第3章 基于深度学习的视觉感知

3.1 深度学习基础

深度学习属于“连接主义”学派，其核心是深度神经网络。神经网络是受到生物神经网络的启发而提出的，其核心的概念是神经元。深度神经网络通过模仿神经网络进行信息的分布式并行处理以实现对复杂函数的逼近，多层的神经网络能够表达出数据的内在特征，从而能够更好地完成认知任务。有关深度学习发展历史、MP 神经元模型和感知机模型的知识在第 2 章已经介绍过，这里不再赘述。

通过将感知机组织成多层神经网络，就能够使其表示复杂函数，研究者们也称这种结构为深度神经网络。深度神经网络的基本结构是前馈神经网络，如图 3-1 所示，在网络结构中，最左边的一层称为**输入层**，其中的神经元称为输入神经元。最右边的一层是**输出层**，其中的神经元称为输出神经元，在输入层和输出层之间的结构层称为**隐藏层**，一个网络中往往有多个隐藏层，从而构成深层的网络结构。在图示结构中，前一层的所有神经元都与下一层的所有神经元相连接，这种结构的网络被称为**全连接神经网络**。全连接神经网络存在的缺点：一是巨大的参数量和计算量；二是全连接神经网络没有考虑图像中的结构信息，从而丢失了平移等操作的不变性。由于这些缺点，使得全连接神经网络很难用于解决实际的应用问题。

卷积神经网络（Convolutional Neural Network，CNN）是一种具有局部连接、权重共享等特性的前馈神经网络，如图 3-2 所示。卷积神经网络参考了生物视觉的感受野（Receptive Field，RF）机制，感受野是指某一层输出结果中的一个元素所对应的输入层的区域。更深的卷积神经网络使特征图中单

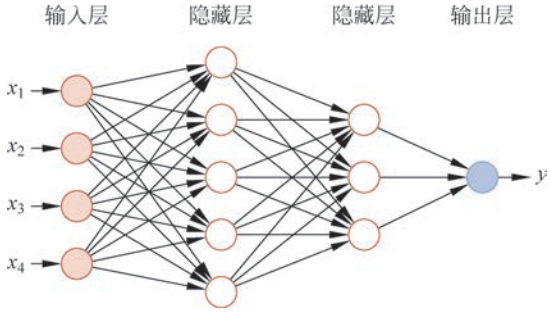

图 3-1 前馈神经网络

个元素的感受野变得更加广阔，从而能够获得输入图像上更大尺寸的特征。基于感受野机制，卷积神经网络就能够通过卷积核参数共享和层间连接的稀疏性获得输入图像上的特征。

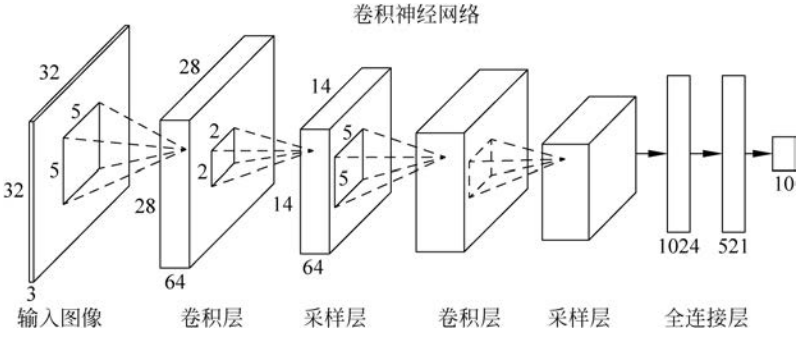

图 3-2 卷积神经网络

1998 年，Yann LeCun 等在之前卷积神经网络的基础上构建了更加完备的卷积神经网络 LeNet-5，其中定义的卷积神经网络的基本框架和基本组件（卷积层、激活层、池化层和全连接层）成为现代卷积神经网络的基础。2006 年，Geoffrey Hinton 基于受限玻尔兹曼机提出了深度置信网络（Deep Belief Network，DBN），通过逐层训练参数与预训练的方法使卷积神经网络可以设计得更复杂、训练效果更好。其后，随着数据规模的扩大和以 GPU 为代表的计算性能提升，卷积神经网络在计算机视觉领域，特别是图像分类、目标检测和语义分割等任务上不断突破，推动了深度学习的飞速发展。

3.2 计算机视觉技术

视觉是人类获取信息的主要途径。计算机视觉（Computer Vision，CV）是指用计算机来模拟人的视觉以获取和处理一系列图像信息，使其成为更适合人眼观察或传送给仪器检测的图像。计算机视觉属于机器学习在视觉领域的应用，是一个多学科交叉的研究领域。当前计算机视觉应用领域包括视频监控、人脸识别、医学图像分析、自动驾驶等。计算机视觉感知技术包括图像分类、目标检测、目标跟踪、图像分割等。

1. 图像分类

图像分类任务是计算机视觉的核心，具有广泛的应用场景。图像分类就是对一组测试图像的类别进行预测，并测量预测结果的准确性，如图 3-3 所示。传统的图像分类处理基于人工设计的各类具有不变性的颜色、形状、纹理等作为特征算子，其中具有代表性的算法有 SIFT、HOG 等，再使用经典的 SVM、AdaBoost 等分类器来完成分类任务。

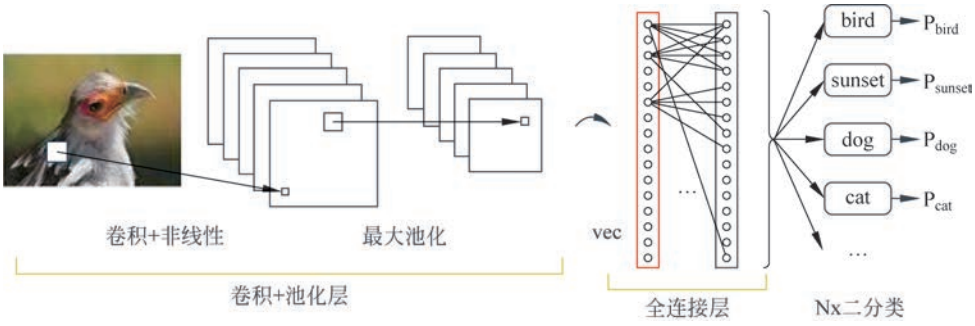

图 3-3　图像分类及 CNN 网络结构

基于深度学习，确切地说是基于卷积神经网络的图像分类方法，随着 2012 年 ImageNet 项目举办的 ILSVRC(ImageNet Large Scale Visual Recognition Challenge)比赛中 AlexNet 网络的夺冠而获得了广泛的关注，此后成为图像分类任务的主流方法。

ImageNet 项目的目标是构建一个用于计算机视觉研究的大型数据集，完整的 ImageNet 数据集有 1400 多万张图片，涵盖了 2 万多个类别的标注与超过百万的边界框的标注，每个类别有 500～1000 张图片。从 2010 年到 2017 年，ImageNet 共举办了 8 届 ILSVRC 比赛，包括图像分类、目标检测等竞赛单元，其中图像分类竞赛使用的是 ImageNet 完整数据集的一个子类，包括 1000 类主要是动物的图片。

ILSVRC 比赛中产生了许多经典的神经网络模型，推动了计算机视觉研究的发展。继 AlexNet 网络之后，2013 年，Matthew D. Zeiler 和 Rob Fergus 在 AlexNet 的基础上，通过使用更小的卷积核和步长，提出了性能更好的网络模型 ZFNet，获得了当年 ILSVRC 分类任务的冠军。2014 年，ILSVRC 分类任务的冠军是 GoogLeNet，其核心是 Inception 模块，由多个 Inception 模块叠加组成，该模块对上一层的输出分别做 1×1 卷积、3×3 卷积、5×5 卷积和 3×3 最大池化 4 种处理。不得不提的是，2014 年获得分类任务第二名的 VGGNet，该网络证明了基于尺寸较小的卷积核，增加网络深度可以有效提升模型的性能。VGGNet 模型结构简单，模型的泛化能力强，到现在依然被很多任务用作图像特征提取。

2015 年，ILSVRC 比赛中最具开创性的工作是在分类、定位、检测及 COCO 的物体检测与语义分割 5 项比赛中全部取得第一名的 ResNet(深度残差网络)模型。ResNet 采用了跨层连接方式，极大地缓解了深层神经网络中的梯度消失问题，因此可以构造更深的网络。很多研究者基于 ResNet 模型，通过构造出更多神经网络层数的模型来进一步提高图像分类的准确率。

2016 年的 ILSVRC 比赛中诞生的经典模型包括 ResNeXt 和 DenseNet。ResNeXt 的核心思想是分组卷积。DenseNet 的核心组件是密集连接模块，在这个

模块中任意两层之间都有直接的连接，这种方法解决了深层网络的梯度消失问题，也加强了特征的传播，并且减少了模型参数。ILSVRC 比赛的最后一个分类任务冠军是 2017 年的 SeNet，它使用“特征重标定”的策略对特征进行处理，通过学习获取每个特征通道的重要性，并根据重要性改变相应特征通道的权重。

2. 目标检测

给定一组图像，图像中每个待测目标各自被标记出类别与位置，这些图像被用作训练集，以训练 CNN 网络具有检测能力。训练完成后，再通过对一组没有标记目标类别和位置的测试图像进行类别及位置预测，并测量预测的准确性结果，这就是目标检测(Object Detection)问题。总的来说，目标检测就是物体检测，需要识别图像中存在的物体并标示出物体所在的位置。目标检测在很多领域都有广泛的应用，如安防、智慧农业、无人驾驶等领域(图 3-4)。

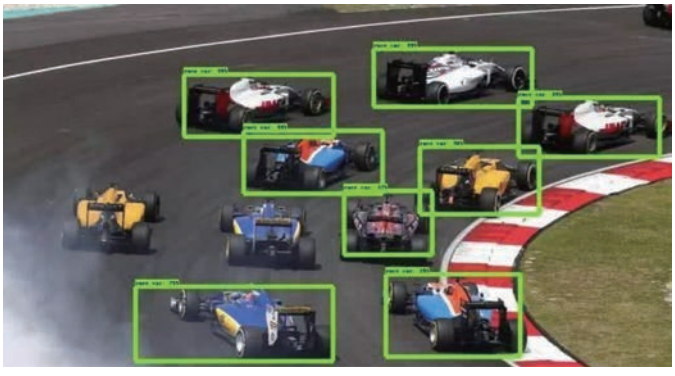

图 3-4　目标检测

针对目标检测的数据集有 Pascal VOC 和 MS COCO 等，其中 COCO 目标检测数据集规模更大，更能体现目标检测方法的性能。在进行多类别物体检测时，需要用很多框来表示物体的位置，有些框可能是准确的，有些框则可能不准确。如果一个框框住了物体并且分类正确，则在该物体的检测上算法是正确的。通常用图像测试集中所有物体检测的准确性，即平均精度均值(mean Average Precision，mAP)来衡量一个目标检测算法的性能。

现阶段基于深度学习的目标检测算法，根据算法实现步骤分为二阶段(Two-stage)目标检测算法和一阶段(One-stage)目标检测算法。Two-stage 目标检测算法将目标检测问题划分为两个阶段，首先，通过候选区域生成算法对输入图像进行待测目标可能所在位置的候选区域预测(Region Proposal)；其次，在第一阶段的基础上对候选区域进行目标分类与位置回归。两阶段算法的代表是 R-CNN 系列算法。One-stage 目标检测算法是一种端到端(End-to-End)的检测方法，该检测方法在输入端输入一幅图像进入卷积神经网络后，仅通过单次的特征提取、池化及边

界框预测等操作即可同时完成待测目标位置回归和类别预测。一阶段算法的代表包括 YOLO 系列及 SSD 算法。

3. 目标跟踪

目标跟踪是指在特定场景跟踪某一个或多个特定感兴趣对象的过程。目标跟踪利用视频或图像序列的上下文信息，对目标的外观和运动信息进行建模，从而对目标运动状态进行预测并标定目标的位置。目标跟踪在无人驾驶领域也很重要，无人驾驶需要在实际场景中对车辆和行人进行跟踪检测。

近年来，深度学习研究人员尝试了使用不同的方法来适应视觉跟踪任务的特征，并且已经探索了很多方法(图 3-5)。在目标跟踪上，初期的应用方式是把网络学习到的特征直接应用到相关滤波或 Struck 的跟踪框架里面，从而得到更好的跟踪结果。本质上卷积输出得到的特征表达，更优于传统的基于特征的跟踪方法，这也是深度学习的优势之一，但同时也带来了计算量的增加。目前很多研究跟踪的框架和方法往往会同时比较两种特征，从而验证跟踪方法或框架的改进与提高，一种是传统的手工特征，另一种就是深度网络学习的特征。网络不同层的卷积输出都可以作为跟踪的特征，对于如何有效利用深度学习的特征，Martin 做了大量的工作，提出了一系列相关的方法，如 C-COT 和 ECO 等。深度学习的另一大优势是端到端的输出，如斯坦福大学的 D. Held 发表在 ECCV2016 上的 GOTURN 方法，目前该方法已经集成到 OpenCV 3.2.0 的开发版本中。牛津大学的 Luca Bertinetto 也提出了多个有影响力的端到端跟踪框架，如 SiameseFC 和 CFNet 等。

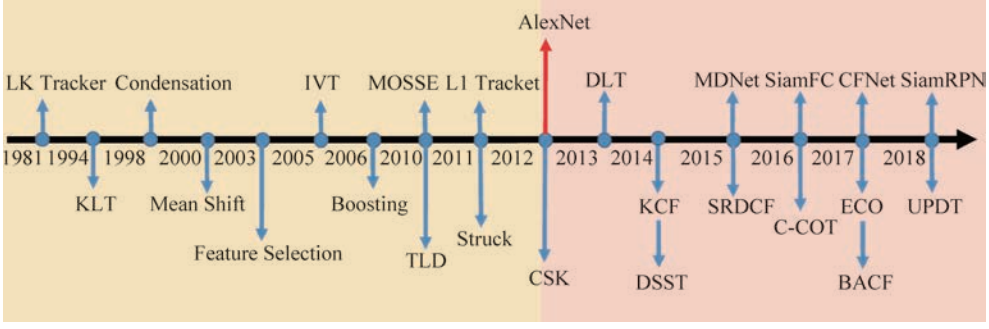

图 3-5　各类代表性目标跟踪算法

(图片来源于李玺，等，2019)

4. 图像分割

图像分割能够提供一种对图形信息进行自动理解的方法，是计算机视觉的核心研究内容，包括语义分割(Semantic Segmentation)、实例分割(Instance Segmentation)和全景分割(Panoptic Segmentation)，三者的区别如图 3-6 所示，其中图 3-6(b)是语义分割，能够对图像上的所有像素点进行分类，图 3-6(c)是实例分

割，需要标示出同一类别的不同个体，图 3-6(d)是全景分割，可以看作语义分割和实例分割的结合，能够对图中的所有物体包括背景进行检测和分割。

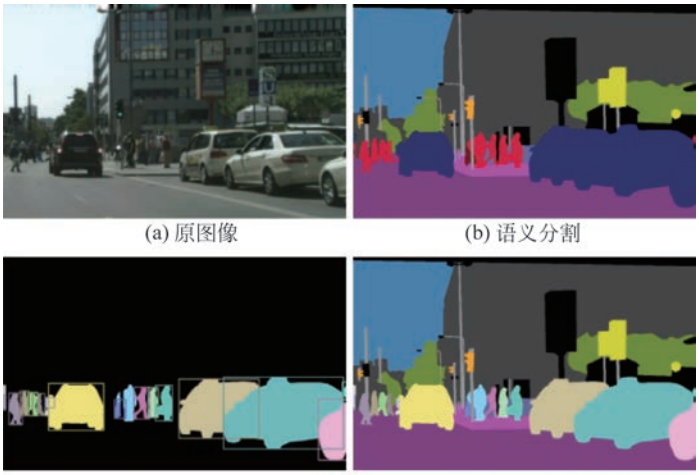

图 3-6 图像分割包含的研究内容

(图片来源于 Kirillov,et al.,2019)

语义分割将整个图像分成一个个像素组，然后对其进行标记和分类，是对图像中的不同目标进行精确的边界划分。特别地，语义分割试图在语义上理解图像中每个像素的角色(如识别它是汽车、摩托车还是其他的类别)。语义分割将图像转换为具有突出显示的特定目标区域的划分，对图像的认知上升到了像素级。

传统的语义分割工作大多基于图像像素的低阶视觉信息，包括基于阈值、基于像素聚类和基于图划分的分割方法等，这些方法的优点是无须进行训练，计算复杂度较小，但是这种方式实现的分割结果精度不高。与其他计算机视觉任务一样，卷积神经网络在语义分割任务上也取得了巨大成功，基于卷积神经网络的语义分割方法不断提高着图像语义分割的精度和速度，有些算法已经能够实现实时的语义分割。语义分割的常用数据集包括 Pascal VOC、MS COCO、BDD100K 和 Cityscapes 等，在语义分割领域代表性的算法包括 FCN、SegNet 和 DeepLab 等。

除语义分割外，实例分割将不同类型的实例进行分类，如用 5 种不同颜色来标记 5 辆汽车。分类任务通常来说就是识别出包含单个对象的图像是什么，但在分割实例时，需要执行更复杂的任务。此时就会看到多个重叠物体和不同背景的复杂景象，不仅需要将这些不同的对象进行分类，而且还要确定对象的边界、差异和彼此之间的关系。全景分割任务是为图像中的每个像素点赋予类别和实例 ID，生成全局的、统一的分割图像，其中，Label 即语义标签，指的是物体的类别，而实例 ID 则对应同类物体的不同编号。实例分割及全景分割的典型算法如图 3-7 所示。

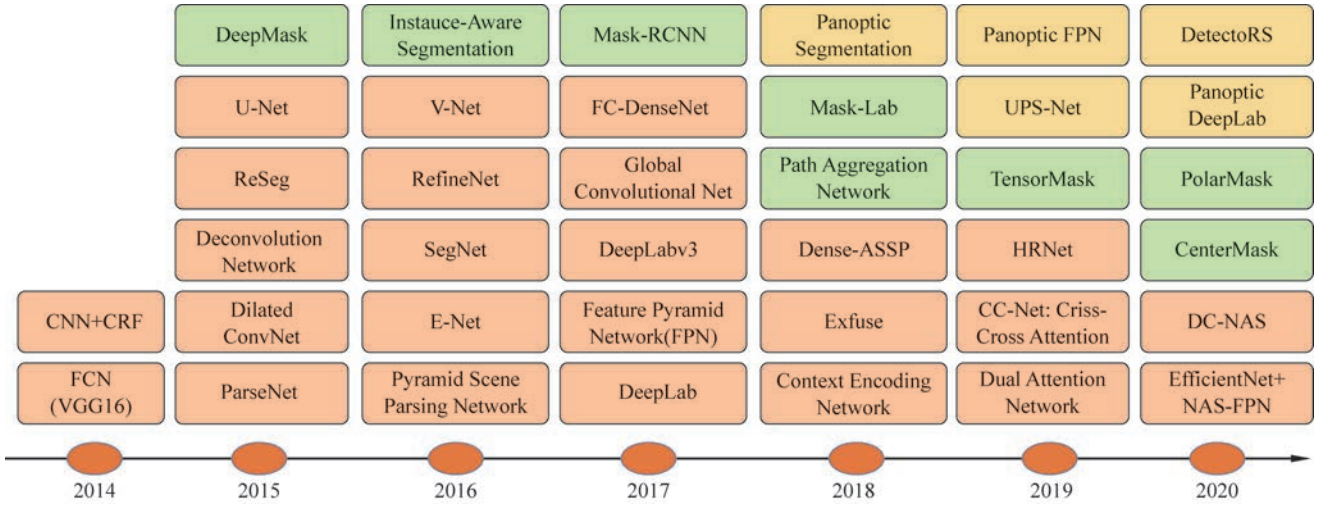

图 3-7　基于深度学习的图像分割算法

（桔色、绿色和黄色分别代表语义、实例和全景分割）

（图片来源于 Minaee，et al.，2021）

3.3 图像分类典型算法

图像分类是计算机视觉的核心任务，实践证明，卷积神经网路具有特征发现的强大能力，特别适合于解决计算机视觉的分类任务。在计算机中图像一般是由三通道的 2D 矩阵数据表示的，实际的图像表示会存在尺度变化、图像变形、图像遮挡、光照条件变化等复杂情况，然而，研究表明，基于卷积神经网络的图像分类算法已经超过了人类的水平，可以应用于实际的问题解决中。

LeNet 是第一个实用的卷积神经网络，确立了卷积神经网络的基本结构，如图 3-8 所示。在用于 MNIST 手写数字识别的 LeNet-5 网络结构中，除输入输出层外，还包括两个卷积层、两个池化层和两个全连接层。C1 为卷积层，包括 6 个(1,5,5)的卷积核，将 32×32 像素的图像输入转换为(6,28,28)的特征图。S2 层为下采样层，对 C1 层的输出数据进行下采样处理。C3 卷积层中卷积核的大小也是 5×5，接着的 S4 层与 S2 层类似。C5 是全连接层，将前面的特征图转换为 120 维的一维向量。F6 是具有 84 个神经元节点的全连接层。当时，LeNet-5 被成功用于 ATM 以对支票中的手写数字进行识别。

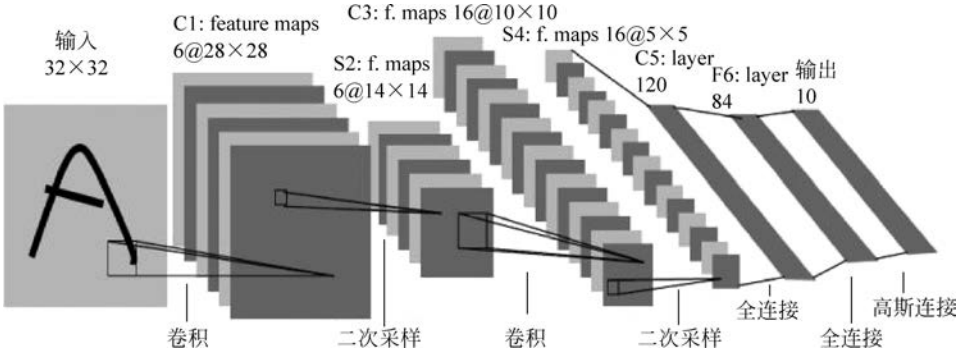

图 3-8 LeNet-5 网络结构

(图片来源于 LeCun,et al.,1998)

AlexNet 是 ILSVRC 2012 图像分类大赛的冠军，以网络提出者的名字命名。AlexNet 的输入图像尺寸为 224×224，输出为 1000 类的全连接层，其网络结构和 LeNet-5 相似，但更深、有更多参数，包括 5 个卷积层和 3 个全连接层，网络结构如图 3-9 所示。

受当时算力的限制，AlexNet 巧妙地将整个网络分割部署在两块 NVIDIA GTX580 GPU 上并行执行。AlexNet 将卷积神经网络的基本原理应用到了更深的网络模型中，扩展了卷积神经网络的性能，其技术关键点包括：①使用了 ReLU 激活函数，使之有更好的梯度特性、训练更快；②使用了随机失活(Dropout)，通过

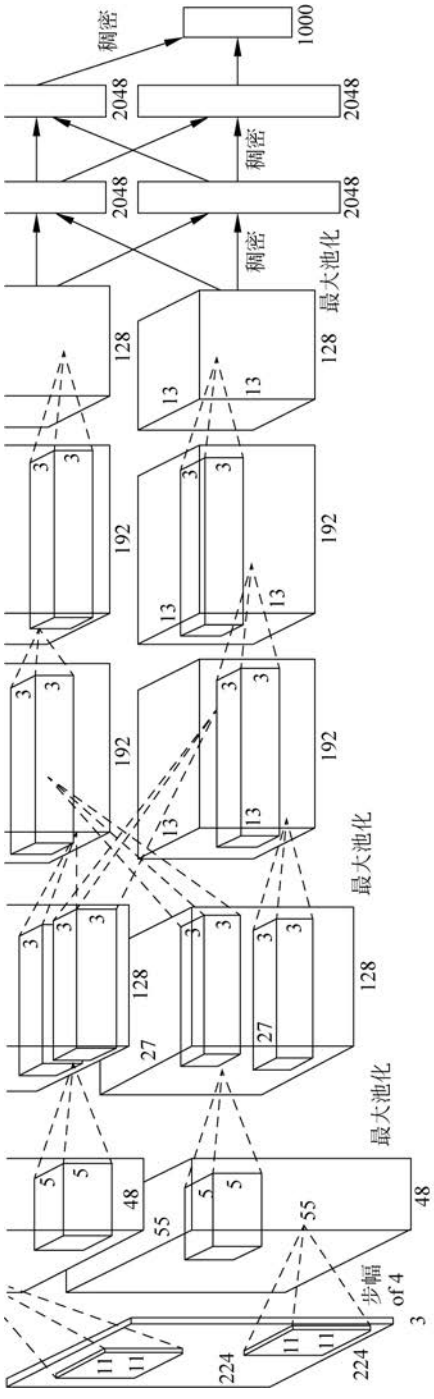

图 3-9　AlexNet 网络结构

（图片来源于 Krizhevsky，et al.，2017）

随机忽略一部分神经元后，可以有效减少模型的过拟合；③数据增强技术，对于图像数据，常用的增强技术包括裁剪、镜像、旋转、缩放以及在图像中加入随机噪声等方法，这些方法都在 AlexNet 网络中被采用以提升系统性能。AlexNet 在 ILSVRC 竞赛中以高出第二名 10%的图像分类性能使人们意识到卷积神经网络在计算机视觉领域研究的优势。同时，在 AlexNet 之后，采用 GPU 进行卷积神经网络训练加速的方法成为学术界和工业界做深度学习研究的主流。

VGG 网络是 ILSVRC 2014 的亚军，其取名源自作者所在的研究组牛津大学的 Visual Geometry Group。VGG-16 的基本架构为：conv1^2（64）→pool1→conv2^2（128）→pool2→conv3^3（256）→pool3→conv4^3（512）→pool4→conv5^3(512)→pool5→fc6(4096)→fc7(4096)→fc8(1000)→softmax。^3 代表重复 3 次。VGG 网络的关键点是：①结构简单，只有 3×3 卷积和 2×2 池化两种配置，并且重复堆叠相同的模块组合，卷积层不改变空间大小，每经过一次池化层，空间大小减半；②参数量大，而且大部分的参数集中在全连接层中。网络名称中的 16 表示它有 16 层 conv/fc 层；③合适的网络初始化和使用批量归一(Batch Normalization)层对训练深层网络很重要(图 3-10)。

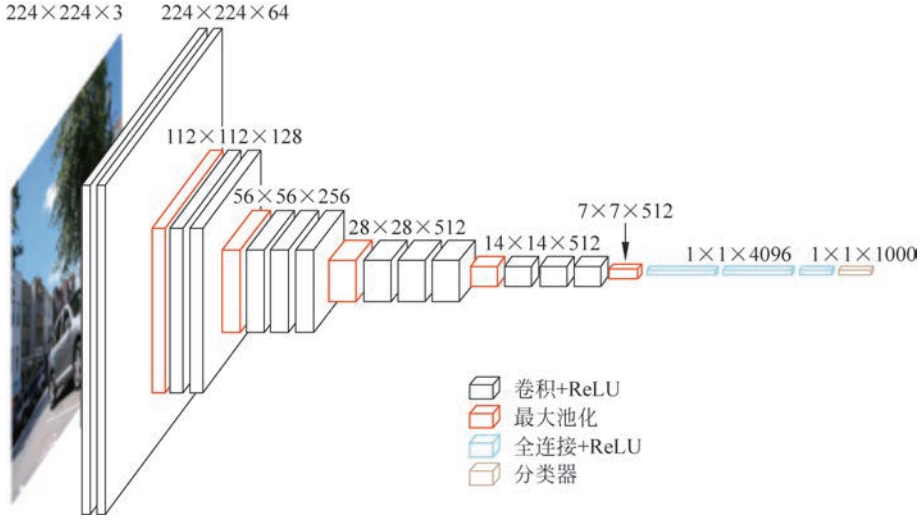

图 3-10　VGG16 网络结构

(图片来源于 Simonyan，et al.，2015)

在原论文中无法直接训练深层 VGG 网络，因此先训练浅层网络，并使用浅层网络对深层网络进行初始化。在 BN 出现之后，伴随其他技术，后续提出的深层网络可以直接得以训练。VGG-19 结构类似于 VGG-16，有略好于 VGG-16 的性能，但 VGG-19 需要消耗更大的资源，因此实际中 VGG-16 使用更多。由于 VGG-16

网络结构十分简单，并且很适合迁移学习，因此至今 VGG-16 仍在广泛使用。

GoogLeNet 网络是 ILSVRC 2014 的冠军，其试图回答在设计网络时究竟应该选用多大尺寸的卷积或者应该选什么样的池化核（图 3-11）。GoogLeNet 网络提出了 Inception 模块，同时用 1×1、3×3、5×5 卷积和 3×3 池化核，并保留所有结果。网络基本架构为：conv1(64)→pool1→conv2^2(64,192)→pool2→inc3(256,480)→pool3→inc4^5(512,512,512,528,832)→pool4→inc5^2(832,1024)→pool5→fc(1000)。GoogLeNet 的关键点是：①多分支分别处理并级联结果；②为了降低计算量，用了 1×1 卷积降维。GoogLeNet 使用了全局平均汇合替代全连接层，使网络参数大幅减少。GoogLeNet 取名源自作者的单位（Google），其中 L 大写是为了向 LeNet 致敬。

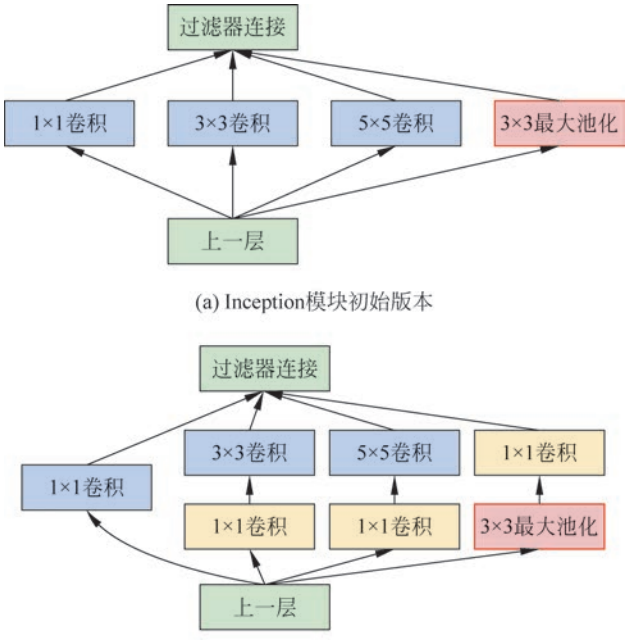

图 3-11　GoogLeNet 网络 Inception 模块

（图片来源于 Szegedy，et al.，2015）

ResNet 网络是 ILSVRC 2015 的冠军。ResNet 旨在解决网络加深后训练难度增大的现象，其提出了 Residual 模块，包含两个 3×3 卷积和一个短路连接。短路连接可以有效缓解反向传播时由于深度过深导致的梯度消失现象，这使得网络加深之后性能不会变差。短路连接是深度学习又一重要思想，除计算机视觉外，短路连接也被用到机器翻译、语音识别/合成领域。此外，具有短路连接的 ResNet 可

以看作许多不同深度而共享参数的网络集成，网络数目随层数指数增加(图 3-12)。

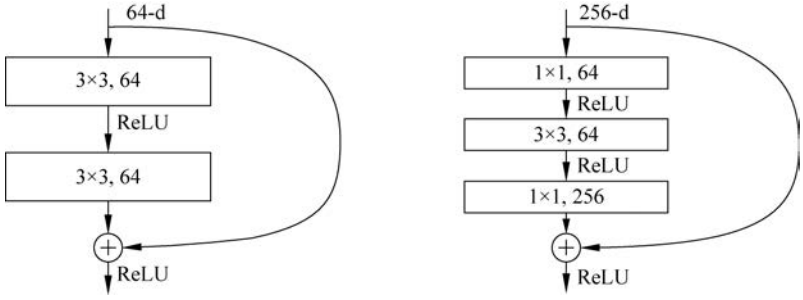

图 3-12 ResNet 网络结构

(图片来源于 He,et al.,2016)

ResNet 的创新点：①使用短路连接，使训练深层网络更容易，并且重复堆叠相同的模块组合；②ResNet 大量使用了批量归一层；③对于很深的网络(超过 50 层)，ResNet 使用了更高效的瓶颈(bottleneck)结构。ResNet 在 ImageNet 上取得了超过人类的准确率。

3.4 目标检测典型算法

3.4.1 两阶段目标检测方法

1. R-CNN 网络

R-CNN 全称 Region-CNN，是由 Ross Girshick 等于 2014 年提出，是第一个成功将基于卷积神经网络的深度学习算法应用于目标检测上的算法(图 3-13)。该算法在 Pascal VOC 2012 数据集上，能够将多类目标检测的 mAP 提升到 53.3%，较之前最好的传统目标检测算法的结果提升了 30%。同时，这篇论文中也指出了迁移学习概念与重要性。神经网络在训练时需要大量的标注数据，获取及标注数据时往往难以尽如人意，出现数据不足的情况，这时就可以采用迁移学习。迁移学习是在缺乏大量标注数据时，将在其他大型数据集训练并保存的神经网络的参数迁移至有需要的网络训练中，其后在小规模特定的数据集中进行网络参数的微调。使用这种方法训练网络可以减少网络对数据的需求，加快神经网络的收敛速度。

R-CNN 目标检测算法，首先通过选择搜索算法(Selective Search)在原图像上生成大量不同尺寸的待测目标可能存在的区域(Region Proposal)；其后，根据这些区域之间的相似性进行区域合并，通过叠加将多个小区域融合成一个较大区域，从而得到许多可能包含待测目标区域的边界框。第三步，将上一步中获得的若干

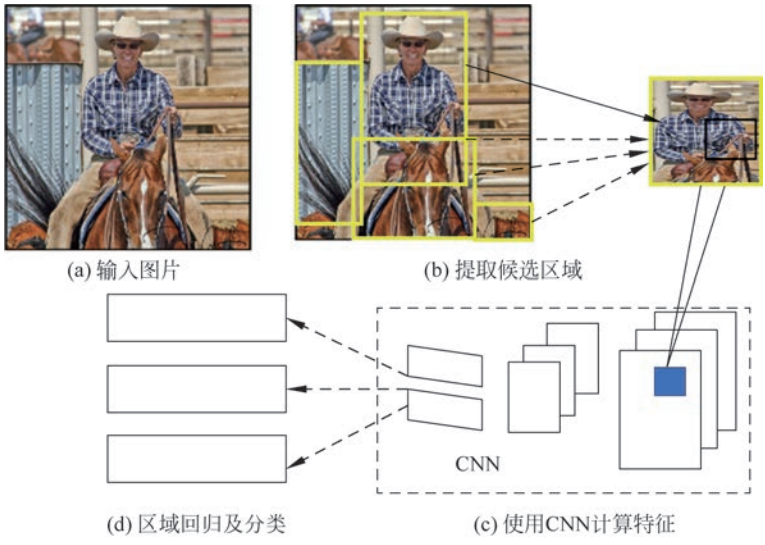

图 3-13　R-CNN 目标检测算法

（图片来源于 Girshick，et al.，2014）

子区域放缩到同一尺寸并将其送入卷积神经网络中进行特征提取。最后，通过支撑向量机（SVM）进行待测目标的分类和线性回归模型进行边界框微调。

2. SPP-net 网络

由于 R-CNN 在进行特征提取前，对选取的所有候选区域生成一个维度相等的向量，对区域强行放缩后，会使图像比例关系遭到破坏，不利于卷积神经网络提取待测目标的语义特征。而且，无论是生成候选区域还是候选区域放缩都是非常耗时的，针对这一缺点提出了 SPP-net。

SPP-net 全称为 Spatial Pyramid Pooling，意为空间金字塔池化，是 Kaiming He 等在 2014 年提出的。SPP-net 在卷积层后加入了空间金字塔池化层，依靠该池化层中的滤波器将卷积层输出的候选区域统一到相同尺寸，摆脱了 R-CNN 中暴力的强制放缩手段，使全连接层获得不失真的同尺寸特征输入。SPP-net 网络提出的空间金字塔池化过程如图 3-14 所示，首先将经过卷积操作后得到的特征图划分为一定大小的栅格；其次对划分好的栅格做最大池化；最后将各层的池化输出结果组合起来送入全连接层完成最后检测。假设有 256×256 的特征图，对该特征图并行 4×4、2×2、1×1 栅格划分与最大池化操作，经过这样的空间金字塔池化后，将会获得多个 16×256、4×256、1×256 维度的特征图，最后把这些特征图连接组合成特征向量送入全连接层。

输入图像经过卷积层后，对获得的特征图进行可视化会发现，原图像中的待测目标所在位置与特征图上所在位置相同。由此可以确定，R-CNN 先进行候选区筛

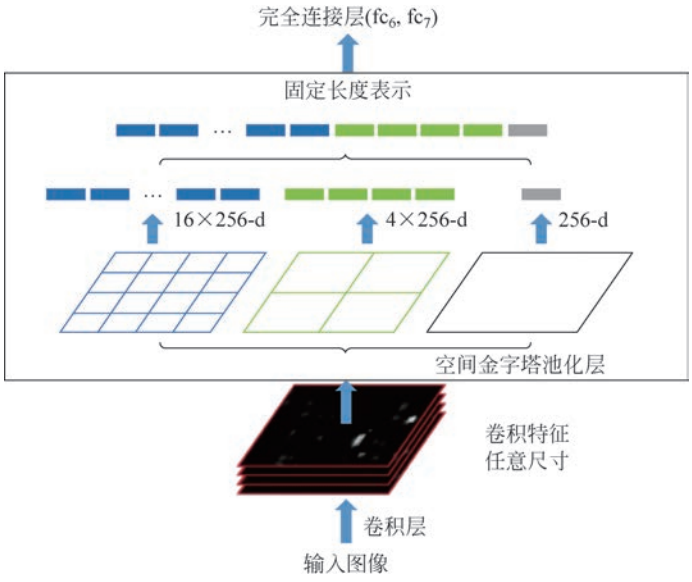

图 3-14 SPP-net 网络检测流程

(图片来源于 He,et al.,2014)

选,再对筛选出来的候选区域进行卷积这一操作流程势必导致原输入图像的部分区域进行了多次特征提取,任务量加大。而 SPP-net 仅对输入图像进行一次完整图像的特征提取,其后在特征图上做候选区域截取与剪裁,并搭配空间金字塔池化,解决了反复提取特征与输入图像尺寸大小不等对网络的不良影响。虽然这种方法看似只对特征提取部分进行了改进,其他模块均与 R-CNN 网络一样,但是与 R-CNN 相比,该模型速度提高了百倍,精度也有了明显提升。

3. Fast R-CNN

继 2014 年提出 R-CNN 之后,Ross Girshick 时隔一年又推出了 R-CNN 算法的改进版本 Fast R-CNN,该算法针对 R-CNN 与 SPP-Net 依然存在的训练过程烦琐与所占存储空间大等缺点进行了改进,实现在保证检测效果的同时节约存储空间,提升检测速度。基于 VGG16 的 Fast R-CNN 模型在训练速度上比 R-CNN 快大约 9 倍,比 SPP-net 快大约 3 倍,在 VOC2012 数据集上的 mAP 大约为 66%。

Fast R-CNN 网络检测流程如图 3-15 所示,输入图片首先经过 5 个卷积层和 2 个下采样层得到 conv5 层特征图和 $2k$ 个候选区域;其次,将上述两者送入 RoI 池化层中。RoI 池化类似空间金字塔池化,不同的是空间金字塔池化输出的特征维度相同、尺度不一,而 RoI 池化只需要同一尺度特征图;最后,直接使用 softmax 分类器替代 SVM 分类,同时利用多任务损失函数将边界框回归也加入到网络中。经作者试验验证,这种将分类任务与位置预测统一为一个多任务模

块，使网络能够更充分地利用特征，实现了特征共享，进一步提升了网络的检测速度。

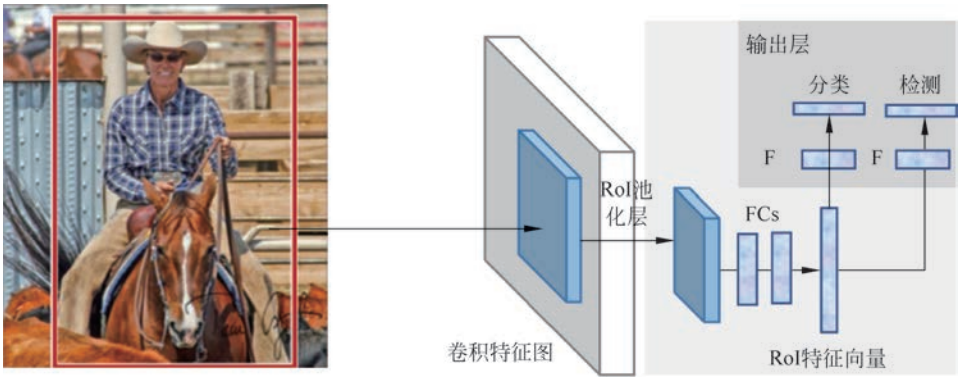

图 3-15　Fast R-CNN 网络检测流程

（图片来源于 Girshick，2015）

4. Faster R-CNN

在 2016 年，R-CNN 作者又提出了新的 Faster R-CNN，其最突出贡献是创造性地提出了 RPN（Region Proposal Network）。RPN 位于卷积神经网络层之后，不再使用 SS（Selective Search）方法生成检测框，而是直接使用 RPN 生成检测框，实现了将候选框提取与神经网络进行特征提取的融合，使综合性能有较大提高。

Faster R-CNN 检测流程主要分为四部分内容，即提取目标特征的卷积层、生成待测目标候选区域的 RPN、调节候选区域尺寸的 RoI 池化层以及全连接分类层（图 3-16）。首先，图像送入 Faster R-CNN；其次，RPN 对输入的特征图进行同时的分任务处理操作。此部分中 RPN 不仅要使用 softmax 分类器对候选区域做正负样本分类，还要计算候选区域与真实边界框的偏移量，其后将结果输入到 Proposal 层中。Proposal 层将计算的偏移量与分类正确的候选区域融合，目的是获取精确的建议候选区域，同时剔除太小和超出边界的候选区域。完成以上两部分工作，整个网络的目标定位任务已基本完成。最后，RoI 池化层对上一步输出的

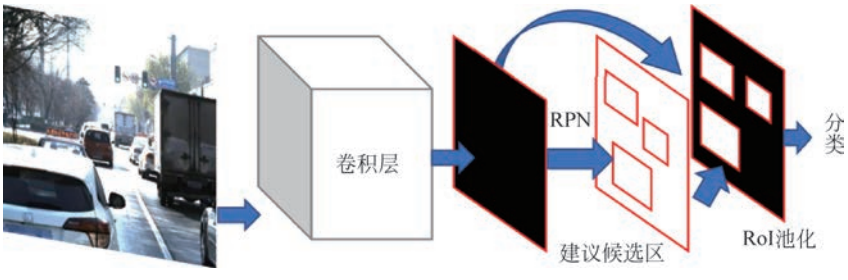

图 3-16　Faster R-CNN 网络检测流程

建议候选区进行池化，本次池化的目的主要是为下一步全连接层做准备，即将上一步获得候选区域统一池化成固定尺寸。最后，使用 softmax 分类器对待测目标进行类别细化，输出定位结果与分类结果。

3.4.2 单阶段目标检测方法

One-stage 类目标检测算法，期望通过一次操作就可以同时完成位置回归和分类任务。这类算法预测推理过程相对简单，由于减少了类似初筛正负样本的操作过程，加快了检测速度，但增加了检测器的难度，降低了检测的精度。随着研究开展，通过不断探索，就目前的一阶段目标检测器来说，在保持较高检测速度的同时，可以达到部分两阶段检测算法的精度。典型的一阶段检测算法有 YOLO 系列、SSD 及其改进算法、RetinaNet、RefineDet 等。

1. YOLO

Joseph Redmon 等提出的 YOLO 系列目标检测算法最早出现于 2016 年的 CVPR 学术会议上，通过不断改进，2018 年推出改进后的 YOLOv3。YOLO 系列算法是典型的端到端检测网络，实现了从原始图片输入到待测目标的位置预测与类别判断。

第一个版本的 YOLO 检测算法的网络结构如图 3-17 所示。YOLOv1 检测时，输入图像首先经过卷积层获得具有丰富语义信息的特征图，根据输出特征图的尺寸对原始图像进行分割，使特征图中的每个单位区域与原图像中的一部分像素块对应。假设输入为 320×320 大小，卷积层输出为 10×10 大小的特征图，则将原始图像与特征图划为 10×10 的网格图像块。特征图中每个 1×1 的单位特征图分别对应于原图像中的 32×32 的图像块，这也是感受野的原理。其次，如果有待测目标的中心落入 10×10 栅格中的任意一个，该栅格就负责检测该待测目标，输出

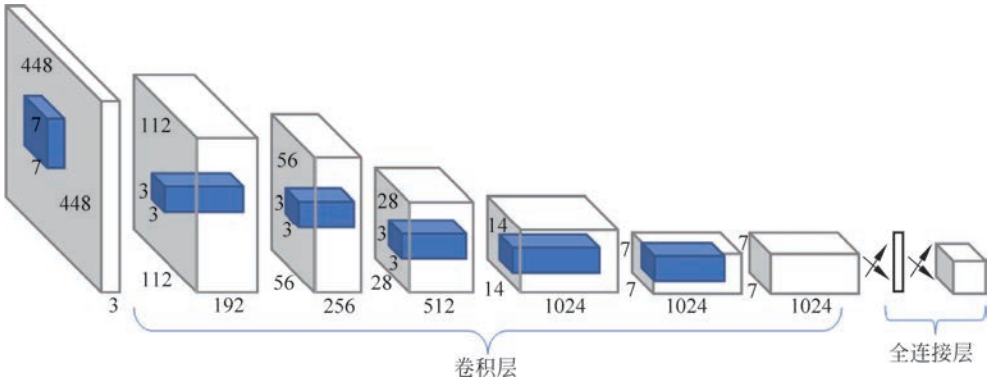

图 3-17 YOLOv1 网络结构

一定个数的边界框和类别概率。最后，各边界框做回归计算，每个边界框共计输出5个值，分别为边界框中心点(x，y)坐标，边界框相对全图的宽高比例以及一个置信度值。其中，置信度值代表了所预测的边界框中含有目标的置信度和该边界框预测准确率的双重信息，计算公式为

$$\text{confidence}=\Pr(\text{object})\cdot \text{IOU}_{\text{pred}}^{\text{truth}} \tag{3-1}$$

第一项 Pr(object)仅有两种结果，即如果有待测目标中心在栅格内，则值为1；否则其值为0。第二项 $\text{IOU}_{\text{pred}}^{\text{truth}}$ 代表的是预测的边界框与真实边界框面积交集比并集，表达的是预测的边界框与真实边界框的重叠度，预测中 IOU 值越大代表预测的结果越准确。

对比 YOLO 系列检测算法可以发现，YOLOv1 检测算法按照栅格来预测目标大大减少了背景误检率。但是 YOLOv1 网络中依然存在不足，虽然每个栅格都可以预测数个边界框，但是最终只输出 IOU 最高的边界框，这注定了一个栅格只能预测出一个物体。如当待测目标较小时，每个栅格将会包含多个待测目标，而 YOLOv1 最终只会输出一个预测边界框，从而导致该网络检测精准度与召回率双低的问题。

YOLOv2 是 Redmon 等根据 YOLOv1 存在的召回率与准确率低的问题，对此算法进行的改进，该篇论文获得 CVPR2017 最佳论文奖，说明 YOLOv2 算法的创新性和性能得到了学术界的认可。作者在论文中提出了多种改进网络检测准确率与召回率的手段，其中较为有效的方法包括设置先验框、根据聚类设置先验框尺寸、新的特征提取卷积层、多尺度预测训练等。YOLOv2 算法由于加入了上述的改进策略，使 mAP 有了显著提升，而检测速度上依然保持着自己作为第一阶段方法的优势，YOLOv2 网络结构如图 3-18 所示。

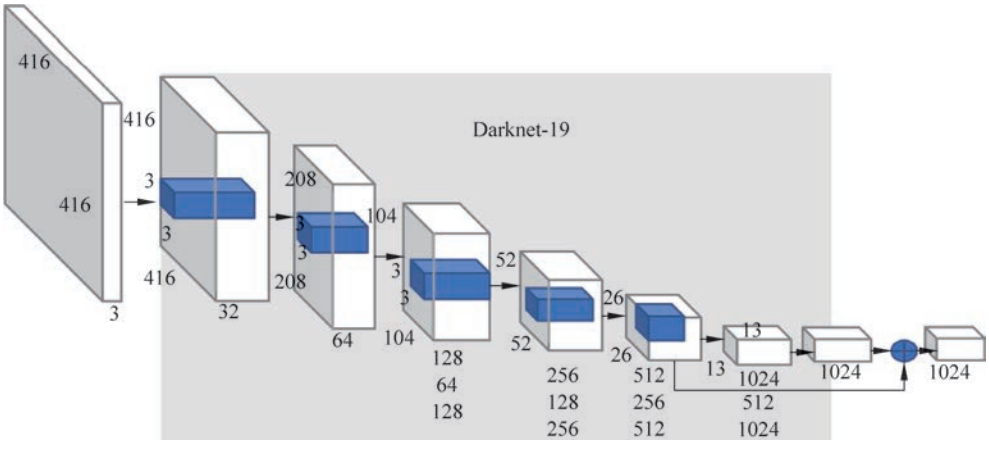

图 3-18　YOLOv2 网络结构

YOLOv2 目标检测网络不同于 YOLOv1 使用栅格直接预测两种比例的边界框，该网络借鉴了 Faster R-CNN 的预设先验框思想，为每个栅格预设 5 种不同尺寸的先验框，使预测框的数量由原来的 98 提升为 845，为使网络能更好地回归真实边界框，作者采用 K-Means 聚类算法代替随机人为设置，通过对数据集进行聚类分析，设置了更贴合真实边界框的先验框，以提高检测精度。YOLOv2 虽然对于 YOLOv1 中存在定位不准确、召回率低等问题进行了改进，但是该检测算法对于小目标的检测能力依然存在可改进的空间，且检测精度仍然有待于提升。

YOLOv3 提出于 2018 年，网络延续了端到端的检测流程，融合了 Faster R-CNN 的预设先验框、ResNet 的跳层连接及 FPN 的多尺度检测等多种先进思想，弥补了 YOLOv1/v2 的不足，是一个兼顾检测速度与精度的目标检测算法。

YOLOv3 目标检测网络中使用 Darknet-53 特征提取骨干网络替换了原 YOLOv2 中的 Darknet-19。Darknet-19 是基于 ResNet 改进的、带有跳跃连接层的特征提取网络，包含 19 个卷积层、5 个最大值池化层。Darknet-53 在 Darknet-19 基础上加深了网络结构，包含了 53 个卷积层。不同于 YOLOv2 仅在一个尺度上进行边界框预测，YOLOv3 借鉴了 FPN 的预设先验框思想，在 3 种不同尺度上各设 3 种不同尺寸的先验框，持续提升预测框的数量，更好地提升了召回率，其网络结构如图 3-19 所示。

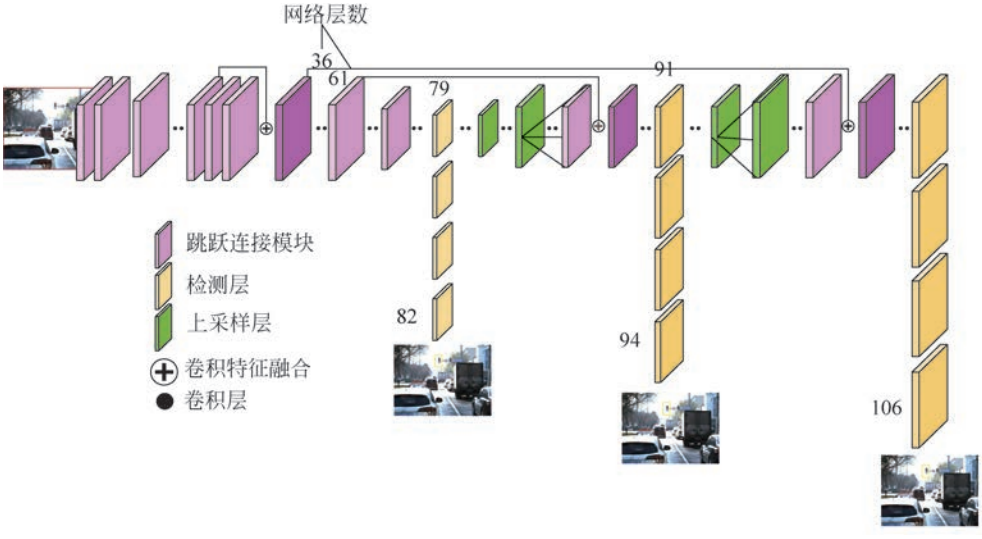

图 3-19　YOLOv3 网络结构

2. SSD

SSD(Single Shot MultiBox Detector)是 Wei Liu 在 ECCV2016 上提出的一阶段目标检测算法，其网络结构如图 3-20 所示。SSD 目标检测算法首先使用

VGG-16 网络模型中包括 conv5_3 之前的卷积层结构，在此基础上又添加了若干卷积层与一个全球平均池化(Global Average Pool)层组成该算法的特征提取网络，完成对输入图像的特征提取任务。其次，在检测部分以 conv4_3、conv7、conv8_2、conv9_2、conv10_2、conv11_2 等 6 个特征图层作为基础，分别在此 6 个特征图上预设了 6 种不同尺度大小的先验框，再由这些先验框对待测目标进行位置回归与类别预测，输出多个符合条件的候选框。最后，将多个检测层的输出结果连接起来，通过非极大值抑制的方法来筛选上一步中生成的候选框，获得最终的位置与分类信息。

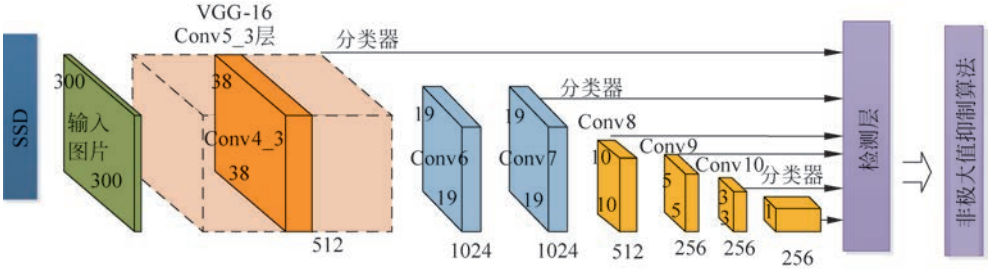

图 3-20　SSD 网络结构图

(图片来源于 Liu，et al.，2016)

SSD 算法可以看作对 YOLOv1 召回率低的缺点进行改进，在召回率与检测精度上较 YOLOv1 有明显提高。但不足之处在于，SSD 目标检测算法需要预先手动设置先验框的初始尺寸和长宽比例，而不能通过学习来获得，导致调试过程非常依赖经验。同时，SSD 算法中使用的最大尺寸特征图为 VGG-16 网络中 conv4_3 层输出的特征图，在进行如交通标识的小目标检测时，由于卷积结构较深，容易忽略小目标的特征，存在检测不到小目标的缺点。

3. Anchor free 目标检测算法和实现

Anchor free 是近两年内蓬勃发展起来的一类目标检测算法。从检测流程上来分析，此类算法与一阶段目标检测算法基本相同。不同的是该类方法抛弃了前面在一阶段和二阶段目标检测网络中常见的对于边界框的依赖，创新性地通过关键点、密集预测等方式完成对目标物体的检测。对比基于边界框的 Anchor 类目标检测算法与 Anchor free 类算法可以看出两种方法各具长处与不足，Anchor 类目标检测算法相对成熟，Anchor free 类目标检测算法尚需更多的试验及实际应用检验。

Anchor free 的思想其实在早期检测网络 YOLO 中就有所体现，后期由于 Anchor 类检测算法较好的性能，Anchor free 类算法的思想并没有引起研究人员的过多关注。Anchor free 类算法抛弃了 YOLO 与 SSD 等算法中使用的边界框描

述待测目标为位置的思想，而是使用一个或多个关键点来描述待测目标的位置。

基于关键点进行目标检测是目标检测领域新的检测手段，其中有代表性的检测网络有基于关键点的 CornerNet、ExtremeNet、CenterNet 等。基于密集预测的 FSAF、FCOS、FoveaBox 等，该类网络多应用于语义分割、深度估计、关键点检测中，并取得了不错的效果，使得 Anchor free 类目标检测算法引发大量关注。

3.5 目标跟踪典型算法

目标跟踪(Object Tracking)是计算机视觉领域的一个重要问题。一般提到“视觉目标跟踪”或“VOT”，往往指的是单目标跟踪。尽管看起来单目标跟踪(Single Object Tracking，SOT)和多目标跟踪(Multi Object Tracking，MOT)只是目标数量上的差异，但它们通用的方法实际上截然不同。

从研究对象上讲，单目标跟踪算法一般是不限类别的，而多目标跟踪一般是仅针对特定类别的物体。从时长上讲，单目标跟踪更多地针对短时间的图像序列，而多目标跟踪一般要处理较长的视频，其中涉及各个目标的出现、遮挡和离开等情况。从实现思路上讲，单目标跟踪更关注如何对目标进行重定位，而常见的多目标跟踪方法往往更多地关注如何根据已检测到的目标进行匹配。

按照初始化方式，常见的多目标跟踪算法一般可分为基于检测的跟踪(Detection-Based Tracking，DBT)和无检测的跟踪(Detection-Free Tracking，DFT)。DBT 要求由一个目标检测器首先将每帧图像中的目标检测出来，而 DFT 要求已知每个目标首次出现的位置，再对每个目标分别进行跟踪(这一点可以看作在同一个视频中进行的多个单目标跟踪)。显然，前者的设定更接近实际应用场景，也是学术界研究的主流。

深度学习的发展和以 GPU 为代表的算力增强带来了视觉算法性能的突飞猛进。在目标跟踪领域中，基于 CNN 并结合相关滤波的端到端深度学习方法在跟踪准确度和系统性能上逐渐超越传统基于相关滤波的方法，成为当前研究的主流方法。

全卷积孪生网络(Fully-Convolutional Siamese Networks)开创了端到端深度学习式相关滤波方法的先河，成为基于深度学习方法进行 SOT 研究的基础。孪生网络的基本思想如图 3-21 所示，其中 φ 就是 CNN 编码器，上、下两个分支(两输入网络，目标跟踪的典型结构)使用的 CNN 不仅结构相同，参数也是完全共享的(其实就是同一个网络)。z 和 x 分别是要跟踪的目标模板图像(尺寸为 127×127)和新的一帧中的搜索范围(尺寸为 255×255)。两者经过同样的编码器后得到各自的特征图，对两者进行互相关运算后，则会同样得到一个响应图(尺寸为 17×17)，

其每一个像素的值对应了 x 中与 z 等大的一个对应区域出现跟踪目标的概率。

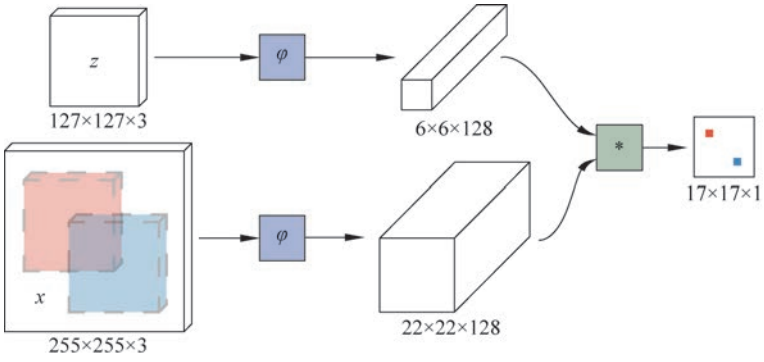

图 3-21　SiamFC 网络结构

（图片来源于 Bertinetto，et al.，2016）

SiamFC 的离线端到端训练使 CNN 模型学习了衡量 x 与 z 的相似性方式，同时由于很好地利用了 GPU 的算力，基于 AlexNet 网络的 SiamFC 可以达到 65FPS 的速度，并保持了不错的准确率，尽管跟踪效果还无法匹敌当时的最优水平。

在 SiamFC 中，原尺寸 127×127 的 z 经过了 5 层 AlexNet 后得到的特征图已经小到 6×6 的尺寸，因为没有填充并且经过了几次池化。照这样下去，再加一个池化层和一个 3×3 卷积层，特征图就要变成 1×1 了。显然，想让网络再深一些，填充是不可避免的。加了填充，网络的确能够变得很深，但是新的问题又出现了——CNN 的平移不变性变得极差，目标的定位经常出现明显的偏移，并且模型对目标的判别能力并没有提高。

SiamRPN 借鉴了目标检测领域常用的 RPN 概念用于预测新图像中目标的尺度，从而解决了 SiamFC 难以应对物体尺度变化的问题，其结构如图 3-22 所示。

SiamRPN 在 x 和 z 经过孪生 CNN 得到各自的特征图后，没有直接对两者进行互相关运算，而是将这两个特征图各自放入 RPN 部分的两个分支中，每个分支中的两个特征图分别经过一个 CNN 再进行互相关运算。RPN 部分的两个分支分别用于进行目标概率的预测和目标边框的回归，并且同样借鉴了目标检测领域的 Anchor 方法，从而降低了目标边框回归的训练难度。

SiamRPN＋＋是在 SiamRPN 工作上的改进，其网络结构如图 3-23 所示，主要改进包括以下 4 点：①使用了微调版的 ResNet-50 主干，极大地优化了特征的提取；②对 ResNet-50 的 3、4、5 阶段的特征分别使用 RPN 进行边框回归与目标定位，并使用带权重的融合方法结合三者的结果；③使用了 Depth-wise 互相关运算，减少参数量，加速了 RPN 部分的运算；④提出了一种 Spatial-aware 的采样策略，从而打破了目标跟踪对 CNN 的严格平移不变性限制。

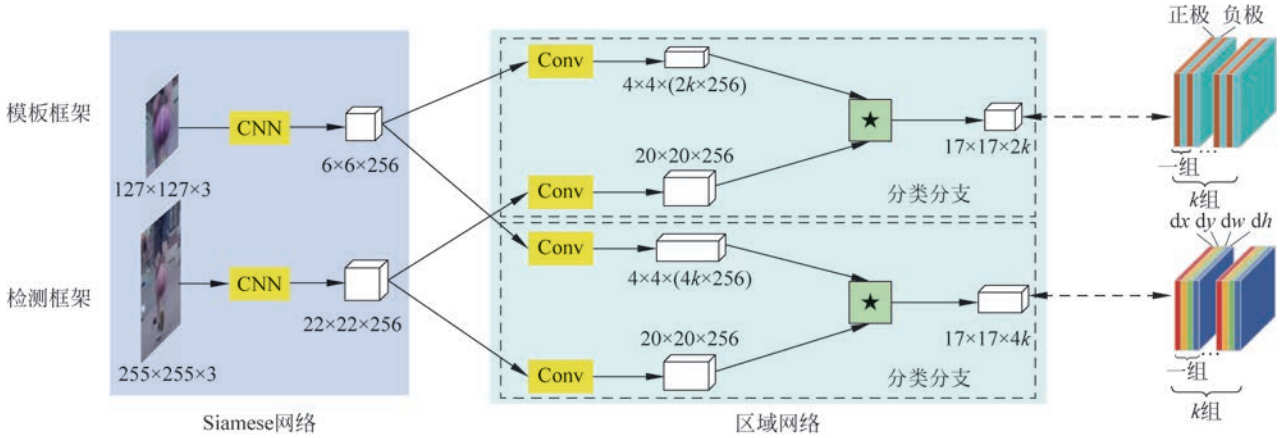

图 3-22　SiamRPN 网络结构图

（图片来源于 Li, et al., 2018）

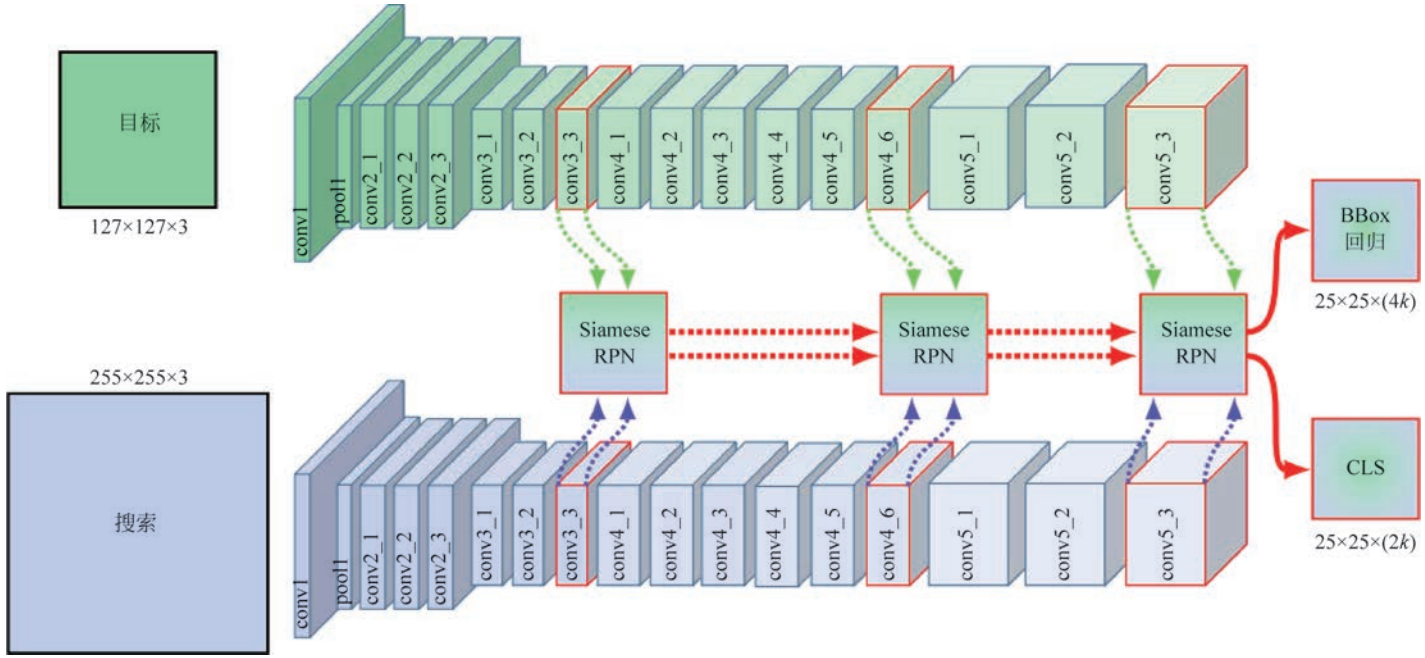

图 3-23　SiamRPN＋＋网络结构

（图片来源于 Li,et al.,2019）

没有填充的网络具有严格的平移不变性，而为了提升性能，加深 CNN 又无法避免填充的出现。作者研究发现，通过在训练样本中人工加入服从均匀分布的随机平移，可在一定程度上打破这种严格平移不变性限制。从模型的预测结果上来看，如果训练数据在一定范围内服从均匀分布，那么理想情况下跟踪器预测的结果也应该更接近均匀分布。通过以上改进，SiamRPN＋＋成为各目标跟踪数据集上的领先算法，基于深度学习的方法在跟踪准确度上达到了最优水平。

SORT 是多目标跟踪中基于检测的跟踪框架，有 4 个基本组件，即目标检测器、状态预测、数据关联和追踪管理，后续的很多研究工作都有类似的框架。SORT 使用 VGG16 主干的 Faster R-CNN 作为目标检测器。对于目标的状态，SORT 简单地使用中心坐标、面积、长宽比以及它们的变化率对目标进行建模，而没有利用任何外观信息。SORT 使用卡尔曼滤波器主动地对目标之后的状态进行预测，并将预测的结果与实际检测到的目标边框进行匹配。

SORT 中追踪与检测的关系被视作二分图，二分图的每一条边的权重由它的两个顶点（分别为一个追踪和一个检测）的 IOU 定义。SORT 使用匈牙利算法在这个二分图中寻找最优匹配，并为匹配设置最小 IOU 阈值，以减少错误的匹配数量。关于追踪的管理，SORT 将匹配失败的追踪保留帧，为匹配失败的检测开启新的追踪并设置其初始状态。

DeepSORT 是 SORT 工作的改进版本，其最大的贡献在于使用了深度 CNN 提取目标的特征以作为匹配标准。DeepSORT 使用 Mahalanobis 距离作为运动特征的相似度标准，以及余弦距离作为外观特征编码的相似度标准，两种相似度通过加权平均来得到总体的相似度。另外，DeepSORT 定义了一种级联式的匹配方法，使得近期活跃度较高的追踪被优先匹配。结果表明，DeepSORT 最近几年在多个公开数据集上一直保持着位于前列的检测性能。

3.6 图像分割典型算法

3.6.1 语义分割算法

最流行的原始方法之一是通过滑动窗口进行块分类，利用每个像素周围的图像块，对每个像素分别进行分类。但是其计算效率非常低，因为不能在重叠块之间重用共享特征。解决方案就是加州大学伯克利分校提出的全卷积网络（Fully Convolutional Network，FCN），它提出了端到端的卷积神经网络体系结构，在没有任何全连接层的情况下进行密集预测，其结构如图 3-24 所示。这种方法允许针对任何尺寸的图像生成分割映射，并且比块分类算法快得多，几乎后续所有的语义分割算法都采用了这种范式。

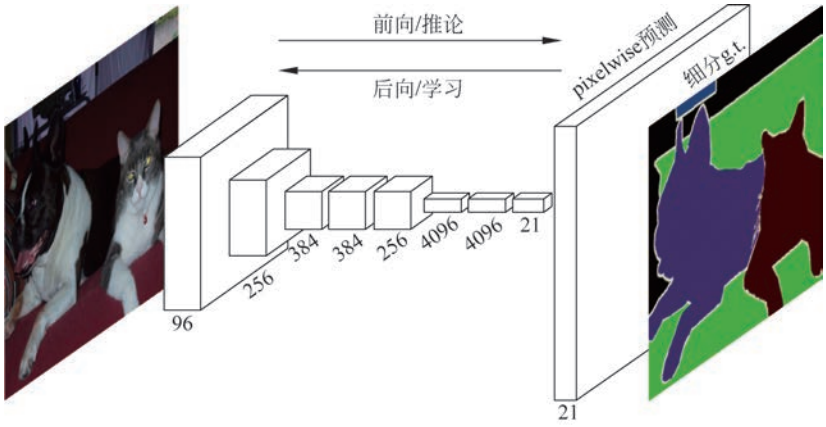

图 3-24　基于 FCN 的语义分割

（图片来源于 Long，et al.，2015）

但是，这也仍然存在一个问题：在原始图像分辨率上进行卷积运算非常昂贵。为了解决这个问题，FCN 在网络内部使用了下采样和上采样：下采样层被称为条纹卷积，而上采样层被称为转置卷积。

语义分割在医学图像分析上有巨大的应用空间，医学图像分析对于视觉任务的要求是严苛的，不仅要识别目标的位置和类别，甚至要求图像中的每个像素都应该有标签。U-Net 网络就是为了医学图像分析而提出的网络结构，如图 3-25 所示，使用了更深的网络结构和跳层连接，能够大幅度提升语义分割的精度。

U-Net 中包含两条串联的路径：压缩路径用来提取图像特征，捕捉上下文，将图像压缩为由特征组成的特征图；扩展路径用来精准定位，将提取的特征解码为与原始图像尺寸一样的分割后的预测图像。

与 FCN 不同的是，U-Net 在上采样过程中保留了大量的特征通道，从而使更多的信息能流入最终复原的分割图像中。另外，为了降低在压缩路径上损失的图像信息，还将压缩路径和扩展路径同尺寸的特征图进行叠加，再继续进行卷积和上采样处理，以此整合更多信息进行图像分割。

此外，U-Net 没有使用 VGG 等 ImageNet 预训练的模型作为特征提取器，原因在于 U-Net 做的是医学图像的二值分割，与 ImageNet 的输出分类完全不同。U-Net 在进行特征融合时，采用的是 Concat，而不是 FCN 中的 Add。Concat 是通道数的增加，Add 是特征图相加，通道数不变。与 Concat 相比，Add 的计算量少很多，但是 Concat 层更多用于不同尺度特征图的语义信息的融合，而 Add 较多使用在多任务问题上。

尽管采用了上采样和下采样层，但由于池化期间的信息丢失，FCN 会生成比较粗糙的分割映射。SegNet 是一种比 FCN（使用最大池化和编码解码框架）更高

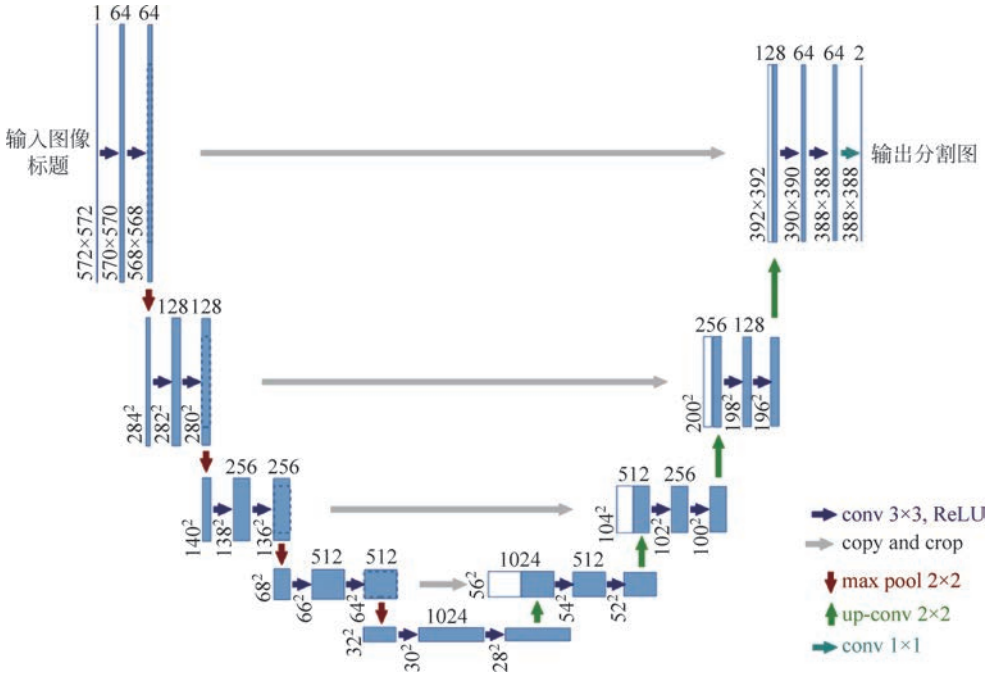

图 3-25 U-Net 网络结构

（图片来源于 Ronneberger，et al.，2015）

效的内存架构，其架构如图 3-26 所示。在 SegNet 解码技术中，从更高分辨率的特征映射中引入了快捷/跳跃连接，以改善上采样和下采样后的粗糙分割映射。

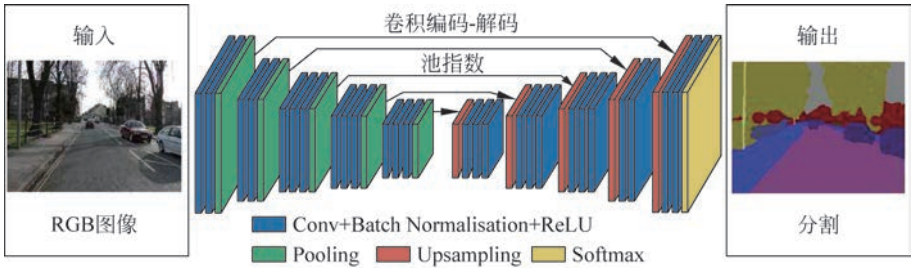

图 3-26 SegNet 网络结构图

（图片来源于 Badrinarayanan，et al.，2017）

SegNet 是一个由编码（左）和解码（右）组成的对称网络。网络根据输入图像中物体的语义信息，把图像中的物体进行分类（如“马路”“汽车”“楼房”等），最后生成一张分割图像。其中，编码本身其实就是一连串的卷积网络，由卷积层、池化层和 Batch Normalization 层组成。卷积层负责获取图像局域特征，池化层对图像进行下采样，并且将尺度不变特征传送到下一层，而 BN 主要对训练图像的分布归一

化，加速学习。解码对缩小后的特征图像进行上采样，然后对上采样后的图像进行卷积处理，目的是完善物体的几何形状，弥补编码中池化层将物体缩小造成的细节损失。

3.6.2　实例分割算法

实例分割既具备语义分割的特点，需要做到像素层面上的分类，又要满足目标检测的需求，定位出相同类别中的不同实例。因此，实例分割可以结合语义分割和目标检测的方法，从不同方向分两个阶段实现实例分割的任务，这类算法的代表是Mask R-CNN。此外，受单阶段目标检测思想的影响，也有单阶段的实例分割算法的研究与实现，如 YOLACT、SOLO 和 PolarMask 等。

Mask R-CNN 通过向 Faster R-CNN 添加一个分支来进行像素级分割，其结构如图 3-27 所示，该分支输出一个二进制掩码，该掩码表示给定像素是否为目标对象的一部分：该分支是基于卷积神经网络特征映射的全卷积网络。将给定的卷积神经网络特征映射作为输入，输出一个矩阵，其中像素属于该对象的所有位置，用 1 表示，其他位置则用 0 表示，也就是二进制掩码。

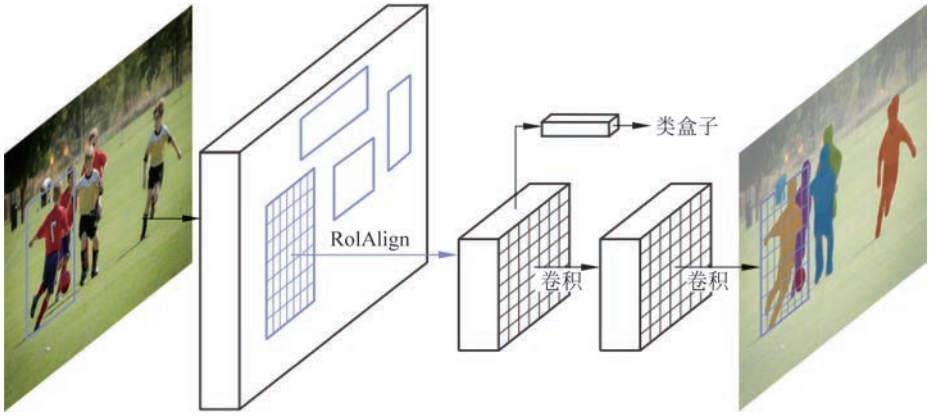

图 3-27　Mask R-CNN 实例分割网络结构

（图片来源于 He，et al.，2017）

另外，当在原始 Faster R-CNN 架构上运行且没有做任何修改时，感兴趣池化区域（RoIPool）选择的特征映射区域和原始图像的区域稍微错开，由于图像分割具有像素级特性，这与边界框不同，自然会导致结果不准确。Mask R-CNN 通过调整 RoIPool 来解决这个问题，使用感兴趣区域对齐（RoIAlign）方法使其变得更精确。本质上，RoIAlign 使用双线性插值来避免舍入误差，这会导致检测和分割不准确。

一旦生成这些掩码，Mask R-CNN 将 RoIAlign 与来自 Faster R-CNN 的分类和边界框相结合，以便进行精确的实例分割。Mask R-CNN 利用 R-CNN 得到的

物体框来区分各个实例，然后针对各个物体框对其中的实例进行分割。显而易见的是，如果框不准，分割结果也会不准。因此，对于一些边缘精度要求高的任务而言，这是一个有待改进的解决方案。

SOLO 属于单阶段实例分割算法，其核心思想是将实例分割问题重新定义为类别感知预测问题和实例感知掩码生成问题，其结构框架如图 3-28 所示。算法首先将图片划分成 $S \times S$ 的网格，如果物体的中心（质心）落在某个网格中，那么该网格就有了两个任务：①分类分支负责预测该物体语义类别；②掩码分支负责预测该物体的实例掩码。这就对应了网络的两个分支。同时，SOLO 在骨干网络后面使用了 FPN，用来应对尺寸。FPN 的每一层后都接上述两个并行的分支，进行类别和位置的预测，每个分支的网格数目也相应不同，小的实例对应更多的网格。

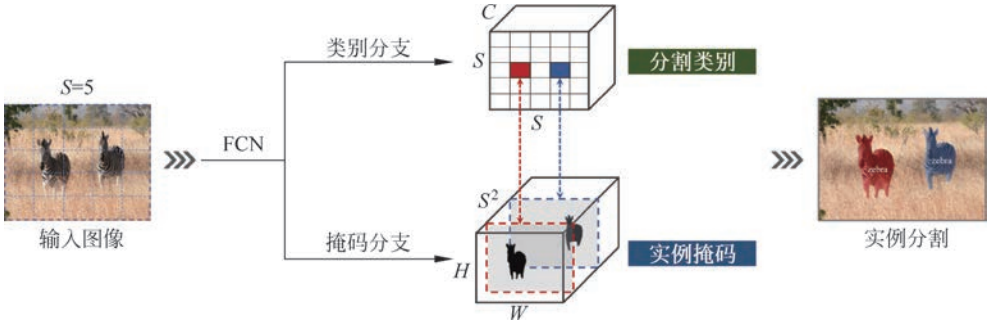

图 3-28　SOLO 实例分割网络核心结构

（图片来源于 Wang，et al.，2021）

参考文献

[1] 阿斯顿，李沐，立顿，等. 动手学深度学习[M]. 北京：人民邮电出版社，2019.

[2] 马飒飒. 人工智能基础[M]. 北京：电子工业出版社，2020.

[3] 贲可荣，张彦铎. 人工智能[M]. 北京：清华大学出版社，2018.

[4] LECUN Y，BOTTOU L，BENGIO Y，et al. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE，1998，86(11)：2278-2324.

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

[6] KRIZHEVSKY A，SUTSKEVER I，HINTON G E. ImageNet classification with deep convolutional neural networks[J]. Communications of the ACM，2017，60(6)：84-90.

[7] ZEILER M D，FERGUS R. visualizing and understanding convolutional networks[C]// European Conference on Computer Vision. Springer，2014：818-833.

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

[9] SIMONYAN K,ZISSERMAN A. Very deep convolutional networks for large-scale image recognition[C]//International Conference on Learning Representations,2015: 1-6.

[10] 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.

[11] JIAO L,ZHANG F,LIU F,et al. A survey of deep learning-based object detection[J]. IEEE Access,2019,7: 128837-128868.

[12] HE K,ZHANG X,REN S,et al. Spatial pyramid pooling in deep convolutional networks for visual recognition[C]//European Conference on Computer Vision,2014: 346-361.

[13] GIRSHICK R,DONAHUE J,DARRELL T,et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C]//IEEE Conference on Computer Vision and Pattern Recognition,2014: 1-8.

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

[15] REN S,HE K,GIRSHICK R,et al. Faster R-CNN: towards real-time object detection with region proposal networks[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence,2017,39(6): 1440-1448.

[16] REDMON J,FARHADI A. YOLO9000: Better,Faster,Stronger[C]//IEEE Conference on Computer Vision and Pattern Recognition,IEEE,2017: 7263-7271.

[17] REDMON J,FARHADI A. YOLOv3: An incremental improvement[J]. arXiv preprint. arXiv: 1804.02767,2018.

[18] LIU W,ANGUELOV D,ERHAN D,et al. SSD: Single shot multibox detector[C]// European Conference on Computer Vision,2016: 21-37.

[19] 言有三. 深度学习之模型设计：核心算法与案例实践[M]. 北京：电子工业出版社,2020.

[20] 李玺,查宇飞,张天柱,等. 深度学习的目标跟踪算法综述[J]. 中国图象图形学报,2019,24(12): 2057-2080.

[21] 张继凯,赵君,张然,等. 深度学习的图像实例分割方法综述[J]. 小型微型计算机系统, 2021,42(1): 161-171.

[22] KIRILLOV A,HE K,GIRSHICK R,et al. Panoptic segmentation[C]//IEEE Conference on Computer Vision and Pattern Recognition,IEEE,2019: 9404-9413.

[23] MINAEE S,BOYKOV Y,PORIKLI F,et al. Image segmentation using deep learning: A survey[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence,2021.

[24] BERTINETTO L,VALMADRE J,HENRIQUES J F,et al. Fully-convolutional siamese networks for object tracking[C]//European Conference on Computer Vision, 2016: 850-865.

[25] LI B,YAN J,WU W,et al. High performance visual tracking with siamese region proposal network[C]//IEEE Conference on Computer Vision and Pattern Recognition,2018: 8971-8980.

[26] LI B,WU W,WANG Q,et al. SiamRPN++: Evolution of siamese visual tracking with very deep networks[C]//IEEE Conference on Computer Vision and Pattern Recognition, 2019: 4282-4291.

[27] LONG J, SHELHAMER E, DARRELL T. Fully convolutional networks for semantic segmentation[C]//IEEE Conference on Computer Vision and Pattern Recognition, 2015: 3431-3440.

[28] RONNEBERGER O, FISCHER P, BROX T. U-Net: Convolutional networks for biomedical image segmentation[C]//International Conference on Medical Image Computing and Computer-assisted Intervention, 2015: 234-241.

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

[30] HE K, GKIOXARI G, DOLLAR P. Mask R-CNN[C]//IEEE International Conference on Computer Vision, 2017: 2961-2969.

[31] WANG X, ZHANG R, SHEN C, et al. SOLO: a simple framework for instance segmentation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2021.