82.5	43.8
ICNet[130]	6.68M	78.1	31.8
ENet[226]	364K	50.1	18.0
ESPNet[36]	364K	50.5	16.2
ESPNetv2[37]	99K	47.0	13.2
ESPNetv2[37]	725K	43.4	14.6
MiniNet	95K	55.7	18.2
MiniNet	211K	55.8	20.5

较，包括 BiSeNet[131]、PSPNet50[13]、SegNet[114]、Dilation8[227]、DeepLabv1[118]、FCN-8s[83]、ICNet[130]、ENet[226] 和 ESPNet[36]。结果如表4.4所示。由于 CamVid 含有 11 个类别，与 Cityscapes（19 类）有所差别，因此每个分割模型的参数数量与表4.3可能略有不同。可以发现 MiniNet 仅需少量参数和少量的 FLOPs 即可达到最优的精度，例如，MiniNet 比 ICNet[130] 少 30 倍的参数，比 PSPNet50[13] 少 220 倍的参数。

Mapillary Vistas 数据集. Mapillary Vistas 数据集[8] 是最新发布的街道场景数据集。本文通过如下方式将 Mapillary Vistas 数据集中的 66 个类别映射到 Cityscapes 数据集[7] 中的 19 个类别：1) 合并“traffic sign front”类和“traffic sign back”类到 Cityscapes 中的“traffic sign”类；2) 合并“bicyclist”、“motorcyclist”和“other rider”

到 Cityscapes 中的“rider”类；3) 忽略 Mapillary Vistas 没有出现在 Cityscapes 中的其他类。因此，得到的新 Mapillary 数据集将具有与 Cityscapes 数据集相同的类别。为了测试分割模型对未见数据的泛化性，本文使用在 Cityscapes 训练集[7]上预训练的模型来对 Mapillary 验证集（2000 张图像）进行评估，模型并不进行微调。

在 ESPNet[36] 中，Mehta 等人提议将 Mapillary 的类别划分为与 Cityscapes 相同的 7 个类别分组。这是不适当的，因为这样的分类将使得看起来完全不一样的物体被分到同一个分组中。例如，“vehicle”分组包括了像“bus”和“car”等类别，但是 Mapillary 中的“boat”类别也被看作了“vehicle”。对于一个在“bus”和“car”类别的数据上预训练的模型，是很难检测出“boat”类别的，因为它们看起来完全不同。

结果总结在表4.5中。尽管 MiniNet 仍然比大规模、参数更多的网络（例如 PSPNet[13] 和 ICNet[130]）差一些，但它始终优于其他轻量级语义分割模型，包括 ENet[226]、ESPNet[36] 和 ESPNetv2[37]。

第四节 小结与讨论

为了应对现实环境中的计算资源经常受限的问题，本文提出了一种新的轻量级语义分割网络 MiniNet，以实现资源自适应的图像理解。MiniNet 通过 SPC 和 SPP 模块实现了多尺度学习，将大多数网络层置于较小的分辨率（1/16 尺度），并试图平衡卷积通道数和网络深度。本文提供了详细的消融实验，以证明各种设计选择的有效性，这将有助于对 MiniNet 的理解。与最新的分割模型相比，MiniNet 能够以更少的参数、更快的速度和更少的 FLOPs 获得更好或相当的准确性。MiniNet 的高效率和小尺寸使其可以部署在移动设备上。将来，作者计划将 MiniNet 应用于其他移动端的视觉任务。

第五章　基于通用的图像属性知识的弱监督图像理解

如第一章中的分析所示，弱监督学习是解决计算机视觉中标注数据不足的重要途经。弱监督学习以通用的图像属性知识作为输入，使用尽量简单的数据标注进行图像理解，是实现知识引导的自适应图像理解的重要技术支撑。本章研究基于弱监督的像素级图像理解，致力于仅依靠图像级监督进行弱监督实例/语义分割，而不是依赖昂贵的像素级蒙版或边界框标注，并通过将所有训练图像的图像级别信息聚合到一个大的知识图中，以利用该图中的语义关系来解决这一难题。具体来说，本章的工作从一些类别无关的、通用的、基于分割的似物性采样（Segment-based Object Proposal, SOP）开始，提出了一个多实例学习（Multiple Instance Learning, MIL）[146] 框架，该框架可以使用带有图像级标签的训练图像以端到端的方式进行训练。对于每个 SOP，此 MIL 框架可以同时计算概率分布和类别感知的语义特征，利用这些信息可以构造一个大型无向图。此图中还包括背景类别，以删除 SOP 中的大量噪声。因此，该图的最佳多路割可以为每个 SOP 分配可靠的类别标签。这些带有指定类别标签的去噪后的 SOP 可以视为训练图像的伪实例分割，用于训练全监督的模型。所提出的方法在弱监督实例分割和语义分割方面都达到了最新的性能。第一节介绍了本章的研究背景和研究动机；第二节提供了与本章相关的研究工作的概述；第三节引入了对问题的公式化表述；第四节介绍了所提出的基于 SOP 的 MIL 框架；第五节介绍了基于多路割的标签分配；第六节对所提出的方法进行了实验验证；第七节对全章进行了总结。

第一节　引言

实例感知语义分割（简称实例分割）专注于同时检测和分割图像中的所有对象实例。由于其巨大的学术和工业价值，使得它成为计算机视觉中最重要的任务之一。实例分割的最新进展是由功能强大的基准系统驱动的，例如 Fast/Faster/Mask R-CNN[11, 106, 179] 和全卷积神经网络[83]。但是，这些深度模型的性能在很大程度上依赖于大量训练数据以及昂贵的逐像素标记。标注此类训练数据一直是将实例分割应用于实际应用中的一个严重瓶颈，因为对大量图像

进行逐像素标注特别耗时。例如，在 Cityscapes 数据集中逐像素标注一张图像需要“平均超过 1.5 小时”[7]。

为了减轻对昂贵的逐像素标注的需求，一些研究使用边界框来放宽监督[142, 144, 145, 147]，其中训练数据可以只是用于物体检测的数据。尽管标记边界框要比标记像素便宜，但由于边界框标记仍然是一件劳动密集型的事情，弱监督物体检测实际上早已是一个经过充分研究的领域[237–239]。本文的工作遵循另一些研究[39–43] 来进一步放宽监督，即**仅使用图像级监督来进行弱监督实例分割**。由于图像级标签的标注成本较低，因此该类方法将使许多实际应用受益。

在弱监督实例分割中，主要挑战之一是将图像的标签分配给每个语义实例，例如，似物性采样的物体推荐[26]。Zhou 等人[39] 试图通过计算从图像分类器[3, 4] 获得的类激活图（Class Activation Map, CAM）[148] 中的类峰值响应来解决这一难题。这些峰值响应可用于查询与类别无关的对象建议，以预测实例遮罩。与 Zhou 等人[39] 类似，许多其他弱监督实例分割方法[40–43] 和弱监督语义分割方法[154, 156, 160, 164, 168, 173, 240–242] 在很大程度上也依赖于 CAM 进行物体识别。但是，CAM 倾向集中于目标对象的小部分具有区分性的区域，并且 CAM 也很难从包含小对象、多个对象和复杂背景的复杂场景中准确定位对象。尽管已经引入了各种技术[152, 156, 161, 243] 来改善 CAM，但 CAM 的天然局限性仍然阻碍了弱监督学习的发展[42]。

基于以上观察，本文提出了一种可以克服这些局限性的新方法。与以前的基于 CAM 的弱监督分割方法不同，这些方法直接使用 CAM 或 CAM 的改进版本进行物体识别[39–43, 152, 154, 156, 160, 164, 168, 173, 240–242]，本文使用 CAM 作为多实例学习（Multiple Instance Learning, MIL）框架中的**监督源之一**来学习训练过程中每张图像的语义信息。因此，CAM 通过提供近似的粗略信息来帮助训练所提出的系统，但是所提出的系统的性能并不完全依赖于 CAM，因为还有其他设计来确保对 MIL 框架进行训练，这在实验中得到了证明。此外，本文提出将所有训练图像的有用信息集成到一个大型的知识图中，并探索该图中的信息以桥接图像级标签和相应的语义实例。这样，所提出的方法不仅考虑了每个图像的固有属性，还考虑了训练数据库的整体数据分布，从而打破了 CAM 在弱监督分割方面的局限性。

具体来说，本文的工作从一些通用的基于分割的似物性采样开始（Segment-based Object Proposal, SOP），例如 selective search[30]，LPO[101]，以及 MCG[26]。

因为这些方法与类别无关，所以它们不依赖任何语义标签。因此，所提出的系统可以仅使用图像级信息将其推广到任何类别。给定一张图像的标签和它的 SOP，本文旨在为每个 SOP 分配正确的类别标签并过滤出噪声采样。为了实现此目标，本文建立了一个 MIL 框架，用于使用图像标签作为监督的图像分类。在此框架中，如果一个 SOP 包含特定类别的对象，则所提出的模型将学习使该 SOP 为相应类别的最终分类概率做出更多贡献。如果一个 SOP 不包含目标类别中的任何物体，则所提出的模型将忽略该 SOP。最后，该 MIL 框架可以为每个 SOP 分配所有目标类别的概率分布并且计算语义特征向量。

通过将训练数据库中的所有图像的 SOP 视为*非终端结点*，并将所有目标类别（包括背景）视为*终端结点*，可以使用产生的概率分布和语义特征向量构造一个无向图。这个大图可以很好地表示训练数据库中每个 SOP 的属性以及所有 SOP 之间的关系。该无向图的最佳*多路割*可以将每个 SOP 与适当的类别标签相关联。删除 SOP 中的噪声后，带有自动分配的标签的其余 SOP 可以用作伪实例分割，以用于训练全监督的模型。由于所提出的方法利用了实例、图像和数据集级别的信息，因此将其称为 LIID，即英文 "**L**everages **I**nstance-, **I**mage- and **D**ataset-level information" 的缩写。

本文在 PASCAL VOC2012 数据集[9] 和 MS-COCO[6] 数据集上进行了广泛的实验，以在各种实验设置下评测所提出的方法。评测结果表明，LIID 在弱监督实例分割和语义分割方面都达到了最新的性能。综上所述，本文的主要贡献有三点：

- 提出了一种新的*多实例学习*（MIL）框架，从而为每个 SOP 计算概率分布并提取语义特征向量。
- 使用产生的概率分布和语义特征构造一个大型无向图，其中目标类别（包括背景）被视为*终端结点*。并进一步提出了一种有效的近似优化算法，可以对该图进行*多路割*以获得伪实例分割。
- 大量实验表明，对于弱监督实例分割和语义分割，所提出的 LIID 始终能够达到最新的性能。

第二节 问题表述

假设有一个训练图像集 $\mathcal{I}=\{I_1,I_2,\cdots,I_N\}$ 和相应的图像级别标签 $\mathcal{Y}=\{Y_1,Y_2,\cdots,Y_N\}$，其中 N 是训练图像的数量。假设 $\mathcal{K}=\{0,1,2,\cdots,K\}$ 是类别

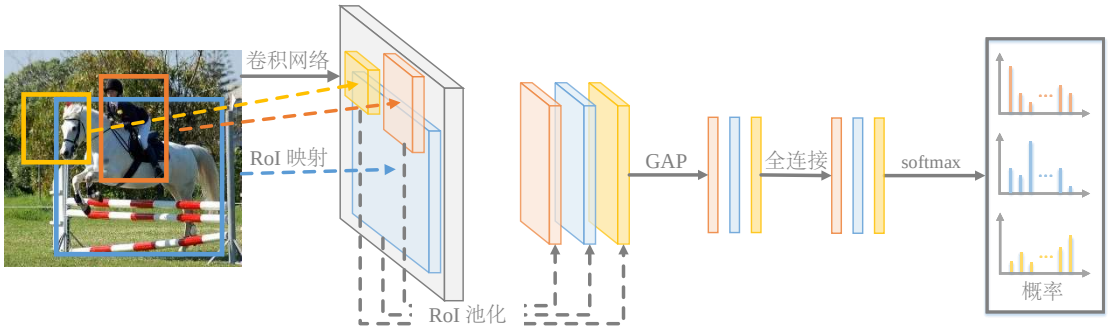


图 5.1 本文针对基于 MIL 的多标签图像分类提出的网络架构。该网络旨在为每个 SOP 同时计算概率分布并提取语义特征。

集合，其中 0 表示背景，K 是目标语义类别的数量。在每个图像都有背景区域的温和假设下，则有 $0 \in Y_i$ 和 $Y_i \subseteq \mathcal{K}$ $(i = 1,2,\cdots,N)$。为方便起见，定义 $\mathcal{K}' = \{1,2,\cdots,K\}$，不包括背景类别[1]。可以将图像 $\mathcal{I}$ 输入到任何自底向上的 SOP 生成方法[26, 30, 101, 104, 111, 224, 244] 中去，以获得 SOP $\mathcal{S} = \{S_1, S_2, \cdots, S_N\}$。假设 $S_i = \{s_i^1, s_i^2, \cdots, s_i^{|S_i|}\}$ $(i = 1,2,\cdots,N)$，并且 s_i^j $(j = 1,2,\cdots,|S_i|)$ 是二进制分割蒙版。注意 $|\cdot|$ 表示一个集合中元素的数量。于是，可以轻易地获得这些基于分割的 SOP 的相应边界框，它们可以表示为 $\mathcal{B} = \{B_1, B_2, \cdots, B_N\}$，其中 $B_i = \{b_i^1, b_i^2, \cdots, b_i^{|S_i|}\}$。

这些与类别无关的 SOP 可能不包含任何语义物体、多个或一个语义物体。不包含完整语义物体和多个物体的 SOP 在本文中被认为是噪声。为了进行实例分割，本文的主要目的是删除 SOP 中的噪声，并为紧密包含一个完整物体的 SOP 分配正确的类别标签。因此，本文的目标可以表述为

$$F(s_i^j) = \begin{cases} 0 & \text{如果 } s_i^j \text{ 是一个噪声采样} \\ k' & \text{如果 } s_i^j \text{ 属于类别 } k' \end{cases}, \tag{5.1}$$

其中 $k' \in \mathcal{K}'$，s_i^j 表示第 i 张图像中的第 j 个 SOP。具有 $F(s_i^j) > 0$ 的采样 s_i^j 将作为图像 I_i 的伪实例分割。图5.2中展示了所提出的用于计算 $F(s_i^j)$ 的解决方案的概述。

[1]为清楚起见，用 $k \in \mathcal{K}$ 中使用 $k' \in \mathcal{K}'$ 分别代表包括背景和不包括背景的类别。

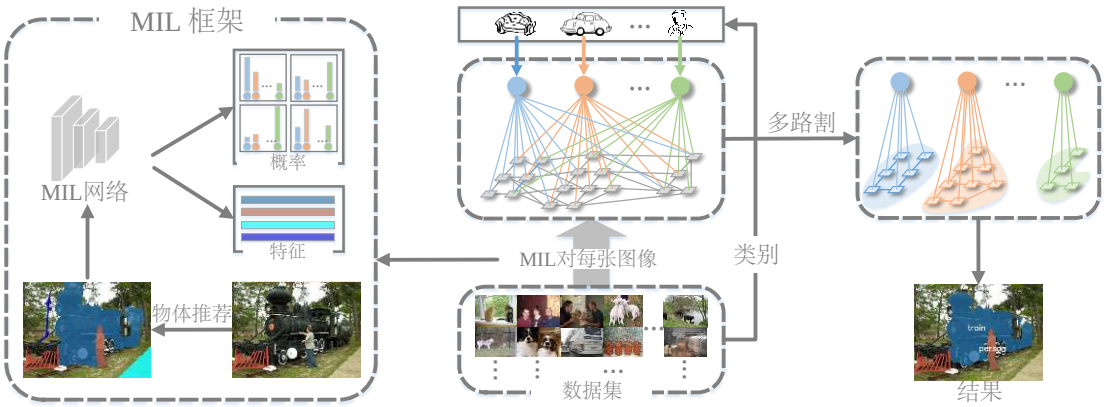


图 5.2 本文所提出的方法的概述。带有图像级别标签的训练图像用于训练基于 MIL 的多标签图像分类网络，如第三节所示。所有训练图像以及相应的 SOP 都被输入到 MIL 网络中，以计算概率分布和语义特征。然后，使用所有训练图像构造一个大型知识图。伪实例分割可以使用改进的多路割算法来获得。

第三节 基于 SOP 的 MIL 框架

给定具有图像级别标签的图像，以前的研究[39–41] 通常会训练用于为目标定位计算 CAM 的多标签图像分类器。然后，他们将 CAM 和 SOP 结合起来以产生伪分割。由于 CAM 的天然局限性，如上所述，训练数据没有得到充分利用。与此不同，本文考虑将 SOP 纳入训练过程，使得每个 SOP 都能学习有用的信息。给定带有图像标签 Y_i 的输入图像 I_i，可知相应的物体推荐 S_i/B_i 包含类别 Y_i，但每个物体推荐分别对应于哪个类别未知。这实际上是多实例学习（Multiple Instance Learning, MIL）的一种情况。因此，本文建立了一个 MIL 框架，该框架以图像和 SOP 作为输入，并以图像级别的标签作为监督。通过训练，该模型有望学习为每个 SOP 生成类概率分布和语义特征向量，并将其用于后续的多路割。本节首先介绍所提出的神经网络架构，然后介绍用于训练基于 SOP 的 MIL 框架的几种损失函数。

5.3.1 网络架构

这一部分介绍为基于 MIL 的多标签图像分类而设计的卷积神经网络。所提出的网络架构展示在图5.1中。在这里，与类别无关的物体推荐是由 MCG 算法[26] 生成的。输入图像 I_i 首先通过骨干网络，即 ResNet50[4]，再使用 SOP S_i 的边界框 B_i 对生成的特征图执行 ROI 池化[179]。ROI 池化之后跟随一个全局平均池化（GAP）层来将每一个 SOP 对应的特征图转换为一个 2048 维的特征向量 $\mathbf{f}_i^j$ $(j=1,2,\cdots,|S_i|)$。然后连接一个全连接层，该全连接层有 $(K+1)$ 个输出 $\mathbf{a}_i^j$

($|\mathbf{a}_i^j| = K+1$)，代表 K 个目标类别以及背景的预测分数。最后，令 $(\mathbf{p}_i^j)_k$ 为在一个 softmax 层之后获得的类别 k 的概率，因而可以有

$$(\mathbf{p}_i^j)_k = \frac{\exp((\mathbf{a}_i^j)_k)}{\sum_{m=0}^{K}\exp((\mathbf{a}_i^j)_m)}, \tag{5.2}$$

其中 $k \in \mathcal{K}$。通过这种设计模式，可以为每个 SOP 计算特征向量 $\mathbf{f}_i^j$ 和概率分布 $\mathbf{p}_i^j$。通过使用适当的损失函数，所提出的神经网络将会为每个 SOP 学习类别感知信息。

5.3.2 基于 SOP 的 MIL 损失函数

对于 MIL 框架的训练，本文提出了几个损失函数以同时推断概率分布并提取 SOP 的语义特征。考虑到 SOP 的标签未知，本文设计了一个基于 CAM 的损失函数来估计每个 SOP 的伪标签，并且本文还设计了一个基于 MIL 的图像分类损失函数来计算每个图像的汇总的概率，以便采用图像标签作为监督源。通过施加这些损失函数来监督概率分布 $\mathbf{p}_i^j$，从而使网络收敛。此外，本文设计了一种基于 MIL 的中心损失函数，以将语义特征向量 $\mathbf{f}_i^j$ 集中在同一类别上，这样，属于同一类别的 SOP 的特征向量 $\mathbf{f}_i^j$ 将具有较小的特征距离。

5.3.2.1 基于 CAM 的损失函数

本文不再像之前的方法[39–41] 那样依赖 CAM 来定位物体，而是通过用 CAM 为每个 SOP 估计伪标签，来将 CAM 用作训练的监督源之一。具体来说，使用具有 K 个独立的交叉熵损失函数的标准 ResNet50[4] 网络，可以训练一个多标签图像分类模型。然后可以使用著名的 CAM 算法[148] 计算图像 I_i 的 CAM $A_i^{k'}$ ($k' \in \mathcal{K}'$)。$A_i^{k'}$ 被标准化为 [0,1] 的范围。令 $\widetilde{y}_i^j$ 表示第 j 个 SOP 的估计类别标签（$j = 1,2,\cdots,|S_i|$）。假设 $(R_i^j)_{k'} = mean(A_i^{k'}[b_i^j]) + max(A_i^{k'}[b_i^j])$ 且 $(R_i^j)_{k'} \in [0,2]$，其中 $A_i^{k'}[b_i^j]$ 表示 SOP b_i^j 在 $A_i^{k'} \in [0,1]$ 中的对应区域。使用计算的 CAM 来估计 $\widetilde{y}_i^j$，如下所示

$$\widetilde{y}_i^j = \begin{cases} 0 & if \ \ \forall k', (R_i^j)_{k'} < \eta \\ \arg\max\limits_{k'} (R_i^j)_{k'} & otherwise \end{cases}, \tag{5.3}$$

其中 η 是一个阈值。因此，可以将 $\widetilde{y}_i^j$ 视为第 j 个 SOP b_i^j 的伪标签，而 $\mathcal{K}\backslash\{\widetilde{y}_i^j\}$ 是除了 $\widetilde{y}_i^j$ 以外的类别集合。将基于 CAM 的损失函数定义为

$$L_{Att}^{(i)} = -\frac{1}{|S_i|}\sum_{j=1}^{|S_i|}\left[\log(\mathbf{p}_i^j)_{\widetilde{y}_i^j} + \frac{1}{K}\sum_{k\in\mathcal{K}\backslash\{\widetilde{y}_i^j\}}\log(1-(\mathbf{p}_i^j)_k)\right]. \tag{5.4}$$

通过这种方式，预训练的多标签图像分类模型可以通过 CAM 来帮助所提出的 MIL 框架的训练。对于 $(R_i^j)_{k'}$ 的计算，本文在 $A_i^{k'}[b_i^j]$ 中使用边界框池化而不是蒙版池化，因为基于边界框的物体推荐往往比基于分割的蒙版（即 SOP）更可靠。如相关研究[26, 30, 101, 224] 中所描述的那样，边界框的生成比蒙版生成要容易得多，因此可以实现更高的准确性。自底向上的方法很难准确地分割物体，而且不准确的蒙版将会对 MIL 的训练有害。下文将通过消融实验在第5.5.2节中进一步证明这种设计的合理性。

5.3.2.2 基于 MIL 的图像分类损失函数

尽管 SOP 的标签是未知的，但是图像中所有 SOP 的学习到的概率分布 $\mathbf{p}_i^j$ 的聚合可以反映网络的分类能力。换句话说，虽然不能直接监督每个 SOP 的概率 $\mathbf{p}_i^j$，但是可以监督一张图像的总体的概率聚合。假设图像 I_i 中每个类的聚合概率为 $(\mathbf{Z}_i)_k$ $(k\in\mathcal{K})$，可以从 $(\mathbf{p}_i^j)_k$ 推断得到。本文使用 Log-Sum-Exp（LSE）函数[245] 来计算 $(\mathbf{p}_i^j)_k$ $(j=1,2,\cdots,|S_i|)$ 的平滑的最大值的估计，而不是 $(\mathbf{p}_i^j)_k$ 的简单的最大值或平均值，这可以表示为

$$(\mathbf{Z}_i)_k = \frac{1}{r}\log\left[\frac{1}{|S_i|}\sum_{j=1}^{|S_i|}\exp(r\,(\mathbf{p}_i^j)_k)\right], \tag{5.5}$$

其中 r 是一个使得 LSE 函数的表现介于最大值和平均值之间的参数。本文根据经验将 r 设置为 5[172]。与简单的最大值相比，LSE 函数不仅可以估计其最大值，而且可以考虑到 $(\mathbf{p}_i^j)_k$ 的所有元素。通过估计的 $(\mathbf{Z}_i)_k$，将基于 MIL 的图像分类损失函数定义为

$$L_{MIL}^{(i)} = -\frac{1}{|Y_i|}\sum_{k\in Y_i}\log((\mathbf{Z}_i)_k) - \frac{1}{|\overline{Y_i}|}\sum_{k\in\overline{Y_i}}\log(1-(\mathbf{Z}_i)_k), \tag{5.6}$$

其中 $\overline{Y_i}$ 是 Y_i 的补集。符合直觉的是，当前类别应出现在 SOP 中，而对缺席类别具有高概率的 SOP 应受到惩罚。

如第二节中所述，假定每个图像都有背景区域，即 $0 \in Y_i$ $(i = 1,2,\cdots,N)$。这个温和的假设对于公式 (5.6) 至关重要。一方面，自底向上算法生成的 SOP 通常包含许多噪声，这些噪声 SOP 并不在目标物体类别中，涵盖其他物体类别甚至非物体区域，因此必须为每张图像包括背景类，以确保神经网络的训练。另一方面，本文的目标要求按照公式 (5.1) 识别并过滤掉这些噪声 SOP，因此必须将背景类别纳入训练中，以学习有关噪声 SOP 的适当信息，然后使用第四节中的技术过滤掉它们。

5.3.2.3 基于 MIL 的中心损失函数

下一个损失函数被设计用于语义特征提取。该训练被期望能使具有相同类别的 SOP 的语义特征的相似性最大化，并且使具有不同类别的 SOP 的相似性最小化。为此，本文引入了基于 MIL 的中心损失函数，以集中具有相似语义的语义特征：

$$\begin{aligned} \widehat{y}_i^j &= \arg\max_k (\mathbf{p}_i^j)_k, \\ L_{Cent}^{(i)} &= \frac{1}{|S_i|}\sum_{j=1}^{|S_i|}\left[1 - \frac{\mathbf{f}_i^j \cdot \mathbf{c}_{\widehat{y}_i^j}}{\|\mathbf{f}_i^j\|_2 \|\mathbf{c}_{\widehat{y}_i^j}\|_2}\right], \end{aligned} \tag{5.7}$$

其中 $\mathbf{c}_k$ 是学习到的第 k 类的输入样本的中心，而 $\|\cdot\|_2$ 表示向量的 ℓ^2 范数。该损失度量了特征向量 $\mathbf{f}_i^j$ 与学习到的类别中 $\mathbf{c}_k$ 之间的 cosine 相似度。在每次训练迭代中，根据语义特征向量 $\mathbf{f}_i^j$ 将 $\mathbf{c}_k$ 更新为

$$\mathbf{c}_{\widehat{y}_i^j}^{new} = \mathbf{c}_{\widehat{y}_i^j}^{old} + \theta \cdot (\mathbf{f}_i^j - \mathbf{c}_{\widehat{y}_i^j}^{old}), \qquad for\ j = 1,2,\cdots,|S_i|, \tag{5.8}$$

其中 θ 是更新速率。因此，任意两个 SOP 之间的相似距离可以通过它们学习到的特征向量 $\mathbf{f}_i^j$ 来计算。

通过以上定义，可以通过以下公式来表示基于 MIL 的多标签图像分类问题的总体损失函数：

$$L^{(i)} = \alpha L_{Att}^{(i)} + \beta L_{MIL}^{(i)} + \gamma L_{Cent}^{(i)}. \tag{5.9}$$

实际上，本文根据经验将 α，β 和 γ 分别设置为 0.5、0.5 和 0.1。所提出的 $L_{Att}^{(i)}$ 可以利用预先训练的多标签图像分类模型来帮助 MIL 的训练，而 $L_{MIL}^{(i)}$ 自然地适合此处的 MIL 训练。因此，$L_{Att}^{(i)}$ 和 $L_{MIL}^{(i)}$ 的系数被均设置为 0.5。对于损失 $L_{Cent}^{(i)}$，其目的是最大程度地减少类内差异，这与图像分类无关，因此为其设置一个小的系数 0.1 以避免它对分类结果的影响。

第四节 基于多路割的标签分配

直观地，考虑所有训练样本的数据分布的预测将比仅考虑单个样本的预测更好。这是因为单个样本可能有偏差或随机误差，但总体数据分布更为可靠。尽管 MIL 框架的训练过程已经利用了所有训练数据，但这只是整体数据分布的间接使用。这里考虑一种直接的方法。具体来说，本文利用一个庞大的知识图，其中包括所有训练图像中的 SOP，以提供一种全局的解决方案。

5.4.1 多路割问题的回顾

在介绍用于 SOP 标签分配的方法之前，本部分将简要回顾多路割问题。先来描述传统的图割。假设有一个连通的无向图 $G=(V,E)$，其中结点集为 V，边集为 E。该图 G 的权重函数可以表示为 $w:E\to\mathbb{R}^+$，其中 $\mathbb{R}^+$ 表示非负实数的集合。交换属性适用于任何结点对 $u\in V$，$u\in V$，即 $w(u,v)=w(v,u)$。通过将 V 划分为不相交的子集 V_1 和 V_2 来定义图割，从而得到边集的子集 $E'\subseteq E$，其中每条边在 V_1 中有一个顶点，另一个顶点在 V_2 中。因此，边的子集 E' 可用于表示该图割。该图割的代价定义为 $\sum_{(u,v)\in E'}w(u,v)$。典型的最小割问题是找到将两个给定结点 $\dot{u}$ 和 $\dot{v}$（这些结点被称为终端结点）分开的具有最小代价的切割，即 $\dot{u}\in V_1$ 和 $\dot{v}\in V_2$。这个最小割问题是最大流问题的对偶，可以在多项式时间内解决。

多路割问题是最小割问题的一种泛化，也被称为多终端切割问题[246–248]。给定一组终端结点 $\hat{E}\subseteq E$，多路割是找到具有最小代价的边的子集 $E'\subseteq E$，删除 E' 将使得终端结点相互隔绝。换句话说，图 $(V,E-E')$ 的任何连通子图都不可能包含 $\hat{E}$ 中的两个终端结点。当只有两个终端，即 $|\hat{E}|=2$ 时，此问题等效于上述在多项式时间内可解决的最小割问题。当存在三个或更多终端，即 $|\hat{E}|\geq 3$ 时，多路割成为 NP 难问题，需要近似算法来解决。下面的小节将 SOP 标签分配表示为多路割问题，并提出了解决该问题的一种简单方案。

5.4.2 知识图的构造

为了计算公式 (5.1) 中的 $F(s_i^j)$，本文构造了一个大知识图，该图不仅包含每个 SOP 的内在属性，而且还包含整个训练数据库中不同 SOP 之间的关系，因为使用了所有训练图像来构造该图。利用该知识图将为每个 SOP 分配一个可靠的类别标签。这里将标签分配过程表示为一个多路割问题，并为该问题引入了一个有效的近似解决方案。训练图像的图割结果将作为图像的伪实例分割，可用

于训练全监督的模型。

如第5.4.1节中所述，本文构造了一个连通的无向图 $G=(V,E)$。具体来说，将所有 SOP s_i^j $(i=1,2,\cdots,N;j=1,2,\cdots,|S_i|)$ 和目标类别 $\mathcal{K}$ $(\mathcal{K}=\{0,1,2,\cdots,K\})$ 看作图的结点，因此 $V=\mathcal{K}\cup S_1\cup S_2\cup\cdots\cup S_N$。此外，令 $\mathcal{K}$ 为终端结点的集合，即 $\hat{E}=\mathcal{K}$。每条边 $(u,v)\in E$ 有一个非负权重

$$w(u,v)=\begin{cases}(\mathbf{p}_i^j)_k & if\ \exists i,j\ \ u=s_i^j; v\in\mathcal{K}\\ 0 & if\ u\in\mathcal{K}, v\in\mathcal{K}\\ \delta\cdot\dfrac{|\mathbf{f}_i^j\cdot\mathbf{f}_{i'}^{j'}|}{\|\mathbf{f}_i^j\|_2\|\mathbf{f}_{i'}^{j'}\|_2} & if\ \exists i,j\ \ u=s_i^j;\ \exists i',j'\ \ v=s_{i'}^{j'}\end{cases}, \tag{5.10}$$

其中 δ 是一个平衡因子。因此，终端结点之间的边缘权重为 0，这是知识图 G 中边的最小权重。SOP 结点和终端结点之间的边的权重就是该 SOP 属于相应类别的预测概率。两个 SOP 结点之间的边的权重是其特征向量的 cosine 相似性[173, 249]，因此具有相似语义内容的 SOP 对将具有较大的 cosine 相似性。通过这种方式，图 G 通过合并在第三节中学习的所有训练图像的概率分布和语义特征，包含了整个训练数据库的知识。

5.4.3 知识图上的多路割

给定带有一组终端结点 $\mathcal{K}$ 的知识图 $G=(V,E)$，本文的目标是找到一种多路割方法，以断开每个终端结点与其余终端结点的连接。也就是说，本文的主要目标是找到具有最小代价的边的子集 $E'\subseteq E$，以便在新图 $(V,E-E')$ 中，任何两个终端结点之间没有连通的路径。经过多路割后，具有相似语义信息的 SOP 的对应结点将落入同一子图中，因为上述多路割通过使图割代价最小化，已使每个子图内的相似度最大化，并使不同子图之间的相似度最小化。每个子图中只有一个终端结点 $k\in\mathcal{K}$，每个 SOP 的伪类别标签就是其对应子图中的终端结点 k。此处，类别 $k=0$ 表示背景或噪声 SOP，因为它不属于目标物体类别。

常用数据集，例如 PASCAL VOC2012[9] 和 MS-COCO[6] 通常具有 $|\mathcal{K}|\geq 3$，即存在三个或更多的物体类别。如第5.4.1节中所讨论的，这里需要一种近似算法来解决上述多路割问题。假设 Δ_K 表示 K-单纯形，因此 $\mathbb{R}^{K+1}$ 中的 K 维凸多面体可以表示为 $\{x\in\mathbb{R}^{K+1}|(x\geq 0)\wedge\sum_k x_k=1\}$。对于 $k,\dot{k}\in\mathcal{K}$，$e^k\in\mathbb{R}^{K+1}$ 表示单位向量，即 $(e^k)_k=1$ 且 $(e^k)_{\dot{k}}=0$ $(\forall k\neq\dot{k})$。根据相关文献[248]，可以制定以

下优化函数来解决多路割问题

$$\begin{aligned} \min_{x} \frac{1}{2} \sum_{(u,v)\in E} w(u,v) \cdot \|x^u - x^v\|_1 \;\; s.t. \\ x^u \in \Delta_K, \;\; \forall u \in V; \\ x^k = e^k, \;\; \forall k \in \mathcal{K}, \end{aligned} \tag{5.11}$$

其中 $\|\cdot\|_1$ 表示 ℓ^1 范数。但是，由于指数级数量的约束[248]，直接求解公式 (5.11) 中的线性规划是不切实际的，尤其是在整个训练数据库上的知识图非常大的情况下。直接的解决方案所需的 CPU 内存和运行时间对于现有设备来说是不可行的。具体来说，直接的解决方案的空间复杂度为 $O(|E||V|^2)$，PASCAL VOC2012[9] 训练集所需的 CPU 内存约为 $10^3 \sim 10^4$ GB，比现有计算机的存储容量大得多。

为了解决这个问题，本文将每个结点 $u \in (V - \mathcal{K})$ 连接到最多三个具有最大的边的权重的其他结点 v ($v \in (V - \mathcal{K})$ 且 $v \neq u$)，而不是将每个结点 $u \in V$ 连接到所有其他节点。可以观察到，在获得的稀疏图中，整个大型知识图将自动划分为许多小的相互不连通的子图，每个子图可以记作 $G_t = (V_t, E_t)$：

$$\begin{aligned} \cup_t V_t = V, \\ \cup_t E_t = E. \end{aligned} \tag{5.12}$$

在多路割问题中，每个子图彼此独立。可以通过将公式 (5.11) 分解为许多项来轻松证明这一点，所分解的每个项代表子图的多路割的代价。这些分解项中的公共图结点仅是终端结点，这不会影响最终的图割结果，因为这些终端结点最终必须落入不同的图割中去。因此，可以单独处理每个子图，以计算其多路割 E'_t。为了解决此线性规划问题，首先使用单纯形方法来求解公式 (5.11)，其结果将使用 IBM-CPLEX[250] 的分支定界法进一步转换为多路割的结果。原始大图的多路割 E' 可以通过下式求得

$$\cup_t E'_t = E'. \tag{5.13}$$

如此，便可以通过计算许多小图来成功地估计大图的多路割。在这里，本文选择为每个节点连接三条边，是因为为每个节点连接四条边将会导致子图过大，如上所述，子图也将很难求解。有了多路割结果，如果 SOP s_i^j 与终端结点 k ($k \in \mathcal{K}$) 属于同一子集，就可以轻松地将公式 (5.1) 中的 $F(s_i^j)$ 分配给类别 k。如

果 $F(s_i^j)=0$，那么 SOP s_i^j 将是噪声，因此将被丢弃。对于 $F(s_i^j)\neq 0$ 的其余 SOP，使用对应的边界框 b_i^j 应用非极大值抑制（Non-Maximum Suppression, NMS），非极大值抑制的重叠率（Intersection-over-Union, IoU）阈值为 0.4，就像物体检测领域中常做的那样[11, 106, 179]。此 NMS 操作解决了多个 SOP 代表同一个物体的情况。最后，将其余的 SOP 和相应的类别标签 $F(s_i^j)$ 作为训练图像的伪真值，便可以训练 Mask R-CNN 模型[11]（使用 ResNet50[4] 作为骨干网络）用于弱监督实例分割，或训练 DeepLab 模型[38]（使用 ResNet101[4] 作为骨干网络）用于弱监督语义分割。

第五节 实验

5.5.1 实验设置

数据集. 本文在 PASCAL VOC2012 数据集[9] 和 MS-COCO 数据集[6] 上对提出的方法进行了评测。请注意，仅图像级标签被用于训练，而不使用像素级的标注。VOC2012 数据集[9] 包含 20 个语义物体类别以及背景类别。本文遵循之前的研究[39–41] 来利用 VOC2012 的 main trainval 子集（不包括 segmentation val 中的图像）来训练本文所提出的 MIL 框架（大约 10K 图像），使用 1449 张 segmentation val 中的图片来评测 LIID 和基准模型。对于消融实验，本文采用 VOC2012 的 main trainval 的子集（不包括 segmentation train+val 中的图像）进行训练，并采用 segmentation train 进行验证。MS-COCO 数据集[6] 包含 80 个语义物体类别。本文遵循之前的研究[173] 在其标准的 trainval 集上进行训练，并在其 test-dev 集上进行评测。

实现细节. 为了公平比较，在训练中，本文采用自底向上的 MCG[26] 算法为每张图像生成 500 个 SOP，然后使用 Wei 等人[219] 的简单过滤法从中为 VOC2012/MS-COCO 选择 20/40 个 SOP。用 PyTorch 来实现基于 MIL 的多标签图像分类模型。将 SGD 优化算法与 step 学习率策略一起使用以训练模型。对于 VOC2012 和 MS-COCO 数据集，初始学习率均为 5×10^{-4}，在 5 个纪元后将其除以 10。使用一张图像的小批量运行 SGD，总共运行 10 个纪元。权重衰减率和动量分别设置为 10^{-4} 和 0.9。在建图的过程中，本文遵循之前的研究[173] 以计算的显著性实例[244] 来作为 SOP。Mask R-CNN[11] 和 DeepLab[38] 的训练使用了网上的公开代码，并且遵循其默认设置。

表 5.1 在 VOC2012 的 segmentation train 数据集[9] 上对不同的 θ 值（在公式 (5.8) 中）和 γ 值（公式 (5.9) 中）的评测结果。每一对结果 w_1/w_2 分别表示没有（w_1）和有（w_2）知识图的结果。

编号	θ	γ	AP_{50}	AP_{75}	ABO
1	0.01	0.05	30.3/33.8	14.9/16.3	36.7/38.7
2	0.01	0.1	**32.5/34.8**	**15.5/16.7**	**38.2/39.4**
3	0.01	0.5	32.4/33.7	15.1/16.1	37.9/39.2
4	0.03	0.1	32.0/33.7	14.9/16.0	38.0/39.3
5	0.03	0.3	31.9/33.7	14.8/16.3	37.6/39.3
6	0.05	0.1	32.3/33.6	15.1/16.3	37.8/39.2
7	0.005	0.5	29.1/32.3	13.6/15.5	36.3/38.5
8	0.05	0.5	31.5/33.2	14.9/16.2	37.7/39.0
9	-	0	31.3/-	15.0/-	37.8/-

表 5.2 在 VOC2012 的 segmentation train 数据集[9] 上对不同的 α 值和 β 值（在公式 (5.9) 中）的评测结果。每一对结果 w_1/w_2 分别表示没有（w_1）和有（w_2）知识图的结果。

编号	α	β	AP_{50}	AP_{75}	ABO
1	1.0	0.0	28.7/31.8	13.9/15.7	35.5/37.9
2	0.8	0.2	31.5/34.0	14.7/16.6	37.4/39.2
3	0.5	0.5	**32.5/34.8**	**15.5/16.7**	**38.2/39.4**
4	0.2	0.8	31.3/32.8	14.8/16.1	36.2/37.6
5	0.0	1.0	18.7/19.3	8.8/9.2	22.9/23.0

表 5.3 在 VOC2012 的 segmentation train 数据集[9] 上对 $(R_i^j)_{k'}$ 中 *mean* 和 *max* 的存在与否（在公式 (5.3) 中）的评测。每一对结果 w_1/w_2 分别表示没有（w_1）和有（w_2）知识图的结果。

编号	mean	max	AP_{50}	AP_{75}	ABO
1	✔	✗	28.7/32.6	13.1/15.5	34.3/37.7
2	✗	✔	32.4/33.6	15.1/16.1	37.7/38.7
3	✔	✔	**32.5/34.8**	**15.5/16.7**	**38.2/39.4**

评测指标. 对于实例分割的评测指标，本文遵循之前的研究[39] 来采用在 IoU 阈值 0.5（AP_{50}）和 0.75（AP_{75}）下的基于蒙版的平均精度（Average Precision, AP）指标（详情请参见原文[6]），以及另一个视角下的平均最佳重叠（Average Best Overlap, ABO）指标（详情请参见原文[30]）。

5.5.2 消融实验

在与其他方法进行比较之前，本节进行了一些消融实验，以评测不同设计选择和参数设置的有效性。如上所述，所有消融实验都是针对 VOC2012 的

表 5.4 在 VOC2012 的 segmentation train 数据集[9] 上对公式 (5.3) 中计算 $(R_i^j)_{k'}$ 时是用边界框池化还是蒙版池化的评测。每一对结果 w_1/w_2 分别表示没有（w_1）和有（w_2）知识图的结果。

编号	物体推荐类型	AP_{50}	AP_{75}	ABO
1	边界框	**32.5/34.8**	**15.5/16.7**	**38.2/39.4**
2	蒙版	30.7/32.4	14.5/15.7	36.8/38.0

表 5.5 在 VOC2012 的 segmentation train 数据集[9] 上对不同 η 值（在公式 (5.3) 中）的评测。每一对结果 w_1/w_2 分别表示没有（w_1）和有（w_2）知识图的结果。

编号	η	AP_{50}	AP_{75}	ABO
1	0.25	28.3/30.8	13.7/15.3	34.6/36.5
2	0.50	30.0/33.4	14.2/16.0	36.5/38.7
3	0.75	**32.5/34.8**	**15.5/16.7**	**38.2/39.4**
4	1.00	30.1/32.5	14.5/16.0	36.4/38.3

表 5.6 在 VOC2012 的 segmentation train 数据集[9] 上对不同 δ 值（在公式 (5.10) 中）的评测结果。

编号	δ	AP_{50}	AP_{75}	ABO
1	1	30.3	14.9	37.0
2	2	34.6	16.3	39.3
3	3	34.8	16.7	39.4
4	5	34.8	16.7	39.4
5	10	34.8	16.6	39.4

segmentation train 数据集[9] 上的弱监督实例分割进行的。在这里，如果没有提及，便不会训练 Mask R-CNN[11] 以节省时间。调整每组超参数时，其他参数将保持为默认值。

中心损失函数 $L_{Cent}^{(i)}$ 的超参设置. 中心损失函数旨在聚合特征向量 $\mathbf{f}_i^j$。超参数 θ（在公式 (5.8) 中）控制每个类别的中心特征向量的更新速度，而参数 γ（在公式 (5.9) 中）控制其对骨干网络的影响。表5.1中展示了 θ 和 γ 的不同设置和相应结果。当 $\gamma = 0$ 时，将省略参数 θ 和相应的知识图（表5.1中的第 9 号）。可以看到，此设置的结果比没有知识图的最佳设置要差，这表明基于 MIL 的中心损失函数（第5.3.2.3节）不仅对于知识图的构建是必要的，而且对 MIL 框架的训练很有帮助。当 $\gamma \neq 0$ 时，θ 和 γ 似乎对不同的值不敏感。$\theta = 0.01$ 和 $\gamma = 0.1$ 的设置可获得更好的性能。因此，本文分别使用 0.01 和 0.1 作为 θ 和 γ 的默认值。

表 5.7 在 VOC2012 的 segmentation train 数据集[9] 上对 LIID 的上界的评测。LIID 的上界使用标注的边界框来过滤和标记 SOP。

编号	真值边界框	AP_{50}	AP_{75}	ABO
1	✗	34.8	16.7	39.4
2	✔	44.9	23.0	39.1

损失函数 $L_{Att}^{(i)}$ 和 $L_{MIL}^{(i)}$ 的平衡因子. 这里评测公式 (5.9) 中损失函数 $L_{Att}^{(i)}$ 和 $L_{MIL}^{(i)}$ 的平衡因子 α 和 β 的效果。结果显示在表5.2中。可以看到，$L_{Att}^{(i)}$ 和 $L_{MIL}^{(i)}$ 对最后的实例分割的贡献都很大。当 $\alpha = 0.5$ 且 $\beta = 0.5$ 时，所提出的方法效果最佳，因此本文将此设置用作默认设置。

$(R_i^j)_{k'}$ 的 *mean* 和 *max* 项. 在公式 (5.3) 中，定义了一个用来估计类别标签 $\widetilde{y}_i^j$ 的辅助项 $(R_i^j)_{k'} = mean(A_i^{k'}[b_i^j]) + max(A_i^{k'}[b_i^j])$，它将在公式 (5.4) 中用于计算 $L_{Att}^{(i)}$。在表5.3中，作者仅使用 $(R_i^j)_{k'}$ 的 *mean* 项，仅 *max* 项、以及同时使用 *mean* 和 *max* 项进行 MIL 训练。第三个实验明显胜过其他两个。

$(R_i^j)_{k'}$ 的边界框或蒙版池化. 在第5.3.2.1节中，直观地分析了为什么在公式 (5.3) 中使用边界框池化而不是蒙版池化来计算 $(R_i^j)_{k'}$ 的原因。这里，在 VOC2012 的 segmentation train/val 数据集[9] 上进行实验以验证边界框池化相对于蒙版池化的优越性。结果显示在表5.4和表5.8中（第 6 号）。可以观察到，蒙版级别的池化会导致性能显著下降，这可能是因为不准确的 SOP 损害了 MIL 框架的训练。

$(R_i^j)_{k'}$ 的阈值 η. 在表5.5中，对公式 (5.3) 中的 $(R_i^j)_{k'}$ 应用不同的阈值 η。尽管 $\eta \in [0,2]$，但是这里仅测试 $\eta \leq 1.00$，因为 $\eta \geq 0.75$ 会导致性能显著下降。阈值 0.75 表现最佳，因此本文将其用作默认设置。

知识图的有效性. 在第四节中，本文使用 MIL 框架的输出来构造一个知识图，该知识图的多路割可以为相应的 SOP 分配类别标签。如果没有知识图，还可以使用 MIL 所学的概率来标记 SOP。在表5.1 - 表5.5中，这里报告了多路割前后的结果。知识图可以在所有情况下提高性能。因此，可以得出结论，知识图对于所提出的 LIID 系统至关重要。

表 5.8 在 VOC2012 的 segmentation val 数据集[9] 上对 Mask R-CNN 训练之后的 LIID 每个组件的评测。符号✗表示删除 LIID 中的一个组件。第一行（编号 1）是 LIID 的默认版本。

编号	策略	AP_{50}	AP_{75}	ABO
1	-	**48.4**	**24.9**	**50.8**
2	CAM-Based Loss $L_{Att}^{(i)}$ ✗	38.3	17.1	45.4
3	MIL Loss $L_{MIL}^{(i)}$ ✗	46.9	24.1	48.1
4	Center Loss $L_{Cent}^{(i)}$ ✗	45.8	23.0	48.6
5	Knowledge Graph ✗	46.1	22.8	48.1
6	$(R_i^j)_{k'}$ (Box $\rightarrow$ Mask)	45.2	22.9	48.9

平衡因子 δ. 在公式 (5.10) 中，使用平衡因子 δ 来控制特征的 cosine 相似度对图的边权重的贡献。在表5.6中，研究了不同的 δ 值的影响。当 $\delta \geq 2$ 时，可以获得了类似的结果。本文将 δ 的默认值设置为 5，因为 $\delta = 5$ 具有略好的性能。

关于 CAM 的讨论. 如果为公式 (5.9) 设置 $\alpha = 1.0$ 和 $\beta = 0.0$ 而不使用第四节中的知识图，则模型将退化为仅依赖 CAM 进行训练。在表5.2中，可以看到，按 AP_{50}、AP_{75} 和 ABO 计，结果分别为 28.7%、13.9% 和 35.5%。使用本文的其他设计，就 AP_{50}、AP_{75} 和 ABO 而言，结果分别提高到了 34.8%、16.7% 和 39.4%。请注意，这个基于 CAM 的简单变体还包括一些本文的有效设计，如表5.3 - 表5.5所示。因此，本文的系统并不是直截了当的。

LIID 的上界. 这里使用标注的真值边界框过滤和标记 SOP 来评测 LIID 的上限。具体来说，如果一个 SOP 的边界框与任何一个真值边界框的 IoU 大于 0.5，则保留该 SOP，并且为其分配与具有最大 IoU 的真值边界框相同的标签。否则，该 SOP 被丢弃。在表5.7中展示了实验结果。LIID 和其上界之间存在很大的性能差距，为将来的改进留有余地。

Mask R-CNN 训练之后的每个组件. 现在继续在 VOC2012 的 segmentation val 数据集[9] 上评测每个组件在 Mask R-CNN 训练之后的影响。具体来说，逐个忽略每个分量，如损失函数或多路割，然后采用生成的伪实例分割来训练 Mask R-CNN。结果汇总在表5.8中。可以观察到，LIID 的每个组件都会对最终性能产生重大影响，因为删除任何组件都会导致性能大幅下降。

表 5.9 在 VOC2012 的 segmentation val 数据集[9] 上比较所提出的 LIID 和其他弱监督实例分割模型。浅色方法[145] 使用边界框作为监督，而其他方法仅使用图像级标签作为监督。

方法		AP_{50}	AP_{75}	ABO
CAM[148]	Rect.	2.5	0.1	18.9
	Ellipse	3.9	0.1	20.8
	MCG	7.8	2.5	23.0
SPN[251]	Rect.	5.2	0.3	23.0
	Ellipse	6.1	0.3	24.0
	MCG	12.7	4.4	27.1
MELM[237]	Rect.	14.6	1.9	26.4
	Ellipse	19.3	2.4	27.0
	MCG	22.9	8.4	32.9
PRM[39]		26.8	9.0	37.6
IAM-S5[40]		28.8	11.9	41.9
Cholakkal 等人[41]		30.2	14.4	44.3
Ahn 等人[42]		46.7	17.4	-
Hsu 等人[145]		58.9	21.6	-
Label-PEnet[43]		30.2	12.9	41.4
LIID (Ours)		**48.4**	**24.9**	**50.8**

5.5.3 VOC2012 上的实例分割

由于仅由图像级别监督的弱监督实例分割是 Zhou 等人[39] 最近提出的问题，所以以前对此问题的研究非常有限[39–43]。因此，本节遵循 Zhou 等人[39] 来基于一些弱监督的物体定位模型[148, 237, 251] 生成的边界框构建一些基准模型。为了获得实例分割，应用了三种简单的蒙版提取策略：i）矩形（Rect），即仅使用边界框作为分割结果；ii）椭圆（Ellipse），即仅填充每个边界框所包含的最大椭圆；iii）MCG，即为每个边界框以最大 IoU 检索 MCG SOP[26]。本节使用训练集的伪实例分割来训练 Mask R-CNN 模型[11]，并将测试结果与现有的模型[39–43, 145] 以及这 9 个基准模型进行比较。

在表5.9中总结了 VOC2012 的 segmentation val 数据集[9] 上的数值的实验结果。请注意，Hsu 等人[145] 使用边界框作为监督，因此直接将其他方法与其进行比较是不公平的。尽管如此，相对于该方法[145] 而言，所提出的 LIID 在度量指标 AP_{75} 上实现了 3.3% 的提高，这证明了 LIID 对于准确分割物体实例的有效性。看到该方法[145] 在指标 AP_{50} 方面胜过 LIID 并不奇怪，因为该方法[145] 使用的边界框先验将极大地帮助其查找物体实例，从而粗略地分割它们，使其与真

图 5.3 PASCAL VOC2012 的 segmentation val 数据集[9] 上的实例分割的定性结果。

值有些重叠。对于图像级监督的方法，所提出的 LIID 在各种评测指标下均达到最佳性能。相比于第二好的方法，即 Ahn 等人[42] 的方法，LIID 的 AP_{50} 和 AP_{75} 分别高 1.7% 和 7.5%。请注意，AP_{75} 是实例分割中最重要的度量指标，因为它反映了检测紧密覆盖物体的能力。AP_{75} 方面的显著提高表明 LIID 擅长准确地分割与真值高度重叠的物体。最近，使用由 MCG 生成的 SOP 的弱监督物体检测模型 MELM[237] 的性能还不错，但比 PRM[39] 和 LIID 差。这证明了弱监督物体检测与弱监督实例分割密切相关，但不能直接应用于弱监督实例分割。在图5.3中展示了一些 LIID 的实例分割结果的示例。可以看到，LIID 可以产生很好的实例分割。即使对于包含多个相同类别实例的图像，也可以很好地分割每个实例。

运行时间和内存消耗. 对于运行时间和内存占用，多路割需要大约 5 分钟的时间和 26 GB 的 CPU 内存才能处理 VOC2012 训练数据集。MIL 框架需要大约 0.02 秒来处理一张图像。因此，每张训练图像的平均运行时间为 $5 \times 60/10\text{K} + 0.02 = 0.05$ 秒。测试图像的运行时间与 Mask R-CNN[11] 相同，因为本文采用伪真值来训练 Mask R-CNN 进行测试。

5.5.4 MS-COCO 上的实例分割

本部分将 LIID 与其他方法[173, 252] 进行比较，这些方法在 MS-COCO 数据集[6] 上报告了弱监督实例分割结果。这里使用与 VOC2012 数据集相同的实验设置为 SOP 分配类别标签，并训练 Mask R-CNN[11] 模型。此外，还报告了三种全

表 5.10 MS-COCO 的 test-dev 数据集[6] 上的实例分割的蒙版 AP。关于度量指标的详细信息可以在[6] 中找到。浅色的方法是全监督的，而[173, 252] 和 LIID 是弱监督的。

方法	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
MNC[136]	24.6	44.3	24.8	4.7	25.9	43.6
FCIS[253]	29.2	49.5	-	7.1	31.3	50.0
Mask R-CNN[11]	35.7	58.0	37.8	15.5	38.1	52.4
Fan 等人[173]	13.7	25.5	13.5	0.7	15.7	26.1
WS-JDS[252]	6.1	11.7	5.5	1.5	7.1	12.2
LIID (Ours)	**16.0**	**27.1**	**16.5**	**3.5**	**15.9**	**27.7**

监督的方法的结果，包括 MNC[136]、FCIS[253] 和 Mask R-CNN[11]。评测结果汇总在表5.10中。所提出的 LIID 的性能明显优于其他弱监督方法[173, 252]，这表明所提出的 LIID 对不同的数据集具有鲁棒性。与 Fan 等人[173] 相比，LIID 的 AP、AP_{50} 和 AP_{75} 分别实现了 2.3%、1.6% 和 3.0% 的性能提升。

5.5.5 弱监督语义分割

上述实验在实例分割上评测了所提出的 LIID，而与之高度相关的另一个挑战性任务是仅在图像级监督下的弱监督语义分割。语义分割可以看作是一个逐像素分类问题，其中每个像素都分配有类别标签。与实例分割不同，语义分割不需要识别具有相同类别的物体。对于训练图像，在每张图像中合并具有相同语义类别的实例分割蒙版。然后，将产生的语义分割视为伪真值，并采用与以前的方法[162, 173, 242, 257, 259] 相同的设置来训练 DeepLab[38] 模型。

表5.11中展示了在 PASCAL VOC2012 的 segmentation val 和 test 数据集[9] 上与最近的方法[42, 43, 149, 150, 152, 154, 156, 160–164, 166, 168, 169, 172, 173, 240–242, 254–260] 用平均重叠率（mean Intersection-over-Union, mIoU）进行比较。为了公平起见，如果原始论文提供了，这里将以 ResNet101[4] 作为骨干网络报告这些方法的结果（最近的方法通常报告 ResNet101 的结果）。除了 10K VOC2012 训练图像以外，某些方法[152, 164, 168, 169, 241, 258] 还使用了额外的训练数据，例如网络抓取的图像[168, 169, 258]、网络抓取的视频[164, 241] 和像素级标签[152]，以提高性能，这已在表5.11中进行了标记。这里提供两种形式的 LIID：一种没有额外的训练数据，另一种在 ImageNet 的简单子集[170] 上进行了预训练。ImageNet 的简单子集[170] 从 ImageNet 数据集[5] 中选择与 PASCAL VOC 具有相同类别的 24K 图像。无论有没有额外的数据，LIID 的表现都优于所有最近的方法。与同时为实例分割和语

表 5.11 在 PASCAL VOC2012 的 segmentation val 和 test 数据集[9] 上对弱监督语义分割的比较。除了 10K VOC2012 训练图像外，某些方法还使用额外的数据进行训练。24K ImageNet 表示[170] 中的简单 ImageNet 图像。4.6K Videos 来自 Web-Crawl 数据集[241]，包括 960K 视频帧。除了图像级别的监督之外，半监督方法[149, 150, 152] 还分别使用像素级别的标签、点和涂鸦作为监督。为了进行公平的比较，使用 ResNet101[4] 作为骨干网络（如果原始论文提供的话）报告了各种方法的结果。“†” 表示使用 Res2Net101[243] 作为主干网络的结果。

方法	发表年份	额外训练数据	mIoU (%)	
			val	test
CCNN[254]	ICCV’15	✗	35.3	-
EM-Adapt[255]	ICCV’15	✗	38.2	39.6
MIL[172]	CVPR’15	✗	42.0	-
SEC[163]	ECCV’16	✗	50.7	51.7
AugFeed[166]	ECCV’16	✗	54.3	55.5
Bearman 等人[149]	ECCV’16	Points	49.1	-
ScribbleSup[150]	CVPR’16	Scribbles	63.1	-
STC[168]	PAMI’17	40K Web	49.8	51.2
Roy 等人[240]	CVPR’17	✗	52.8	53.7
Oh 等人[256]	CVPR’17	✗	55.7	56.7
AE-PSL[154]	CVPR’17	✗	55.0	55.7
WebS-i2[169]	CVPR’17	19K Web	53.4	55.3
Hong 等人[241]	CVPR’17	4.6K Videos	58.1	58.7
DCSP[242]	BMVC’17	✗	60.8	61.9
DSRG[162]	CVPR’18	✗	61.4	63.2
MCOF[257]	CVPR’18	✗	60.3	61.2
AffinityNet[161]	CVPR’18	✗	61.7	63.7
Wei 等人[156]	CVPR’18	✗	60.4	60.8
GAIN[152]	CVPR’18	1464 Pixel	60.5	62.1
Shen 等人[258]	CVPR’18	80K Web	63.0	63.9
Fan 等人[173]	ECCV’18	✗	63.6	64.5
Fan 等人[173]	ECCV’18	24K ImageNet	64.5	65.6
Ahn 等人[42]	CVPR’19	✗	63.5	64.8
FickleNet[259]	CVPR’19	✗	64.9	65.3
Label-PEnet[43]	ICCV’19	✗	-	57.2
Lee 等人[164]	ICCV’19	4.6K Videos	66.5	67.4
SSDD[260]	ICCV’19	✗	64.9	65.5
OAA[160]	ICCV’19	✗	65.2	66.4
LIID (Ours)	**-**	✗	**66.5**	**67.5**
LIID (Ours)	**-**	**24K ImageNet**	**67.8**	**68.3**
LIID† (Ours)	**-**	✗	**69.4**	**70.4**

图 5.4 PASCAL VOC2012 的 segmentation val 数据集[9] 上语义分割的定性结果。从上到下：原始图像，真值和 LIID 的预测结果，底部三行重复此顺序。

义分割而设计的 Fan 等人[173] 相比，当都使用 24K ImageNet 的简单图像[170] 作为额外的训练数据时，LIID 比 Fan 等人[173] 在 val 和 test 集合上的 mIoU 分别高 3.3% 和 2.7%。这再次证明了 LIID 相对于 Fan 等人[173] 的改进既不琐碎也不简单。最近的新方法[164] 使用包含 960K 视频帧的 4.6K 视频[241] 作为额外的训练数据，比 LIID 多 40 倍。但是，LIID 的性能仍然比它更好，这证明了 LIID 的优越性。图5.4中展示了一些语义分割结果的示例。结合第5.5.3节和第5.5.4节，可以得出结论：对于弱监督实例分割和语义分割，LIID 均达到了最佳的性能。

第六节 小结与讨论

本章致力于基于图像级监督的弱监督实例分割问题，以解决计算机视觉中的图像理解所面临的标注不足的问题。本章所提出的系统 LIID 开始于一些通用

的 SOP，LIID 用一个 MIL 框架为每个 SOP 同时预测概率分布并提取语义特征向量。然后，LIID 使用获得的信息为所有训练图像构造一个大型知识图。最后，LIID 用一种改进的多路割算法，为每个 SOP 分配一个类别标签，属于背景类别的 SOP 将被视为噪声数据而被删除。因此，LIID 利用了实例、图像和数据集级别的信息来检索 SOP 并为其分配正确的标签。与以前的方法相比，LIID 在弱监督实例分割和语义分割方面都可以实现更好的性能。此外，LIID 对 PASCAL VOC2012 和 MS-COCO 数据集使用相同的超参数，这表明 LIID 的超参数对于不同的数据集具有较好的鲁棒性。将来，作者将尝试将所提出的基于 SOP 的 MIL 框架和多路割表示应用于其他弱监督的视觉任务。

第六章 总结与展望

本文研究工作的核心内容是如何用有限的数据、有限的计算资源、有限的人工标注使机器能够理解无限复杂的真实世界，研究方法是通用的图像属性知识、轻量级卷积神经网络、以及弱监督学习，主要的研究成果包括图像边缘检测方法、图像过分割方法、图像显著性检测方法、似物性采样方法等通用的图像属性知识提取技术，以及基于轻量级卷积神经网络的图像理解方法、基于弱监督学习的图像理解方法。作者将本文的研究成果用于自适应图像语义分割、弱监督实例分割、弱监督语义分割等计算机视觉的高层应用中，取得了很好的效果。

第一节 工作总结

计算机视觉旨在使机器能够像人一样处理和理解现实中的开放式环境，从而辅助人工智能系统做出决策。现有的很多计算机视觉技术已经在典型的实验室场景下（例如公开的测试数据集上）取得了惊人的效果，甚至在某些核心指标上超越了人类标注的平均水平，但是将这些技术应用于现实的开放式环境下仍然具有很大的挑战，现有的技术普遍难以应对复杂多变的现实场景、更多的对象类别等问题。

本文在第一章分析了计算机视觉目前面临的主要挑战，并将其归纳为三点：有限的数据、有限的计算资源、有限的人工标注。通过对问题的分析，作者决定采用“数据不够，知识来凑”的方式，模仿人类视觉系统的先验知识，利用通用的图像属性知识（包括图像边缘、图像过分割、图像显著性、似物性采样等）来解决机器学习中数据有限的问题。通过研究基于轻量级卷积神经网络的图像理解来降低深度卷积神经网络的计算机，从而解决现实条件下计算资源受限的问题。以通用的图像属性知识作为基础，通过研究基于弱监督学习的图像理解，来缓解深度学习对大量带有标注的训练数据的需求，从而解决标注有限的问题。第二章对国内外在相关领域的研究现状进行了介绍，并简单概述了现有的研究工作所面临的问题。

第三章针对通用的图像属性知识提取技术提出了一系列新颖的方法。先在第三章第一节提出了一种基于更丰富卷积特征的图像边缘检测方法，该方法首

次将深度卷积神经网络的所有卷积层生成的特征用于模型预测，是第一个在著名的 BSDS500 数据集[25] 上以实时的速度超越人类标注的边缘检测方法，该边缘检测方法及其利用所有卷积层特征的思想已经被广泛应用于各种计算机视觉任务[1]。然后，在第三章第二节提出了一种基于分层特征选择的图像过分割方法，该方法从较小的图像超像素开始，按层次地合并超像素，逐渐生成大区域的分割。在每一层中，通过学习自适应地选择合适地手工设计的特征并赋予其权重。所提出的方法尽量使用可以在 GPU 上实现的并行特征来加快运行速度，使得最终的系统能够高速地进行图像过分割。和语义分割不同，图像过分割的每个区域没有固定的标签，因而无法直接应用传统的深度学习技术。为此，又提出了一种基于深度嵌入学习的图像过分割方法，该方法使用所提出的深度嵌入学习为每个超像素学习深度特征，以替换上述方法中的手工设计的特征，从而显著提高了性能，并保持了较快的运行速度。接着，在第三章第三节通过对现有的显著性检测技术的分析，在理论和实验上都证明了现存的基于深监督线性融合的显著性检测是非最优的，并进而提出了基于深监督非线性融合的显著性检测方法。只需一个简单的基础网络，所提出的方法就相对于以往的复杂模型提高了显著性检测精度。由于几乎所有最近性能最优的显著性检测方法都是基于深监督线性融合的，所提出的深监督非线性融合的概念对显著性检测具有重要意义。最后，在第三章第四节提出了一种用深度卷积神经网络来精炼传统方法生成的物体推荐，以得到少量且高质量的物体推荐。传统方法通过巧妙的算法设计，可以对图像进行密集的似物性采样，但是由于缺少高层的语义信息，生成的物体推荐数量往往太多，其中包含着大量的噪声。所提出的方法结合了传统的似物性采样方法和深度卷积神经网络的优势，用卷积神经网络对传统方法生成的物体推荐进行精炼，得到少量且高质量的物体推荐。

第四章提出了一种用于语义分割的轻量级卷积神经网络。与以往用于图像分类的轻量级卷积神经网络不同，本文利用语义分割高度依赖于多尺度特征的特点，设计了一种进行多尺度学习的轻量级卷积神经网络，该方法以极少的参数、极快的速度获得了可观的性能。与以往基于轻量级神经网络的语义分割方法相比，所提出的方法的设计思想更为简洁，效果却也更好。

第五章以图像边缘、图像过分割、图像显著性、似物性采样等通用的图像属性知识为基础，提出了一种基于弱监督学习的实例分割和语义分割方法。所提

[1]论文发表三年，谷歌学术显示其已获得 380 余次引用，并被应用于我国最大的风电设备企业金风科技的实时风力发电机监测以及国际知名数据公司 SuperAnnotate。

出的方法仅使用图像级别的标签，设计了一个多实例学习框架为似物性采样生成的每个物体推荐学习类别概率和语义特征，并用提取的类别概率和语义特征为整个训练数据集建立一个大图，该图的每个结点代表一个物体推荐。通过将类别看作终端结点，构建了图的多路割问题，将图进行切割，以删除物体推荐中的噪声，并为剩下的物体推荐赋予类别标签。将得到的结果作为伪真值，即可用于训练强监督的多实例分割模型。若忽略每张图像中同类别不同个体的差别，即可训练强监督的语义分割模型。最终，所提出的弱监督实例分割和弱监督语义分割均取得了当前最好的结果，这也表明本文基本达到了预定的研究目标。

第二节 未来工作的展望

从自然图像中提取通用的图像属性知识，并根据其对图像进行自适应的分析和理解，是解决计算机视觉面临的数据有限、资源受限、标注有限的挑战，使机器能够像人一样对开放式环境进行理解的重要步骤。这些通用属性知识的获取将为快速智能的图像识别和理解提供重要的基础。在这个大方向上，本文仅仅只是个开始，还有很多课题有待更深入的研究。这里仅列出其中几个大的方面：

1. 第三章利用所有卷积层的特征进行图像边缘检测，在公开指标上获得了较好的效果。但是，所获得的图像边缘仍然较粗，如何在不损害性能的情况下预测更细的边缘，将是一个很有意义的话题。
2. 第三章所提出的分别基于分层特征选择和基于深度嵌入学习的图像过分割方法以非常小的超像素作为输入，然后进行合并，从而加快了处理速度。超像素也蕴含了比单独的像素点更丰富的信息，因而所提出的方法也获得了可观的性能。但是，这也意味着所提出的方法将受限于所使用的输入的图像超像素。如何避免这种限制，并在不牺牲速度的情况下提升精度，将是作者下一阶段对图像过分割的研究目标。
3. 第三章从理论和实验上分析了现有的显著性物体检测方法中常用的深监督线性融合对显著性检测是非最优的，并进而提出了一种基于深监督非线性融合的简单网络，所提出的简单的方法就获得了较好的性能。如何结合现有方法中某些巧妙的设计与所提出的深监督非线性融合以进一步提升检测性能是一个值得深入研究和探索的问题。
4. 第四章根据语义分割依赖于多尺度特征学习的特点，设计了一种基于多尺度学习的轻量级网络，所提出的轻量网络在参数量、精度、速度、计算量

等方面取得了比已有方法更好的权衡。但是，所提出轻量网络的精度仍然与大规模、高精度的方法存在一定的差距，如何进一步提高轻量级语义分割的精度，从而提升在应用中的体验，将是一个非常值得进一步研究的方向。

参考文献

[1] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. ImageNet classification with deep convolutional neural networks. [C] // Neural Information Processing Systems, 2012: 1097–1105.

[2] SZEGEDY C, LIU W, JIA Y, et al. Going deeper with convolutions. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2015: 1–9.

[3] SIMONYAN K, ZISSERMAN A. Very deep convolutional networks for large-scale image recognition. [C] // International Conference on Learning Representations, 2015.

[4] HE K, ZHANG X, REN S, et al. Deep residual learning for image recognition. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2016: 770–778.

[5] RUSSAKOVSKY O, DENG J, SU H, et al. ImageNet large scale visual recognition challenge. [J]. International Journal on Computer Vision, 2015, 115 (3): 211–252.

[6] LIN T.-Y, MAIRE M, BELONGIE S, et al. Microsoft COCO: Common objects in context. [C] // European Conference on Computer Vision, 2014: 740–755.

[7] CORDTS M, OMRAN M, RAMOS S, et al. The Cityscapes dataset for semantic urban scene understanding. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2016: 3213–3223.

[8] NEUHOLD G, OLLMANN T, ROTA BULO S, et al. The Mapillary Vistas dataset for semantic understanding of street scenes. [C] // IEEE International Conference on Computer Vision, 2017: 4990–4999.

[9] EVERINGHAM M, ESLAMI S A, VAN GOOL L, et al. The PASCAL visual object classes challenge: A retrospective. [J]. International Journal on Computer Vision, 2015, 111 (1): 98–136.

[10] HE K, ZHANG X, REN S, et al. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. [C] // IEEE International Conference on Computer Vision, 2015: 1026–1034.

[11] HE K, GKIOXARI G, DOLLÁR P, et al. Mask R-CNN. [C] // IEEE International Conference on Computer Vision, 2017: 2961–2969.

[12] LIN T.-Y, GOYAL P, GIRSHICK R, et al. Focal loss for dense object detection. [J]. IEEE Transactions on Pattern Recognition and Machine Intelligence, 2020, 42 (2): 318–327.

[13] ZHAO H, SHI J, QI X, et al. Pyramid scene parsing network. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2017: 2881–2890.

[14] NIH. How to keep your sight for life. [J]. MedlinePlus, 2008, 3 (3): 12.

[15] FELLEMAN D J, VAN ESSEN D C. Distributed hierarchical processing in the primate cerebral cortex. [C] // Cereb cortex. Citeseer, 1991.

[16] FERGUS R, PERONA P, ZISSERMAN A. Object class recognition by unsupervised scale-invariant learning. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2003: 264–271.

[17] HOCHSTEIN S, AHISSAR M. View from the top: Hierarchies and reverse hierarchies in the visual system. [J]. Neuron, 2002, 36 (5): 791–804.

[18] DICARLO J J, ZOCCOLAN D, RUST N C. How does the brain solve visual object recognition? [J]. Neuron, 2012, 73 (3): 415–434.

[19] SERRE T. Hierarchical models of the visual system. [J]. Encyclopedia of Computational Neuroscience, 2015: 1309–1318.

[20] TREISMAN A M, GELADE G. A feature-integration theory of attention. [J]. Cognitive Psychology, 1980, 12 (1): 97–136.

[21] SHEN W, WANG X, WANG Y, et al. DeepContour: A deep convolutional feature learned by positive-sharing loss for contour detection. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2015: 3982–3991.

[22] HWANG J.-J, LIU T.-L. Pixel-wise deep learning for contour detection. [J]. ArXiv preprint arXiv:1504.01989, 2015.

[23] XIE S, TU Z. Holistically-nested edge detection. [J]. International Journal on Computer Vision, 2017, 125 (1-3): 3–18.

[24] FELZENSZWALB P F, HUTTENLOCHER D P. Efficient graph-based image segmentation. [J]. International Journal on Computer Vision, 2004, 59 (2): 167–181.

[25] ARBELÁEZ P, MAIRE M, FOWLKES C, et al. Contour detection and hierarchical image segmentation. [J]. IEEE Transactions on Pattern Recognition and Machine Intelligence, 2011, 33 (5): 898–916.

[26] PONT-TUSET J, ARBELAEZ P, BARRON J T, et al. Multiscale combinatorial grouping for image segmentation and object proposal generation. [J]. IEEE Transactions on Pattern Recognition and Machine Intelligence, 2017, 39 (1): 128–140.

[27] CHENG M.-M, MITRA N J, HUANG X, et al. Global contrast based salient region detection. [J]. IEEE Transactions on Pattern Recognition and Machine Intelligence, 2015, 37 (3): 569–582.

[28] JIANG H, WANG J, YUAN Z, et al. Salient object detection: A discriminative regional feature integration approach. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2013: 2083–2090.

[29] LIU N, HAN J, YANG M.-H. PiCANet: Learning pixel-wise contextual attention for saliency detection. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2018: 3089–3098.

[30] UIJLINGS J R, van de SANDE K E, GEVERS T, et al. Selective search for object recognition. [J]. International Journal on Computer Vision, 2013, 104 (2): 154–171.

[31] ZITNICK C L, DOLLÁR P. Edge Boxes: Locating object proposals from edges. [C] // European Conference on Computer Vision, 2014: 391–405.

[32] HOWARD A G, ZHU M, CHEN B, et al. MobileNets: Efficient convolutional neural networks for mobile vision applications. [J]. ArXiv preprint arXiv:1704.04861, 2017.

[33] SANDLER M, HOWARD A, ZHU M, et al. MobileNetV2: Inverted residuals and linear bottlenecks. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2018: 4510–4520.

[34] MA N, ZHANG X, ZHENG H.-T, et al. ShuffleNet v2: Practical guidelines for efficient CNN architecture design. [C] // European Conference on Computer Vision, 2018: 116–131.

[35] POUDEL R P, BONDE U, LIWICKI S, et al. ContextNet: Exploring context and detail for semantic segmentation in real-time. [C] // British Machine Vision Conference, 2018.

[36] MEHTA S, RASTEGARI M, CASPI A, et al. ESPNet: Efficient spatial pyramid of dilated convolutions for semantic segmentation. [C] // European Conference on Computer Vision, 2018: 552–568.

[37] MEHTA S, RASTEGARI M, SHAPIRO L, et al. ESPNetv2: A light-weight, power efficient, and general purpose convolutional neural network. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2019: 9190–9200.

[38] CHEN L.-C, PAPANDREOU G, KOKKINOS I, et al. DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. [J]. IEEE Transactions on Pattern Recognition and Machine Intelligence, 2018, 40 (4): 834–848.

[39] ZHOU Y, ZHU Y, YE Q, et al. Weakly supervised instance segmentation using class peak response. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2018: 3791–3800.

[40] ZHU Y, ZHOU Y, XU H, et al. Learning instance activation maps for weakly supervised instance segmentation. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2019: 3116–3125.

[41] CHOLAKKAL H, SUN G, KHAN F S, et al. Object counting and instance segmentation with image-level supervision. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2019: 12397–12405.

[42] AHN J, CHO S, KWAK S. Weakly supervised learning of instance segmentation with inter-pixel relations. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2019: 2209–2218.

[43] GE W, GUO S, HUANG W, et al. Label-PEnet: Sequential label propagation and enhancement networks for weakly supervised instance segmentation. [C] // IEEE International Conference on Computer Vision, 2019: 3345–3354.

[44] ROBINSON G S. Color edge detection. [J]. Optical Engineering, 1977, 16 (5): 165479–165479.

[45] SOBEL I. Camera models and machine perception. [R]. Stanford Univ Calif Dept of Computer Science, 1970.

[46] CANNY J. A computational approach to edge detection. [J]. IEEE Transactions on Pattern Recognition and Machine Intelligence, 1986, 8 (6): 679–698.

[47] DOLLÁR P, TU Z, BELONGIE S. Supervised learning of edges and object boundaries. [C] // IEEE Conference on Computer Vision and Pattern Recognition. Vol. 2, 2006: 1964–1971.

[48] REN X. Multi-scale improves boundary detection in natural images. [C] // European Conference on Computer Vision, 2008: 533–545.

[49] KONISHI S, YUILLE A L, COUGHLAN J M, et al. Statistical edge detection: Learning and evaluating edge cues. [J]. IEEE Transactions on Pattern Recognition and Machine Intelligence, 2003, 25 (1): 57–74.

[50] MARTIN D R, FOWLKES C C, MALIK J. Learning to detect natural image boundaries using local brightness, color, and texture cues. [J]. IEEE Transactions on Pattern Recognition and Machine Intelligence, 2004, 26 (5): 530–549.

[51] SHI J, MALIK J. Normalized cuts and image segmentation. [J]. IEEE Transactions on Pattern Recognition and Machine Intelligence, 2000, 22 (8): 888–905.

[52] LIM J J, ZITNICK C L, DOLLÁR P. Sketch tokens: A learned mid-level representation for contour and object detection. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2013: 3158–3165.

[53] DOLLÁR P, ZITNICK C L. Fast edge detection using structured forests. [J]. IEEE Transactions on Pattern Recognition and Machine Intelligence, 2015, 37 (8): 1558–1570.

[54] GANIN Y, LEMPITSKY V. N^4-Fields: Neural network nearest neighbor fields for image transforms. [C] // Asian Conference on Computer Vision, 2014: 536–551.

[55] LI Y, PALURI M, REHG J M, et al. Unsupervised learning of edges. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2016: 1619–1627.

[56] ALPERT S, GALUN M, BRANDT A, et al. Image segmentation by probabilistic bottom-up aggregation and cue integration. [J]. IEEE Transactions on Pattern Recognition and Machine Intelligence, 2012, 34 (2): 315–327.

[57] MAIRE M, YU S X. Progressive multigrid eigensolvers for multiscale spectral segmentation. [C] // IEEE International Conference on Computer Vision, 2013: 2184–2191.

[58] TAYLOR C J. Towards fast and accurate segmentation. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2013: 1916–1922.

[59] ACHANTA R, SHAJI A, SMITH K, et al. SLIC superpixels compared to state-of-the-art superpixel methods. [J]. IEEE Transactions on Pattern Recognition and Machine Intelligence, 2012, 34 (11): 2274–2282.

[60] CHENG J, LIU J, XU Y, et al. Superpixel classification based optic disc and optic cup segmentation for glaucoma screening. [J]. IEEE Transactions on Medical Imaging, 2013, 32 (6): 1019–1032.

[61] ZHANG L, GAO Y, XIA Y, et al. Representative discovery of structure cues for weakly-supervised image segmentation. [J]. IEEE Transactions on Multimedia, 2014, 16 (2): 470–479.

[62] CHANG J, WEI D, FISHER J W. A video representation using temporal superpixels. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2013: 2051–2058.

[63] SUN J, PONCE J. Learning discriminative part detectors for image classification and cosegmentation. [C] // IEEE International Conference on Computer Vision, 2013: 3400–3407.

[64] REN Z, SHAKHNAROVICH G. Image segmentation by cascaded region agglomeration. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2013: 2011–2018.

[65] GONG C, TAO D, LIU W, et al. Saliency propagation from simple to difficult. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2015: 2531–2539.

[66] YANG C, ZHANG L, LU H, et al. Saliency detection via graph-based manifold ranking. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2013: 3166–3173.

[67] WANG L, LU H, RUAN X, et al. Deep networks for saliency detection via local estimation and global search. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2015: 3183–3192.

[68] LI G, YU Y. Visual saliency based on multiscale deep features. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2015: 5455–5463.

[69] LEE G, TAI Y.-W, KIM J. Deep saliency with encoded low level distance map and high level features. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2016: 660–668.

[70] CHEN S, TAN X, WANG B, et al. Reverse attention for salient object detection. [C] // European Conference on Computer Vision, 2018: 234–250.

[71] ZHANG L, DAI J, LU H, et al. A bi-directional message passing model for salient object detection. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2018: 1741–1750.

[72] WANG T, ZHANG L, WANG S, et al. Detect globally, refine locally: A novel approach to saliency detection. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2018: 3127–3135.

[73] ISLAM M A, KALASH M, BRUCE N D. Revisiting salient object detection: Simultaneous detection, ranking, and subitizing of multiple salient objects. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2018: 7142–7150.

[74] ZHANG P, WANG D, LU H, et al. Amulet: Aggregating multi-level convolutional features for salient object detection. [C] // IEEE International Conference on Computer Vision, 2017: 202–211.

[75] WANG T, BORJI A, ZHANG L, et al. A stagewise refinement model for detecting salient objects in images. [C] // IEEE International Conference on Computer Vision, 2017: 4019–4028.

[76] HOU Q, CHENG M.-M, HU X, et al. Deeply supervised salient object detection with short connections. [J]. IEEE Transactions on Pattern Recognition and Machine Intelligence, 2019, 41 (4): 815–828.

[77] LI G, YU Y. Deep contrast learning for salient object detection. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2016: 478–487.

[78] LUO Z, MISHRA A K, ACHKAR A, et al. Non-local deep features for salient object detection. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2017: 6609–6617.

[79] LIU N, HAN J. DHSNet: Deep hierarchical saliency network for salient object detection. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2016: 678–686.

[80] WU Z, SU L, HUANG Q. Cascaded partial decoder for fast and accurate salient object detection. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2019: 3907–3916.

[81] ZHANG L, ZHANG J, LIN Z, et al. CapSal: Leveraging captioning to boost semantics for salient object detection. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2019: 6024–6033.

[82] HARIHARAN B, ARBELÁEZ P, GIRSHICK R, et al. Hypercolumns for object segmentation and fine-grained localization. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2015: 447–456.

[83] SHELHAMER E, LONG J, DARRELL T. Fully convolutional networks for semantic segmentation. [J]. IEEE Transactions on Pattern Recognition and Machine Intelligence, 2017, 39 (4): 640–651.

[84] RONNEBERGER O, FISCHER P, BROX T. U-Net: Convolutional networks for biomedical image segmentation. [C] // Medical Image Computing and Computer Assisted Intervention, 2015: 234–241.

[85] WANG L, WANG L, LU H, et al. Saliency detection with recurrent fully convolutional networks. [C] // European Conference on Computer Vision, 2016: 825–841.

[86] ZHANG P, WANG D, LU H, et al. Learning uncertain convolutional features for accurate saliency detection. [C] // IEEE International Conference on Computer Vision, 2017: 212–221.

[87] HU P, SHUAI B, LIU J, et al. Deep level sets for salient object detection. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2017: 2300–2309.

[88] LI X, YANG F, CHENG H, et al. Contour knowledge transfer for salient object detection. [C] // European Conference on Computer Vision, 2018: 355–370.

[89] LEE C.-Y, XIE S, GALLAGHER P, et al. Deeply-supervised nets. [C] // Artificial Intelligence and Statistics, 2015: 562–570.

[90] WANG W, ZHAO S, SHEN J, et al. Salient object detection with pyramid attention and salient edges. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2019: 1448–1457.

[91] FENG M, LU H, DING E. Attentive feedback network for boundary-aware salient object detection. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2019: 1623–1632.

[92] LIU J.-J, HOU Q, CHENG M.-M, et al. A simple pooling-based design for real-time salient object detection. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2019: 3917–3926.

[93] WU R, FENG M, GUAN W, et al. A mutual learning method for salient object detection with intertwined multi-supervision. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2019: 8150–8159.

[94] BYLINSKII Z, JUDD T, OLIVA A, et al. What do different evaluation metrics tell us about saliency models? [J]. IEEE Transactions on Pattern Recognition and Machine Intelligence, 2018, 41 (3): 740–757.

[95] CONG R, LEI J, FU H, et al. Review of visual saliency detection with comprehensive information. [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2019, 29 (10): 2941–2959.

[96] LOWE D G. Distinctive image features from scale-invariant keypoints. [J]. International Journal on Computer Vision, 2004, 60 (2): 91–110.

[97] MANEN S, GUILLAUMIN M, VAN GOOL L. Prime object proposals with randomized Prim's algorithm. [C] // IEEE International Conference on Computer Vision, 2013: 2536–2543.

[98] RANTALANKILA P, KANNALA J, RAHTU E. Generating object segmentation proposals using global and local search. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2014: 2417–2424.

[99] RAHTU E, KANNALA J, BLASCHKO M. Learning a category independent object detection cascade. [C] // IEEE International Conference on Computer Vision, 2011: 1052–1059.

[100] ENDRES I, HOIEM D. Category-independent object proposals with diverse ranking. [J]. IEEE Transactions on Pattern Recognition and Machine Intelligence, 2014, 36 (2): 222–234.

[101] KRAHENBUHL P, KOLTUN V. Learning to propose objects. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2015: 1574–1582.

[102] ALEXE B, DESELAERS T, FERRARI V. Measuring the objectness of image windows. [J]. IEEE Transactions on Pattern Recognition and Machine Intelligence, 2012, 34 (11): 2189–2202.

[103] ZHANG Z, TORR P H. Object proposal generation using two-stage cascade SVMs. [J]. IEEE Transactions on Pattern Recognition and Machine Intelligence, 2016, 38 (1): 102–115.

[104] CHENG M.-M, LIU Y, LIN W.-Y, et al. BING: Binarized normed gradients for objectness estimation at 300fps. [J]. Computational Visual Media, 2019, 5 (1): 3–20.

[105] LU C, LIU S, JIA J, et al. Contour box: Rejecting object proposals without explicit closed contours. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2015: 2021–2029.

[106] REN S, HE K, GIRSHICK R, et al. Faster R-CNN: Towards real-time object detection with region proposal networks. [C] // Neural Information Processing Systems, 2015: 91–99.

[107] PINHEIRO P O, COLLOBERT R, DOLLÁR P. Learning to segment object candidates. [C] // Neural Information Processing Systems, 2015: 1990–1998.

[108] PINHEIRO P O, LIN T.-Y, COLLOBERT R, et al. Learning to refine object segments. [C] // European Conference on Computer Vision, 2016: 75–91.

[109] KUO W, HARIHARAN B, MALIK J. DeepBox: Learning objectness with convolutional networks. [C] // IEEE International Conference on Computer Vision, 2015: 2479–2487.

[110] CHEN X, MA H, WANG X, et al. Improving object proposals with multi-thresholding straddling expansion. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2015: 2587–2595.

[111] ZHANG Z, LIU Y, CHEN X, et al. Sequential optimization for efficient high-quality object proposal generation. [J]. IEEE Transactions on Pattern Recognition and Machine Intelligence, 2018, 40 (5): 1209–1223.

[112] HE S, LAU R W. Oriented object proposals. [C] // IEEE International Conference on Computer Vision, 2015: 280–288.

[113] NOH H, HONG S, HAN B. Learning deconvolution network for semantic segmentation. [C] // IEEE International Conference on Computer Vision, 2015: 1520–1528.

[114] BADRINARAYANAN V, KENDALL A, CIPOLLA R. SegNet: A deep convolutional encoder-decoder architecture for image segmentation. [J]. IEEE Transactions on Pattern Recognition and Machine Intelligence, 2017, 39 (12): 2481–2495.

[115] CHEN L.-C, YANG Y, WANG J, et al. Attention to scale: Scale-aware semantic image segmentation. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2016: 3640–3649.

[116] LIN G, MILAN A, SHEN C, et al. RefineNet: Multi-path refinement networks for high-resolution semantic segmentation. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2017: 1925–1934.

[117] XIA F, WANG P, CHEN L.-C, et al. Zoom better to see clearer: Human and object parsing with hierarchical auto-zoom net. [C] // European Conference on Computer Vision, 2016: 648–663.

[118] CHEN L.-C, PAPANDREOU G, KOKKINOS I, et al. Semantic image segmentation with deep convolutional nets and fully connected CRFs. [C] // International Conference on Learning Representations, 2015.

[119] CHEN L.-C, PAPANDREOU G, SCHROFF F, et al. Rethinking atrous convolution for semantic image segmentation. [J]. ArXiv preprint arXiv:1706.05587, 2017.

[120] CHEN L.-C, ZHU Y, PAPANDREOU G, et al. Encoder-decoder with atrous separable convolution for semantic image segmentation. [C] // European Conference on Computer Vision, 2018: 801–818.

[121] YANG M, YU K, ZHANG C, et al. DenseASPP for semantic segmentation in street scenes. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2018: 3684–3692.

[122] ZHANG H, DANA K, SHI J, et al. Context encoding for semantic segmentation. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2018: 7151–7160.

[123] HUANG Z, WANG X, HUANG L, et al. CCNet: Criss-cross attention for semantic segmentation. [C] // IEEE International Conference on Computer Vision, 2019: 603–612.

[124] ZHU Z, XU M, BAI S, et al. Asymmetric non-local neural networks for semantic segmentation. [C] // IEEE International Conference on Computer Vision, 2019: 593–602.

[125] WU Z, SHEN C, HENGEL A V D. Wider or deeper: Revisiting the resnet model for visual recognition. [J]. Pattern Recognition, 2019, 90: 119–133.

[126] YU C, WANG J, PENG C, et al. Learning a discriminative feature network for semantic segmentation. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2018: 1857–1866.

[127] LIU Z, LI X, LUO P, et al. Semantic image segmentation via deep parsing network. [C] // IEEE International Conference on Computer Vision, 2015: 1377–1385.

[128] ZHENG S, JAYASUMANA S, ROMERA-PAREDES B, et al. Conditional random fields as recurrent neural networks. [C] // IEEE International Conference on Computer Vision, 2015: 1529–1537.

[129] ROMERA E, ALVAREZ J M, BERGASA L M, et al. ERFNet: Efficient residual factorized convnet for real-time semantic segmentation. [J]. IEEE Transactions on Intelligent Transportation Systems, 2018, 19 (1): 263–272.

[130] ZHAO H, QI X, SHEN X, et al. ICNet for real-time semantic segmentation on high-resolution images. [C] // European Conference on Computer Vision, 2018: 405–420.

[131] YU C, WANG J, PENG C, et al. BiSeNet: Bilateral segmentation network for real-time semantic segmentation. [C] // European Conference on Computer Vision, 2018: 325–341.

[132] TREML M, ARJONA-MEDINA J, et al. Speeding up semantic segmentation for autonomous driving. [C] // MLITS, Neural Information Processing Systems Workshop, 2016.

[133] POHLEN T, HERMANS A, MATHIAS M, et al. Full-resolution residual networks for semantic segmentation in street scenes. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2017: 4151–4160.

[134] LO S.-Y, HANG H.-M, CHAN S.-W, et al. Efficient dense modules of asymmetric convolution for real-time semantic segmentation. [J]. ArXiv preprint arXiv:1809.06323, 2018.

[135] HARIHARAN B, ARBELÁEZ P, GIRSHICK R, et al. Simultaneous detection and segmentation. [C] // European Conference on Computer Vision, 2014: 297–312.

[136] DAI J, HE K, SUN J. Instance-aware semantic segmentation via multi-task network cascades. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2016: 3150–3158.

[137] LIU S, QI L, QIN H, et al. Path aggregation network for instance segmentation. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2018: 8759–8768.

[138] CHEN K, PANG J, WANG J, et al. Hybrid task cascade for instance segmentation. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2019: 4974–4983.

[139] ARNAB A, TORR P H. Pixelwise instance segmentation with a dynamically instantiated network. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2017: 441–450.

[140] BAI M, URTASUN R. Deep watershed transform for instance segmentation. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2017: 2858–2866.

[141] KIRILLOV A, LEVINKOV E, ANDRES B, et al. InstanceCut: from edges to instances with multicut. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2017: 5008–5017.

[142] KHOREVA A, BENENSON R, HOSANG J H, et al. Simple does it: Weakly supervised instance and semantic segmentation. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2017: 876–885.

[143] ROTHER C, KOLMOGOROV V, BLAKE A. GrabCut: Interactive foreground extraction using iterated graph cuts. [J]. ACM Transactions on Graphics, 2004, 23 (3): 309–314.

[144] LI Q, ARNAB A, TORR P H. Weakly- and semi-supervised panoptic segmentation. [C] // European Conference on Computer Vision, 2018: 106–124.

[145] HSU C.-C, HSU K.-J, TSAI C.-C, et al. Weakly supervised instance segmentation using the bounding box tightness prior. [C] // Neural Information Processing Systems, 2019: 6582–6593.

[146] DIETTERICH T G, LATHROP R H, LOZANO-PÉREZ T. Solving the multiple instance problem with axis-parallel rectangles. [J]. Artificial Intelligence, 1997, 89 (1): 31–71.

[147] HU R, DOLLÁR P, HE K, et al. Learning to segment every thing. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2018: 4233–4241.

[148] ZHOU B, KHOSLA A, LAPEDRIZA A, et al. Learning deep features for discriminative localization. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2016: 2921–2929.

[149] BEARMAN A, RUSSAKOVSKY O, FERRARI V, et al. What's the point: Semantic segmentation with point supervision. [C] // European Conference on Computer Vision, 2016: 549–565.

[150] LIN D, DAI J, JIA J, et al. ScribbleSup: Scribble-supervised convolutional networks for semantic segmentation. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2016: 3159–3167.

[151] SONG C, HUANG Y, OUYANG W, et al. Box-driven class-wise region masking and filling rate guided loss for weakly supervised semantic segmentation. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2019: 3136–3145.

[152] LI K, WU Z, PENG K.-C, et al. Tell me where to look: Guided attention inference network. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2018: 9215–9223.

[153] SINGH K K, LEE Y J. Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization. [C] // IEEE International Conference on Computer Vision, 2017: 3544–3553.

[154] WEI Y, FENG J, LIANG X, et al. Object region mining with adversarial erasing: A simple classification to semantic segmentation approach. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2017: 1568–1576.

[155] KIM D, CHO D, YOO D, et al. Two-phase learning for weakly supervised object localization. [C] // IEEE International Conference on Computer Vision, 2017: 3534–3543.

[156] WEI Y, XIAO H, SHI H, et al. Revisiting dilated convolution: A simple approach for weakly- and semi-supervised semantic segmentation. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2018: 7268–7277.

[157] ZHANG X, WEI Y, FENG J, et al. Adversarial complementary learning for weakly supervised object localization. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2018: 1325–1334.

[158] DURAND T, MORDAN T, THOME N, et al. WILDCAT: Weakly supervised learning of deep convnets for image classification, pointwise localization and segmentation. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2017: 642–651.

[159] DURAND T, THOME N, CORD M. Exploiting negative evidence for deep latent structured models. [J]. IEEE Transactions on Pattern Recognition and Machine Intelligence, 2018, 41 (2): 337–351.

[160] JIANG P.-T, HOU Q, CAO Y, et al. Integral object mining via online attention accumulation. [C] // IEEE International Conference on Computer Vision, 2019: 2070–2079.

[161] AHN J, KWAK S. Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2018: 4981–4990.

[162] HUANG Z, WANG X, WANG J, et al. Weakly-supervised semantic segmentation network with deep seeded region growing. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2018: 7014–7023.

[163] KOLESNIKOV A, LAMPERT C H. Seed, expand and constrain: Three principles for weakly-supervised image segmentation. [C] // European Conference on Computer Vision, 2016: 695–711.

[164] LEE J, KIM E, LEE S, et al. Frame-to-frame aggregation of active regions in web videos for weakly supervised semantic segmentation. [C] // IEEE International Conference on Computer Vision, 2019: 6808–6818.

[165] CHENG M.-M, MITRA N J, HUANG X, et al. SalientShape: Group saliency in image collections. [J]. The Visual Computer, 2014, 30 (4): 443–453.

[166] QI X, LIU Z, SHI J, et al. Augmented feedback in semantic segmentation under image level supervision. [C] // European Conference on Computer Vision, 2016: 90–105.

[167] SHIMODA W, YANAI K. Distinct class-specific saliency maps for weakly supervised semantic segmentation. [C] // European Conference on Computer Vision, 2016: 218–234.

[168] WEI Y, LIANG X, CHEN Y, et al. STC: A simple to complex framework for weakly-supervised semantic segmentation. [J]. IEEE Transactions on Pattern Recognition and Machine Intelligence, 2017, 39 (11): 2314–2320.

[169] JIN B, SEGOVIA M V O, SÜSSTRUNK S. Webly supervised semantic segmentation. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2017: 1705–1714.

[170] HOU Q, MASSICETI D, DOKANIA P K, et al. Bottom-up top-down cues for weakly-supervised semantic segmentation. [C] // International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR), 2017: 263–277.

[171] SALEH F, ALIAKBARIAN M S, SALZMANN M, et al. Built-in foreground/background prior for weakly-supervised semantic segmentation. [C] // European Conference on Computer Vision, 2016: 413–432.

[172] PINHEIRO P O, COLLOBERT R. From image-level to pixel-level labeling with convolutional networks. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2015: 1713–1721.

[173] FAN R, HOU Q, CHENG M.-M, et al. Associating inter-image salient instances for weakly supervised semantic segmentation. [C] // European Conference on Computer Vision, 2018: 371–388.

[174] ULLMAN S, BASRI R. Recognition by linear combinations of models. [J]. IEEE Transactions on Pattern Recognition and Machine Intelligence, 1991, 13 (10): 992–1006.

[175] FERRARI V, FEVRIER L, JURIE F, et al. Groups of adjacent contour segments for object detection. [J]. IEEE Transactions on Pattern Recognition and Machine Intelligence, 2008, 30 (1): 36–51.

[176] ARBELAEZ P, MAIRE M, FOWLKES C, et al. From contours to regions: An empirical evaluation. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2009: 2294–2301.

[177] XIAOFENG R, BO L. Discriminatively trained sparse code gradients for contour detection. [C] // Neural Information Processing Systems, 2012: 584–592.

[178] LEORDEANU M, SUKTHANKAR R, SMINCHISESCU C. Generalized boundaries from multiple image interpretations. [J]. IEEE Transactions on Pattern Recognition and Machine Intelligence, 2014, 36 (7): 1312–1324.

[179] GIRSHICK R. Fast R-CNN. [C] // IEEE International Conference on Computer Vision, 2015: 1440–1448.

[180] BERTASIUS G, SHI J, TORRESANI L. DeepEdge: A multi-scale bifurcated deep network for top-down contour detection. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2015: 4380–4389.

[181] JIA Y, SHELHAMER E, DONAHUE J, et al. Caffe: Convolutional architecture for fast feature embedding. [C] // ACM International Conference on Multimedia, 2014: 675–678.

[182] LIU Y, LEW M S. Learning relaxed deep supervision for better edge detection. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2016: 231–240.

[183] YANG J, PRICE B, COHEN S, et al. Object contour detection with a fully convolutional encoder-decoder network. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2016: 193–202.

[184] KOKKINOS I. Pushing the boundaries of boundary detection using deep learning. [C] // International Conference on Learning Representations, 2016.

[185] MOTTAGHI R, CHEN X, LIU X, et al. The role of context for object detection and semantic segmentation in the wild. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2014: 891–898.

[186] COMANICIU D, MEER P. Mean shift: A robust approach toward feature space analysis. [J]. IEEE Transactions on Pattern Recognition and Machine Intelligence, 2002, 24 (5): 603–619.

[187] HALLMAN S, FOWLKES C C. Oriented edge forests for boundary detection. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2015: 1732–1740.

[188] BERTASIUS G, SHI J, TORRESANI L. High-for-low and low-for-high: Efficient boundary detection from deep object features and its applications to high-level vision. [C] // IEEE International Conference on Computer Vision, 2015: 504–512.

[189] SILBERMAN N, HOIEM D, KOHLI P, et al. Indoor segmentation and support inference from RGBD images. [C] // European Conference on Computer Vision, 2012: 746–760.

[190] GUPTA S, ARBELAEZ P, MALIK J. Perceptual organization and recognition of indoor scenes from RGB-D images. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2013: 564–571.

[191] GUPTA S, GIRSHICK R, ARBELÁEZ P, et al. Learning rich features from RGB-D images for object detection and segmentation. [C] // European Conference on Computer Vision, 2014: 345–360.

[192] JUNEJA M, VEDALDI A, JAWAHAR C, et al. Blocks that shout: Distinctive parts for scene classification. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2013: 923–930.

[193] KOHLI P, TORR P H, et al. Robust higher order potentials for enforcing label consistency. [J]. International Journal on Computer Vision, 2009, 82 (3): 302–324.

[194] WANG S, LU H, YANG F, et al. Superpixel tracking. [C] // IEEE International Conference on Computer Vision, 2011: 1323–1330.

[195] FARABET C, COUPRIE C, NAJMAN L, et al. Learning hierarchical features for scene labeling. [J]. IEEE Transactions on Pattern Recognition and Machine Intelligence, 2013, 35 (8): 1915–1929.

[196] HOIEM D, EFROS A A, HEBERT M. Geometric context from a single image. [C] // IEEE International Conference on Computer Vision, 2005: 654–661.

[197] HU S.-M, ZHANG F.-L, WANG M, et al. PatchNet: A patch-based image representation for interactive library-driven image editing. [J]. ACM Transactions on Graphics, 2013, 32 (6): 196.

[198] LI K, ZHU Y, YANG J, et al. Video super-resolution using an adaptive superpixel-guided auto-regressive model. [J]. Pattern Recognition, 2016, 51: 59–71.

[199] SONG X, ZHANG J, HAN Y, et al. Semi-supervised feature selection via hierarchical regression for web image classification. [J]. Multimedia Systems, 2016, 22 (1): 41–49.

[200] RUSSELL C, KOHLI P, TORR P H, et al. Associative hierarchical crfs for object class image segmentation. [C] // IEEE International Conference on Computer Vision, 2009: 739–746.

[201] CHEN T, CHENG M.-M, TAN P, et al. Sketch2Photo: Internet image montage. [J]. ACM Transactions on Graphics, 2009, 28 (5): 124.

[202] REN C Y, PRISACARIU V A, REID I D. GSLICr: SLIC superpixels at over 250Hz. [J]. ArXiv preprint arXiv:1509.04232, 2015.

[203] STORATH M, WEINMANN A. Fast partitioning of vector-valued images. [J]. SIAM Journal on Imaging Sciences, 2014, 7 (3): 1826–1852.

[204] RIVEST J, CABANAGH P. Localizing contours defined by more than one attribute. [J]. Vision Research, 1996, 36 (1): 53–66.

[205] LIU W, RABINOVICH A, BERG A C. ParseNet: Looking wider to see better. [C] // International Conference on Learning Representations, 2016.

[206] HARIHARAN B, ARBELÁEZ P, BOURDEV L, et al. Semantic contours from inverse detectors. [C] // IEEE International Conference on Computer Vision, 2011: 991–998.

[207] PONT-TUSET J, MARQUES F. Supervised evaluation of image segmentation and object proposal techniques. [J]. IEEE Transactions on Pattern Recognition and Machine Intelligence, 2016, 38 (7): 1465–1478.

[208] COUR T, BENEZIT F, SHI J. Spectral segmentation with multiscale graph decomposition. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2005: 1124–1131.

[209] GAO Y, WANG M, ZHA Z.-J, et al. Visual-textual joint relevance learning for tag-based social image search. [J]. IEEE Transactions on Image Processing, 2013, 22 (1): 363–376.

[210] MAHADEVAN V, VASCONCELOS N. Saliency-based discriminant tracking. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2009.

[211] REN Z, GAO S, CHIA L.-T, et al. Region-based saliency detection and its application in object recognition. [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2014, 24 (5): 769–779.

[212] ZUND F, PRITCH Y, SORKINE-HORNUNG A, et al. Content-aware compression using saliency-driven image retargeting. [C] // IEEE International Conference on Image Processing, 2013: 1845–1849.

[213] WANG W, SHEN J, YU Y, et al. Stereoscopic thumbnail creation via efficient stereo saliency detection. [J]. IEEE Transactions on Visualization and Computer Graphics, 2016, 23 (8): 2014–2027.

[214] WANG W, SHEN J, PORIKLI F. Saliency-aware video object segmentation. [J]. IEEE Transactions on Pattern Recognition and Machine Intelligence, 2018, 40 (1): 20–33.

[215] TONG N, LU H, RUAN X, et al. Salient object detection via bootstrap learning. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2015: 1884–1892.

[216] WANG L, LU H, WANG Y, et al. Learning to detect salient objects with image-level supervision. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2017: 136–145.

[217] YAN Q, XU L, SHI J, et al. Hierarchical saliency detection. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2013: 1155–1162.

[218] MOVAHEDI V, ELDER J H. Design and perceptual validation of performance measures for salient object segmentation. [C] // IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2010: 49–56.

[219] WEI Y, XIA W, LIN M, et al. HCP: A flexible CNN framework for multi-label image classification. [J]. IEEE Transactions on Pattern Recognition and Machine Intelligence, 2016, 38 (9): 1901–1907.

[220] LEE Y J, GRAUMAN K. Predicting important objects for egocentric video summarization. [J]. International Journal on Computer Vision, 2015, 114 (1): 38–55.

[221] WU J, YU Y, HUANG C, et al. Deep multiple instance learning for image classification and auto-annotation. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2015: 3460–3469.

[222] LI D, HUANG J.-B, LI Y, et al. Weakly supervised object localization with progressive domain adaptation. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2016: 3512–3520.

[223] EVERINGHAM M, VAN GOOL L, WILLIAMS C K, et al. The PASCAL Visual Object Classes (VOC) challenge. [J]. International Journal on Computer Vision, 2010, 88 (2): 303–338.

[224] KRÄHENBÜHL P, KOLTUN V. Geodesic object proposals. [C] // European Conference on Computer Vision, 2014: 725–739.

[225] HOSANG J, BENENSON R, DOLLÁR P, et al. What makes for effective detection proposals? [J]. IEEE Transactions on Pattern Recognition and Machine Intelligence, 2015, 38 (4): 814–830.

[226] PASZKE A, CHAURASIA A, KIM S, et al. ENet: A deep neural network architecture for real-time semantic segmentation. [J]. ArXiv preprint arXiv:1606.02147, 2016.

[227] YU F, KOLTUN V. Multi-scale context aggregation by dilated convolutions. [C] // International Conference on Learning Representations, 2016.

[228] YU F, WANG D, SHELHAMER E, et al. Deep layer aggregation. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2018: 2403–2412.

[229] BROSTOW G J, SHOTTON J, FAUQUEUR J, et al. Segmentation and recognition using structure from motion point clouds. [C] // European Conference on Computer Vision, 2008: 44–57.

[230] IOFFE S, SZEGEDY C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. [C] // International Conference on Machine Learning, 2015: 448–456.

[231] CHOLLET F. Xception: Deep learning with depthwise separable convolutions. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2017: 1251–1258.

[232] WANG L, OUYANG W, WANG X, et al. Visual tracking with fully convolutional networks. [C] // IEEE International Conference on Computer Vision, 2015: 3119–3127.

[233] PASZKE A, GROSS S, CHINTALA S, et al. Automatic differentiation in PyTorch. [C] // Neural Information Processing Systems Workshop, 2017.

[234] KINGMA D P, BA J. Adam: A method for stochastic optimization. [C] // International Conference on Learning Representations, 2015.

[235] FU J, LIU J, TIAN H, et al. Dual attention network for scene segmentation. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2019.

[236] NAIR V, HINTON G E. Rectified linear units improve restricted Boltzmann machines. [C] // International Conference on Machine Learning, 2010: 807–814.

[237] WAN F, WEI P, JIAO J, et al. Min-entropy latent model for weakly supervised object detection. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2018: 1297–1306.

[238] ZHANG X, FENG J, XIONG H, et al. Zigzag learning for weakly supervised object detection. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2018: 4262–4270.

[239] SHEN Y, JI R, ZHANG S, et al. Generative adversarial learning towards fast weakly supervised detection. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2018: 5764–5773.

[240] ROY A, TODOROVIC S. Combining bottom-up, top-down, and smoothness cues for weakly supervised image segmentation. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2017: 3529–3538.

[241] HONG S, YEO D, KWAK S, et al. Weakly supervised semantic segmentation using web-crawled videos. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2017: 7322–7330.

[242] CHAUDHRY A, DOKANIA P K, TORR P H. Discovering class-specific pixels for weakly-supervised semantic segmentation. [C] // British Machine Vision Conference, 2017.

[243] GAO S, CHENG M.-M, ZHAO K, et al. Res2Net: A new multi-scale backbone architecture. [J]. IEEE Transactions on Pattern Recognition and Machine Intelligence, 2019.

[244] FAN R, CHENG M.-M, HOU Q, et al. S4Net: Single stage salient-instance segmentation. [J]. Computational Visual Media, 2020, 6 (2): 191–204.

[245] MURPHY K P. Machine learning: A probabilistic perspective. [M]. MIT press, 2012.

[246] DAHLHAUS E, JOHNSON D S, PAPADIMITRIOU C H, et al. The complexity of multiterminal cuts. [J]. SIAM Journal on Computing (SICOMP), 1994, 23 (4): 864–894.

[247] GARG N, VAZIRANI V V, YANNAKAKIS M. Approximate max-flow min-(multi)cut theorems and their applications. [J]. SIAM Journal on Computing, 1996, 25 (2): 235–251.

[248] CĂLINESCU G, KARLOFF H, RABANI Y. An improved approximation algorithm for multiway cut. [J]. Journal of Computer and System Sciences (JCSS), 2000, 60 (3): 564–574.

[249] WEN Y, ZHANG K, LI Z, et al. A discriminative feature learning approach for deep face recognition. [C] // European Conference on Computer Vision, 2016: 499–515.

[250] BLIEK1Ú C, BONAMI P, LODI A. Solving mixed-integer quadratic programming problems with IBM-CPLEX: A progress report. [C] // RAMP Symposium, 2014: 16–17.

[251] ZHU Y, ZHOU Y, YE Q, et al. Soft proposal networks for weakly supervised object localization. [C] // IEEE International Conference on Computer Vision, 2017: 1841–1850.

[252] SHEN Y, JI R, WANG Y, et al. Cyclic guidance for weakly supervised joint detection and segmentation. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2019: 697–707.

[253] LI Y, QI H, DAI J, et al. Fully convolutional instance-aware semantic segmentation. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2017: 2359–2367.

[254] PATHAK D, KRAHENBUHL P, DARRELL T. Constrained convolutional neural networks for weakly supervised segmentation. [C] // IEEE International Conference on Computer Vision, 2015: 1796–1804.

[255] PAPANDREOU G, CHEN L.-C, MURPHY K P, et al. Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation. [C] // IEEE International Conference on Computer Vision, 2015: 1742–1750.

[256] OH S J, BENENSON R, KHOREVA A, et al. Exploiting saliency for object segmentation from image level labels. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2017.

[257] WANG X, YOU S, LI X, et al. Weakly-supervised semantic segmentation by iteratively mining common object features. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2018: 1354–1362.

[258] SHEN T, LIN G, SHEN C, et al. Bootstrapping the performance of webly supervised semantic segmentation. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2018: 1363–1371.

[259] LEE J, KIM E, LEE S, et al. FickleNet: Weakly and semi-supervised semantic image segmentation using stochastic inference. [C] // IEEE Conference on Computer Vision and Pattern Recognition, 2019: 5267–5276.

[260] SHIMODA W, YANAI K. Self-supervised difference detection for weakly-supervised semantic segmentation. [C] // IEEE International Conference on Computer Vision, 2019: 5208–5217.

致谢

衷心感谢我的导师程明明教授在科研上对我的指导和帮助，在我职业选择上的细心规划，并为我树立了为人处世的榜样。程老师多年来在各方面对我的悉心教诲，使我受益良多。感谢新加坡科技研究局的张乐博士、澳大利亚阿德莱德大学的边佳旺同学、德国波恩大学的李仕杰同学，在与他们多年的科研合作中，我学到了很多，也收获了很多。感谢吴宇寰、顾宇超、林铮、梅杰、许刚等实验室的同学，你们给我带来了很多欢乐，使得枯燥的博士生涯不是那么枯燥。感谢我的父母、姐姐、女朋友邱宇一直以来的陪伴、支持和鼓励，你们是我奋斗的动力。最后，感谢所有帮助过我、关心过我的老师、同学、亲人和朋友，谢谢你们！

个人简历　在学期间发表的学术论文与研究成果

个人简历

1994 年 6 月 6 日出生于安徽省蒙城县。

2012 年 9 月考入南开大学计算机学院计算机科学与技术专业，2016 年 6 月本科毕业并获得工学学士学位。

2016 年 9 月免试进入南开大学计算机学院计算机科学与技术专业攻读博士学位至今。

在学期间发表的学术论文与研究成果

[1] Ming-Ming Cheng*, **Yun Liu***, Qibin Hou, Jiawang Bian, Philip Torr, Shi-Min Hu, and Zhuowen Tu. HFS: Hierarchical Feature Selection for Efficient Image Segmentation. European Conference on Computer Vision (ECCV), 2016: 867-882. (CCF B 类会议)

[2] **Yun Liu**, Ming-Ming Cheng, Xiaowei Hu, Kai Wang, and Xiang Bai. Richer Convolutional Features for Edge Detection. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017: 5872-5881. (CCF A 类会议)

[3] **Yun Liu**, Peng-Tao Jiang, Vahan Petrosyan, Shi-Jie Li, Jiawang Bian, Le Zhang, and Ming-Ming Cheng. DEL: Deep Embedding Learning for Efficient Image Segmentation. International Joint Conference on Artificial Intelligence (IJCAI), 2018: 864-870. (CCF A 类会议)

[4] Ziming Zhang, **Yun Liu**, Xi Chen, Yanjun Zhu, Ming-Ming Cheng, Venkatesh Saligrama, and Philip H.S. Torr. Sequential Optimization for Efficient High-Quality Object Proposal Generation. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2018, 40(5): 1209-1223. (SCI 一区、CCF A 类期刊)

[5] **Yun Liu**, Ming-Ming Cheng, Xiaowei Hu, Jia-Wang Bian, Le Zhang, Xiang Bai,

and Jinhui Tang. Richer Convolutional Features for Edge Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2019, 41(8): 1939–1946. (SCI 一区、CCF A 类期刊)

[6] Ming-Ming Cheng*, **Yun Liu***, Wen-Yan Lin, Ziming Zhang, Paul L. Rosin, and Philip Torr. BING: Binarized Normed Gradients for Objectness Estimation at 300fps. Computational Visual Media (CVM), 2019, 5(1): 3-20. (EI 期刊)

[7] **Yun Liu***, Yu-Huan Wu*, Yunfeng Ban, Huifang Wang, and Ming-Ming Cheng. Rethinking Computer-aided Tuberculosis Diagnosis. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020: 2646-2655. (CCF A 类会议)

[8] **Yun Liu**, Shi-Jie Li, and Ming-Ming Cheng. RefinedBox: Refining for Fewer and High-quality Object Proposals. Neurocomputing, 2020, 406: 106-116. (SCI 二区期刊)

[9] **Yun Liu***, Yu-Huan Wu*, Peisong Wen, Yujun Shi, Yu Qiu, Ming-Ming Cheng. Leveraging Instance-, Image- and Dataset-Level Information for Weakly Supervised Instance Segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2020. (SCI 一区、CCF A 类期刊，已接收)

[10] **Yun Liu***, Yu-Chao Gu*, Xin-Yu Zhang*, Weiwei Wang, Ming-Ming Cheng. Lightweight Salient Object Detection via Hierarchical Visual Perception Learning. IEEE Transactions on Cybernetics (TCYB), 2020. (SCI 一区期刊，已接收)

[11] Shijie Li, Yazan Abu Farha, **Yun Liu**, Ming-Ming Cheng, Juergen Gall. MS-TCN++: Multi-Stage Temporal Convolutional Network for Action Segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2020. (SCI 一区、CCF A 类期刊，已接收)

[12] Le Zhang, Zenglin Shi, Ming-Ming Cheng, **Yun Liu**, Jia-Wang Bian, Joey Tianyi Zhou, Guoyan Zheng, and Zeng Zeng. Nonlinear Regression via Deep Negative Correlation Learning. IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2020. (SCI 一区、CCF A 类期刊，已接收)