第 5 章

目 标 检 测

5.1 目标检测任务介绍

5.1.1 目标检测任务

目标检测是计算机视觉的一个热门研究方向，广泛应用于智能视频监控、机器人导航、工业检测、航空航天等诸多领域，通过目标检测可以减少对人力资本的消耗，具有重要的现实意义。目标检测还是目标跟踪、人脸识别、人流统计、目标实例分割等任务的基础，对这些问题的解决起着至关重要的作用。

如图 5.1(a)所示的图像分类解决的是图像中的目标是什么的问题，图像中只有一个目标。图 5.1(b)图像分类和定位解决的是图像中的目标是什么及在图像中的位置，同样图像中只有一个目标。图 5.1(c)目标检测解决的是图像中都有哪些目标，各属于哪种分类及在图像中的位置的问题。在目标检测的图像中一般会有多个目标，目标相互间的尺寸差异可能很大。目标检测可以看作是多个目标的定位和分类任务。图像中的目标也称为前景，其他的部分称为背景。

扫码查看彩图

（a）分类　　（b）分类+定位　　（c）目标检测

图 5.1　图像识别和目标检测

目标检测任务可以概括为两个方面。

（1）目标位置预测。目标位置通常用外接矩形表示，称为边界框(Bounding Box，BBox)。边界框的参数包括中心坐标和宽高(x，y，w，h)，也有按照左上角和右下角(x_{lt}，y_{lt}，x_{rb}，y_{rb})来确定边界框的。

（2）目标类别预测。类别预测可分为单标签分类和多标签分类。单标签分类的类别之间是互斥的关系，目标只能属于一种类别，如猫、狗、电视等。分类时一般采用 Softmax 分类器，计算出目标属于各个类别的概率，各类别概率之和为 1，其中概率得分最大的类别就是目标所属类别。多标签分类的类别之间可能存在从属关系，如人、男人、女人。分类时一

般采用二分类器，计算出目标对每一个类别的是或否的概率，目标可能属于一个或多个类别。

目标检测任务最早是采用传统的图像处理算法来完成的，经历了 VJ 检测算法、HOG 检测算法、DPM 检测算法。如图 5.2 所示，伴随着 AlexNet 等分类网络的出现，基于卷积神经网络的目标检测算法近年来也开始蓬勃发展。

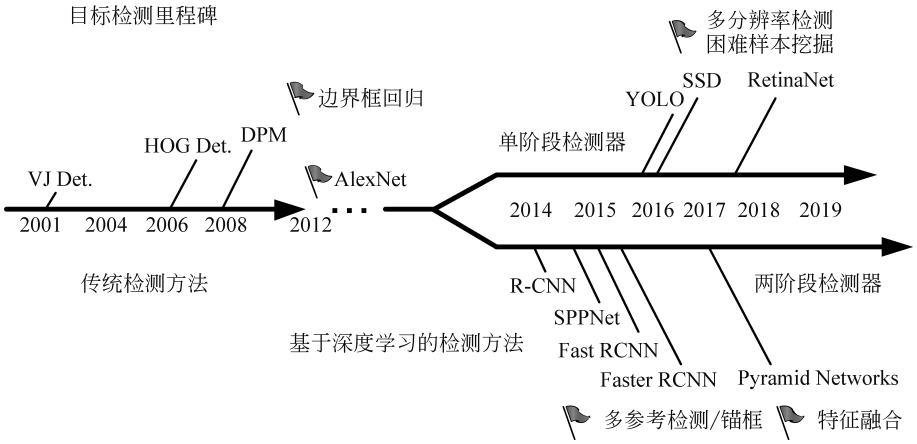


图 5.2　目标检测算法发展里程碑

基于卷积神经网络的主流目标检测算法框架分为两阶段目标检测算法（Two-Stage）与单阶段目标检测算法（One-Stage）。两阶段目标检测算法的代表有 R-CNN 系列、SPPNet、FPN 等，起步略早，单阶段目标检测算法的代表有 YOLO 系列、SSD、RetinaNet 等。两阶段目标检测算法如图 5.3 所示，第一步生成目标区域候选框，第二步进行候选区域的目标分类与定位。单阶段目标检测算法只需要一个步骤就可以完成目标分类和定位。

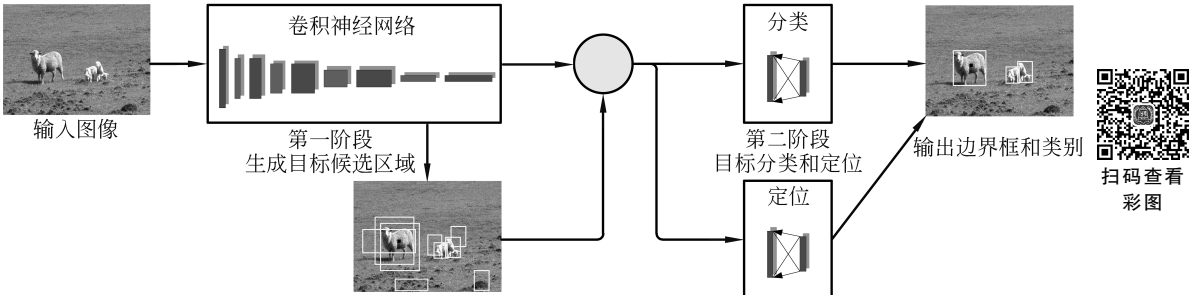


图 5.3　两阶段目标检测算法示意图

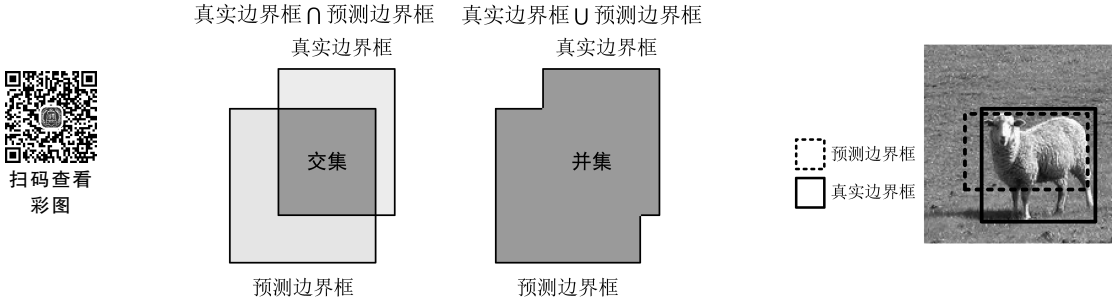
由于图像中的目标往往具有不同姿态、不同视角、不同大小、天气光照等条件复杂多样，即使在技术相对发达的今天，目标检测这一视觉任务仍然是非常具有挑战性的课题。

5.1.2　预备知识

1. 交并比（Intersection over Uion，IoU）

交并比表示预测边界框（prediction）与真实边界框（ground truth，训练样本的标注数据）的重合程度，即二者的交集和并集的比值。IoU 值越大，表示两者的重合程度越高。在

理想情况下，当预测边界框与真实边界框完全重合时，IoU 为 1；当两边界框没有交集时，IoU 为 0，如图 5.4 所示。



扫码查看彩图

$$\mathrm{IoU}_{\mathrm{pred}}^{\mathrm{truth}} = \frac{\text{真实边界框} \cap \text{预测边界框}}{\text{真实边界框} \cup \text{预测边界框}}$$

图 5.4　IoU 计算方法

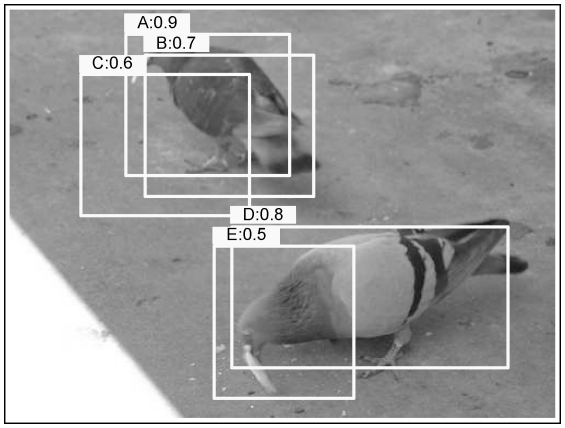
2. 非极大值抑制（Non Maximum Suppression，NMS）

在目标检测的过程中，算法对同一个目标可能预测出多个重叠的边界框，使用 NMS 非极大值抑制算法对预测边界框进行筛选，消除冗余的预测边界框，得到最终预测边界框。

以图 5.5 目标检测为例，图中两只鸽子为检测目标，分别有 3 个和 2 个预测边界框，每个预测边界框都有置信度得分，置信度得分用于评估目标在边界框内可能性，置信度得分越高的预测边界框越接近真实边界框。通过 NMS 算法去除每个目标的冗余预测边界框，保留置信度最大的预测边界框。NMS 的计算过程如下：

（1）将所有预测边界框按照置信度或概率得分排序。

（2）图 5.5 中 A 的置信度得分 0.9 为最高，计算 A 与所有剩余预测边界框 B、C、D、E 的 IoU，剔除 IoU≥0.5 的预测边界框 B、C，剩余预测边界框 D、E，此时 A 成为 NMS 筛选得到的第一个结果。



扫码查看彩图

图 5.5　NMS 非极大值抑制算法

（3）找出下一个置信度最大的预测边界框，D 的置信度得分 0.8，计算 D 与剩余预测边界框 E 的 IoU，剔除 IoU≥0.5 的候选框 E，此时 E 成为 NMS 筛选得到的第二个结果。

(4) 按照过程(3)的规则不断迭代,直至没有剩余的预测边界框。

本例中,预测边界框 A、D 为 NMS 边界框筛选后的结果。

3. 上采样(upsampling)

在卷积神经网络中,池化操作会缩小特征图的尺寸,实现特征选择和信息过滤的目的,这个过程称为下采样(downsampling)或降采样(subsampling)操作。在目标检测和图像分割算法中,需要将深层的小尺寸特征图放大到中、浅层特征图或原图的尺寸,来实现特征融合或映射的目的,这样的过程是上采样(upsampling)操作。上采样操作可以通过传统插值、反采样、反池化、转置卷积等方法实现。

传统插值方法一般有最近邻插值(nearest)、双线性插值(bilinear)等算法。传统插值方法、反采样、反池化等方法是预先定义好的操作,计算简单,没有可学习参数。转置卷积与插值法不同,需要学习一些参数,更能体现神经网络学习的优点。

1) 最近邻插值法(nearest)

最近邻插值法指在待求元素的所有近邻元素中,将距离待求元素最近的邻元素值赋给待求元素。最近邻插值法会造成插值后生成的图像在灰度上不连续,有明显的锯齿现象。

2) 双线性插值法(bilinear)

利用待求元素的 4 个邻元素的值在两个方向上作线性内插,如图 5.6 所示。

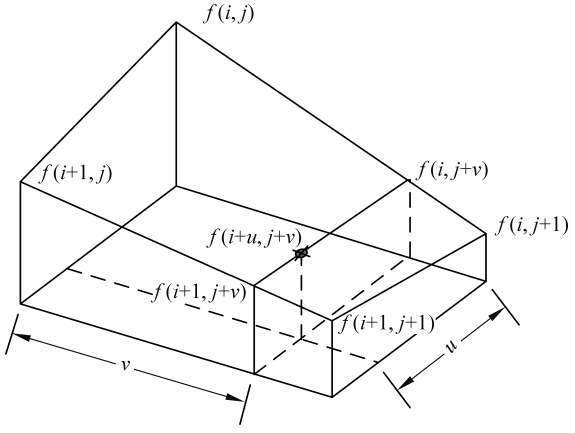


图 5.6　双线性插值

其中:$f(i,j)$、$f(i,j+1)$、$f(i+1,j)$、$f(i+1,j+1)$为 4 个已知元素的值;$(i+u,j+v)$为插入元素的坐标;u,v 为插入元素距(i,j)元素的距离,取值范围在 0～1。双线性插值的计算方法如公式(5.1)所示。双线性插值法虽然算法较为复杂,但得到的插入值连续、平滑,在上采样算法中应用较为广泛。

$$f(i+u,j+v)=(1-u)(1-v)f(i,j)+v(1-u)f(i,j+1)+u(1-v)f(i+1,j)+uvf(i+1,j+1) \tag{5.1}$$

3) 反采样法(unsampling)

反采样法就是直接将特征图的元素复制在扩充的特征图空白区块上,如图 5.7 所示。

4) 反池化法(unpooling)

反池化法是将特征图元素复制到下采样之前特征图各区块原先最大值的位置,其他位置用“0”填充,反池化是对最大池化操作结果进行的上采样操作,需要在最大池化操作时记

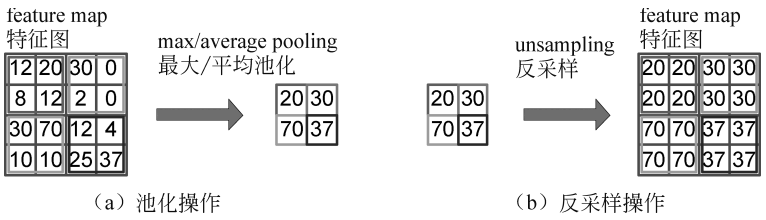


图 5.7　池化和反采样操作

录区块最大值的位置，如图 5.8 所示。

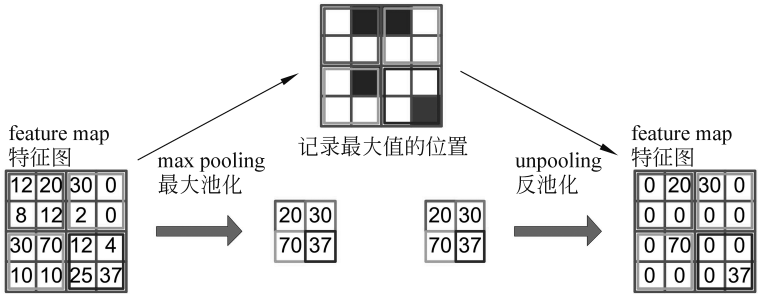


图 5.8　反池化

5）转置卷积法（transposed convolution）

转置卷积能够起到上采样的作用，需要注意的是转置卷积不是卷积的逆运算，而是一种卷积。在网络中，转置卷积的卷积核不是预先设定的，而是与其他卷积核一样，是通过学习得到的。

转置卷积的运算步骤如图 5.9 所示：

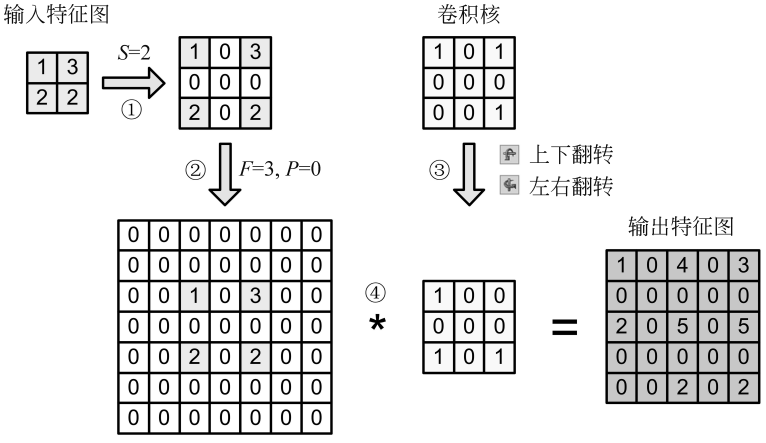


图 5.9　转置卷积的运算步骤

（1）在输入特征图元素之间插入（$S-1$）行和列，填充值为 0，S 为步长。

（2）在输入特征图四周扩展（$F-P-1$）行和列，填充值为 0，F 为卷积核大小，P 为零填充。

（3）将卷积核参数上下、左右翻转。

（4）做步长为 1 的正常卷积运算。

转置卷积运算后得到的输出特征图宽和高为

$$W_o = S \times (W_i - 1) + F - 2 \times P$$

$$H_o = S \times (H_i - 1) + F - 2 \times P$$

其中：W_i、H_i 为输入特征图的宽和高；W_o、H_o 为输出特征图的宽和高；S 为步长；P 为填充；F 为卷积核高和宽。

转置卷积法的计算过程比较烦琐，感兴趣的读者可以参阅论文 *A guide to convolution arithmetic for deep learning*（论文地址：https://arxiv.org/pdf/1603.07285.pdf）。

深度学习框架提供了转置卷积的 API 函数。

（1）TensorFlow 的转置卷积函数：conv2d_transpose(x,kernel,output_shape,strides,padding)。

（2）PyTorch 的转置卷积函数：ConvTranspose2d(in_channels,out_channels,kernel_size,stride,bias)。

5.1.3　评估准则

正样本：在所有检测样本中，与目标的真实样本对应的样本为正样本。例如：猫狗分类中，在统计猫的数量时，所有的真实为猫的样本为正样本；在目标检测任务中，目标预测边界框与真实样本边界框的 IoU 超过设定阈值时确定为正样本。

负样本：在所有检测样本中，与目标的真实样本不同的样本为负样本。例如：猫狗分类中，在统计猫的数量时，所有的真实类别为狗的样本为负样本；在目标检测任务中，目标预测边界框与真实样本边界框的 IoU 小于设定阈值时确定为负样本。

混淆矩阵的判断方法如表 5.1 所示。

表 5.1　混淆矩阵的判断方法

样本的真实属性	预测/判断/检测的样本类别	
	判断为正样本	判断为负样本
正样本	真正例	假负例
负样本	假正例	真负例

真正例（True Positives，TP）：将正样本判断成正样本，判断是正确的。在目标检测中，预测样本与真实样本的边界框 IoU 大于设定阈值（一般为 0.5）的边界框为真正例。

假正例（False Positives，FP）：将负样本判断成正样本，判断是错误的。在目标检测中，预测样本与真实样本的边界框 IoU 小于设定阈值的边界框（即将背景预测为目标）为假正例。

真负例（True Negatives，TN）：将负样本判断成负样本，判断是正确的。

假负例（False Negatives，FN）：将正样本判断成负样本，判断是错误的。在目标检测中，未检测出的目标为假负例。

为便于理解，以目标检测为例：在一张图像中，有 3 只猫和 7 条狗，要求检出图片中所有的猫，即猫为检测对象，是正样本，狗为负样本。预测结果有 6 只猫，其中真实类别为猫的有 2 只，真实类别为狗的有 4 只。

在上例中，真正例 TP=2，假正例 FP=4，真负例 TN=3，假负例 FN=1。

1. 准确率(Accuracy)

准确率指在所有的检测样本中，正确检测出的样本总数占总检测样本数的比例。公式为

$$\text{Accuracy}=\frac{\text{正确检测出的样本总数}}{\text{总检测样本数}}=\frac{\text{TP}+\text{TN}}{\text{TP}+\text{FN}+\text{FP}+\text{TN}}$$

上例预测的准确率为

$$\text{Accuracy}=\frac{2+3}{2+1+4+3}=50\%$$

2. 精度(Precision)

精度(又称查准率)是指在所有检测出的正样本中，正确检测出的正样本所占的比例。公式为

$$\text{Precision}=\frac{\text{正确检测出的正样本数量}}{\text{所有检测出的正样本数量}}=\frac{\text{TP}}{\text{TP}+\text{FP}}$$

上例预测猫的精度为

$$\text{Precision}=\frac{2}{2+4}=33.3\%$$

3. 召回率(Recall)

召回率(又称查全率)是指正确检测出的正样本占正样本总数的比例。公式为

$$\text{Recall}=\frac{\text{正确检测的正样本}}{\text{所有真实正样本的数量}}=\frac{\text{TP}}{\text{TP}+\text{FN}}$$

上例预测猫的召回率为

$$\text{Recall}=\frac{2}{2+1}=66.7\%$$

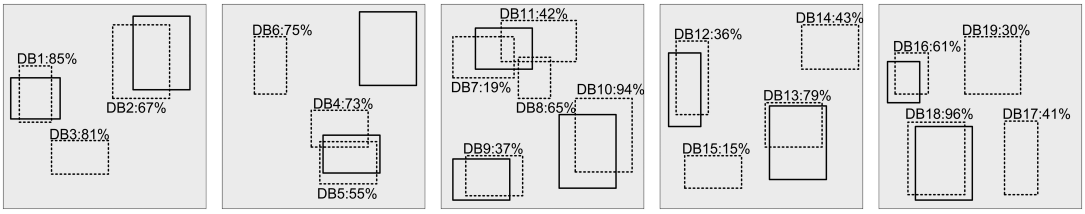
召回率越高，漏检的样本就越少。

4. 平均精度(Average Precision，AP)

如前所述，目标检测的正、负样本是由预测边界框与真实边界框的 IoU 来确定的。当两者 IoU 大于阈值时，预测边界框被确定为正样本，否则确定为负样本。如果一个真实边界框存在多个预测边界框与其 IoU 大于阈值，则选取两者 IoU 最大的预测边界框为正样本，其余为负样本。IoU 阈值不同，得到的正负样本数量不同，由此得到的精度和召回率也不同。在性能评估中，IoU 阈值通常取 0.5 或 0.75。

如图 5.10 所示，对 5 张图像中的目标进行检测，实线框为真实样本的边界框，虚线框为检测样本的预测边界框。5 张图像中共有 11 个真实样本，预测输出 19 个检测样本，即 TP+FP=19。预测边界框的顶部标签为预测边界框号+置信度。以 0.5 为 IoU 阈值可得 DB1、DB2、DB5、DB7、DB9、DB10、DB12、DB13、DB16、DB18 预测边界框为正样本，正确检测出的真实样本数为 10 个，即 TP=10，未检测出的真实样本数为 1 个，即 FN=1，TP+FN=11。

将所有检测出的预测边界框按照置信度降序排序，并计算 TP 累计、FP 累计、精度和召回率。如表 5.2 所示。



图像1　图像2　图像3　图像4　图像5

图 5.10　检测样本与真实样本

表 5.2　精度、召回率计算

图像	预测边界框	置信度	TP	FP	TP 累计	FP 累计	精度 Precision	召回率 Recall	预测框累计数
图像 5	DB18	96%	1		1	0	1.00	0.09	1
图像 3	DB10	94%	1		2	0	1.00	0.18	2
图像 1	DB1	85%	1		3	0	1.00	0.27	3
图像 1	DB3	81%		1	3	1	0.75	0.27	4
图像 4	DB13	79%	1		4	1	0.80	0.36	5
图像 2	DB6	75%		1	4	2	0.67	0.36	6
图像 2	DB4	73%		1	4	3	0.57	0.36	7
图像 1	DB2	67%	1		5	3	0.63	0.45	8
图像 3	DB8	65%		1	5	4	0.56	0.45	9
图像 5	DB16	61%	1		6	4	0.60	0.55	10
图像 2	DB5	55%	1		7	4	0.64	0.64	11
图像 4	DB14	43%		1	7	5	0.58	0.64	12
图像 3	DB11	42%		1	7	6	0.54	0.64	13
图像 5	DB17	41%		1	7	7	0.50	0.64	14
图像 3	DB9	37%	1		8	7	0.53	0.73	15
图像 4	DB12	36%	1		9	7	0.56	0.82	16
图像 5	DB19	30%		1	9	8	0.53	0.82	17
图像 3	DB7	19%	1		10	8	0.56	0.91	18
图像 4	DB15	15%		1	10	9	0.53	0.91	19

在表 5.2 中“TP 列”和“FP 列”，当 TP＝1 时表示预测边界框为真正例，FP＝1 时表示预测边界框为假正例。“TP 累计”列和“FP 累计”列表示按照行顺序累计 TP 和 FP 的数量。“精度 Precision”列和“召回率 Recall”列的计算是一个不断累计的过程，而不是每个预测框独立计算的。每行中精度 Precision＝该行的 TP 累计/该行的预测框累计数，每行中召回率 Recall＝该行的 TP 累计/真实样本数。

PR 曲线(Precision-Recall)是以召回率为横轴、以精度为纵轴绘制的曲线，根据表 5.2 的数据绘制 PR 曲线如图 5.11(a)所示。

平均精度 AP 是对一个类别目标的检测精度的平均值，是 PR 曲线下的面积。PR 曲线

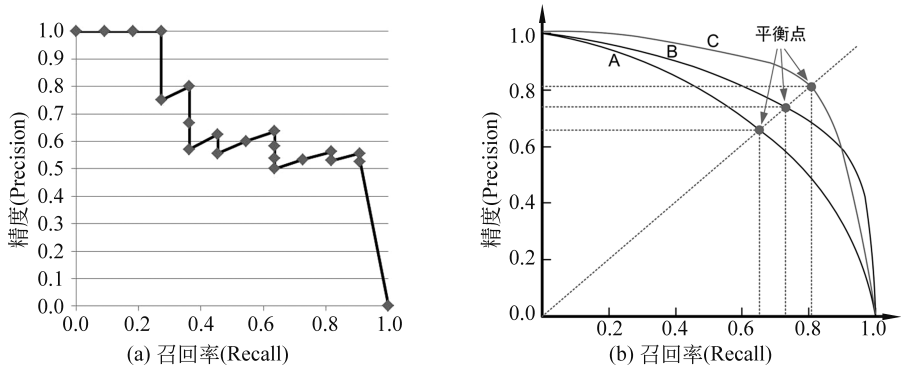


图 5.11 PR 曲线

下的面积越大，AP 值越高，检测效果越好。在实际应用中，需要先对 PR 曲线进行平滑处理，然后再计算 AP 的值。

如图 5.11(b)所示，如果由检测模型 B 得到的 PR 曲线外包围检测模型 A 的曲线，则模型 B 的平均精度优于模型 A。当两个模型的 PR 曲线发生交叉时，如图中模型 C 和模型 B，则由曲线原点开始绘制斜率为 1(P＝R)的直线，直线与模型 PR 曲线的交点称为平衡点(BEP)，平衡点越大的模型平均精度 AP 越高，图中模型 C 的 AP 值高于模型 B。

对于同一个检测模型，IoU 阈值设置不同时，得到的 AP 也不同，在 MS COCO 数据集中采用 IoU 阈值＝0.5 和 0.75 来计算 AP 值，分别为评估指标AP_{50}和AP_{75}，评估指标 AP 是 IoU 阈值从 0.5～0.95 按照间隔 0.05 计算出各个 IoU 阈值下的 AP 值后再取平均得到，除此之外还有针对大、中、小目标的 AP_L、AP_M、AP_S，如图 5.12 所示。

	backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Faster R-CNN+++ [5]	ResNet-101-C4	34.9	55.7	37.4	15.6	38.7	50.9
Faster R-CNN w FPN [8]	ResNet-101-FPN	36.2	59.1	39.0	18.2	39.0	48.2

图 5.12 目标检测模型的平均精度指标

5. 多类别平均精度(mean Average Precision，mAP)

综合衡量检测效果，是所有类别 AP 值的平均值。值的大小在[0，1]上，越大越好。

6. 帧率(Frame Per Second，FPS)

每秒处理图像的数量，反映了模型算法的速度。

5.2 两阶段目标检测算法

两阶段目标检测算法以 R-CNN 系列为代表，包括 R-CNN、Fast R-CNN、Faster R-CNN，除此之外还有 SPPNet、Pyramid Network 等。这类算法在第一阶段需要通过区域建议(region proposal)先在图像上产生一些可能包含目标的候选区域，第二阶段对每个候选区域使用 CNN 网络进行分类与回归，得到目标的类别和精细边界框。

5.2.1 R-CNN

R-CNN 将区域建议算法与 CNN 相结合，将 CNN 从图像分类任务扩展到目标检测任

务，确定了候选区域生成＋分类定位的两级目标检测网络结构。

如图 5.13 所示，R-CNN 算法流程分为 5 个步骤：

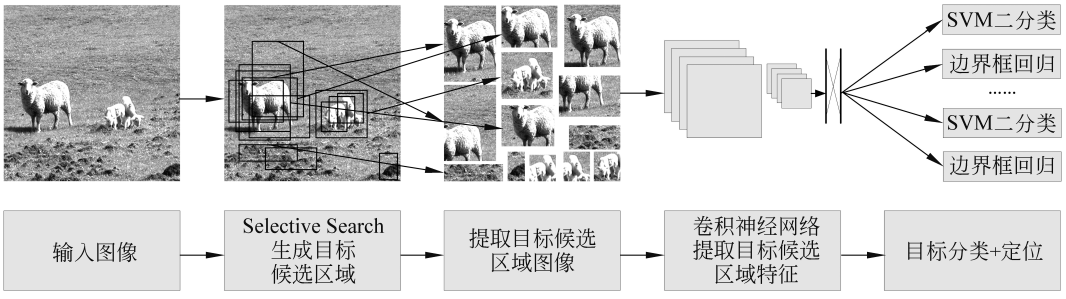
(1) 在待测图像上采用 Selective Search 传统算法，产生 1000～2000 个可能包含目标的候选区域图像。

(2) 将每一个候选区域图像缩放为 227×227 固定尺寸，输入到已训练好的卷积神经网络模型进行特征提取。网络包括卷积层、池化层和 2 个全连接层，输出 4096 维特征向量。

(3) 将每一个候选区域的特征输入到 SVM 分类器进行二分类(是/否)，判断候选区域的对象是否属于该类别。一个 SVM 分类器只能判断一个分类类别，因此需要将候选区域特征遍历所有类别的 SVM 分类器。

(4) 对每一类候选区域进行 NMS 非极大值抑制，剔除冗余候选框，对于同一目标保留该类得分最高的候选框。

(5) 使用边界框回归器精细修正候选边界框位置。



扫码查看彩图

图 5.13　R-CNN 算法流程

R-CNN 算法存在的问题：

(1) R-CNN 训练阶段多，过程烦琐，需要预训练、微调 CNN 网络、训练 SVM 分类器和训练边界框回归器。

(2) R-CNN 使用 Selective Search 这一传统方法进行候选区域提取，产生 1000～2000 个候选区域，这个过程计算量大，耗时很长。

(3) R-CNN 需要对每个候选区域独立进行 CNN 特征提取和 SVM 分类，计算量非常大，如图 5.14 所示。

(4) 对候选区域图像的裁剪和缩放操作不可避免地带来图像失真。

5.2.2　Fast R-CNN

针对 R-CNN 的不足，Girshick 等于 2015 年提出了 Fast R-CNN，旨在提高训练和测试的速度，同时检测精度也得到了提升。

如图 5.15 所示，Fast R-CNN 的算法流程步骤如下：

(1) 与 R-CNN 相同，Fast R-CNN 采用 Selective Search 传统算法在待测图像上产生 1000～2000 个可能包含目标的候选区域图像或称为感兴趣区域图像(Region Of Interest，ROI)。

(2) 将整个图像通过一个卷积神经网络提取全图特征。

(3) 将图像的 ROI 映射到特征图上，通过裁剪得到 ROI 特征图，将 ROI 特征图通过兴