第 1 章 预备知识

近年来，深度学习在多个领域取得了显著进展，其中计算机视觉无疑是最为突出的应用之一。作为图像处理和计算机视觉中的关键任务，图像分割在深度学习的推动下，正经历从传统的图像处理技术向深度学习方法主导的重大转变。本章作为全书的基础部分，旨在阐述深度学习、计算机视觉和图像分割的核心概念，并介绍书中所使用的主要工具和框架。此外，本章还将概述全书的内容结构和章节安排，帮助读者更好地掌握阅读本书所需的基础知识。

1.1 深度学习与计算机视觉

在近 10 年里，以机器学习（Machine Learning, ML）和深度学习（Deep Learning, DL）为代表的人工智能（Artificial Intelligence, AI）技术一直是学术界和工业界追捧的热点。2022 年，OpenAI 推出新一代由大型语言模型（Large Language Models, LLM）驱动的 AI 对话系统 ChatGPT 之后，大模型、生成式人工智能（Generative AI）和通用人工智能（Artificial General Intelligence, AGI）等概念将 AI 的发展推向了新的高潮。从我们的日常生活来看，AI 技术也普及到各个角落，从高铁站闸机的人脸识别系统到汽车最新的自动驾驶辅助，从手机的语音交互功能到实时的机器翻译，从个性化的用户推荐到各种电子竞技游戏，从 AI 绘画到视频生成，无一不体现了 AI 技术的快速发展。

人工智能、机器学习和深度学习三者之间有着层级递进的关系（如图 1-1 所示）。人工智能是一个广泛的领域，而机器学习是其核心分支之一，深度学习则是机器学习中的关键技术。相较于传统的机器学习模型，如线性模型、决策树、支持向量机和集成学习等，深度学习则聚焦于神经网络（neural networks）的研究与应用。深度学习依靠高度层次化的神经网络结构、大量的训练数据以及日益增长的计算资源，得以在近几年大放异彩，在很多应用场景中效果都超越了传统的机器学习方法。

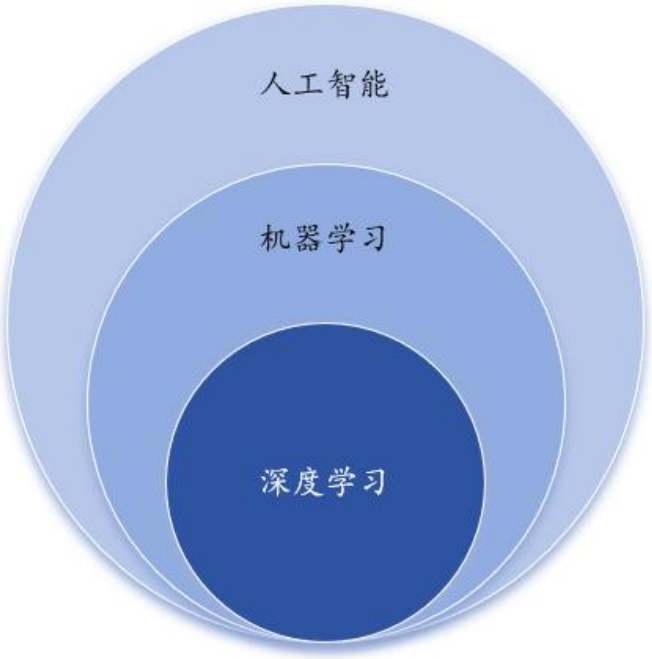

图 1-1 人工智能、机器学习与深度学习的关系

在深度学习的众多应用中，计算机视觉（Computer Vision, CV）无疑是研究最为广泛且深入的一个方向。图像和视频已经成为我们生活中不可或缺的内容载体，而计算机视觉的目标则是赋予计算机“看”与“理解”图像的能力，从而识别和分析视觉数据。通过深度学习的神经网络模型，计算机能够通过大量的数据训练，准确地从图像中提取信息。这项技术不仅仅改变了传统的图像处理方法，更推动了自动驾驶、医学影像诊断、安防系统等多个领域的应用革新。基于深度学习的 CV 技术也不断由卷积神经网络（Convolutional Neural Networks, CNN）、Transformer 再到如今的多模态融合等趋势演变。

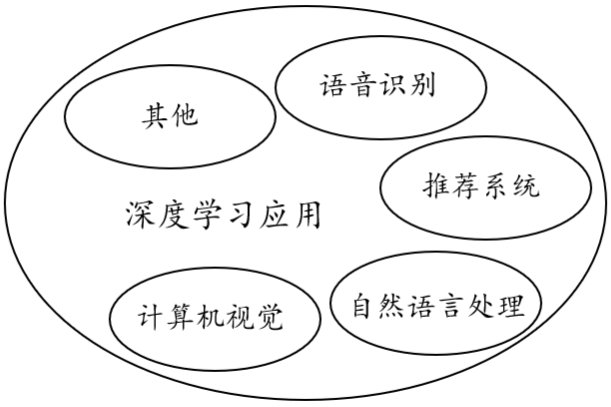

图 1-2 深度学习的应用领域

1.2 计算机视觉的三大任务

传统的计算机视觉应用方向广泛，单从应用上来看不局限于三大任务，所以这里的“计算机视觉的三大任务”指的是基于深度学习的三大任务，其他各细分研究方向大多都是在此三大任务上延伸展开。

- 图像分类（image classification）聚焦于整张图像，对整张图片进行分类，判断图中主要目标物体是猫还是狗，如图 1-3 左图所示。
- 目标检测（object detection）定位于图像具体区域，判断图中每个目标物体的类别及其所在图中的位置。如图 1-3 中间的图所示，需要判断猫和狗的类别及其对应的坐标位置。
- 图像分割（image segmentation）细化到每一个像素，判断每个物体所属的像素边界，如图 1-3 右图所示，将猫和狗的像素边界描绘分割出来。

可见三大任务之间明显存在着一种递进的层级关系，即从图像整体到图像中的局部对象再到图像中每一个像素，如图 1-3 所示。

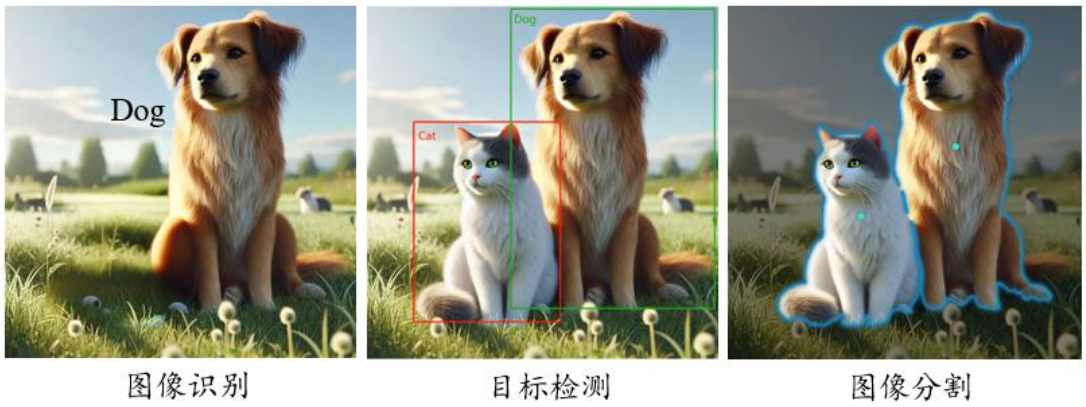

图 1-3 计算机视觉三大任务

本节简单回顾一下近年来这三大任务的研究进展情况。

1.2.1 图像分类

从传统的图像处理和机器学习的角度来看，图像分类可以看作一种模式识别、决策理论以及统计分类器的应用范畴。深度学习大规模介入图像分类领域是在

2012 年的 ILSVRC（ImageNet 大规模视觉识别挑战赛，如图 1-4 所示），当年由 Alex Krizhevsky 提出的 AlexNet 将该项比赛 Top 5 错误率降至 16.4%，这一成绩引起了当时学术界和工业界的极大关注。

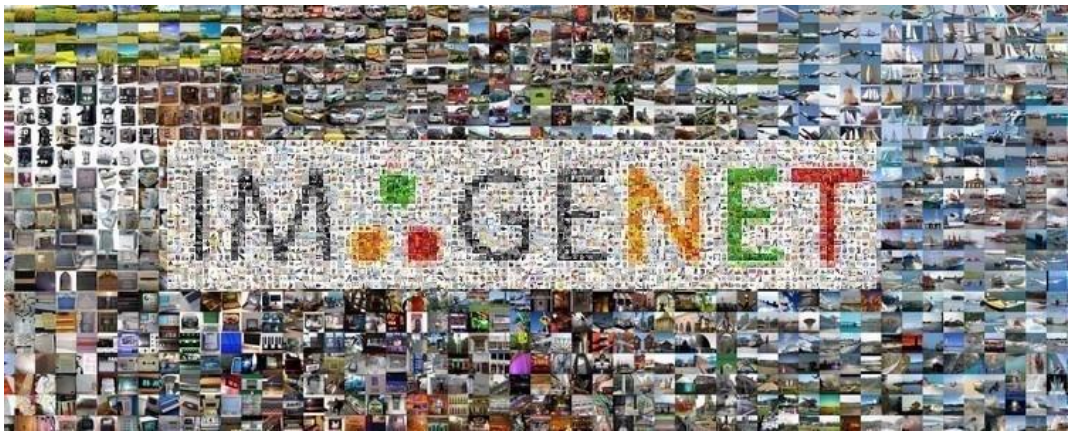

图 1-4 ImageNet 项目

在此后历年的 ILSVRC 挑战赛上，都有层出不穷的深度卷积网络结构设计提出，比如 2014 年的 Inception 和 VGG、2015 年的 ResNet、2016 年的 ResNeXt、2017 年的 SENet 等，图像分类正式进入深度学习主导的时代。ILSVRC 挑战赛历年经典模型如表 1-1 所示。

表 1-1　ILSVRC 历年经典模型

时间	网络名称	Top 5 成绩（错误率）	论文
2012	AlexNet	16.42%	*ImageNet Classification with Deep Convolutional Neural Networks*
2013	ZFNet	13.51%	*Visualizing and understanding convolutional networks*
2014	GoogLeNet	6.67%	*Going Deeper with Convolutions*
	VGG	6.8%	*Very deep convolutional networks for large-scale image recognition*
2015	ResNet	3.57%	*Deep Residual Learning for Image Recognition*
2016	ResNeXt	3.03%	*Aggregated Residual Transformations for Deep Neural Networks*
2017	SENet	2.25%	*Squeeze-and-Excitation Networks*

得益于 ImageNet 项目和大规模计算资源的快速发展，各种图像分类的网络结构层出不穷，不断刷新着 ImageNet 上的 SOTA（state of the art）表现。其中，

像 NasNet、DenseNet 和 EfficientNet 等网络在提高图像分类精度的同时也兼顾了效率和复杂度。从算法设计的角度来看，层级化的 CNN 是深度学习图像分类任务的核心基础。CNN 是一种深度前馈神经网络，通过卷积计算实现自动化的特征提取，主要由卷积层、池化层和全连接层构成，能够有效提取图像中的语义信息，并将这些信息浓缩后输入分类器进行训练。CNN 的核心优势在于它可以自动捕捉图像中的局部特征，并通过层层卷积逐步提取更高层次的语义信息。经过这些年的发展，许多经典的网络架构对计算机视觉领域产生了深远的影响。例如，VGG 网络提出了简洁的平铺式（plain）卷积结构，通过堆叠多个相同大小的卷积层来提高模型深度；ResNet 引入残差连接（residual connections），解决了深度网络中的梯度消失问题；DenseNet 通过密集连接（dense connections）实现了跨层特征复用，进一步提高了模型的性能和效率；Inception 系列则通过 1×1 卷积来减少参数量，同时提高计算效率；SENet 通过提出“squeeze and excitation”模块，在特征通道之间引入了自适应加权机制，进一步提升了模型的表示能力；而 EfficientNet 通过复合缩放策略在网络深度、宽度和分辨率三个维度上进行平衡扩展，实现了更高效的模型性能与精度的提升。这些网络架构的创新不仅推动了图像分类任务的进步，也为其他下游的计算机视觉任务，如目标检测、语义分割、图像生成等，提供了用于提取图像特征的骨干网络。比如图像分割的 UNet 网络可以使用 VGG16 作为编码器的骨干网络，Deeplab v3 使用 ResNet-101 作为其编码器的主干网络等。

此外，基于注意力机制（attention）的 Transformer 结构在计算机视觉任务中展现了强大的能力。基于 Vision Transformer（ViT）的图像分类是近年来深度学习图像分类领域的一个重要突破，它打破了长期以来 CNN 主导的局面，将 Transformer 模型的优势引入到图像处理任务中。ViT 的设计思想源自自然语言处理（Natural Language Processing, NLP）中的 Transformer 架构，它通过自注意力机制（self-attention）来捕捉序列中的全局关系。与 CNN 不同，CNN 是通过局部卷积操作来提取图像的局部特征，而 ViT 直接将图像处理为一个序列，然后利用自注意力机制来学习图像中不同区域之间的全局依赖关系。与 CNN 作为骨干网络相似的是，ViT 也可以作为骨干网络用于下游的视觉任务，比如用于检测的 DETR，用于分割的 TransUNet 等。

目前基于深度学习的图像分类技术已广泛应用于各行各业，比如医学影像的自动分类筛查；在智能监控中，图像分类技术用于分类异常行为、识别特定人物或物体，实现更精准的安全监控；在电商平台上，图像分类可以用于自动识别和分类商品，提升用户体验；在农业生产中，图像分类可以基于农业图像，识别作物的健康状况，检测病虫害等，通过无人机或地面摄像设备采集图像，深度学习模型可以快速分类并判断作物是否受到病害影响等。

1.2.2 目标检测

随着深度学习在图像分类领域的迅速普及，许多基于图像分类的下游视觉任务得到了蓬勃发展，其中目标检测是一个典型的应用。简而言之，目标检测不仅需要识别图像中的物体类别，还要精确定位这些物体在图像中的具体位置。自从深度学习技术崛起后，CNN 逐渐取代了传统的目标检测算法，成为该领域的主导方法。从两阶段（two-stage）的 R-CNN 系列算法到单阶段（one-stage）的 YOLO 系列算法，基于 CNN 的目标检测方法不断提升精度和速度，持续刷新该领域的 SOTA。

两阶段目标检测算法的代表是 R-CNN 系列，其核心思想是先生成图像中的候选区域（proposal regions），然后对这些区域进行分类。R-CNN 系列包括了多种重要的改进算法，如 R-CNN、SPP-Net、Fast R-CNN、Faster R-CNN 和 Mask R-CNN 等。这类方法通过分离候选区域的生成和分类过程，极大地提高了检测精度。R-CNN 最初使用选择性搜索算法生成候选区域，再通过 CNN 对每个区域进行分类和回归；SPP-Net 引入了空间金字塔池化（Spatial Pyramid Pooling, SPP），提高了处理不同大小图像的能力；Fast R-CNN 则进一步优化了训练速度和空间利用效率；Faster R-CNN 通过引入区域候选网络（Region Proposal Network, RPN），将候选区域生成与分类整合为一个网络，大幅提升了效率；而 Mask R-CNN 则在 Faster R-CNN 的基础上添加了分割任务，使其不仅可以进行目标检测，还能实现精确的实例分割。图 1-5（a）展示了两阶段目标检测网络的基本结构。这类网络首先通过一个专门的模块生成候选区域，然后将这些候选区域送入 CNN 进行进一步的特征提取和分类，最终完成目标检测任务。

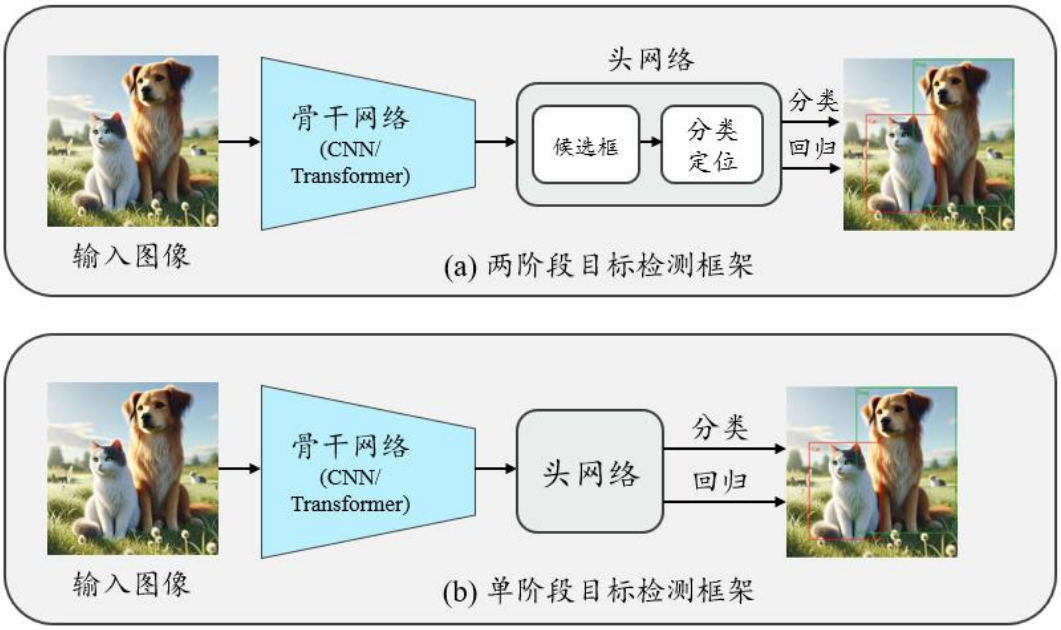

图 1-5 两阶段和单阶段目标检测框架

虽然两阶段算法在检测精度上表现优异，但其多步骤的处理过程导致推理速度相对较慢，难以满足实时性场景下的使用需求。为了解决这一问题，YOLO（You Only Look Once）系列算法（单阶段目标检测算法）提供了一种简化的方案，其核心思想是在输入图像后直接输出目标物体的类别和对应的坐标位置，而不再像两阶段算法那样先生成候选框。这种“一步到位”的设计使得 YOLO 系列算法在速度上具备显著优势，能够很好地满足实时检测的需求。YOLO 系列算法包括多个版本的更新迭代，截止到本书完成时已有 10 个版本，每个版本都在精度和速度之间取得了更好的平衡，具体如表 1-2 所示。其简化的网络结构如图 1-5（b）所示，展现了单阶段目标检测的简化流程。YOLO 系列通过将整个图像划分为网格，每个网格直接预测目标物体的类别以及相应的边界框坐标，这种设计显著提高了检测速度。

表 1-2 YOLO v1-10 各版本汇总

版本	时间	特点	论文
v1	2016	初代模型，将目标检测作为边界框预测和类别概率的单一回归问题。	*You Only Look Once: Unified, Real-Time Object Detection*
v2	2016	引入了 Anchor boxes、多尺度检测、批归一化等技术	*YOLO9000: Better, Faster, Stronger*

v3	2018	引入了 FPN 结构进行多尺度检测，支持检测小目标，使用逻辑回归进行分类。	*YOLOv3: An Incremental Improvement*
v4	2020	引入了 CSPNet 和 Mosaic 数据增强等优化技术，提升了训练效率和推理速度。	*YOLOv4: Optimal Speed and Accuracy of Object Detection*
v5	2020	便捷的部署和轻量化设计，并支持导出到多种平台。	/
v6	2021	更加注重小型化和轻量化，适用于资源受限的设备。	*YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications*
v7	2022	引入了动态标签分配和新型损失函数设计，在实时性能和精度之间取得平衡。	*YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors*
v8	2023	集成了语义分割和检测功能，进一步优化了模型推理速度和精度。	/
v9	2023	继续减少冗余，提高计算效率，并增强推理速度。	*YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information*
v10	2024	引入了 NMS-free 训练策略和双标签分配机制，进一步提升了检测精度和推理效率。	*YOLOv10: Real-Time End-to-End Object Detection*

相比于传统的图像分类任务，目标检测的难点在于图像中聚焦的是局部目标物体，而不仅仅是整体图像。检测算法不仅要识别物体的类别，还要准确定位这些物体的位置。这就要求模型能够同时兼顾全局和局部信息，且在各种复杂场景中保持较高的检测精度。基于深度学习的目标检测技术如今已经在多个领域中得到了广泛应用，尤其是在安防监控、自动驾驶、工业制造等领域。安防领域的摄像头监控系统利用目标检测技术自动识别人群中的异常行为；自动驾驶领域则依赖目标检测技术实时识别道路上的行人、车辆和交通标志；在工业制造中，目标检测用于自动化检测生产线上是否有缺陷产品。这些成熟的应用证明了目标检测技术的有效性和可靠性。

1.2.3 图像分割

在目标检测的基础上，图像分割进一步细化了任务的粒度，不再仅仅是对物

体进行分类和定位，而是深入到图像中的每一个像素点，进行像素级别的分类。因此，图像分割不仅识别图像中的物体，还能够精确地勾勒出每个物体的轮廓，提供更加细致的视觉理解。由于图像分割是本书的主题，在深入介绍基于深度学习的分割算法之前，有必要首先对传统的图像分割方法进行梳理和简要介绍。从图像处理的角度来看，图像分割的核心在于将图像划分为构成它的不同子区域或独立的物体。这一过程主要依赖于图像灰度值的两个基本特性：不连续性和相似性。对于不连续的灰度值，传统的图像分割方法通常基于灰度值的突变进行处理，例如通过检测图像中明显的边界来分割区域。而对于相似的灰度值，传统方法则根据预定义的规则将图像划分为一组具有相似特征的区域。 传统的图像分割方法主要包括 5 个大的分类：

（1）基于边缘检测的图像分割算法：这种方法通过识别图像中灰度值发生明显变化的区域（即边缘）来进行分割。这类方法的优点是能够有效识别物体的边界，但在处理噪声较大的图像时可能表现不佳。

（2）基于阈值处理的图像分割算法：这种方法通过设定一个或多个灰度阈值，将图像划分为若干部分。该方法简单且计算效率高，但适用于图像灰度分布相对均匀的场景，对于复杂图像的分割效果有限。

（3）基于区域生长的图像分割算法：该方法从图像中的种子像素开始，通过逐步扩展相似区域来完成分割。这类方法在保持区域连贯性方面效果较好，但计算复杂度较高，且对初始种子选择较为敏感。

（4）基于形态学的图像分割算法：基于数学形态学的运算（如膨胀、腐蚀、开运算、闭运算等）来处理图像中的结构元素。这类方法常用于处理二值图像的分割问题，适合处理噪声和形状不规则的目标。

（5）基于图论的图像分割算法：通过构建图结构来表示图像，将像素视为图的节点，基于图的切割方法进行分割。这类方法灵活性较高，但通常计算复杂度较大，适用于对精度要求较高的场景。

上述图像分割算法各有优势与局限，在接下来的第 2 章中，我们将对这些算法进行更加系统的回顾与举例，为后续深度学习图像分割技术的介绍奠定基础。图 1-6 为 OTSU 算法（一种基于阈值的图像分割算法）的分割效果，其中左图为原图，右图为 OTSU 算法分割效果。

图 1-6 OSTU 算法图像分割效果

相较于目标检测只关注图像中的局部区域，基于深度学习的图像分割则更为精细，它将每一个像素点作为研究对象，并对每个像素赋予一个语义标签。因此，图像分割的一个基本类型称为语义分割（semantic segmentation）。除了语义分割，实例分割（instance segmentation）和全景分割（panoptic segmentation）两种其他重要的分割方式也有广泛的应用。那么，它们之间有什么区别和联系呢？

（1）语义分割的任务是对图像中的每个像素进行分类，并将具有相同类别的像素统一标记为同一个标签。也就是说，它只关注像素属于什么类别，而不关心属于同一类别的不同个体之间的区别。例如，如果图像中有多个人，语义分割的结果是所有人的像素都会被统一标记为“人”这个类别，而不会区分他们是不同的个体。语义分割的重点是整体场景的理解，适用于对场景分类有需求的任务。

（2）实例分割在语义分割的基础上更进一步，不仅对每个像素进行分类，还要区分同一类别中的不同个体。它结合了语义分割和目标检测的特性，既需要对物体进行像素级别的分割，又要标识每个对象的独立性。例如，在一张图像中有多个人，实例分割的结果不仅会标记出“人”这个类别，还会将不同的个体（每个人）单独分割出来，即使他们属于同一类“人”。

（3）全景分割结合了语义分割和实例分割的优势，是最全面的分割任务类型。它不仅能够像实例分割那样区分可数的物体个体（如多个“人”或“车”），还能够像语义分割那样对不可数的背景部分（如天空、草地）进行分类。因此，全景分割旨在提供对整个场景的完整理解，既能区分出不同的实例对象，又能对

整个图像中所有像素进行有效的分类，无论这些像素是否属于可数的对象。

可见，实例分割和全景分割都是以语义分割技术为基础的。语义分割、实例分割和全景分割具体如图 1-7（b）、（c）和（d）图所示。

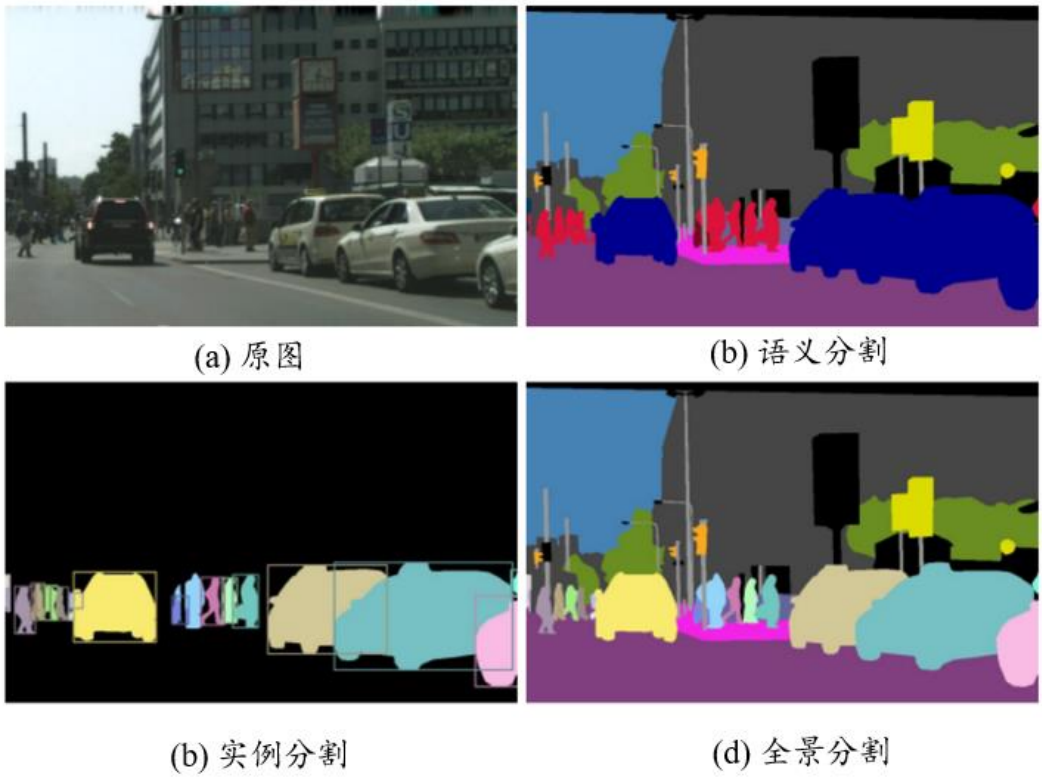

图 1-7 语义分割、实例分割与全景分割

实例分割和全景分割的基础都是语义分割，因此本书将以语义分割为核心展开讨论。在语义分割发展的早期，为了使深度学习能够处理像素级别的分类任务，研究人员对传统的 CNN 进行了一些调整，将分类网络中的全连接层去除，取而代之的是全卷积网络（Fully Convolutional Networks, FCN）。这种网络结构能够生成密集的像素级预测，从而为语义分割问题提供了一个有效的解决方案。随着研究的深入，UNet 的提出为语义分割奠定了基于“编解码”结构的 U 形网络的基础。UNet 是一种基于 CNN 的网络结构，主要由三个部分组成：编码器（encoder）、解码器（decoder）和跳跃连接（skip connection）。由于其网络架构呈现出 U 形，因此被命名为 UNet。编码器负责提取图像中的特征，逐层下采样，而解码器则通过逐层上采样逐步恢复图像的空间分辨率。跳跃连接则用于将编码器的低层次特征与解码器对应的高层次特征相结合，以提升分割精度。语义分割迄今为止最

重要的两个设计有两个：

（1）U 形编解码结构。以 UNet 为代表的 U 形结构已经成为语义分割领域的主导模型架构。基于 UNet 的改进层出不穷，例如应用于 3D 图像的 VNet，以及嵌套式 UNet 结构的 UNet++。这些创新进一步增强了网络对不同任务和场景的适应能力。此外，还有诸如 SegNet、RefineNet、HRNet 和 FastFCN 等变体，这些网络在 UNet 的基础上对编码解码架构进行了优化，进一步提升了分割的精度和效率。

（2）空洞卷积设计。以 DeepLab 系列为代表的设计引入了空洞卷积（dilated convolutions），使得网络在不增加参数和计算量的情况下扩大感受野，从而更好地捕捉上下文信息。除了 DeepLab 系列，PSPNet 通过金字塔池化模块来进一步增强对多尺度上下文信息的建模能力。这些设计使得语义分割网络在应对复杂场景时表现更加出色。随着模型的基线（baseline）效果不断提升，语义分割任务的主要矛盾也逐渐从下采样损失恢复像素逐渐演变为如何更有效地利用上下文信息。

在近年来的研究中，Transformer 结构被引入语义分割领域，这使得基于 ViT 的语义分割网络设计逐渐流行。传统的 CNN 虽然具备强大的特征提取能力，但由于其平移不变性和对长距离依赖建模能力的限制，使其在捕捉全局上下文信息时存在一定的局限性。而 Transformer 擅长处理长距离依赖，能够更好地捕捉序列中的全局信息，因此逐渐受到关注并在图像分割领域得到广泛应用。例如，像 SETR（Segmentation Transformer）、TransUNet、SwinUNet、TransAttUNet 和 LeViT-UNet 等模型，结合了 Transformer 的全局特性和 CNN 的局部特性，展示了 Transformer 在语义分割任务中的巨大潜力。这些模型通过引入 Transformer 架构，大幅提升了网络捕捉全局信息的能力，进一步推动了语义分割技术的发展。

除此之外，大模型等相关概念的兴起也正在对深度学习图像分割领域产生着剧变，正在推动图像分割研究向更智能、更通用的方向发展。随着分割一切模型（Segment Anything Model, SAM）的提出，交互式分割（interactive segmentation）、可提示分割（promptable segmentation）等概念和范式对图像分割的研究产生了深刻的影响，相关图像分割基座模型（foundation models）不断涌现。SAM 是一种基于大模型的图像分割技术，旨在通过一个通用模型来解决几乎所有图像分割任

务。这种模型不仅能够处理传统的语义分割和实例分割任务，还可以根据用户的输入和提示，对几乎任何类型的图像或对象进行分割。SAM 的核心目标是实现“分割一切”，不再局限于某种特定的任务或数据集。未来的分割模型将不仅仅是用于特定的分割任务，而是能够灵活适应多样化的分割需求，甚至可能在跨领域应用中取得显著进展。

1.3 图像分割代码实现工具

本书代码实现涉及到 Python、C++和 JavaScript 三种编程语言，其中 Python 主要用于深度学习图像分割的代码实现和案例演示，主要框架基于 PyTorch，是全书的主要实现工具。C++主要用于基于 OpenCV 的传统图像分割算法的实现，以及部分深度学习图像分割模型的部署。而 JavaScript 仅在深度学习图像分割模型的 web 端部署时会涉及到。

1.3.1 OpenCV

OpenCV (Open Source Computer Vision Library) 是一个由 Intel 在 1999 年发起的开源跨平台计算机视觉和机器学习库，旨在为图像处理和计算机视觉应用提供高效的工具。OpenCV 用 C/C++编写，同时也提供了 Python、Java、MATLAB 等其他语言的接口，是目前计算机视觉领域最广泛使用的库之一。OpenCV 提供了非常丰富的图像处理和计算机视觉算法，包括图像处理、特征检测、视频分析、三维重建、机器学习、深度学习等，强大的功能支持、完善的技术生态和开源的社区性质，使得 OpenCV 在自动驾驶、工业视觉、医学影像、智能安防等领域内都有着广泛的应用。OpenCV 为传统的图像分割算法也提供了大量可直接调用的函数，本节仅对 OpenCV 做一个简单介绍，基于 OpenCV 的图像分割算法我们将在第 2 章进行详细的介绍。

OpenCV 图像处理模块 imgproc 涵盖了常用的图像处理功能，该模块主要包括基础部分（basic）、变换部分（Transformations）、直方图部分（Histograms）、绘制轮廓部分（Contours）和其他部分（Others）。代码 1-1 是一个 C++版本的 OpenCV 代码示例，主要用于读取、展示和保存图像。

代码 1-1　OpenCV 代码示例（C++）

```
// 包含 OpenCV 的头文件
#include <opencv2/opencv.hpp>
#include <iostream>

int main(int argc, char** argv)
{
    // 读取图像
    cv::Mat img = cv::imread("example.jpg");

    // 检查图像是否加载成功
    if (img.empty())
    {
        std::cout << "图像加载失败，请检查路径。" << std::endl;
        return -1;
    }

    // 创建窗口
    cv::namedWindow("显示图像", cv::WINDOW_AUTOSIZE);

    // 在窗口中展示图像
    cv::imshow("显示图像", img);

    // 等待用户按下任意键后关闭窗口
    cv::waitKey(0);

    // 另存为 PNG 格式
    cv::imwrite("example.png", img);

    return 0;
}
```

代码 1-1 对应的 Python 版本写法如代码 1-2 所示。

代码 1-2　OpenCV 代码示例（Python）

```
# 导入 opencv 库
import cv2
```

```
import sys
# 读取图片
img = cv2.imread("example.jpg")
if img is None:
    sys.exit("Could not read the image.")
# 显示图片
cv2.imshow("Display window", img)
# 保存图片为 png 格式
cv2.imwrite("example.png", img)
```

对上述读取图像的例子分别给出了 C++和 Python 的代码版本，本书中涉及图像处理的例子一般都用 C++代码给出，涉及深度学习的图像分割都用 Python 代码给出。C++具有很高的执行速度，能够更好地利用计算机硬件的性能。对于计算密集型的图像处理任务（如实时处理、大规模数据处理、3D 图像处理），C++的执行效率通常远高于 Python。但 Python 的代码量更少，能够快速实现功能，适合快速原型设计、算法测试和实验验证，并且对于机器学习和深度学习的支持，Python 远超 C++。在实际的项目开发中，很多时候我们会选择 C++和 Python 结合的方案。例如，利用 C++实现高性能的底层算法，而通过 Python 实现上层的调用和业务逻辑，既兼顾了性能，也提高了开发效率。

1.3.2 PyTorch

PyTorch 是由 Facebook AI Research (FAIR)团队开发的开源深度学习框架，于 2016 年发布。凭借其动态计算图的特性、直观易用的 API、强大的调试能力，以及丰富的社区支持，PyTorch 迅速崛起，成为研究和应用领域中的主流深度学习框架之一。近年来，PyTorch 的影响力甚至已经超越了 TensorFlow 等框架，成为许多研究人员和开发者的首选工具。从功能上看，PyTorch 以动态计算图为基础，使得代码调试更为灵活，强大的张量计算能力和全面的 API 支持，使得 PyTorch 在 CV、NLP 和强化学习（Reinforcement Learning, RL）等领域都有着广泛的研究应用。除了在研究领域的广泛应用，PyTorch 在生产环境中的表现也十分出色。通过 TorchScript，PyTorch 允许将动态计算图转换为可在生产环境中高

效运行的静态图，适用于大规模部署。此外，PyTorch 支持与 ONNX（Open Neural Network Exchange）的集成，可以方便地与其他框架互操作，确保模型的跨平台兼容性。这些特性使得 PyTorch 不仅能满足研究中的灵活需求，还能在工业级应用中实现稳定和高效的部署。

本书主要关注 PyTorch 在搭建一个深度学习图像分割项目过程中的主要范式及其相关功能，包括数据流的导入（DataLoader）、数据增强（torchvision.transforms）、模型搭建（torch.nn）、模型训练、验证和推理，以及模型部署（libtorch）等方法流程。代码 1-3 是使用 PyTorch 搭建的一个 LeNet5 网络示例，可以看到基于 torch.nn 模块能够非常快速的搭建网络模型。

代码 1-3　PyTorch LeNet5 网络搭建示例

```
# 导入 torch.nn 相关模块
import torch.nn as nn
import torch.nn.functional as F
# 搭建一个 LeNet5
class Net(nn.Module):
    def __init__(self):
        super(Net, self).__init__()
        # 卷积层 1
        self.conv1 = nn.Conv2d(3, 6, 5)
        # 池化层
        self.pool = nn.MaxPool2d(2, 2)
        # 卷积层 2
        self.conv2 = nn.Conv2d(6, 16, 5)
        # 全连接层 1
        self.fc1 = nn.Linear(16 * 5 * 5, 120)
        # 全连接层 2
        self.fc2 = nn.Linear(120, 84)
        # 全连接层 3
        self.fc3 = nn.Linear(84, 10)

    # 定义网络的前向推理流程
    def forward(self, x):
        x = self.pool(F.relu(self.conv1(x)))
        x = self.pool(F.relu(self.conv2(x)))
```

```
        x = x.view(-1, 16 * 5 * 5)
        x = F.relu(self.fc1(x))
        x = F.relu(self.fc2(x))
        x = self.fc3(x)
        return x
```

本节仅对 PyTorch 做概要性介绍，从本书第 13 章的实战部分开始，我们将对 PyTorch 在深度学习图像分割项目中具体的使用方式进行详细的介绍和举例。如果想要更深入的学习掌握 PyTorch，读者可以参考 PyTorch 的官方教程：https://pytorch.org/tutorials/。

1.4 本书内容安排

本书定位于深度学习图像分割，主要从理论和实战两个方面来进行介绍。全书分为理论和实战两个部分，除第 1 章作为预备知识和第 16 章进行全书总结之外，理论部分总共 11 章（第 2-12 章），实战部分总共 3 章（第 13-15 章）。全书总共 16 章，各章内容安排如下。

第 1 章（本章）为预备知识。主要对深度学习、计算机视觉、图像分割等相关概念和知识脉络进行梳理，对全书用到的两个主要工具 OpenCV 和 PyTorch 进行简单介绍，并对全书内容安排进行提要。

第 2~12 章为全书理论部分。其中第 2 章对传统图像分割算法和 OpenCV 应用进行介绍；第 3 章对深度学习图像分割进行总揽，包括语义分割、实例分割和全景分割，对基于深度学习的图像分割任务进行详细阐述；第 4~11 章为全书理论核心内容，对深度学习图像分割的技术组件、三种代表性的分割网络结构设计、3D 图像分割、实例分割、图像分割大模型和非全监督分割等进行详细阐述；第 12 章则是梳理图像分割领域的经典数据集，为实战部分提供数据基础。

第 13~15 章为全书实战部分。实战部分则是基于 PyTorch 进行全流程的深度学习图像分割项目展示。其中第 13 章主要是对 PyTorch 图像分割代码框架进行演示，基于 PASCAL VOC 公开数据集，对图像分割项目的数据流、模型搭建、实验管理、可视化、快速推理和模型部署等模块进行讲解；第 14 章聚焦于 2D 的

医学图像分割，全面展示 2D 医学图像分割项目的实战细节；第 15 章则是切换到 3D 医学图像分割。

第 16 章为全书总结。主要对全书内容进行总结以及对深度学习图像分割未来进行展望。

全书内容安排如图 1-8 所示。

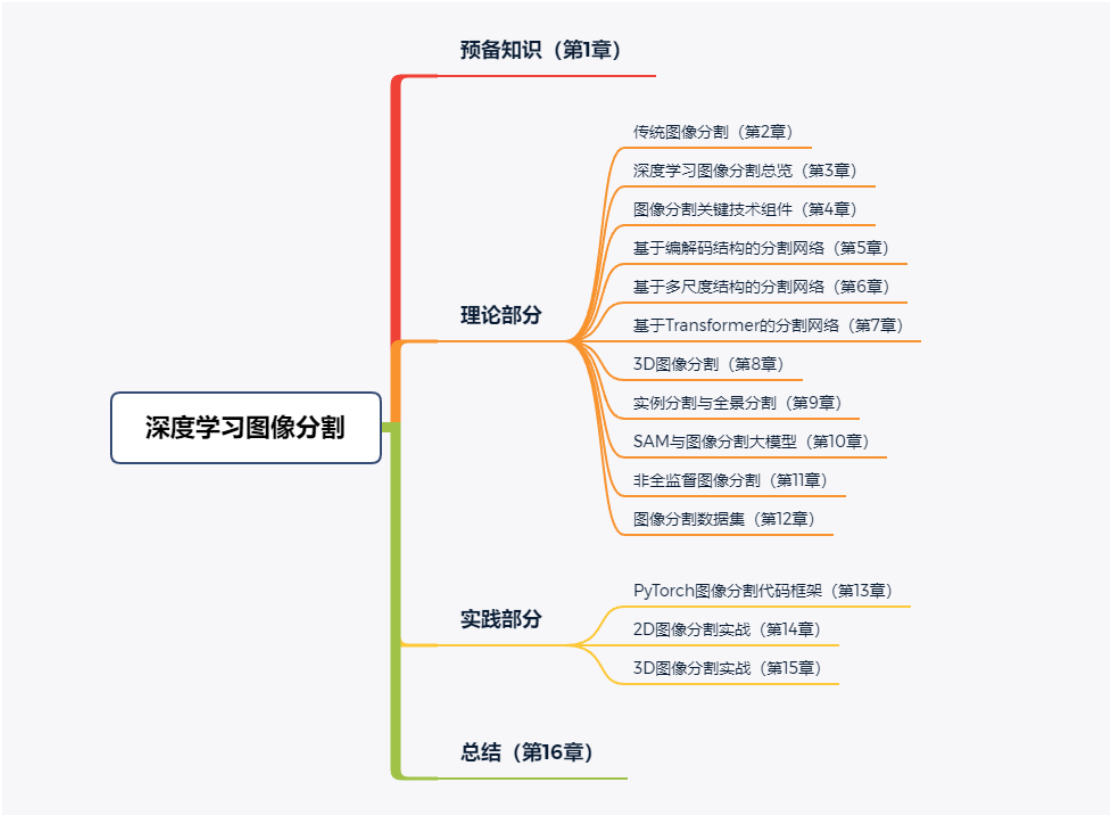

图 1-8 深度学习图像分割内容导图

1.5 本章小结

本章对深度学习图像分割的预备知识，包括深度学习与计算机视觉的相关概念和知识体系进行了梳理，对计算机视觉的三大任务：图像分类、目标检测和图像分割在近年来的进展进行了介绍，了解到三大任务之间的渐进式关系，即图像分类聚焦于整张图像的语义识别，目标检测需要定位到图像中的具体对象类别，而图像分割则是需要关注图像中每一个像素的类别划分。然后我们对全书所用到的两个基本实现框架，OpenCV 和 PyTorch 进行了简介，认识到二者在图像分割实战项目中的重要性。最后对全书的内容进行前情提要和内容安排，方便读者更

好地阅读本书。

参考文献

[1] LeCun Y, Bengio Y, Hinton G. Deep learning[J]. nature, 2015, 521(7553): 436-444.

[2] Esteva A, Chou K, Yeung S, et al. Deep learning-enabled medical computer vision[J]. NPJ digital medicine, 2021, 4(1): 5.

[3] Liu Y, Zhang Y, Wang Y, et al. A survey of visual transformers[J]. IEEE Transactions on Neural Networks and Learning Systems, 2023.

[4] Zhang C, Liu L, Cui Y, et al. A comprehensive survey on segment anything model for vision and beyond[J]. arXiv preprint arXiv:2305.08196, 2023.

[5] Garcia-Garcia A, Orts-Escolano S, Oprea S, et al. A review on deep learning techniques applied to semantic segmentation[J]. arXiv preprint arXiv:1704.06857, 2017.

[6] Guo Y, Liu Y, Georgiou T, et al. A review of semantic segmentation using deep neural networks[J]. International journal of multimedia information retrieval, 2018, 7: 87-93.