

16 目标检测，计算机视觉训练技巧

概要

- 目标检测
 - 边界框和锚框
 - 交并比
 - 区域卷积神经网络(R-CNN)
 - 单发多框检测(SSD)
- 计算机视觉训练技巧

目标检测

不仅仅是“看懂”，而且要“定位”

➤ 不仅仅是“看懂”，更是“定位”

➤ 图像分类 (Classification): "这张图里有一只猫。"

➤ 目标检测 (Object Detection): "这张图里有一只猫，它在坐标 (x, y, w, h) 的边界框内。"

➤ 任务的精确定义

➤ 输入: 一张图像

➤ 输出: 一个包含 N 个检测结果的列表

➤ 边界框表示法:

➤ 核心挑战

➤ 如何在单次前向传播中，高效地搜索并定位图像中数量不定、尺度各异、长宽比多样的目标？

图像分类



目标检测

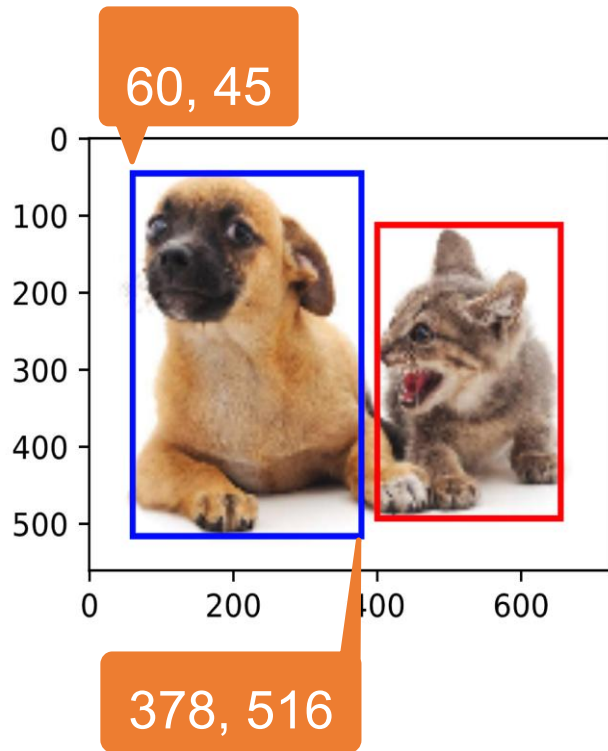


预备基础

边界框

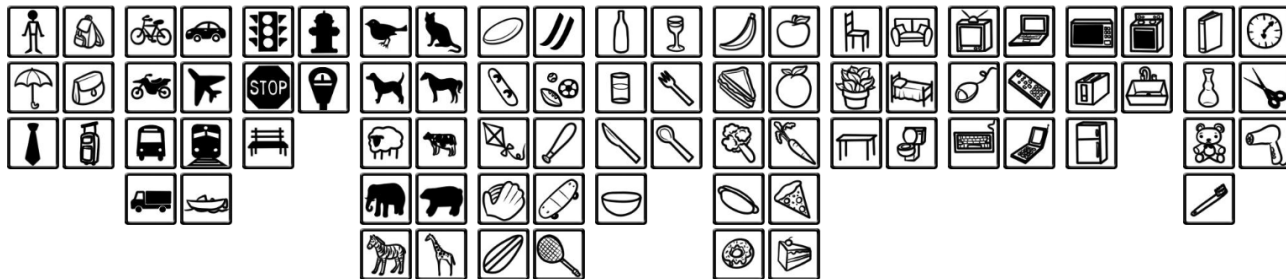
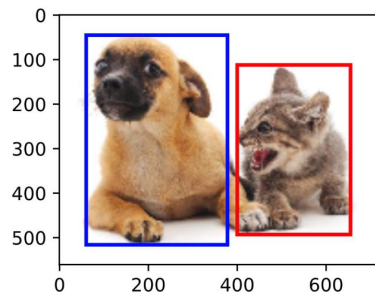
➤ 边界框定义

- (左上角x, 左上角y, 右下角x, 右下角 y)
- (中心(x; y)坐标, 框的宽度和高度)



目标检测数据集

- 每行都有一个目标物体
 - 图像名称，物体类别，边界框
- COCO 数据集(cocodataset.org)
 - 80 个物体
 - 330K 个图像
 - 1.5M 目标物体



锚框

➤ 目标物体检测算法

➤ 原则：在输入图像中采样大量的区域，然后判断这些区域中是否包含我们感兴趣的目标，并调整区域边界从而更准确地预测目标的真实边界框

➤ 候选区域，锚框(anchor box)

➤ 以每个像素为中心，生成多个缩放比和宽高比(aspect ratio)不同的边界框

➤ 预测每个锚框是否包含目标物体

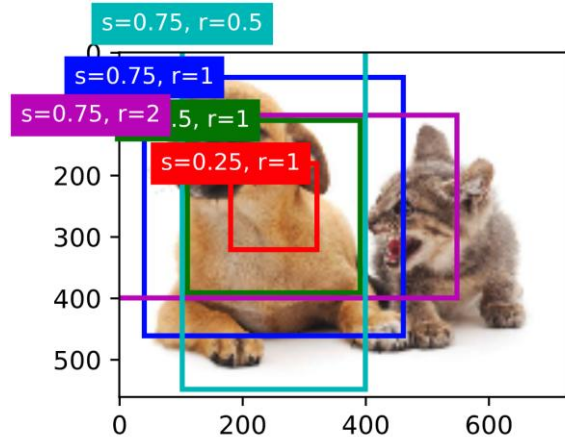
➤ 如果是，预测从锚框到实际边界框的偏移量

➤ 输入图像的高度为 h ，宽度为 w

➤ 以图像每个像素为中心生成不同形状的锚框

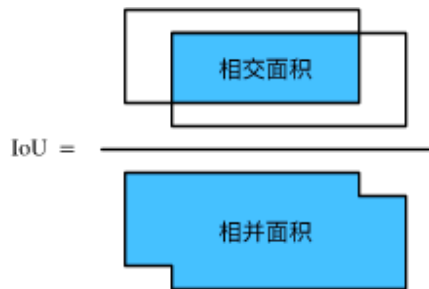
➤ 缩放比 $s \in (0,1]$ ，宽高比 $r > 0$

➤ 锚框的宽度和高度分别为 $hs\sqrt{r}$ 和 $hs/\sqrt{r}$



指标：交并比（IoU）

- Jaccard系数，给定集 A 和 B
- 两边界框的杰卡德系数称为交并比
 - intersection over union, IoU
 - $J(A, B) = \frac{|A \cap B|}{|A \cup B|}$
- IoU测量两个框之间的相似性
- 0表示不重叠
 - 1表示完全相同



在训练数据中标注锚框

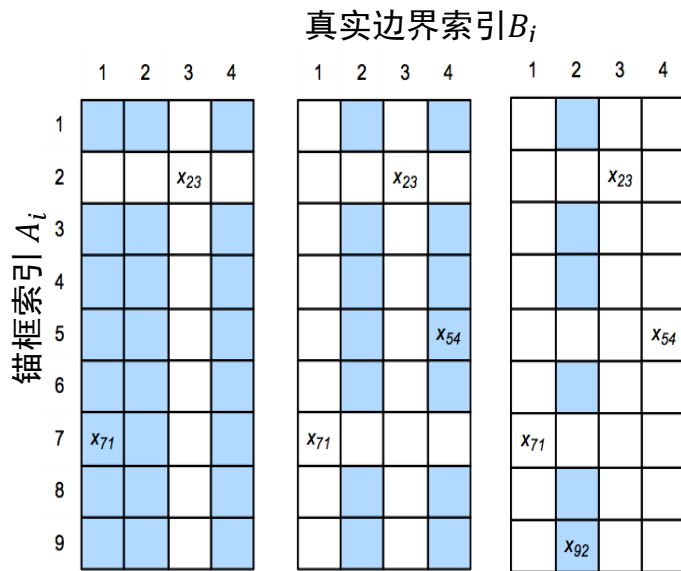
- 在训练集中，将每个锚框视为一个训练样本
- 为了训练目标检测模型，需要每个锚框的
 - 类别（class）
 - 偏移量（offset）

将标签分配给锚框

- 在预测时，为每个图像生成多个锚框，预测所有锚框的类别和偏移量，根据预测的偏移量调整它们的位置，最后输出符合特定条件的预测边界框
- 将每个锚框视为一个训练样本。标记每个锚框 $B_1, \dots, B_{n_b}$
 - 类别
 - 真实边界框相对锚框的偏移量
- 存在大量锚框 $A_1, \dots, A_{n_a}$
 - 导致大部分负面例子

将标签分配给锚框

- 锚框 $A_1, \dots, A_{n_a}$, 真实边界框 $B_1, \dots, B_{n_b}$, $n_a \geq n_b$
 - 矩阵 $\mathbf{X} \in \mathbb{R}^{n_a \times n_b}$
 - x_{ij} : 锚框 A_i 和真实边界框 B_j 的IoU
- 设 $\mathbf{X}$ 中最大IoU为 x_{23} , 将真实边界框 B_3 分配给锚框 A_2 。然后丢弃矩阵第2行和第3列所有元素
- 剩余最大元素 x_{71} , 锚框 A_7 分配真实边界框 B_1
- 继续直到丢弃掉 $\mathbf{X}$ 中 n_b 列中的所有元素
 - 此时已为这 n_b 个锚框各自分配一个真实边界框
- 遍历剩下 $n_a - n_b$ 个锚框
 - 给定任何锚框 A_i , 在矩阵 $\mathbf{X}$ 的第 i 行中找到与 A_i 的IoU最大的真实边界框 B_j , 当此IoU大于给定阈值时, 将 B_j 分配给 A_i

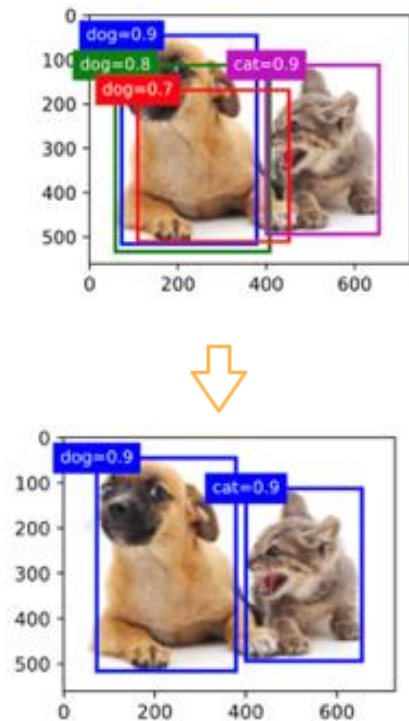


标记类别和偏移量

- 假设一个锚框A被分配了一个真实边界框B
 - 锚框A的类别将被标记为与B相同
 - 锚框A的偏移量根据B和A中心坐标的相对位置以及这两个框的相对大小进行标记
- 如一个锚框未被分配真实边界框，将锚框的类别标记为背景（background）
- 背景类别的锚框通常被称为负类锚框，其余的被称为正类锚框

输出预测边界框

- 当有许多锚框时，可能会输出许多相似的具有明显重叠的预测边界框，都围绕着同一目标。采用非极大值抑制(non-maximum suppression, NMS)合并属于同一目标的类似的预测边界框
 - 每个锚框生成一个边界框预测
 - 选择得分最高的那个预测
 - 所有其他预测与所选预测相比，如果 $\text{IoU} > \theta$ ，则删除
 - 重复，直到选中或删除所有内容
 - 合并属于同一目标的类似的预测边界框
- 如一锚框未分配真实边界框，将该锚框类别标记为“背景”
 - 背景类别的锚框通常被称为“负类”锚框，其余的被称为“正类”锚框



两大主流范式：Two-Stage vs. One-Stage

➤ 两阶段检测器 (Two-Stage Detectors)

- 哲学：“粗筛-精选” (Propose-then-Refine)
- 流程：
 - 阶段一：生成一系列可能包含目标的候选区域 (Region Proposals)
 - 阶段二：对每个候选区域进行精确的分类和边界框回归
- 代表：R-CNN, Fast R-CNN, Faster R-CNN
- 特点：精度高，但结构相对复杂，速度较慢

➤ 单阶段检测器 (One-Stage Detectors)

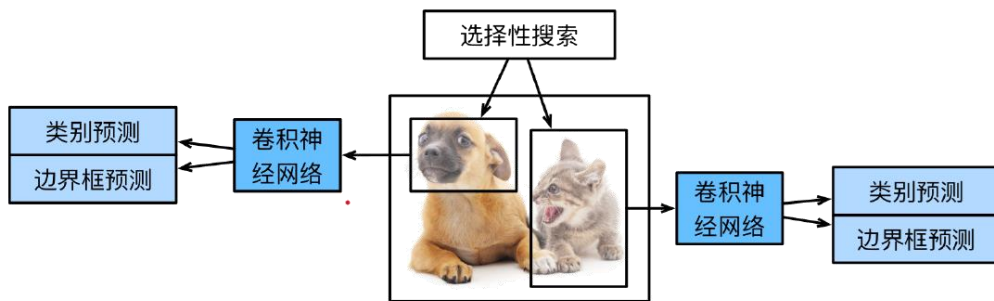
- 哲学：“一步到位” (Single Shot)
- 流程：直接在特征图上密集采样，一次性预测所有位置的类别和边界框
- 代表：SSD, YOLO
- 特点：速度快，结构简洁，是实时检测的首选，近年来精度也取得了巨大进步

区域卷积神经网络 R-CNN

R-CNN: 深度学习检测的开创者

➤ R-CNN(region-based CNN, R-CNN)工作流程

- 区域提议 (Region Proposal): 使用选择性搜索 (Selective Search) 等传统计算机视觉算法, 从图像中提取约2000个候选区域。(此步在CNN之外, 是性能瓶颈)
- 特征提取 (Feature Extraction): 将每个候选区域独立地缩放(warp)到固定尺寸, 然后送入一个预训练的CNN (如AlexNet) 进行前向传播, 提取特征
- 分类 (Classification): 使用一系列线性的SVM分类器, 判断每个区域的特征属于哪个类别
- 回归 (Regression): 训练一个线性回归模型, 对边界框位置微调, 使其更接近真实目标



区域卷积神经网络(R-CNN)

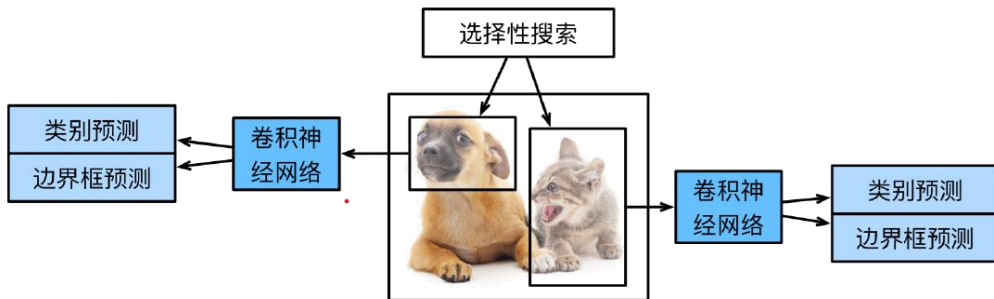
➤ R-CNN的问题

➤ 速度慢！

➤ 对每个提议区域，卷积神经网络的前向传播（特征提取）是独立的，而没有共享计算。由于这些区域通常有重叠，独立的特征抽取会导致重复的计算

➤ 非端到端：训练过程是分步的（提议、CNN、SVM、回归器），无法联合优化。

➤ Fast R-CNN (Girshick, 2015)对R-CNN的主要改进之一，是仅在整张图景上执行卷积神经网络的前向传播



Fast R-CNN: 特征共享

➤ 共享特征提取

➤ 输入一张图像，输出形状为 $1 \times c \times h_1 \times w_1$

➤ 区域投影

➤ 将选择性搜索生成的候选区域，投影到这个特征图上，得到对应的特征区域。

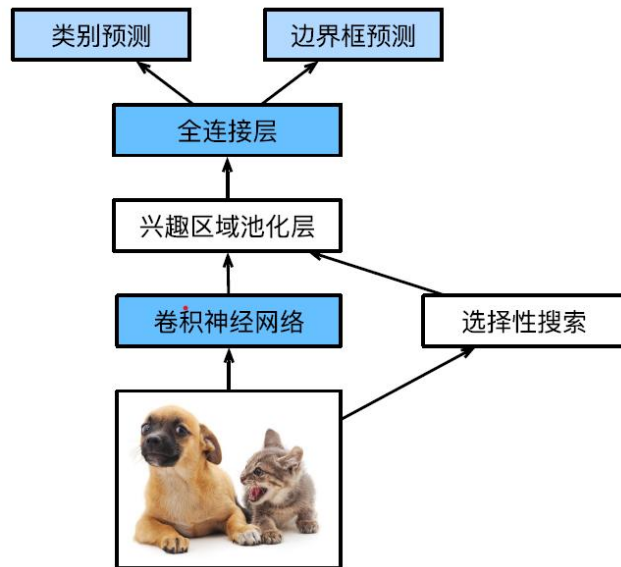
➤ 抽取出形状相同的特征(如高度 h_2 和宽度 w_2)

➤ 兴趣区域汇聚层(RoI pooling)

➤ 将卷积神经网络的输出和提议区域作为输入，输出连结后的各提议区域抽取的特征，形状为 $n \times c \times h_2 \times w_2$

➤ 统一预测头

➤ 将固定尺寸的特征向量送入全连接层，最后通过两个并行的输出层，同时进行分类 (Softmax) 和边界框回归。



RoI汇聚层(兴趣区域汇聚层)

- 全连接层要求输入是固定长度的向量，但候选区域在特征图上大小不一。如何连接这两者？
 - RoI Pooling 将可变尺寸输入转换为固定尺寸输出的桥梁
- 兴趣区域汇聚层(region of interest pooling, RoI汇聚层)
 - 将该区域划分为一个固定数量($h_2 \times w_2$)个个子窗口，然后在每个子窗口内执行Max Pooling
 - 得到一个固定尺寸 ($h_2 \times w_2$) 的特征图

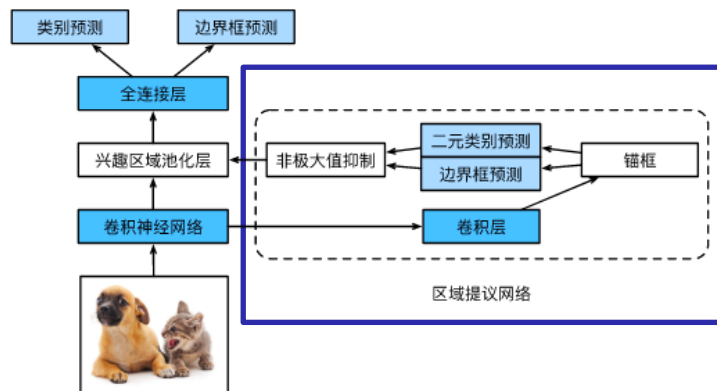
0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

2 × 2 兴趣区
域池化层

5	6
9	10

Faster R-CNN

- 区域提议网络 (Region Proposal Network, RPN)
 - RPN是一个小型的全卷积网络，它在共享的特征图上滑动，并预测一系列候选区域
- 完整架构:
 - 共享主干网络 (Backbone): 提取整图特征
 - RPN分支: 在特征图上生成与目标无关的候选区域
 - Fast R-CNN检测头: 使用RPN生成的区域，执行与Fast R-CNN相同的分类和回归任务
- Faster R-CNN是第一个真正意义上端到端、高性能的深度学习目标检测框架

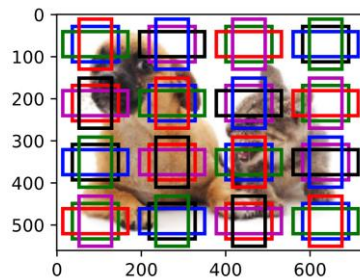


区域提议网络的计算步骤

- 使用填充为1的 3×3 的卷积层变换卷积神经网络的输出，并将输出通道数记为 c 。这样，卷积神经网络为图像抽取的特征图中的每个单元均得到一个长度为 c 的新特征
- 以特征图每个像素为中心，生成多个不同大小和宽高比的锚框并标注它们
- 使用锚框中心单元长度为 c 的特征，分别预测该锚框的二元类别（含目标还是背景）和边界框
- 使用非极大值抑制，从预测类别为目标的预测边界框中移除相似的结果。最终输出的预测边界框即是兴趣区域汇聚层所需的提议区域

深度剖析：锚框 (Anchor Box)

- 锚框，提供一组预定义的、多尺度、多长宽比的“回归基准”
- 在特征图的每个位置，预设k个不同尺寸和形状的锚框
 - 例如：3种尺度 $\times$ 3种长宽比 = 9个锚框。
- RPN的任务不再是“从无到有”找框，而是回答两个更简单的问题
 - 对于每个锚框，它包含的是前景（目标）还是背景？
 - 如果是前景，如何对这个锚框进行微调（平移、缩放），才能完美地框住真实目标？
- 锚框将无约束搜索问题，转化为了一个基于大量高质量“先验”的、更易于学习的分类和回归问题
 - 对于每个像素，生成以该像素为中心的锚框。n 个 $s_1, \dots, s_n$ 和 m 个比率 $r_1, \dots, r_m$ ，生成 $n+m-1$ 个锚框
 - $(s_1, r_1), (s_2, r_1), \dots, (s_n, r_1), (s_1, r_2), \dots, (s_1, r_m)$



Mask R-CNN: 从检测框到像素级分割

➤超越边界框，实现实例分割 (Instance Segmentation)，区分每个目标的精确轮廓

➤RoI Align (兴趣区域对齐层):

➤解决RoI Pooling的特征不对齐问题

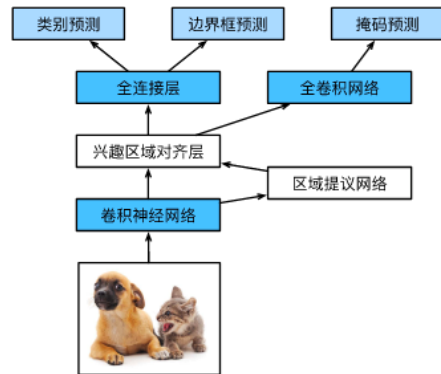
➤放弃取整操作，使用双线性插值在特征图上精确采样，确保输出的特征与原始RoI在空间上严格对齐

➤Mask预测头 (Mask Head):

➤在Faster R-CNN的分类和回归头之外，增加一个并行的全卷积网络 (FCN) 分支

➤该分支对每个RoI输出一个二值的像素级掩码 (Mask)，实现实例分割

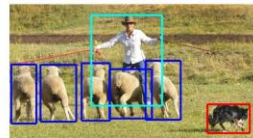
➤Faster R-CNN的框架具有强大的可扩展性，可以通过增加新的预测头来完成更复杂的视觉任务



COCO



(a) Image classification



单发多框检测(SSD)

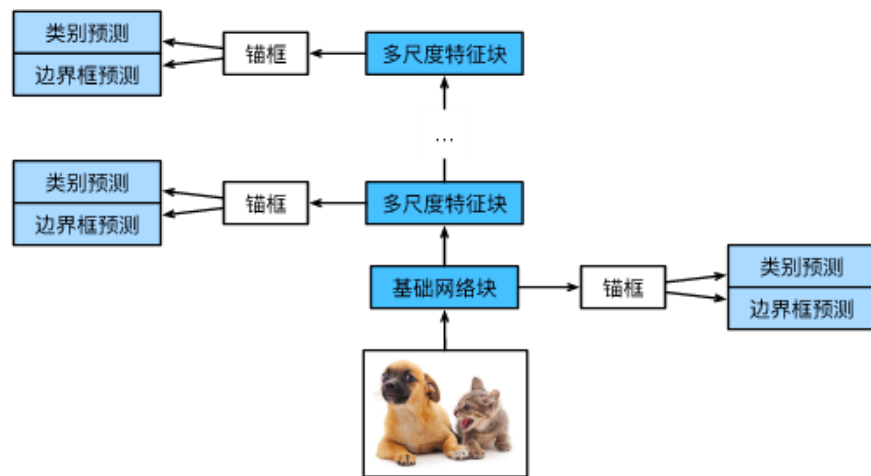
单阶段检测器 - SSD: 单次多尺度检测

- CNN的特征图天然形成了一个“金字塔”结构，不同深度的特征图具有不同的感受野和分辨率。可以利用这一点来检测不同大小的目标
- SSD工作机制
 - 基础网络 (Base Network): 使用一个标准的分类网络（如VGG, ResNet）作为主干
 - 多尺度特征图: 主干网络后，添加一系列逐步降低分辨率的卷积层，形成特征金字塔
 - 并行预测: 在多个不同尺度的特征图上，同时应用卷积预测器（类似于RPN），为每个位置的锚框预测类别和边界框偏移
 - 浅层、高分辨率特征图: 感受野小，用于检测小目标
 - 深层、低分辨率特征图: 感受野大，用于检测大目标
- 概念简洁，速度极快，通过多尺度特征融合了单阶段的速度和两阶段的多尺度思想

单发多框检测(SSD)模型

➤ 单发多框检测为多尺度目标检测模型

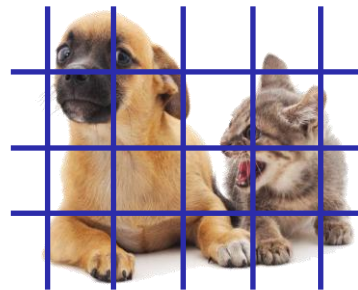
- 首先，基础网络从原始图像中抽取特征，使它输出的高和宽较大，用于检测尺寸较小目标
- 接下来，每一多尺度特征块将上一层提供的特征图高和宽缩小(如减半)，使特征图中每个单元在输入图像上的感受野更大
- 最后，越靠近顶部的多尺度特征块输出的特征图越小，基于特征图生成的锚框也越少，特征图中每个单元感受野越大，因此更适合检测尺寸较大的目标预测每个锚框的类和边界框



你只需看一次(YOLO)

➤ YOLOv1 (Joseph Redmon et al., 2015) 工作机制

- 划分网格: 将输入图像划分为一个 $S \times S$ 的网格
- 网格单元负责制: 如果一个真实目标中心点落入某个网格单元, 那么该单元负责预测该目标
- 统一预测: 每个网格单元独立地预测
 - B个边界框及其置信度 (confidence)。置信度 = $P(\text{Object}) * \text{IoU}(\text{pred}, \text{truth})$
 - C个类别概率 $P(\text{Class}_i | \text{Object})$
- 最终预测: 最终的类别置信度 = 边界框置信度 $\times$ 类别概率。



➤ 演进

- YOLOv1: 速度极快, 但对小目标和密集目标检测效果不佳, 且未使用锚框
- 后续版本吸收了锚框、多尺度预测等思想, 在保持高速的同时, 大幅提升了检测精度, 成为工业界应用最广泛的模型系列之一

核心公共组件：IoU 与 NMS

➤ 交并比 (Intersection over Union, IoU)

➤ 两个边界框交集的面积 / 并集的面积。 $IoU(A, B) = \frac{|A \cap B|}{|A \cup B|}$

➤ 衡量两个边界框重叠程度的归一化指标，取值范围 [0, 1]

➤ 三大核心作用

➤ 定义训练样本: 通过计算锚框与真实框的IoU阈值，来判定锚框是正样本 (high IoU)、负样本还是忽略样本

➤ 评估模型性能: 计算预测框与真实框的IoU，若大于阈值则认为是一次正确检测。是计算mAP指标的基础

➤ 过滤冗余预测: NMS的核心判据

➤ 非极大值抑制 (Non-Maximum Suppression, NMS)

➤ 单个目标可能被多个高置信度的边界框同时检测到

➤ 算法:

➤ 按置信度对所有预测框进行排序

➤ 选择置信度最高的框 M，保留它

➤ 遍历其余框，如果某个框 B 与 M 的IoU超过设定的阈值，则抑制 (删除) 该框 B

➤ 从剩余的框中，重复步骤2和3，直到所有框都被处理

➤ 为每个目标只保留一个最佳的预测框，得到干净的检测结果

计算机视觉训练技巧

数据与标签层面的正则化

➤ Mixup

- 神经网络倾向于在训练样本之间学习到剧烈变化的决策边界。Mixup通过在样本之间进行线性插值，鼓励模型学习更平滑、更简单的函数，从而提高泛化能力
- 随机抽取两个样本 (x_i, y_i) 和 (x_j, y_j) ，通过一个随机数 $\lambda \in [0,1]$ 生成新样本：
 - $\tilde{x} = \lambda x_i + (1 - \lambda)x_j$
 - $\tilde{y} = \lambda y_i + (1 - \lambda)y_j$

➤ 标签平滑 (Label Smoothing)

- One-hot标签 $[0, 0, 1, 0]$ 会驱使模型输出的logit趋向于无穷大，导致模型过分自信 (Over-confident)，降低了对噪声数据和未知数据的泛化能力
- 将硬性的1和0标签，软化为一个略微模糊的目标。例如，对于一个K分类问题，将1替换为 $1 - \epsilon$ ，将0替换为 $\epsilon/(K - 1)$ 。
 - 相当于给模型的“自信”设置了一个上限，强迫它保留一部分概率给其他类别

混合训练数据

- 随机选择两个例子 i 和 j ，采样随机数 $\lambda \in [0,1]$
- 计算： $x = \lambda + x_i(1 - \lambda)x_j$ ， $y = \lambda y_i + (1 - \lambda)y_j$
- 例如：



bittern	0
...	0
otter	0
...	0
analog_clock	1

* 0.9 +



bittern	1
...	0
otter	0
...	0
analog_clock	0

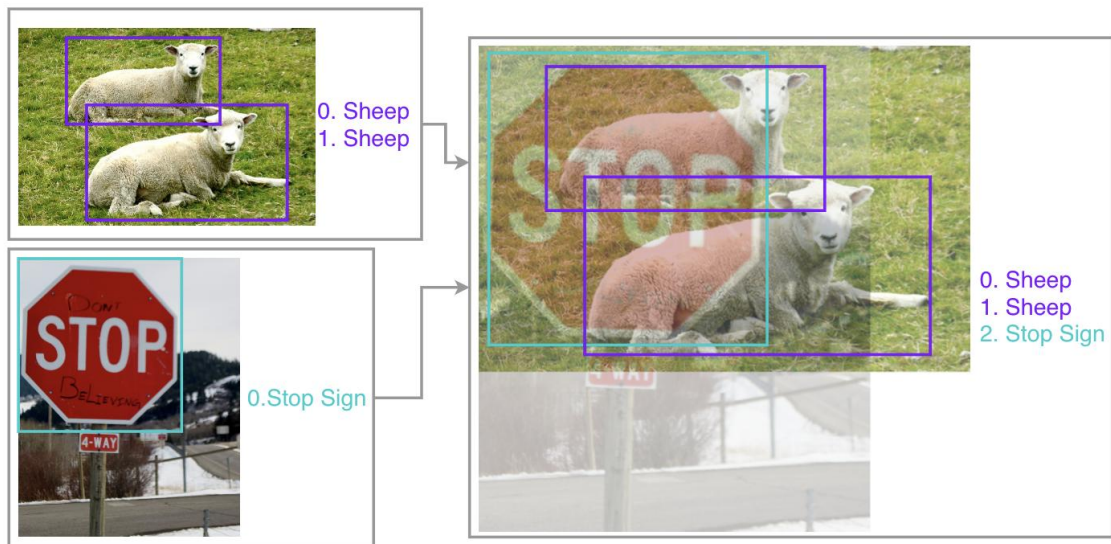
* 0.1 =



bittern	0.1
...	0
otter	0
...	0
analog_clock	0.9

混合训练数据

➤ 也适用于目标物体检测模型



标签平滑化

- 假设标签 $\mathbf{y} \in \mathbb{R}^n$ 是one-hot编码

$$y_i = \begin{cases} 1 & \text{if belongs to class } i \\ 0 & \text{otherwise} \end{cases}$$

- 用 softmax 逼近 0/1 值很难，标签平滑化

$$y_i = \begin{cases} 1 - \epsilon & \text{if belongs to class } i \\ \epsilon / (n - 1) & \text{otherwise} \end{cases}$$

- 常用 $\epsilon = 0.1$

学习率预热

➤ 学习率预热 (Learning Rate Warmup)

- 动机: 训练初期, 模型参数是随机的, 远离最优解。此时若使用较大的学习率, 可能导致梯度爆炸或振荡, 破坏了初始权重结构
- 机制: 在训练开始的几个epoch, 使用一个非常小的学习率, 然后线性或非线性地增加到预设的基础学习率。这给了模型一个“稳定适应”的阶段

学习率预热：余弦衰减

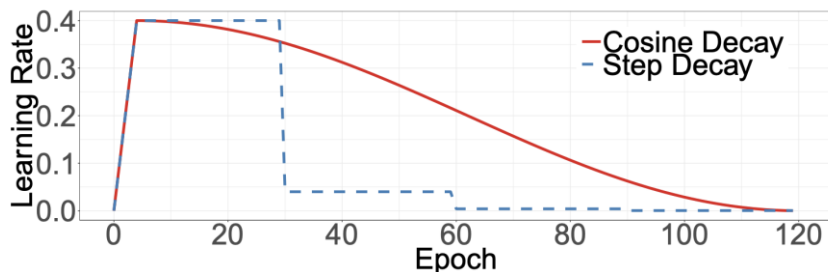
➤ 余弦学习率衰减 (Cosine Learning Rate Decay)

- 随着训练进行，模型逐渐接近最优点。此时需要减小学习率，以更精细的“步长”在最优点附近探索，避免“跨过”最优点
- 学习率按照余弦函数的形式，从初始值平滑地衰减到接近零。相比于阶梯式下降，它在训练后期能更稳定地收敛到更好的局部最优

$$\eta_t = \frac{1}{2} \left(1 + \cos\left(\frac{t\pi}{T}\right) \right) \eta_{initial}$$

➤ 假设在总 T 次迭代(批次)中，余弦衰减计算迭代时的学习率 t

$$\eta_t = 1/2 \left(1 + \cos(t\pi/T) \right) \eta$$



硬件与批次相关的策略

➤ 同步批量归一化 (Synchronized Batch Normalization, SyncBN)

- BatchNorm的有效性依赖于对“足够大”的batch计算准确的均值和方差。但在目标检测等任务中，由于输入图像尺寸大，单个GPU的内存只能容纳很小的batch size（如1-2张图），导致统计量噪声极大，严重影响BN效果
- 在多GPU训练时，将所有GPU上的样本视为一个大的“逻辑Batch”，跨GPU同步计算全局的均值和方差，然后再用这个全局统计量对每个GPU上的数据进行归一化

➤ 随机多尺度训练 (Random Multi-Scale Training)

- 现实中目标尺寸各异。如果只用固定尺寸的图像训练，模型对尺度变化的鲁棒性会受限
- 在每个训练迭代中，从一个预设的尺寸范围中随机选择一个尺寸，并将整个batch的图像缩放到该尺寸进行训练。这是一种非常有效的数据增强，强迫模型学习尺度不变性特征

总结

- 范式演进。目标检测的发展史是一部不断追求效率与精度平衡的历史
 - 从R-CNN的计算冗余，到Fast R-CNN的特征共享，再到Faster R-CNN的端到端提议，两阶段方法将精度推向极致
 - SSD和YOLO开创的单阶段范式，则通过结构简化和并行预测，实现了实时检测的可能
- 核心设计思想
 - 特征共享: 避免重复计算的基石
 - 锚框 (Anchors): 将无约束回归问题转化为基于先验的分类和微调问题
 - 特征金字塔 (Feature Pyramid): 利用CNN的层级结构解决多尺度挑战
 - 端到端: 将所有组件纳入一个统一框架进行联合优化，是提升性能的关键
- 现代SOTA模型往往是混合范式的产物，并大量依赖于高级训练策略。理解这些技巧背后的动机，是复现和超越顶尖结果的必要条件