

通用视觉框架OpenMMLab
第3讲 目标检测与MMDetection(上)

陈恺
2021年4月

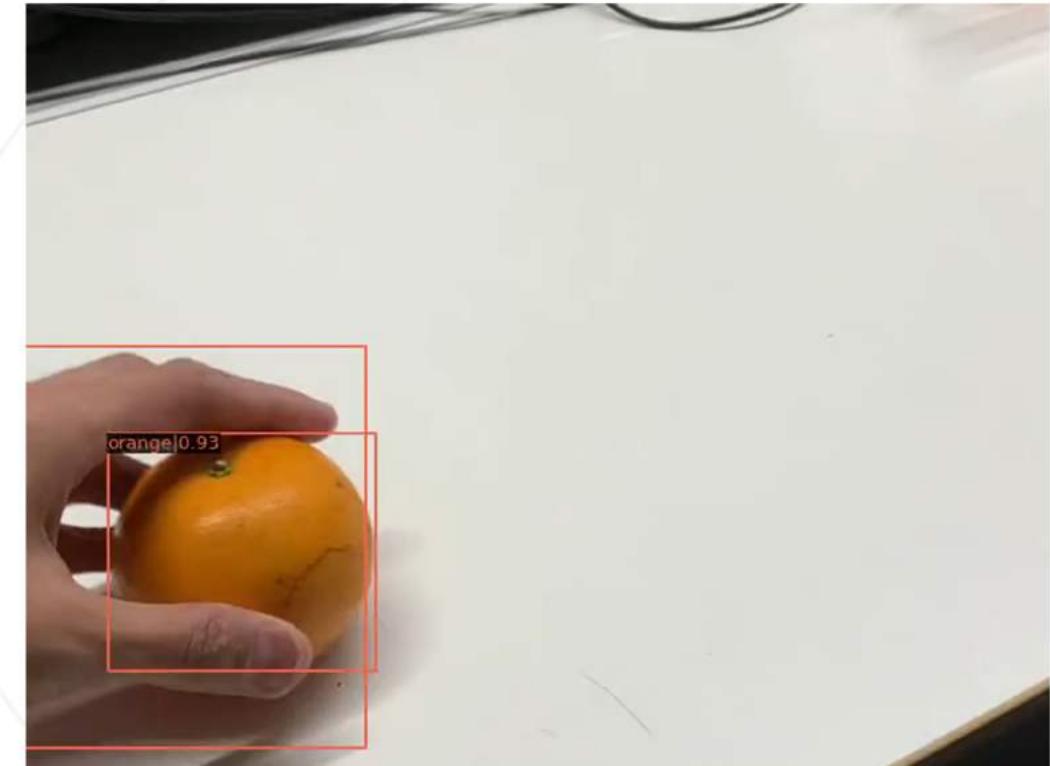
图像分类：

图像中通常只有一个（主要）物体
只需要进行类别预测



目标检测：

图像中有不定数目的物体
分类同时还需要定位物体

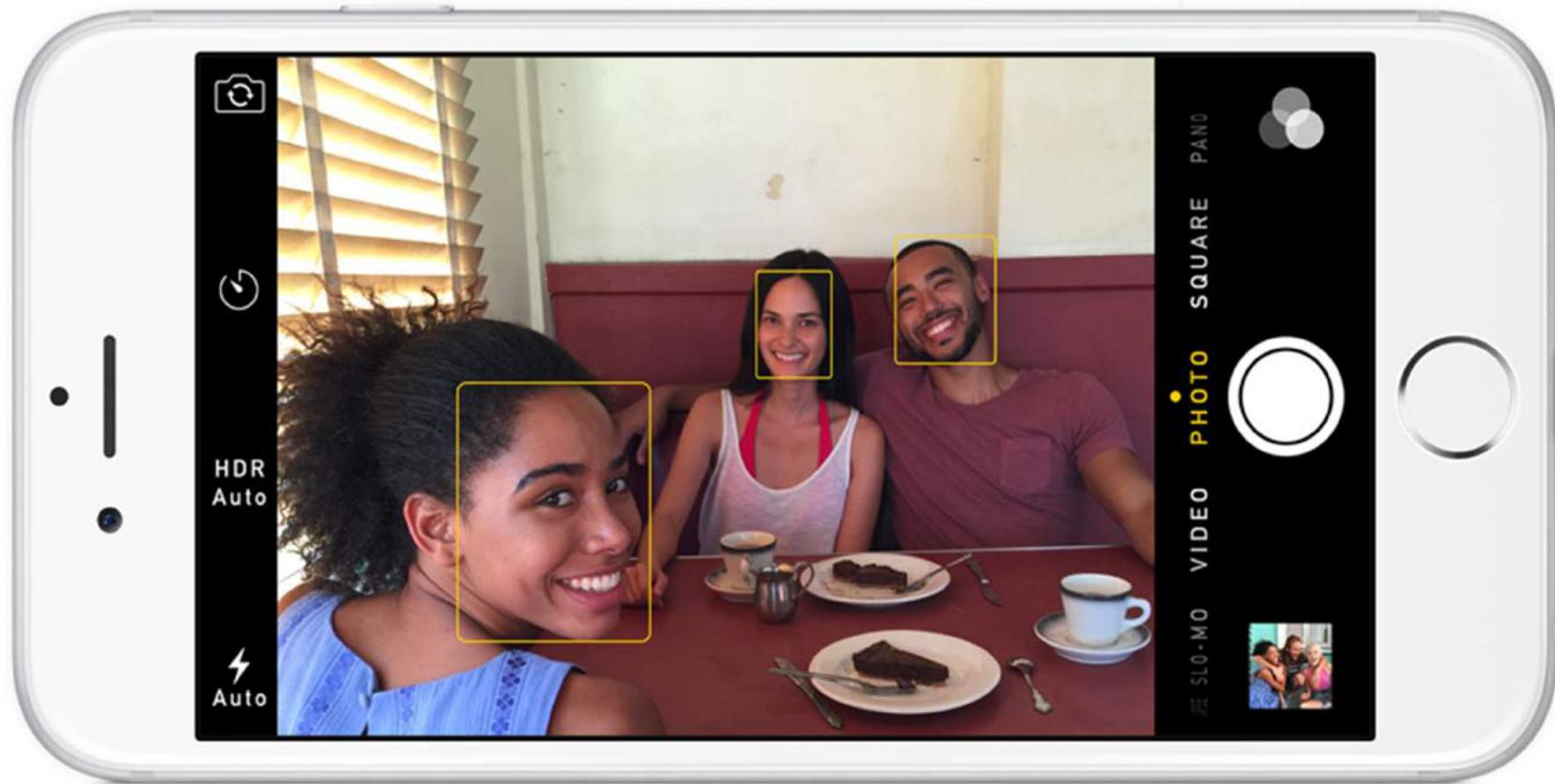


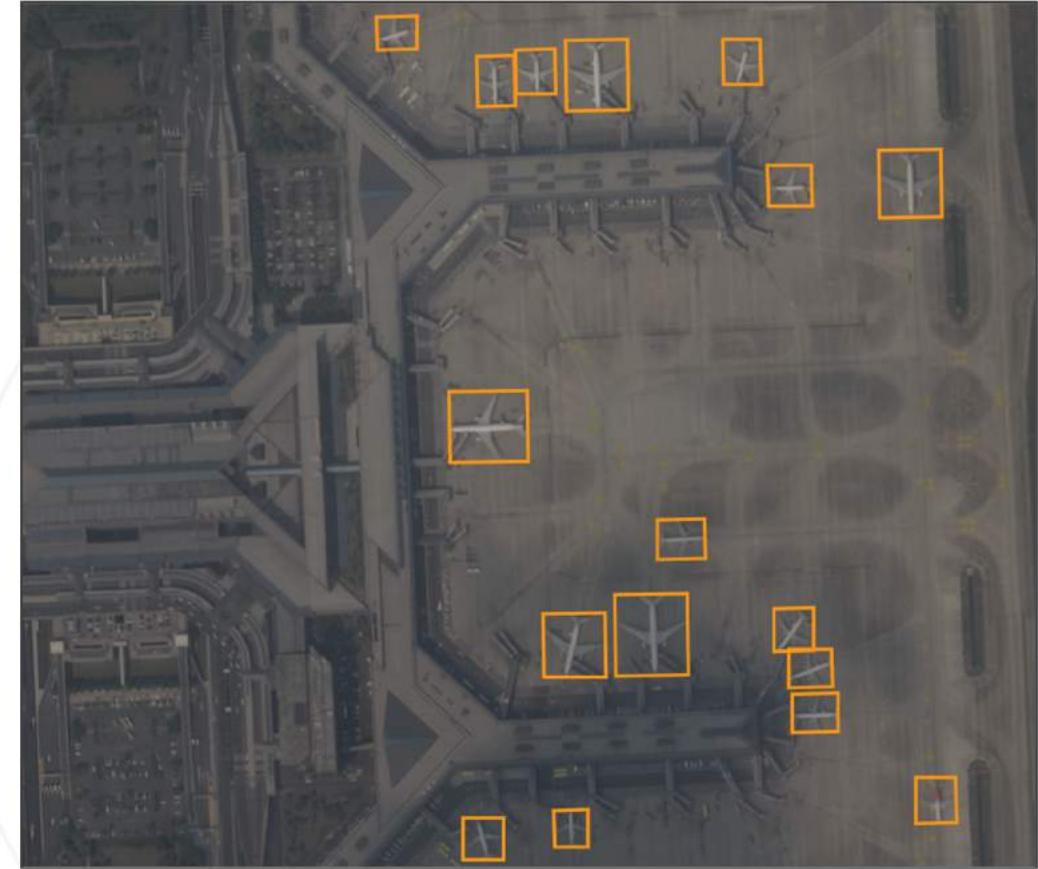


微信扫一扫

拍照中的人脸检测

OpenMMLab





▶ 本节内容：

- 目标检测的基本思想
- 两阶段算法 Two-staged Methods
 - RCNN 算法
 - Fast RCNN 算法
 - Faster RCNN 算法
 - FPN 算法
- 实践 MMDetection 1
 - 使用预训练模型进行推理

▶ 下节预告：

- 一阶段算法
- 无锚点算法
- 实例分割
- 检测算法的评估方法
- 实践 MMDetection 2
 - 训练目标检测模型

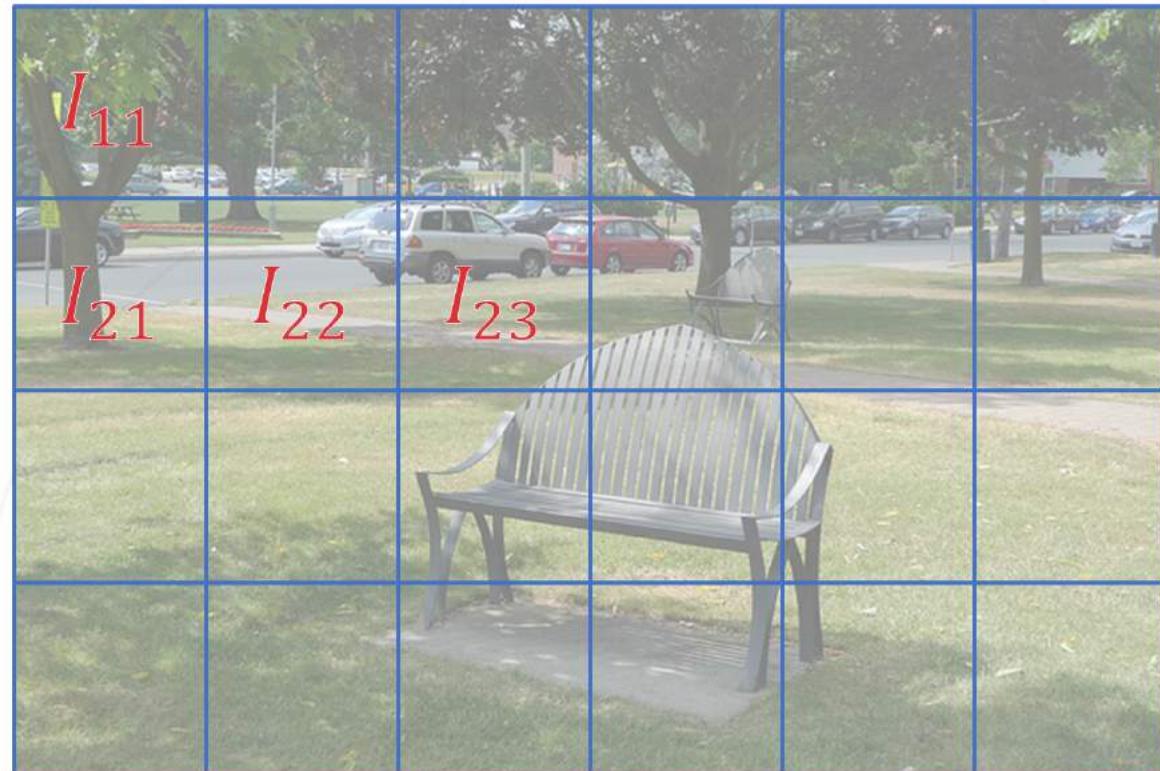
目标检测的基本思路

假设已经具备了图像分类算法 $C(I)$

1. 将图像切分成多块
2. 用分类算法 $C(I_{ij})$ 对每个图像块进行预测
3. 检测结果=(分类结果, 图像块位置)

问题:

过于粗糙, 无法检测分块边界上的物体



物体区域

类别: $C(I_{23}) = \text{car}$

位置: $L(I_{23}) = (3w, 2h)$

背景区域

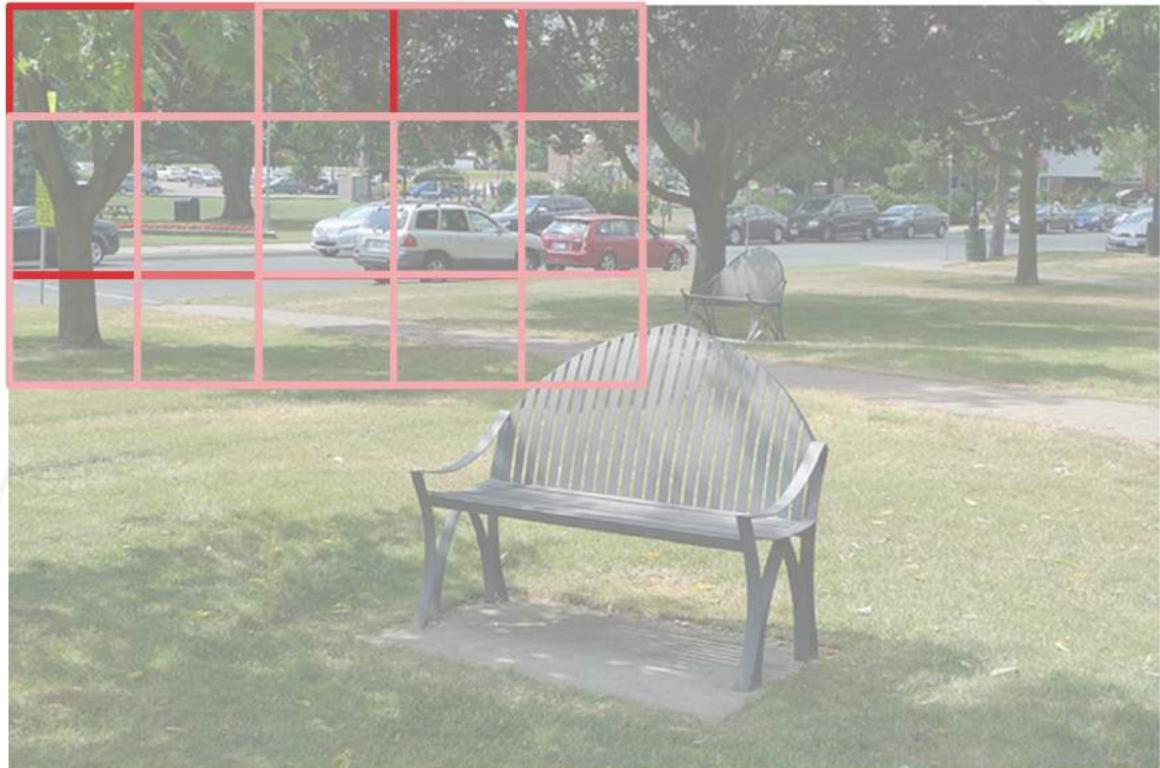
类别: $C(I_{22}) = \text{background}$

位置: 不关心

问题：图像分块过于粗糙，无法检测分块边界上的物体

改进：

1. 使用重叠的窗口，覆盖更多可能出现物体的位置
2. 用分类算法 $C(I)$ 检测每个图像块
3. 检测结果=(分类结果，滑窗位置)



边界框回归 bounding box regression

? 问题：滑窗边界与物体精确边界有偏差

改进：分类的同时预测物体的精确位置
→ 边界框回归

分类模型 $C(I_{ij})$ 预测物体类别

回归模型 $R(I_{ij})$ 预测物体精确位置相对于窗的偏差

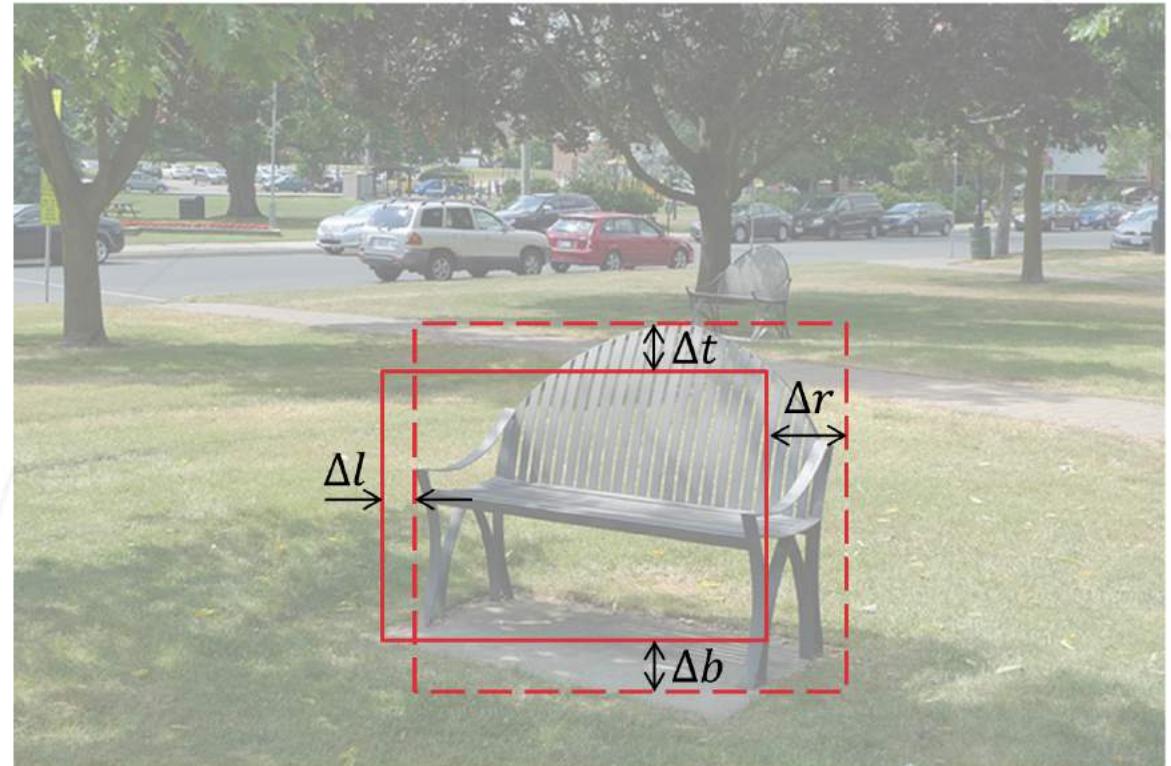
检测结果 = $(C(I_{ij}), \text{窗位置} + R(I_{ij}))$

多任务学习 (multi-task learning)



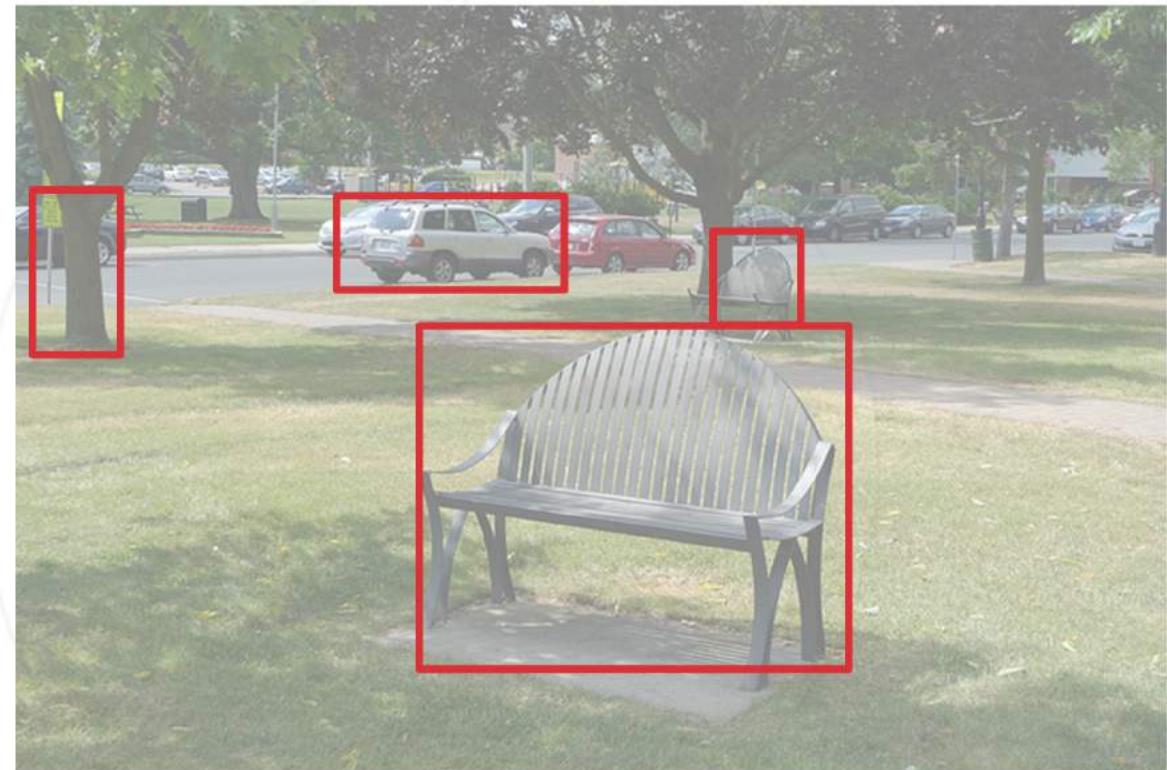
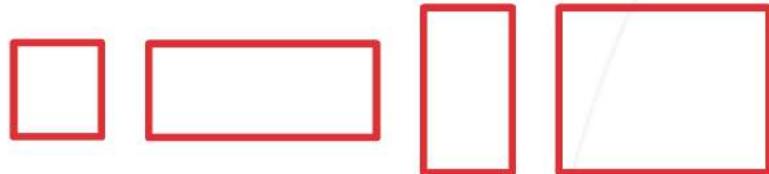
$$C(I_{ij})$$

$$R(I_{ij}) = (\Delta l, \Delta r, \Delta t, \Delta b)$$



? 问题：不同物体大小不同，长宽比不同

□ 改进：使用大小、长宽比不同的滑窗



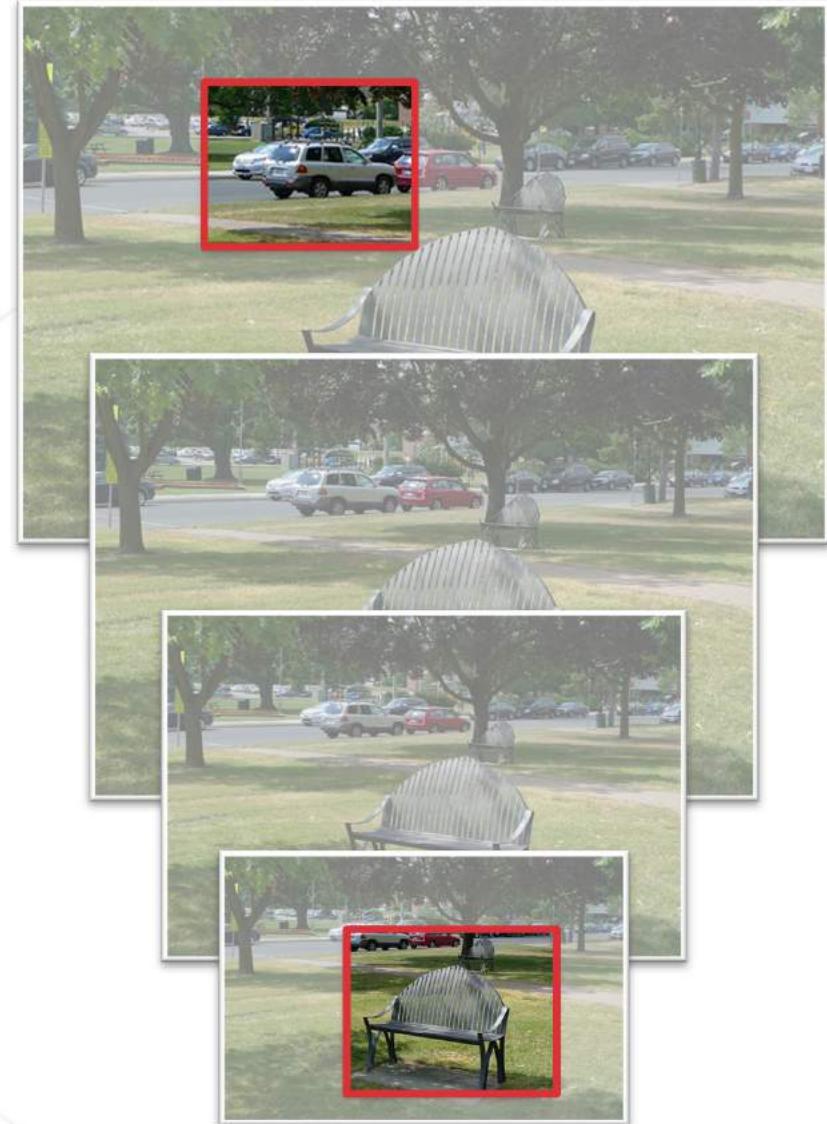
？ **问题：**不同物体大小不同，长宽比不同

□ **解决方案：**将图像缩放到不同大小，构建图像金字塔 (Image Pyramid)。

相同大小的窗口在不同尺寸的图象上可以检测不同尺寸的物体。

图像逐级缩小

可检测的物体逐级放大



滑窗 = 空间上的密集预测

考虑 800x600 的图像，使用 80x60 的窗，步长 10 像素滑动

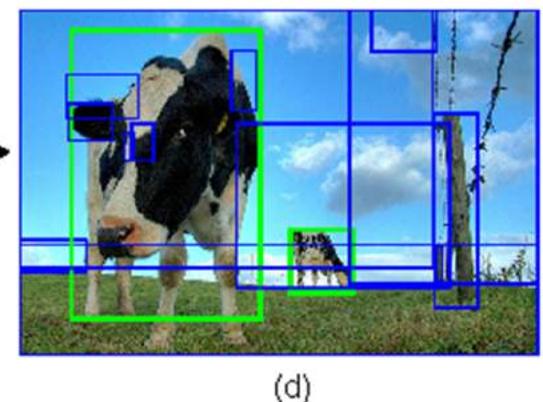
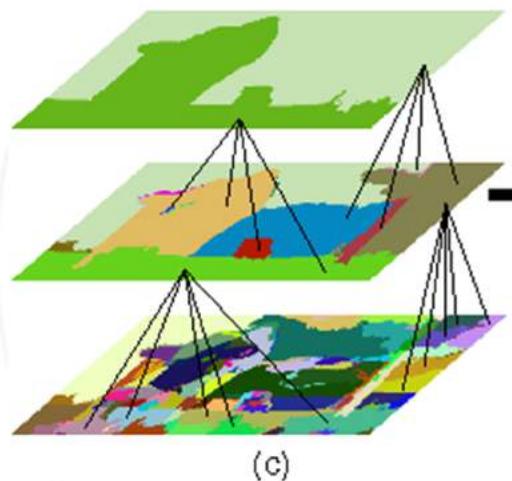
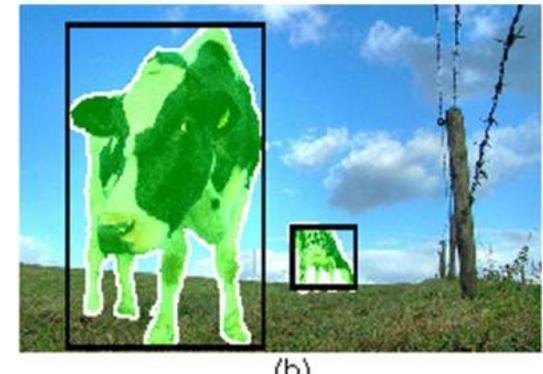
需要在 4800 个窗上进行分类预测

- × 每个位置5个尺度的窗
- × 每个尺度3个长宽比
- ≈ 检测一张图像需要完成数万次图像分类预测

难以满足实时检测的需求

分析：

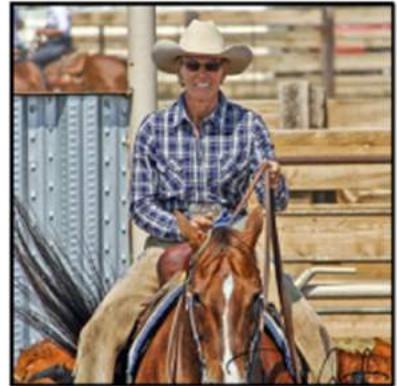
大量窗口都落在不包含物体的边界区域。可以先通过简单快速的方法找出可能含物体的区域。



Selective Search 算法：

使用贪心算法，将空间相邻且特征相似的图像块逐步合并到一起，形成可能包含物体的区域，称为提议区域或提议框。

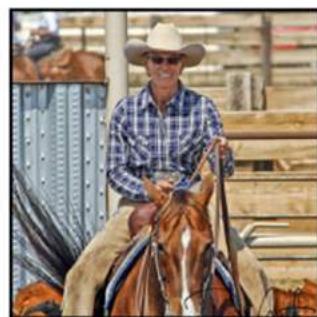
两阶段目标检测算法



输入图片

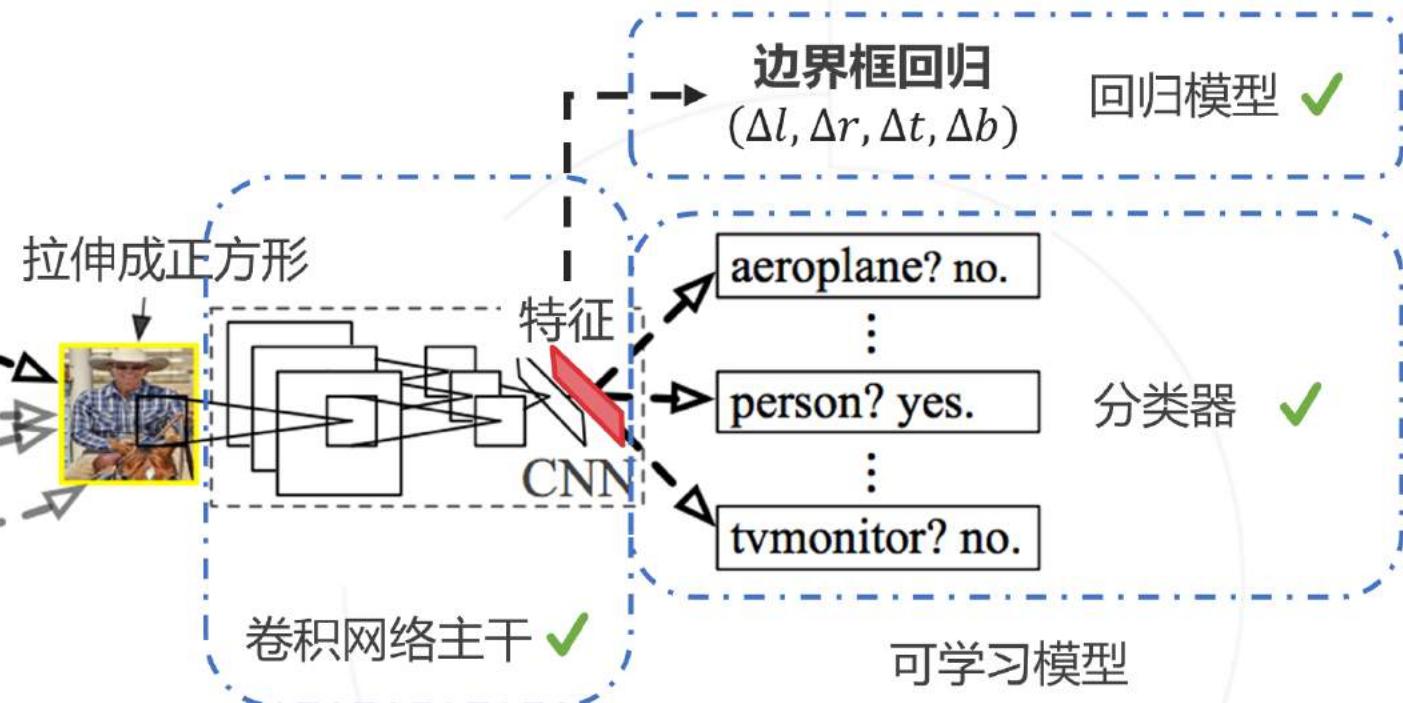
R-CNN 算法包含两个步骤，因此也称为两阶段方法

模型中的哪些模块需要学习？



区域提议 ×

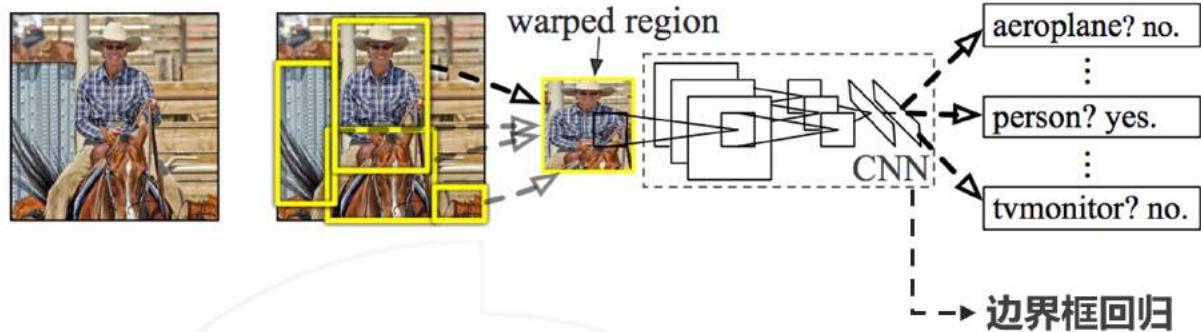
固定算法
无可学习参数



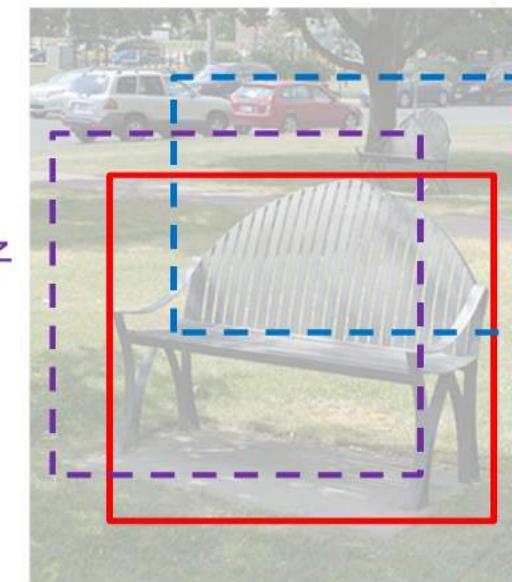
如何生成分类和回归的目标值？

对于提议框 P 和标注框 B

- 如果二者重叠较大
分类目标值 := B 的类别标注
回归目标值 := 编码后的偏差值
- 如果二者重叠较小
分类目标值 := 背景
回归分支不计算 LOSS



提议框 P1
重叠较大
类别:=椅子



提议框 P2
重叠较小
类别:=背景

标注框 B
类别=椅子

为使回归模型更易于学习，通常使用如下编码策略对边界框偏移量进行编码：

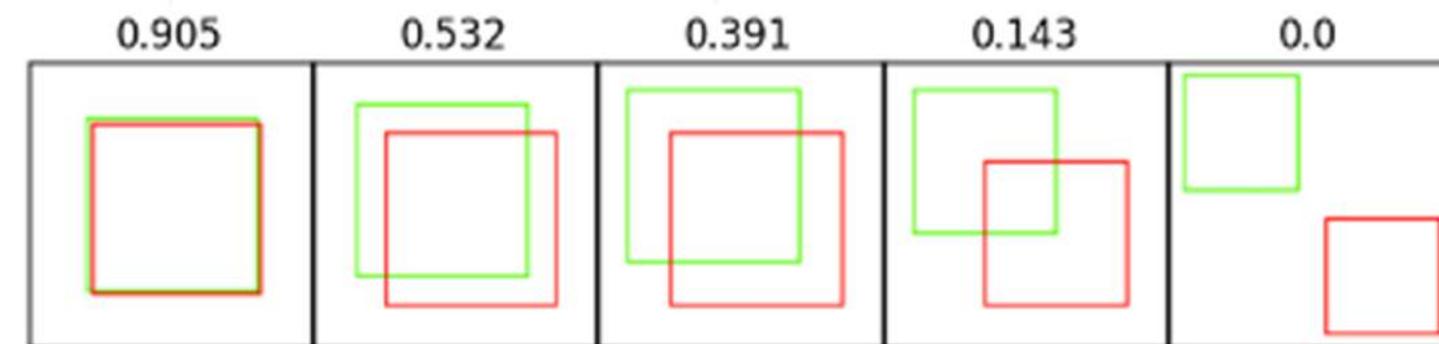
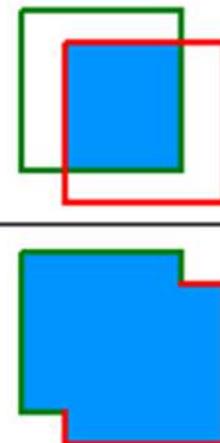
对于提议框 $P = (p_x, p_y, p_h, p_w)$ 和真值框 $B = (b_x, b_y, b_h, b_w)$

边界框的偏移量，即回归模型的预测目标为：

$$(t_x, t_y, t_h, t_w) = \underbrace{\left(\frac{b_x - p_x}{p_w}, \frac{b_y - p_y}{p_h}, \log\left(\frac{b_w}{p_w}\right), \log\left(\frac{b_h}{p_h}\right) \right)}_{\text{位置偏差基于边长归一化}} \quad \underbrace{\text{尺度比例的对数}}$$

交并比 IoU 定义为两个矩形框交集面积与并集面积的比值，是对两个矩形框重合度的衡量。

$$IOU = \frac{\text{area of overlap}}{\text{area of union}}$$



检测算法有时会针对单个物体给出多个相近的检测框。

在所有重叠框中，只需要保留置信度最高的。

非极大值抑制 (NMS) 算法：

输入：检测器产生的一系列检测框 $P = \{P_1, \dots, P_n\}$ 及对应的置信度

$s = \{s_1, \dots, s_n\}$, IoU 阈值 t

步骤：

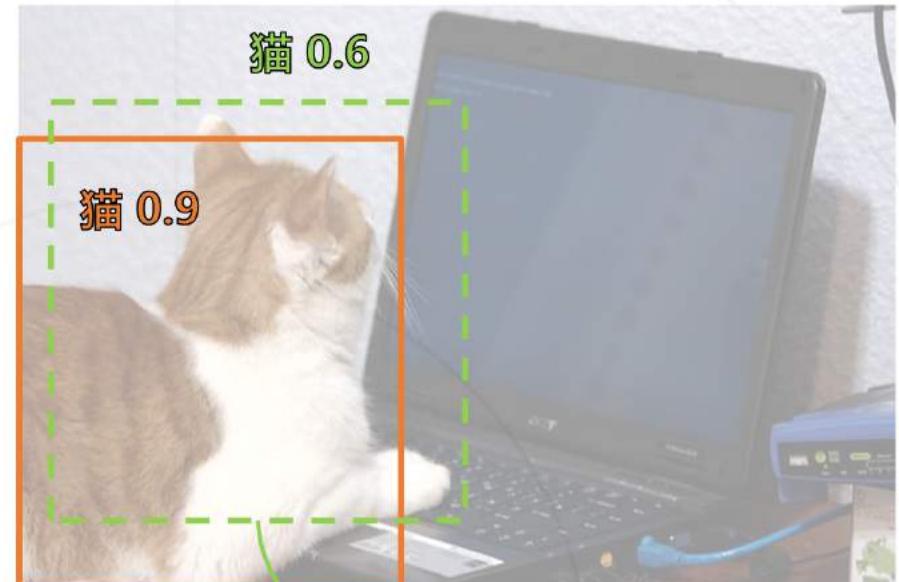
1. 初始化结果集 $R = \emptyset$

2. 重复直至 P 为空集

① 找出 P 中置信度最大的框 P_i 并加入 R

② 从 P 中删除 P_i 以及与 P_i 交并比大于 t 的框

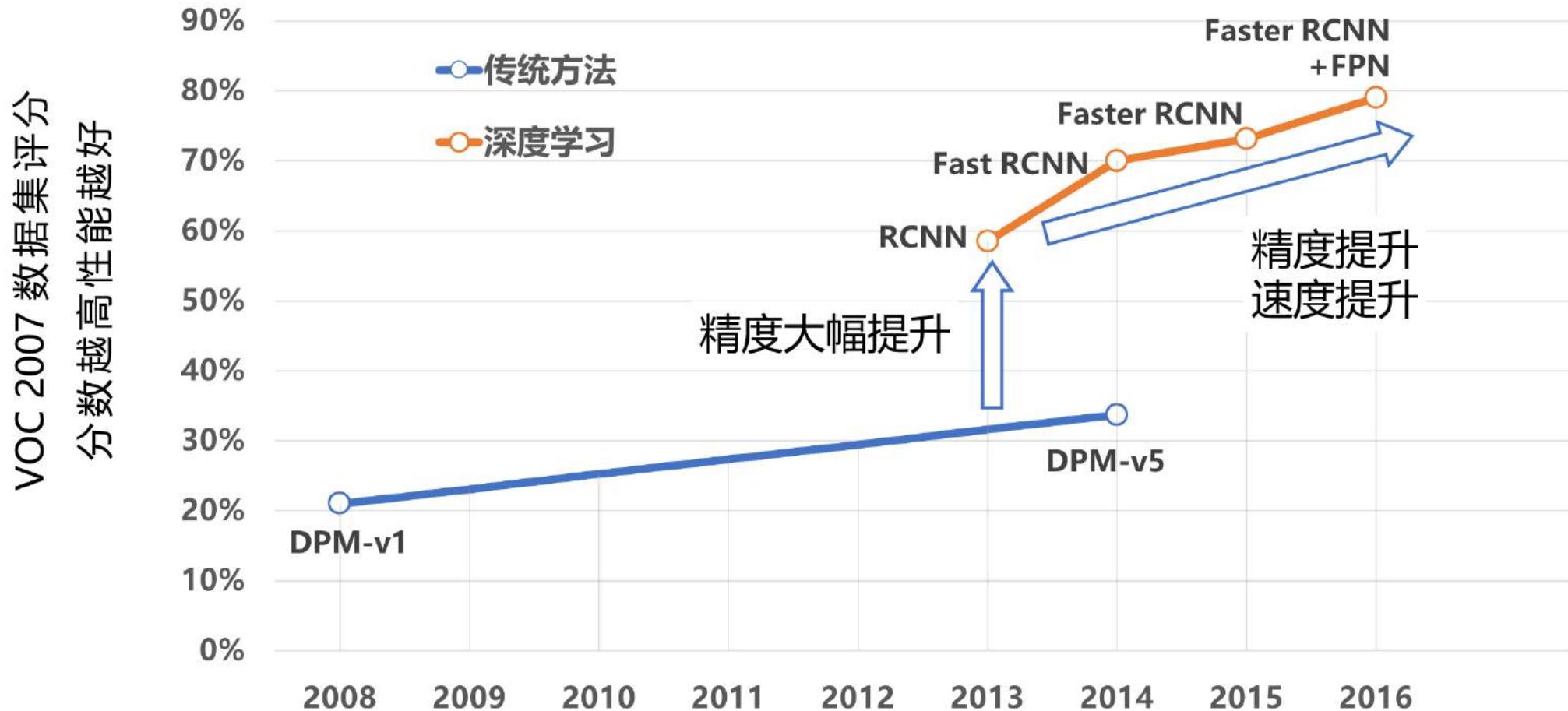
输出：结果集 R



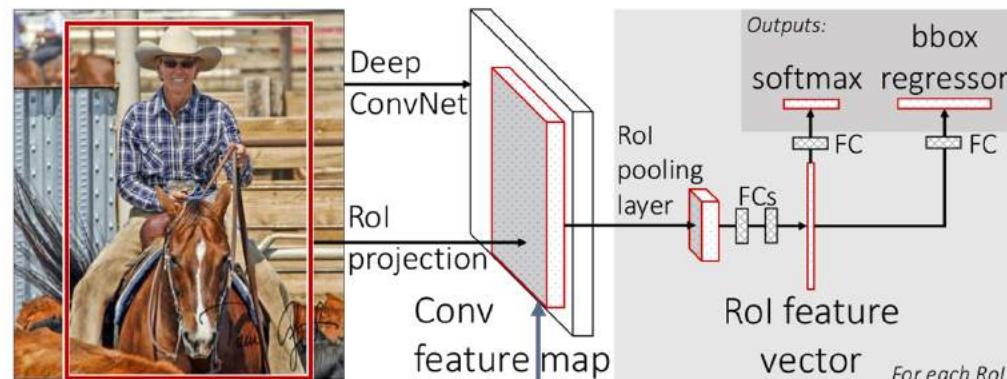
删除绿色框，因为它和置信度更高的橙色框重叠。

- 卷积网络使用 AlexNet
- 使用 Selective Search 产生提议框
- 非端到端训练，先训练卷积网络的主干部分，再基于卷积网络的特征训练 SVM 分类器
- 非多任务学习，先训练分类模型，再边界框回归单独训练
- 根据配置不同，单图推理几秒~几十秒

RCNN 相比于传统方法的提升



Fast RCNN 2014



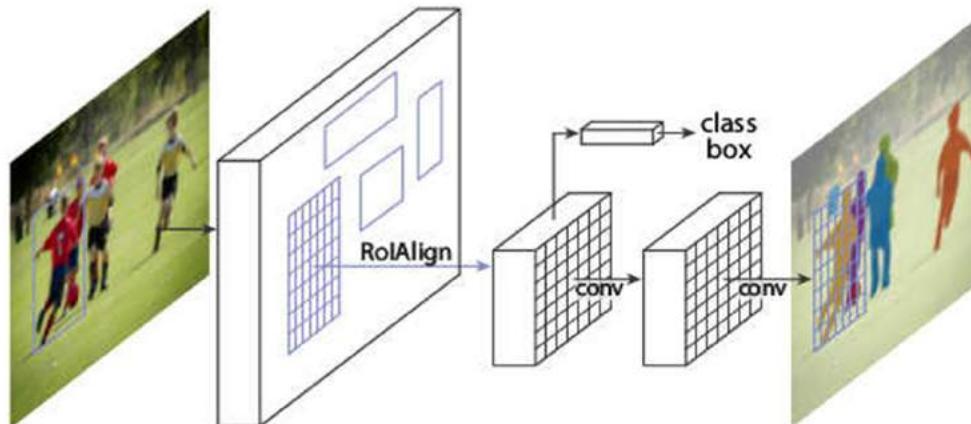
提出了 ROI Pooling 方法，把区域从图像移动到特征图上，大幅降低了计算量。

Faster RCNN 2015



提出了 RPN 网络，用于替换传统方法，产生区域提议，进一步提高效率。

Mask RCNN 2017

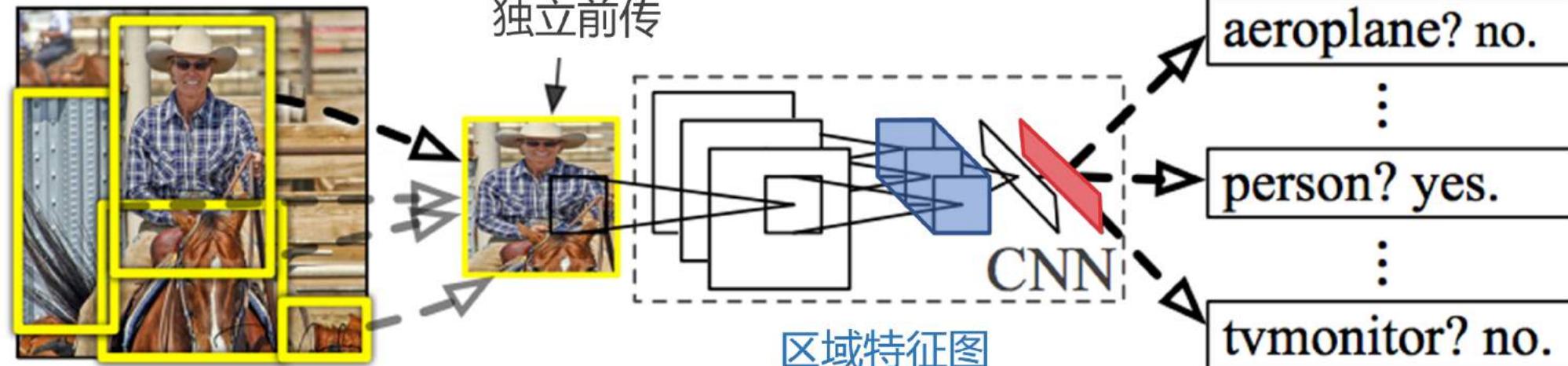


提出了 ROI Align 算法升级替换 ROI Pooling。
加入用于实例分割的分支。

问题：数千个提议框

数千次全图CNN前传

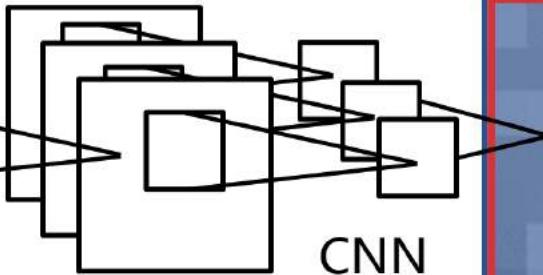
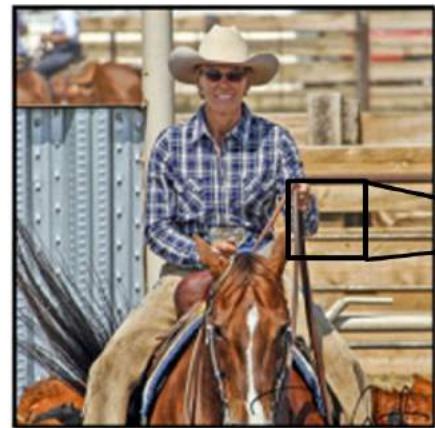
慢！



分析：提议框大量重叠

重叠区域大量重复卷积计算

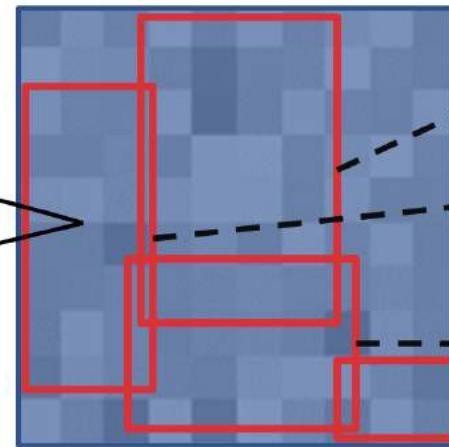
改进思路：整张图片输入主干网络，共享卷积计算，再在特征图上完成裁框



全图输入卷积网络

单次前传

全图特征图



类别预测 + 边界框回归
类别预测 + 边界框回归
类别预测 + 边界框回归

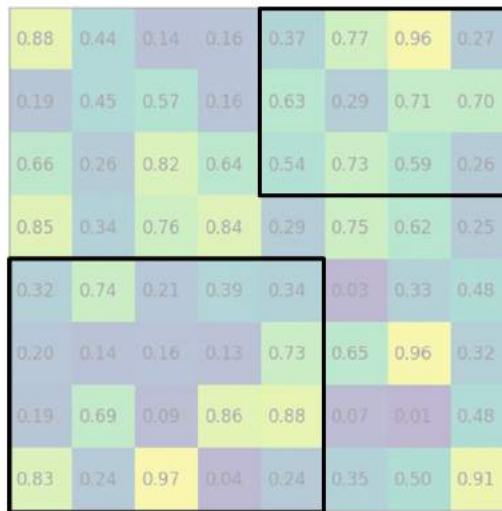
基于区域特征图的**轻量**预测

不同框重叠的部分共享计算

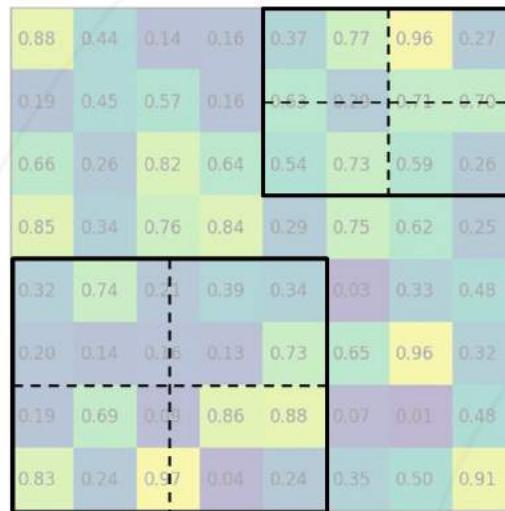
问题 1：如何将提议框映射到特征图上？→ 按照几何比例缩放

问题 2：如何将提议框内不定尺寸的特征图变成固定尺寸？→ RoI Pooling

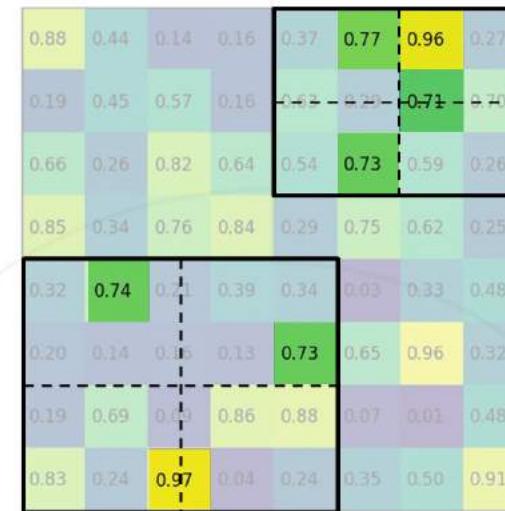
全图特征图与提议框



切分提议区域



各区域最大值



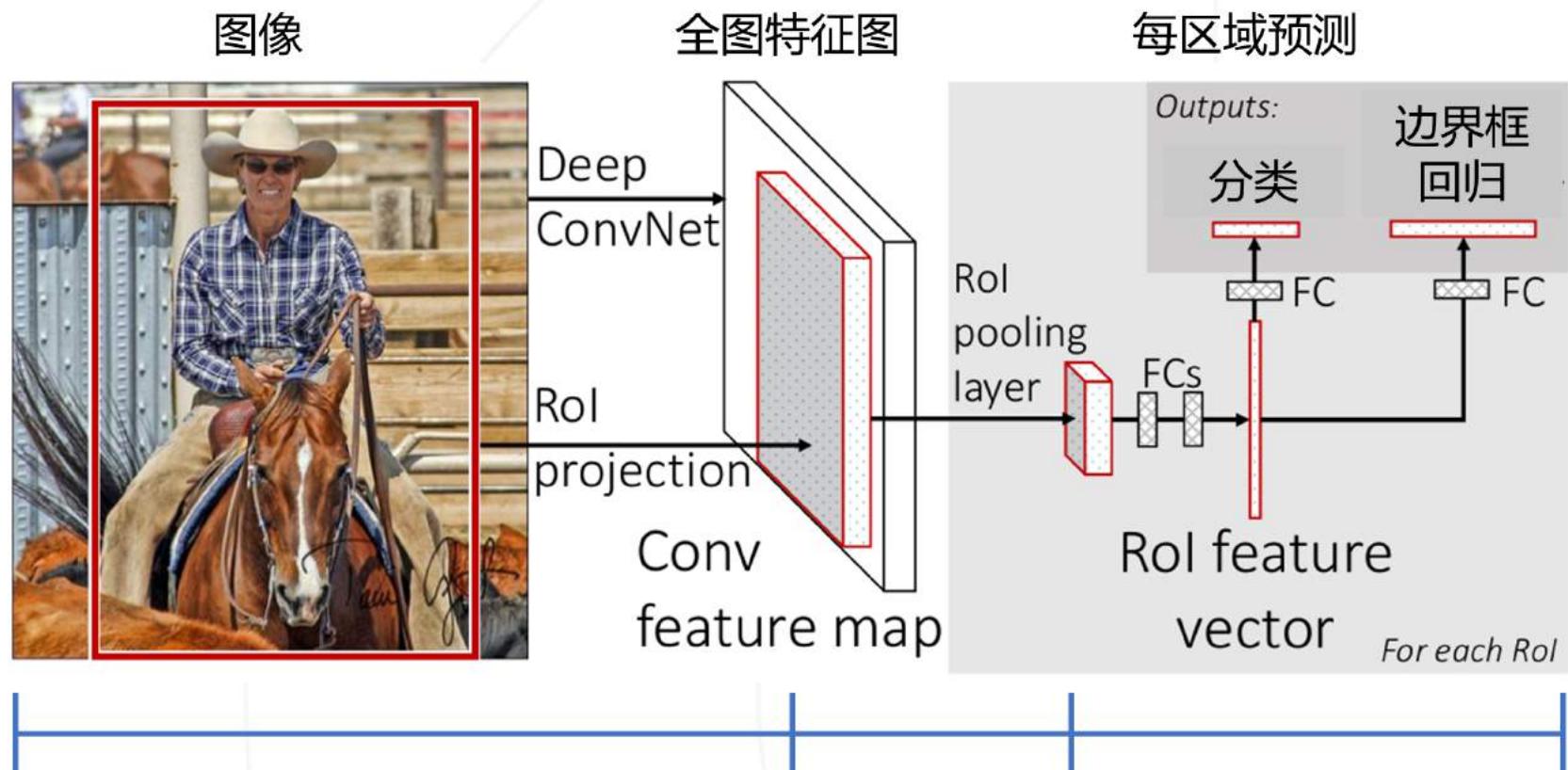
池化结果

| | |
|------|------|
| 0.77 | 0.96 |
| 0.73 | 0.71 |
| 0.74 | 0.73 |
| 0.97 | 0.97 |

做法：将提议区域切分成固定数目的格子（上图中 2×2 ，实际常用 7×7 ），对于每个格子：

- 如果格子边界不在整数坐标，则膨胀至整数坐标
- 通过 Max Pooling 得到格子的输出特征

作用：将任意尺寸的提议区域映射至固定尺寸的特征图，同时保留图像特征

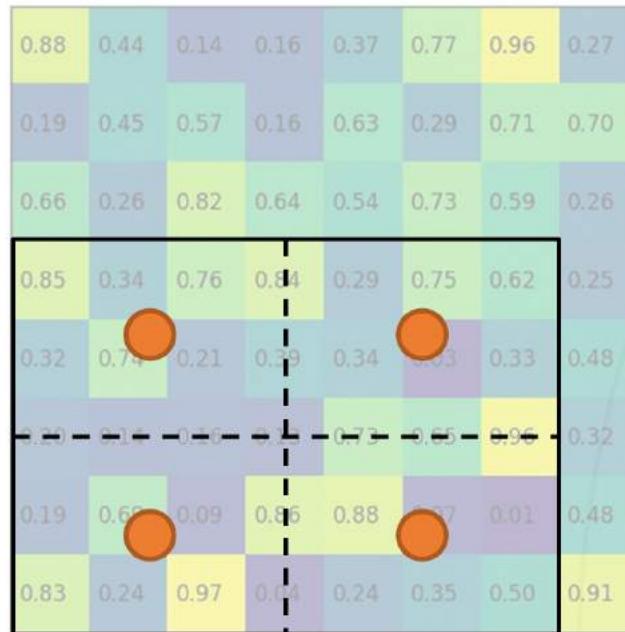


1. 将原图上的提议框按照几何比例缩放到特征图上

2. 利用 RoI Pooling，将不同尺寸的特征图变换到相同大小

3. 基于相同尺寸的特征图进行分类和回归

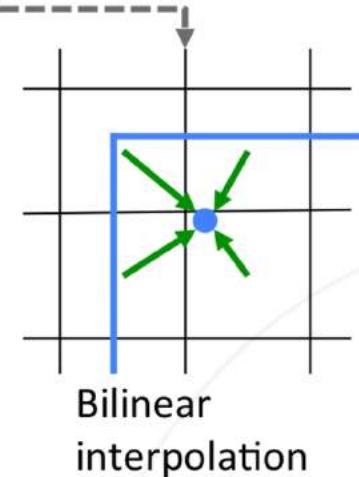
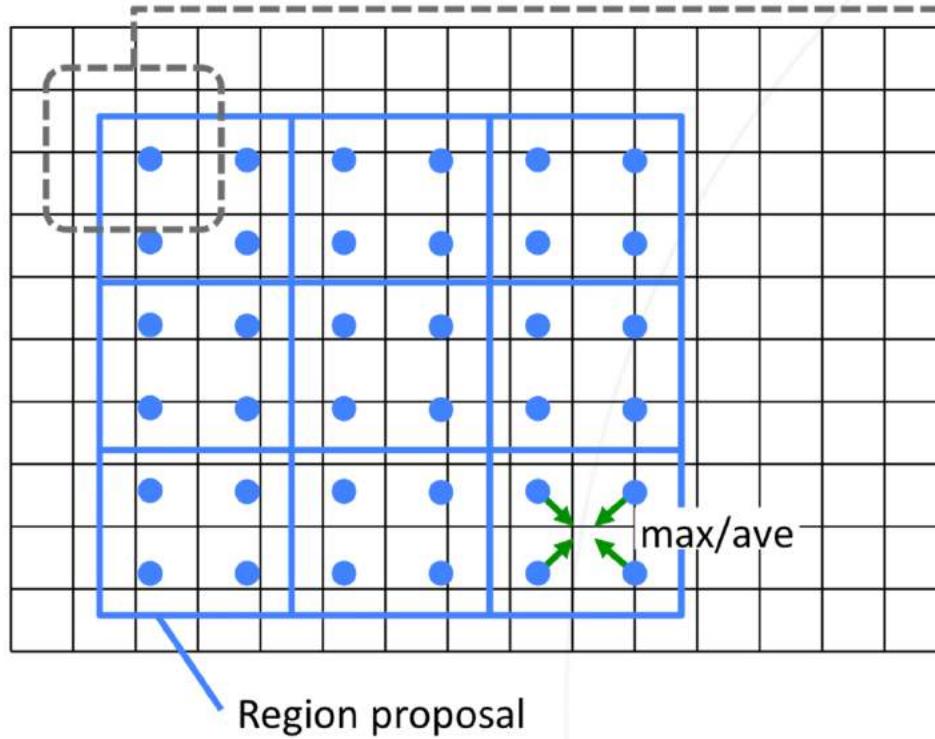
$\text{RoI} = \text{Region of Interest}$
提议框内的图像区域



问题: RoI Pooling 对非整数边界框取整，产生位置偏差

思路: 为保留空间精度，使用“非整数坐标”的特征

方法: 插值 -> RoI Align

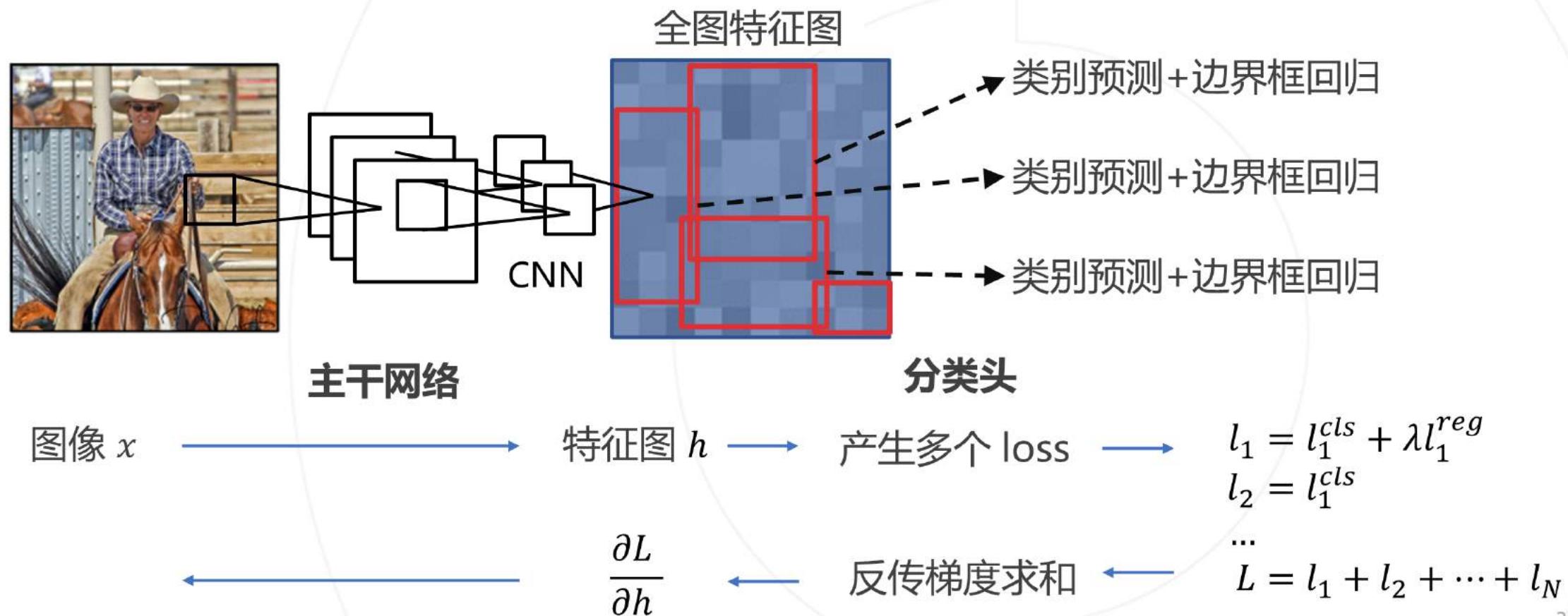


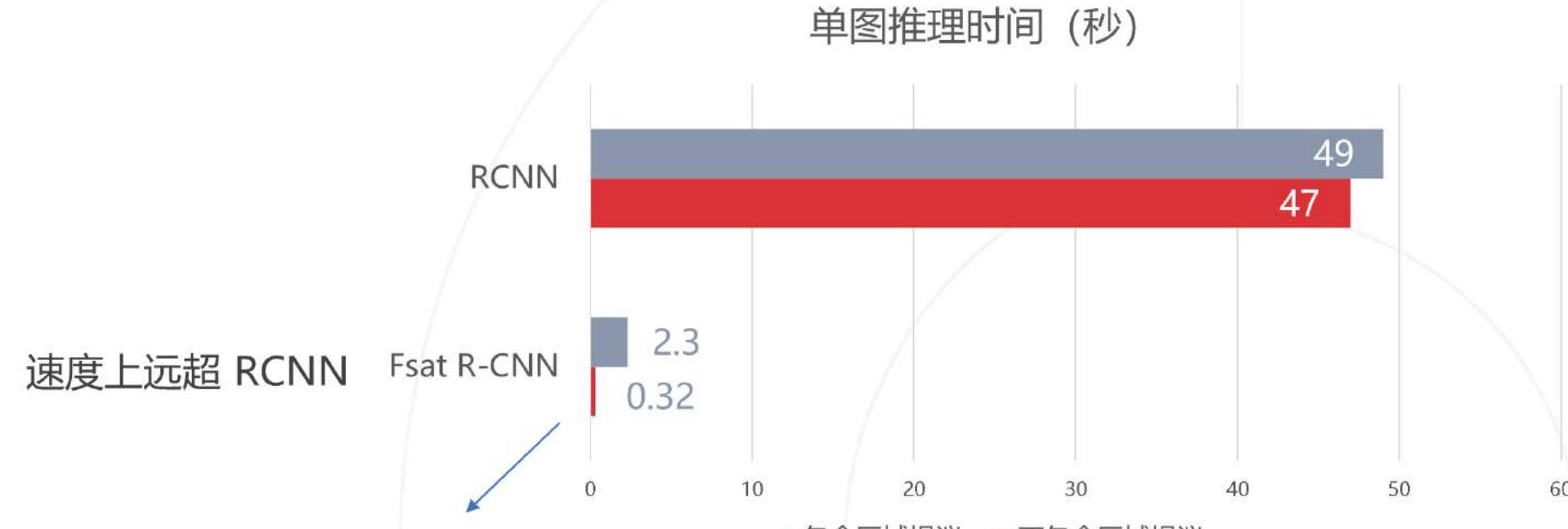
- 将提议区域切成固定数目的格子，例如 7×7
- 在每个格子中，均匀选取若干采样点，如 $2 \times 2 = 4$ 个
- **通过插值方法得到每个采样点处的精确特征**
- 所有采样点做 Pooling 得到输出结果

ROI Align 比 ROI Pooling 在位置上更精细。

与 RCNN 相同，比较真值框与提议框的 IoU，确定提议框的分类目标和回归目标

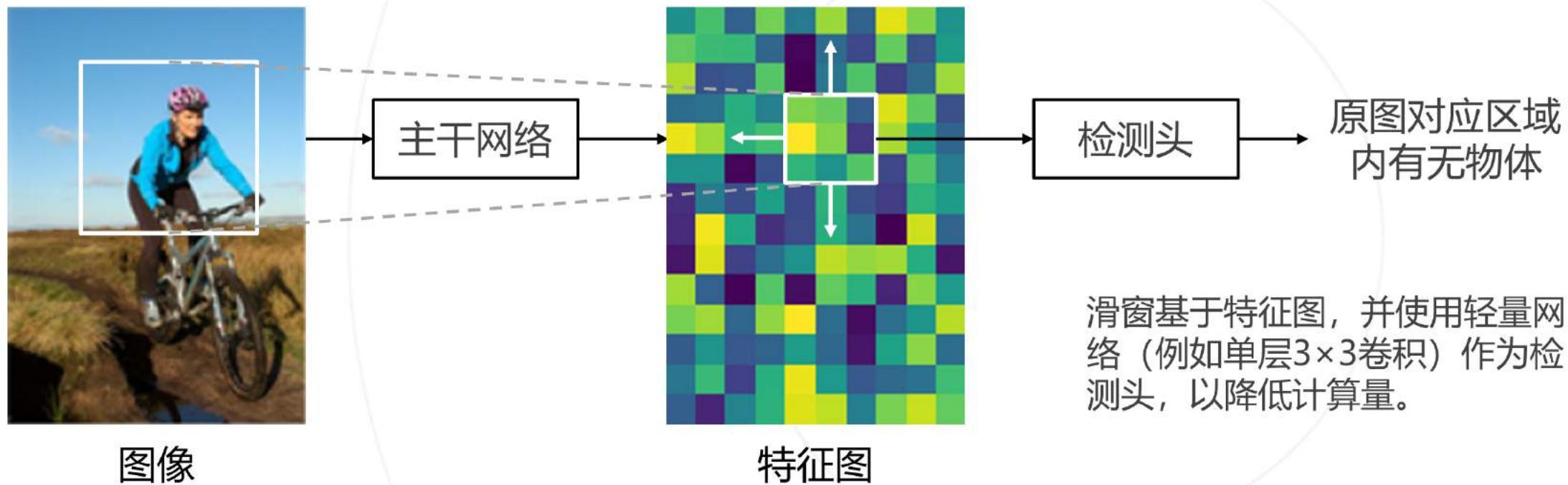
通过 RoI Pooling 的前传和反传实现端到端训练





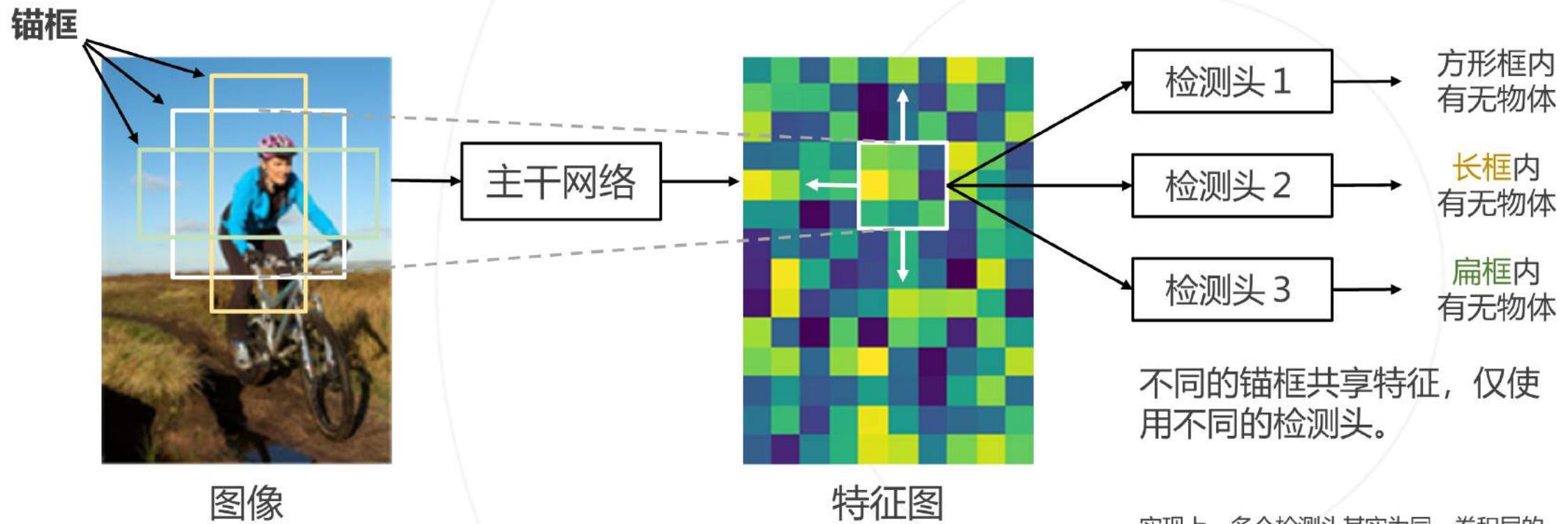
解决方案：使用卷积网络产生提议框，并与检测器共享主干网络结构

- 区域提议 = 在图中找到有物体的框 → 单类别的物体检测问题
- 采用滑窗的思路，进行空间密集预测



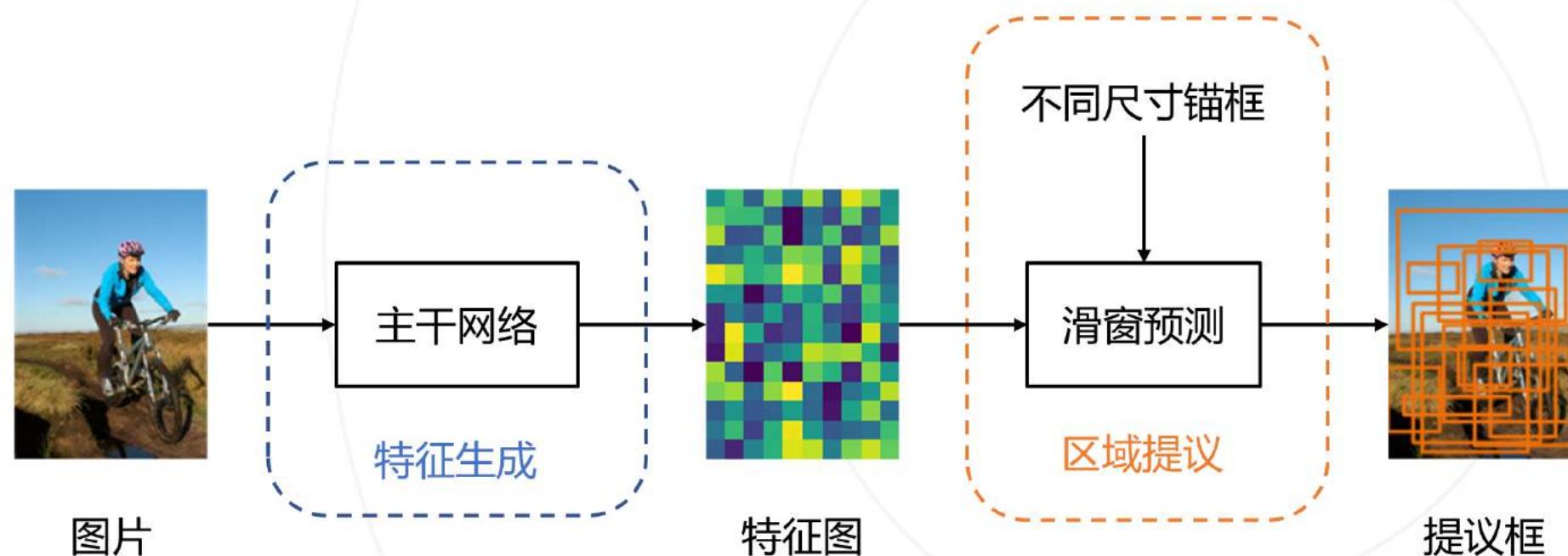
如何处理不同大小的物体？

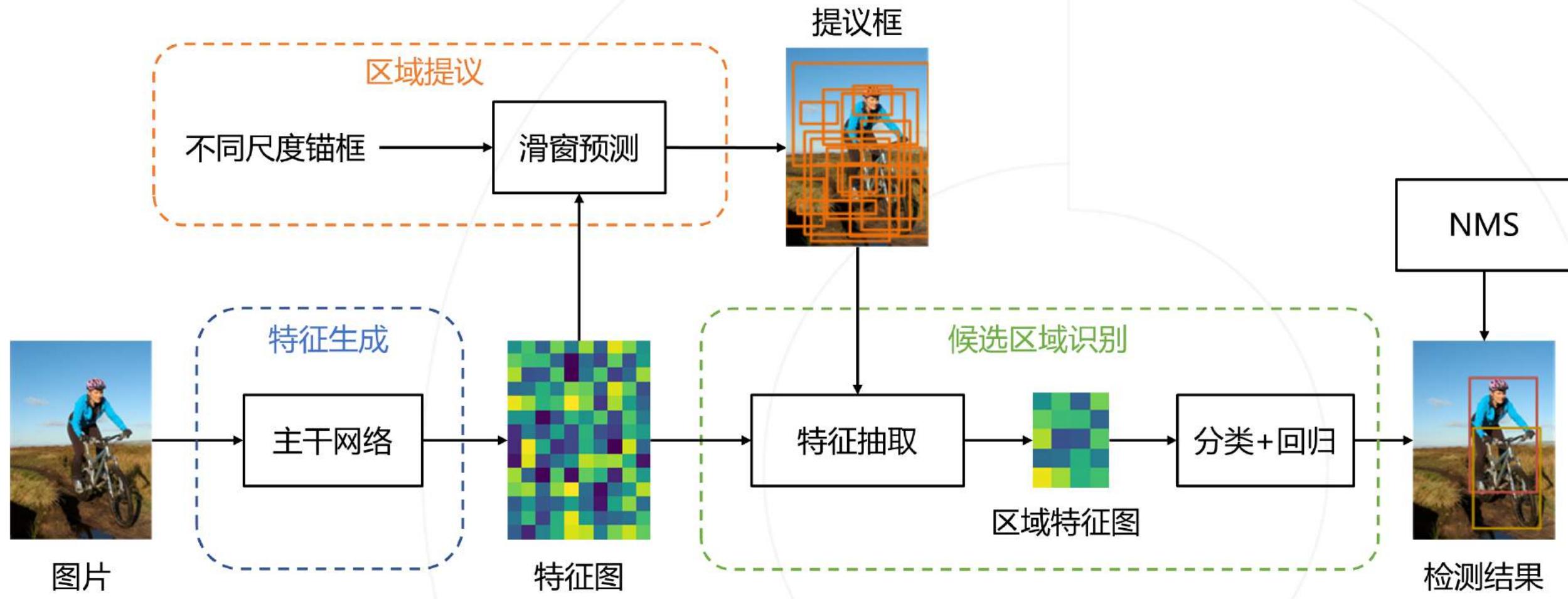
在原图上设置不同大小的假想框，用不同的检测头检测对应框中是否出现物体，称为锚框。



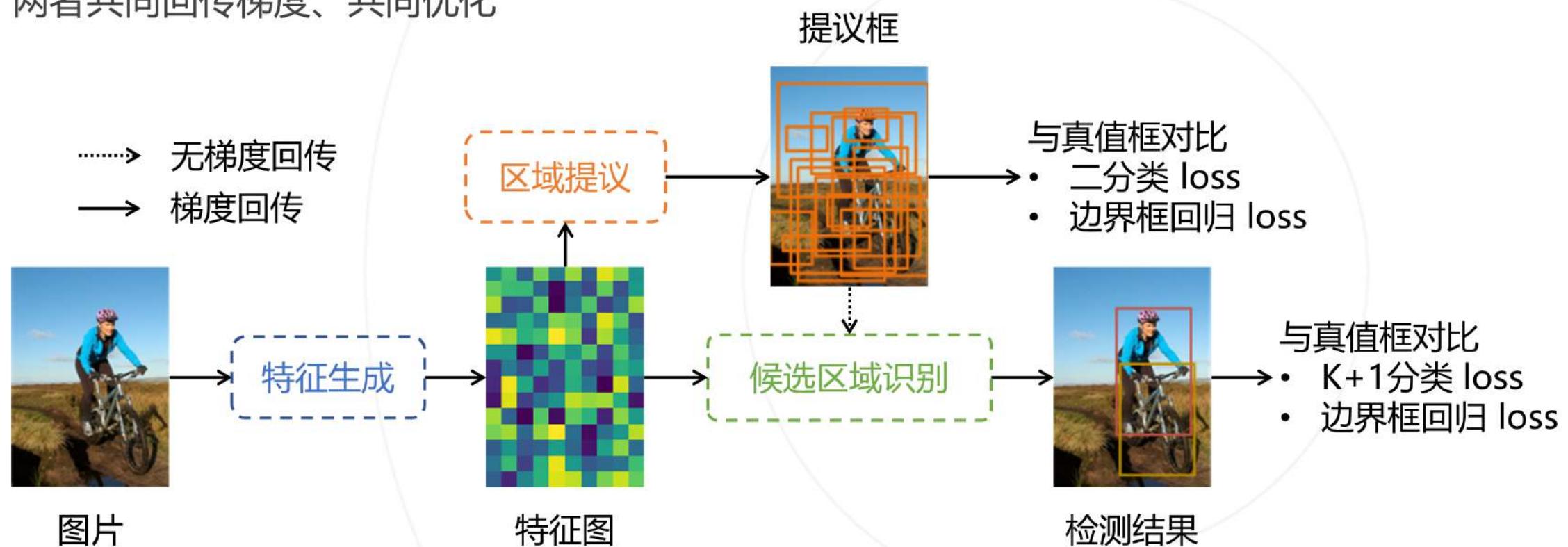
推理：

1. 主干网络产生特征图
2. 不同的检测头产生区域提议（分类+边界框回归）
3. 集合所有提议框，做 NMS





- Faster RCNN = RPN + Fast RCNN
- 每部分产生分类和回归 loss
- 两者共同回传梯度、共同优化



到此为止，模型基于单级特征图进行预测，通常是主干网络最后一层或倒数第二层。

问题：

高层次特征空间降采样率较大 → 小物体信息丢失

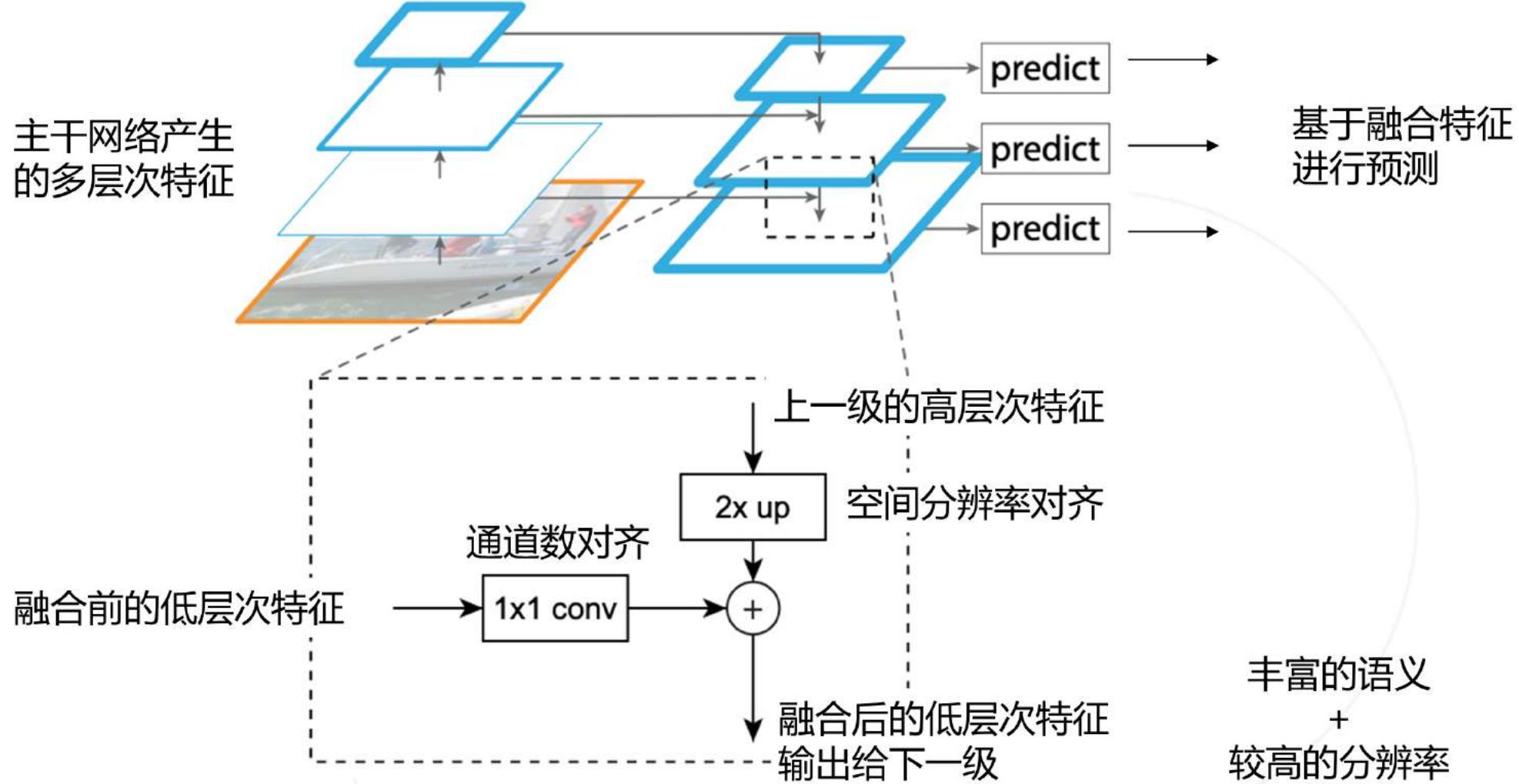
解决思路：

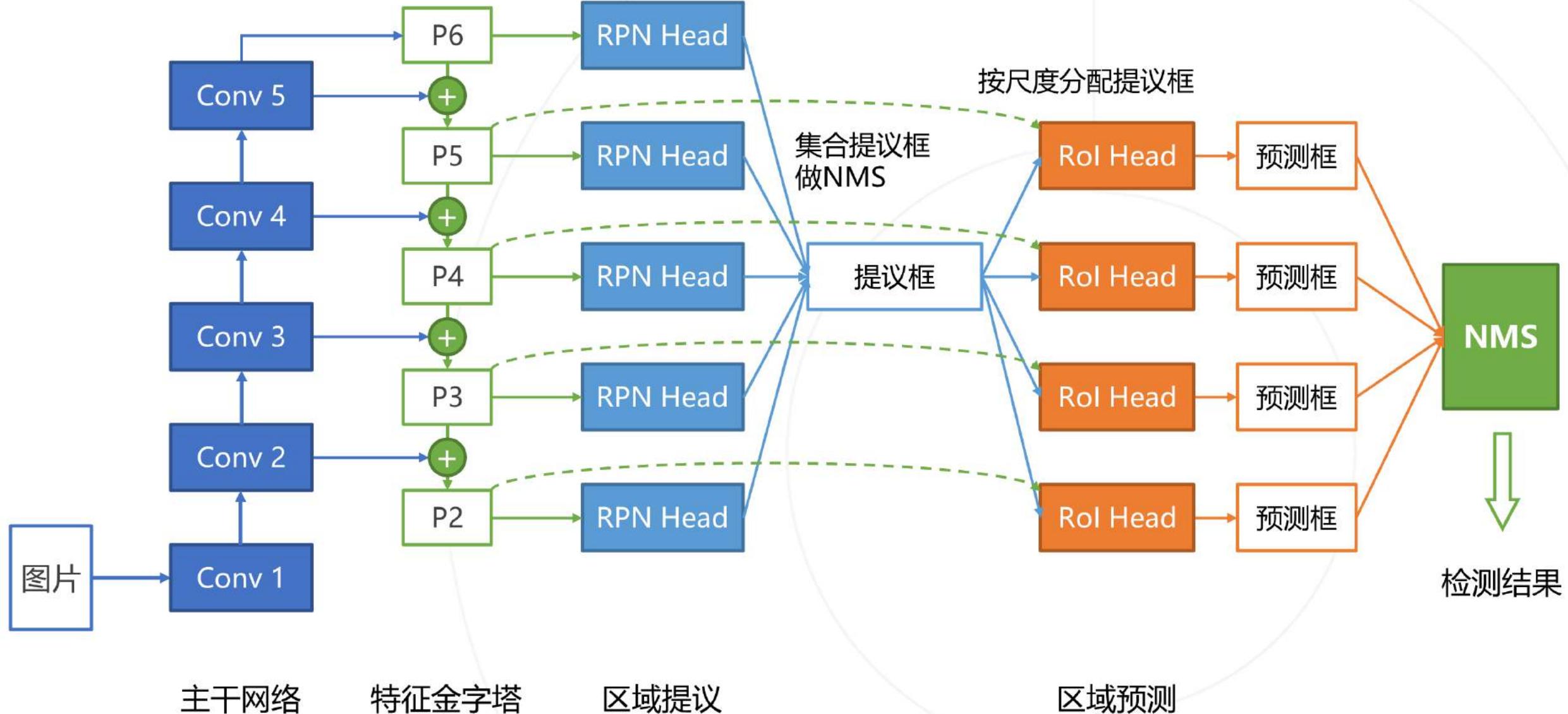
- 基于低一层特征图预测 → 低层特征语义信息薄弱
- 图像金字塔 → 低效率

} 融合高+低层次的特征



特征金字塔网络 FPN





MMDetection 介绍



任务支持

目标检测

实例分割

覆盖广泛

~380 个
预训练模型

~60 篇
论文复现

常用学术数据集

算法丰富

两阶段检测器

一阶段检测器

级联检测器

无锚点检测器

Transformer

- 2018-10 发布
- 2019-07 v1.0
- 2020-05 v2.0

使用方便

训练工具

测试工具

推理 API

科研论文



2019 年 6 月至今

谷歌学术引用 **超过 370 次**；
仅计算机视觉三大顶会上
被 **超过 50 篇论文** 作为基础代码库；

工业落地



商汤、腾讯、阿里、华为、
国内外初创公司，.....

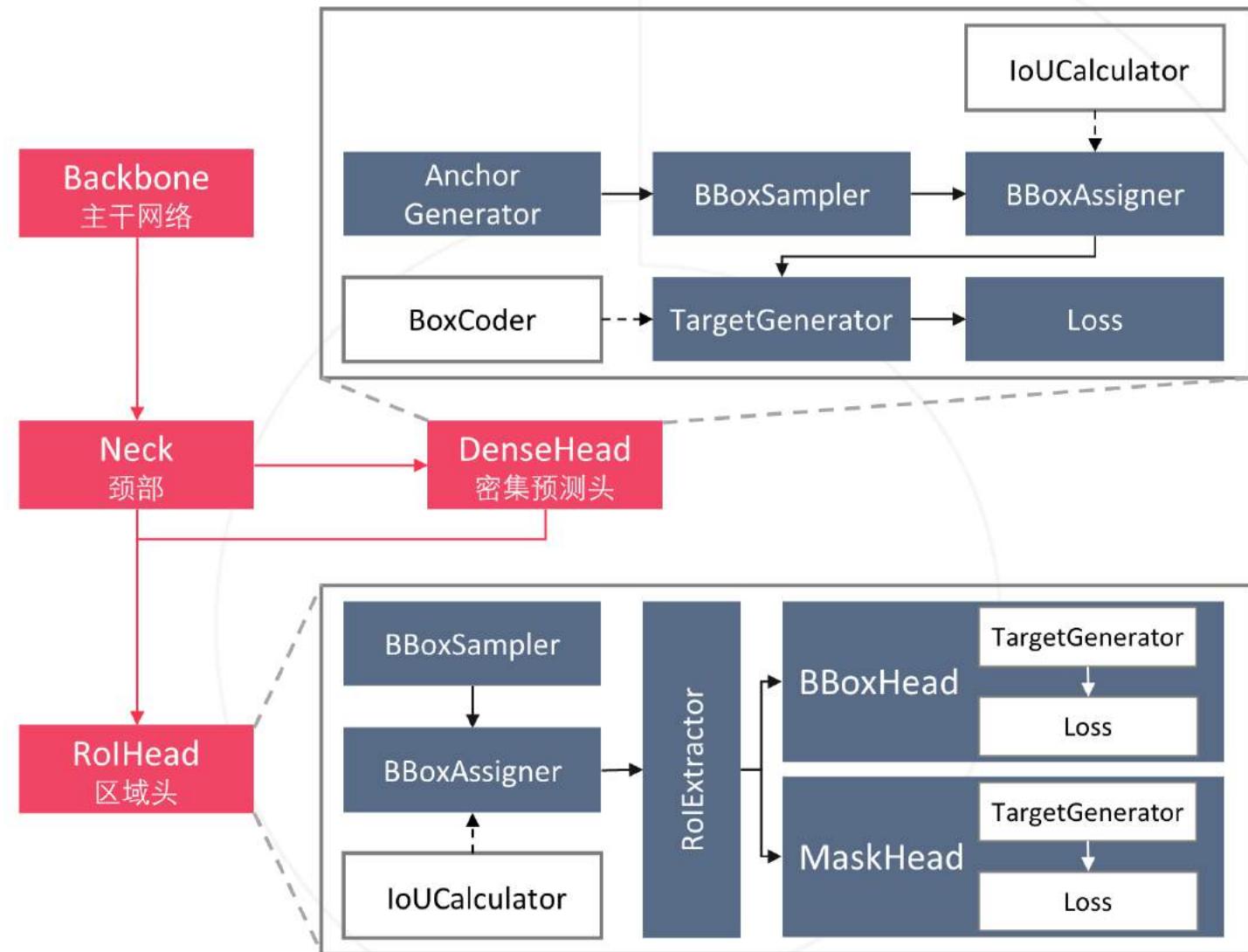


学术比赛



COCO 2018 实例分割**冠军**
COCO 2019 实例分割**冠军**
Open Images 2019 物体检测**冠军**
Global Wheat Detection**冠军**
Crowd Human 人体检测**冠军**
Materialist(FGVC6) 2019**冠军**

模块化设计是 OpenMMLab 的重要原则。在 MMDetection 中，我们将不同模型按照功能模块进行分解，方便用户自由组合和拓展。



open-mmlab / mmdetection

| | |
|----------------|------------|
| 📁 .dev_scripts | |
| 📁 .github | |
| 📁 configs | 配置文件 |
| 📁 demo | |
| 📁 docker | |
| 📁 docs | |
| 📁 mmdet | 核心工具包 |
| 📁 requirements | |
| 📁 resources | |
| 📁 tests | |
| 📁 tools | 训练、推理、测试工具 |

<https://github.com/open-mmlab/mmdetection>

| | |
|---------------|-------------------------|
| .. | |
| 📁 apis | 训练、推理、测试的高层次 API |
| 📁 core | anchor、bbox、mask 等模块的实现 |
| 📁 datasets | 数据集支持、数据预处理与数据增强 |
| 📁 models | 检测模型的实现 |
| 📁 utils | 辅助工具 |
| 📄 __init__.py | |
| 📄 version.py | |

下节课继续