

基于深度学习的目标检测方法

山世光

中科院计算所

深度学习时代的目标检测

■ 检测模型的变迁

□ 经验驱动：手工设计 → 数据驱动：自动学习

	深度学习之前 (2001~2013)	深度学习时代 (2013~现在)
检测框位置搜索	滑动窗口 (Exhaustive)	基于Proposal分类→从特征回归(Anchor → Anchor-free)
检测框大小和长宽比	固定大小和长宽比	回归模型，精调初始BBox
特征表示	手工设计的特征	采用深度网络建模，从数据自动学习特征+分类器（高度非线性）；关键是目标GT的定义方式【BBox, Corner, Center, Extreme】
分类器	线性模型，简单非线性模型	
模型架构	浅层模型，分而治之	所有功能集成在一个端到端的模型中
尺度搜索	图像/特征金字塔	图像/特征金字塔

几个CV任务的区分

Semantic Segmentation



GRASS, CAT,
TREE, SKY

No objects, just pixels

Classification
+ Localization



CAT

Single Object

Object
Detection



DOG, DOG, CAT

Multiple Object

Instance
Segmentation



DOG, DOG, CAT

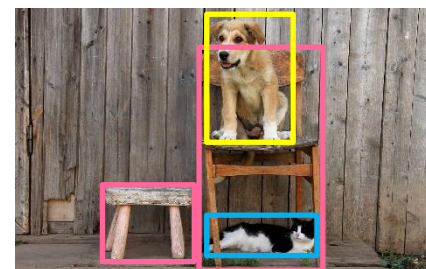
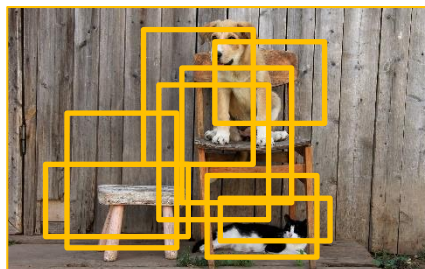
This image is CC0 public domain.

深度学习时代的目标检测

■ 主流的检测框架

- 两阶段检测器 (Two-stage detector)
- 单阶段检测器 (One-stage detector)

两阶段

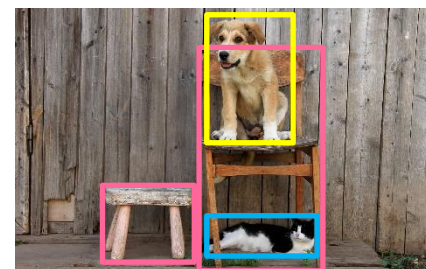


第一阶段: Region Proposal 第二阶段: 分类 & 调整检测框

单阶段



Single-Shot: 模型直接输出检测结果

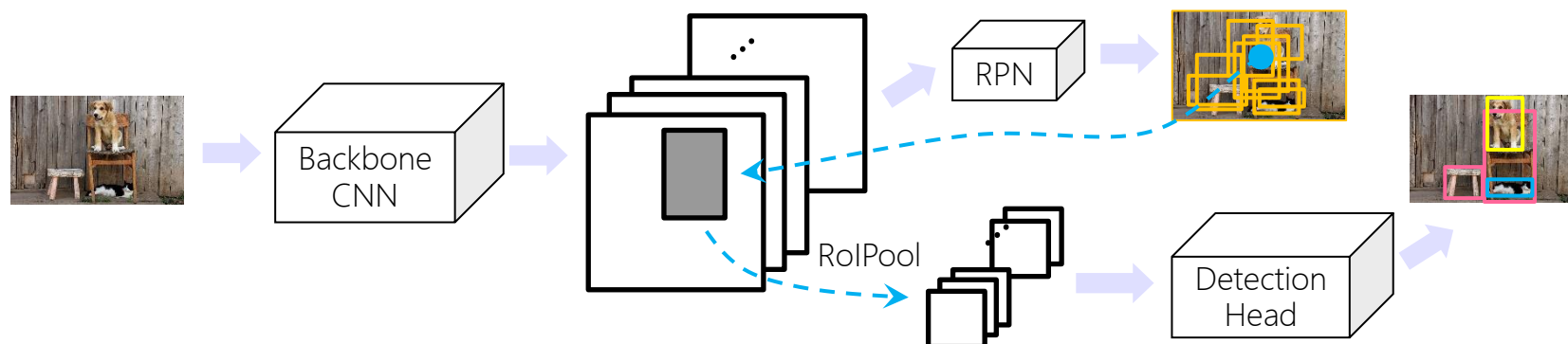


深度学习时代的目标检测

■ 主流的检测框架：典型方法

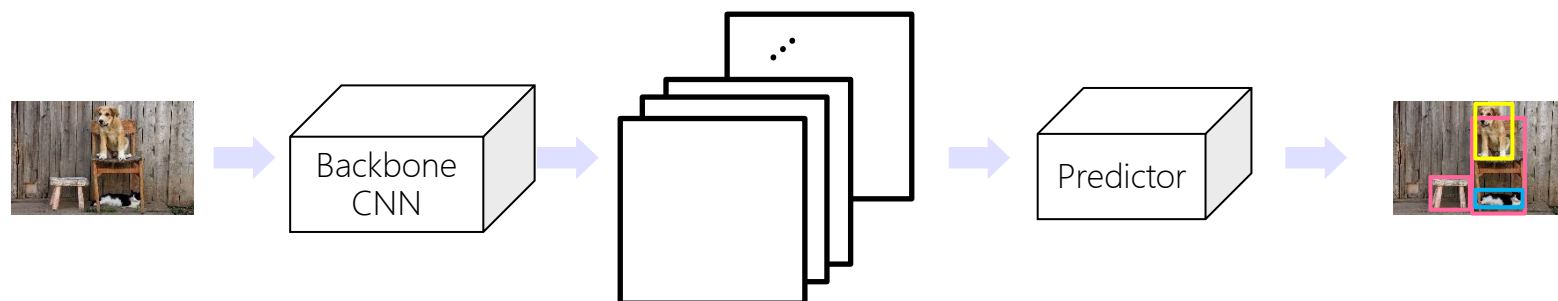
R-CNN → SPPNet, Fast R-CNN → Faster R-CNN → Mask R-CNN...

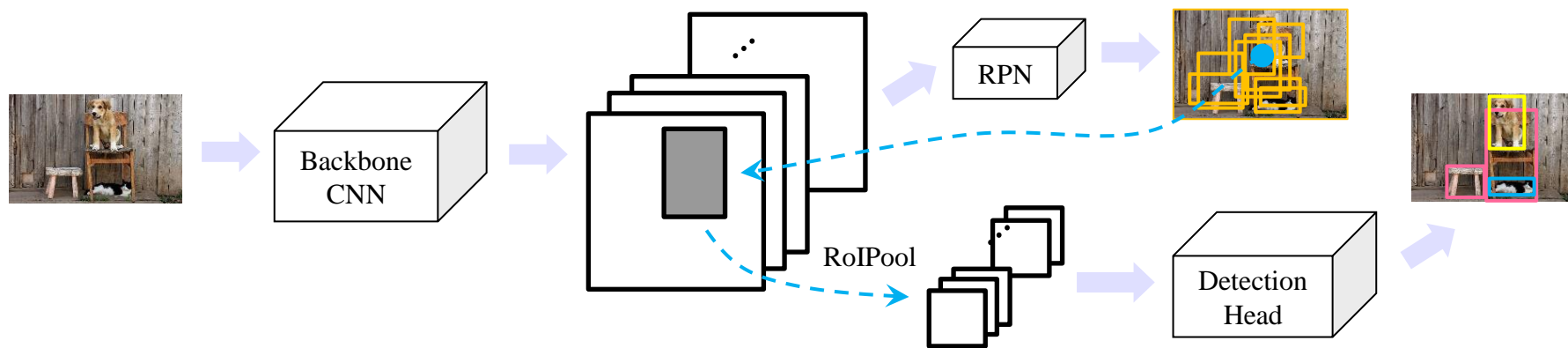
两阶段



Overfeat, DenseBox → YOLO & SSD → CornerNet, ExtremeNet, FSAF, FCOS...

单阶段



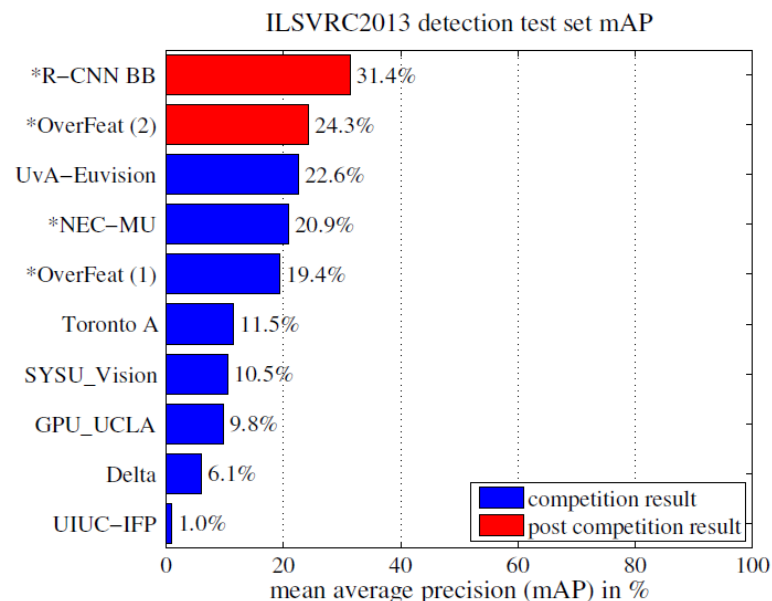
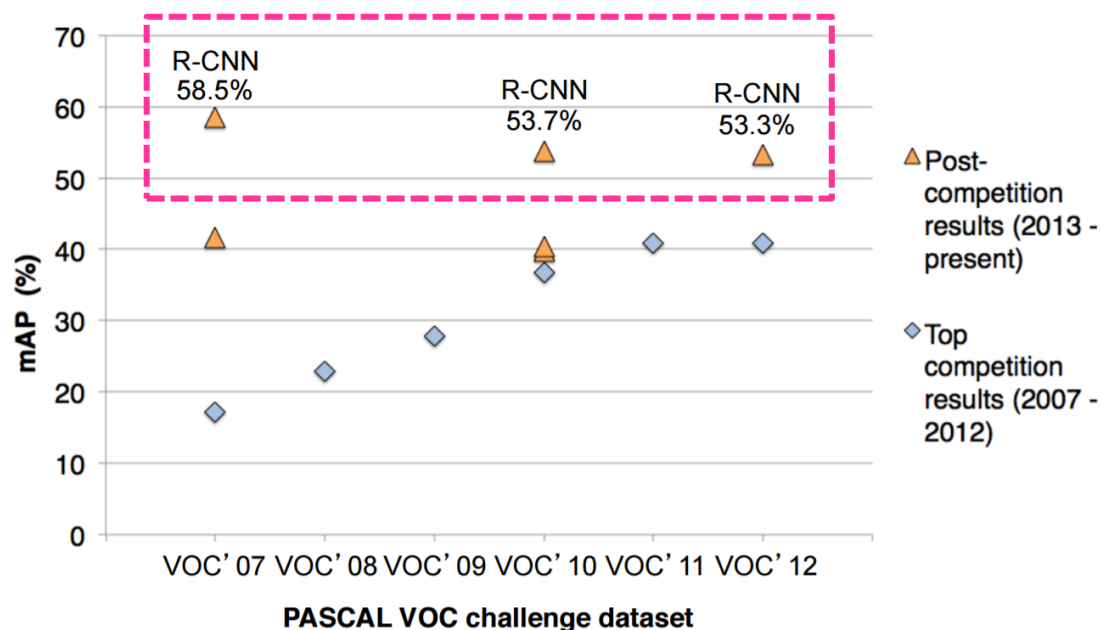


R-CNN → SPPNet, Fast R-CNN → Faster R-CNN → Mask R-CNN...

两阶段检测器

■ 两阶段检测器：R-CNN Regions with CNN features

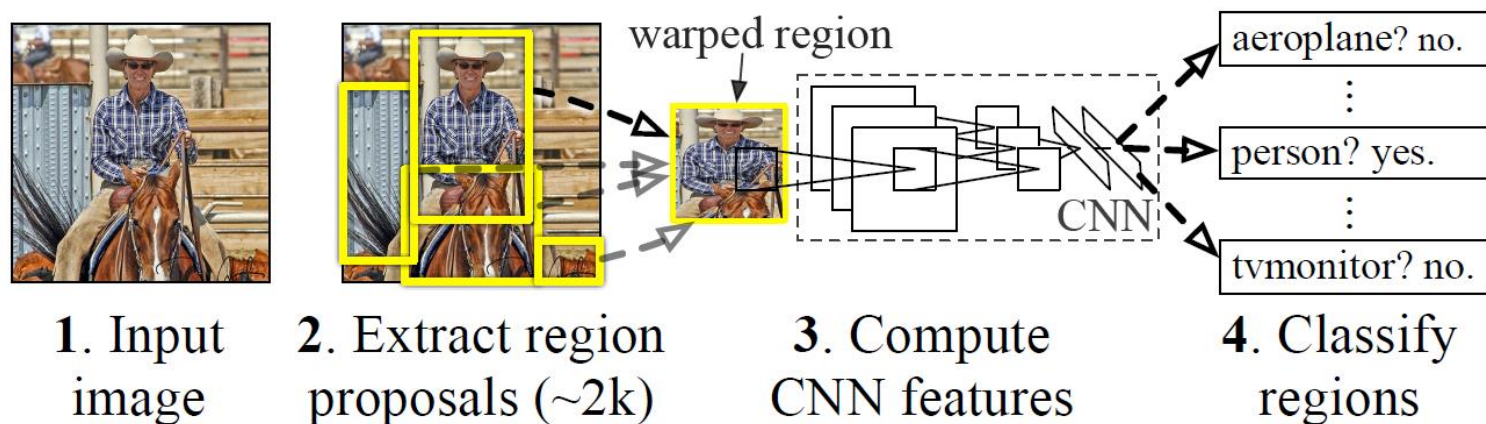
- 最早将CNN用于目标检测的工作之一
- 大幅提升了目标检测的精度



R-CNN

■ 两阶段检测器：R-CNN

□ 基本框架：以CNN作为特征提取器



□ 高度非线性的深度网络具有很强的建模能力，但是：

- 计算复杂度高 → 仅生成少量Region Proposal
- 训练需要大量标注数据 → 有监督预训练 + 领域特定微调

■ 两阶段检测器：R-CNN

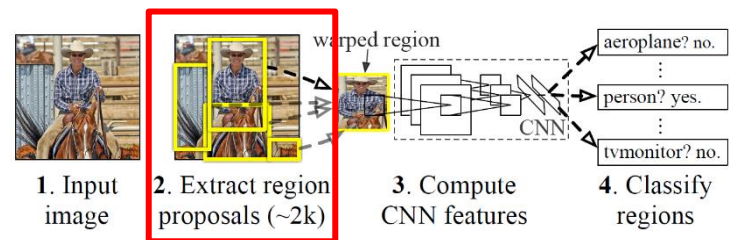
□ 第一步：生成少量Region Proposal

■ 专门的候选窗口生成方法： *Selective Search*

- 无监督：没有训练过程，不需要带标注的数据
- 数据驱动：根据图像特征生成候选窗口
- 基于图像分割任务

□ *Selective Search*

■ 对图像进行分割，每个分割区域生成一个对应的外接矩形框



图像来源: http://vision.stanford.edu/teaching/cs231b_spring1415/slides/ssearch_schuyler.pdf

■ 两阶段检测器：R-CNN

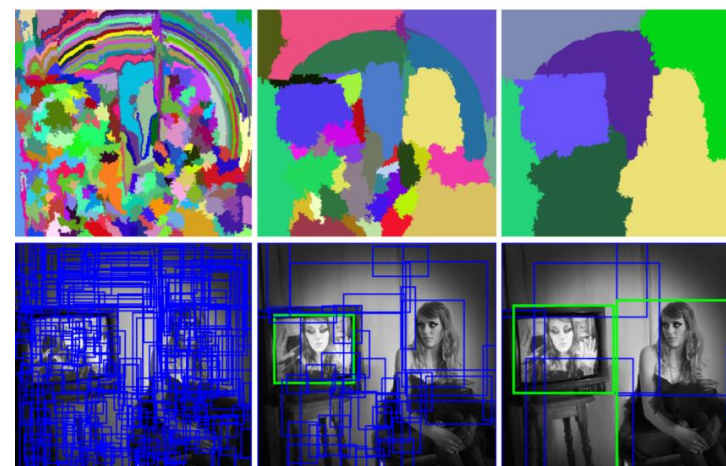
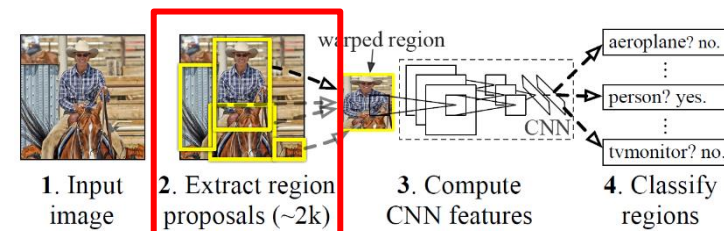
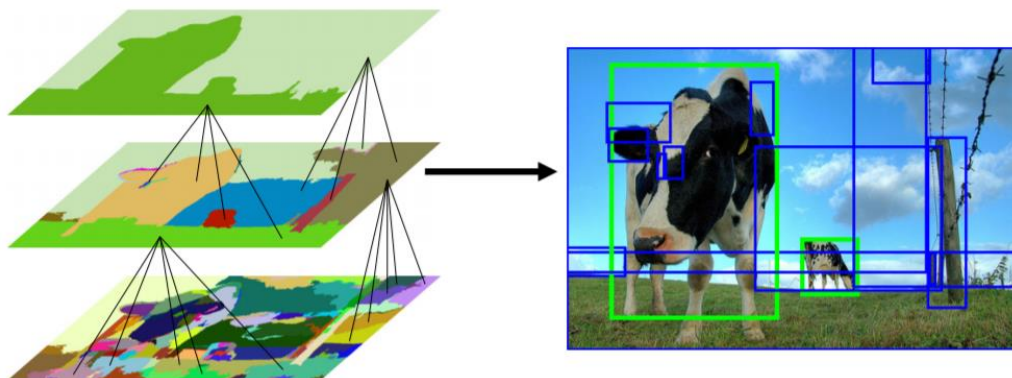
□ 第一步：生成少量Region Proposal

■ 专门的候选窗口生成方法：Selective Search

- 无监督：没有训练过程，不需要带标注的数据
- 数据驱动：根据图像特征生成候选窗口
- 基于图像分割任务

□ Selective Search

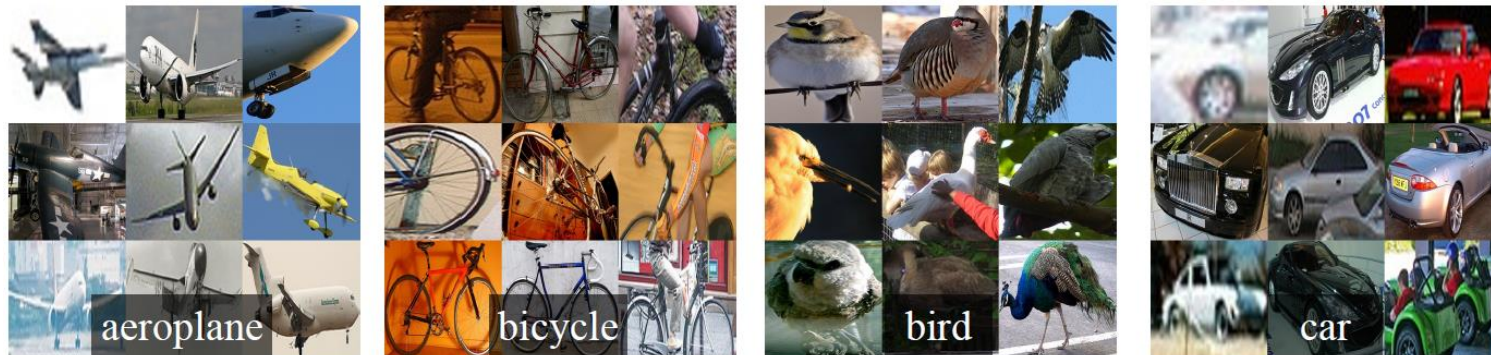
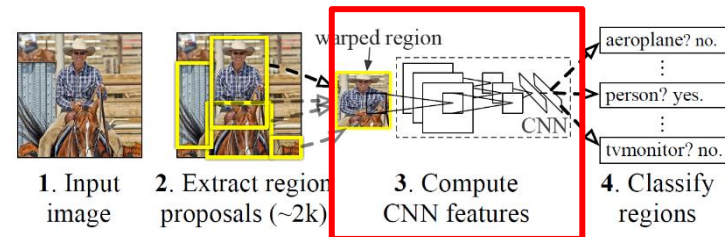
■ 基于相似度进行层次化地区域合并



■ 两阶段检测器：R-CNN

□ 第二步：用CNN提取Region Proposal特征

- 将不同大小的Region Proposal缩放到相同大小： 227×227
 - 进行些许扩大以包含少量上下文信息

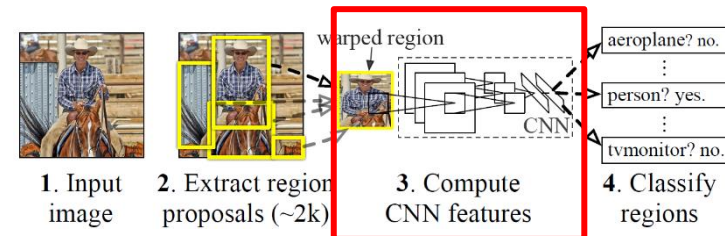


- 将所有窗口送入AlexNet提取特征
- 以最后一个全连接层的输出作为特征表示：4096维

R-CNN

■ 两阶段检测器：R-CNN

□ 第二步：用CNN提取Region Proposal特征



■ 有监督预训练 *Pretraining*

- 图像分类任务：ImageNet, 1000类, 仅有图像类别标注
- 数据量：120万张图像

■ 针对目标任务进行微调 *Fine-tuning*

- 目标检测任务：Pascal VOC, 20类, 有物体边框标注
- 数据量：仅有数千或上万张图像

■ 微调是可选步骤，但其有助于进一步提升检测精度

- 用大量数据预训练的模型，其提取的特征已经有较好的迁移能力
- 另：关于预训练

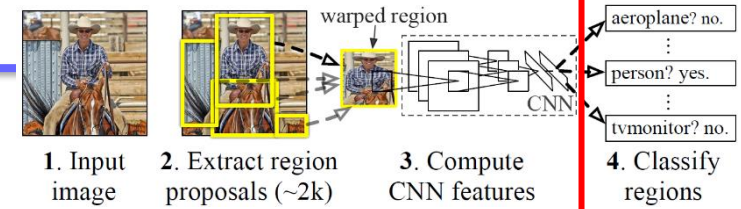
Rethinking ImageNet Pre-training

Kaiming He Ross Girshick Piotr Dollár

Facebook AI Research (FAIR)

R-CNN

■ 两阶段检测器：R-CNN



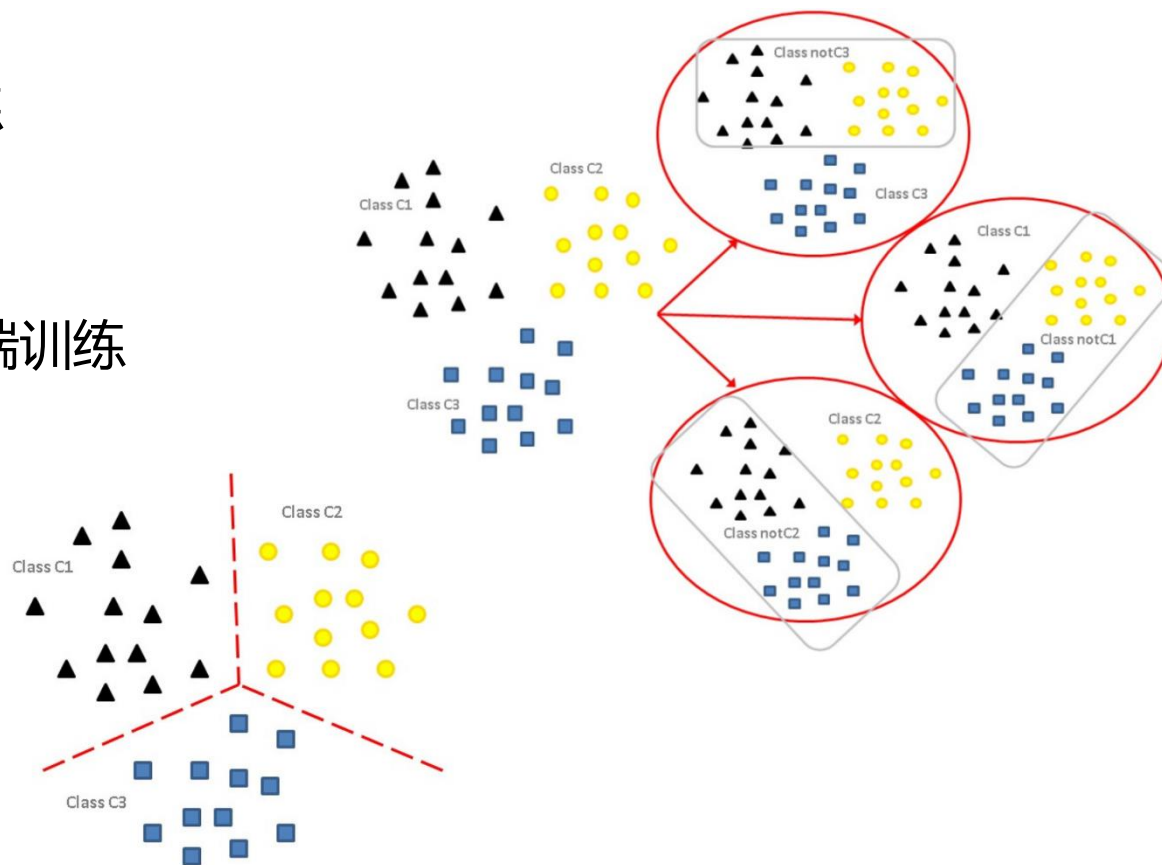
□ 第三步：对Region Proposal进行分类

■ 线性SVM分类器

- 针对每个类别单独训练
- 两类分类：one-vs-all

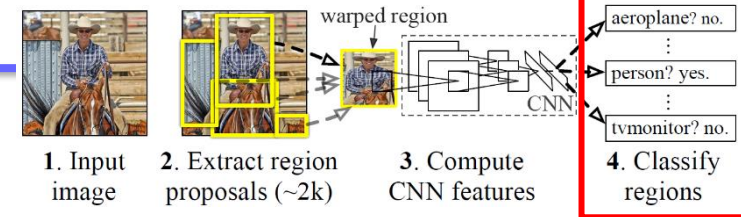
■ Softmax

- 和整个CNN一起端到端训练
- 所有类别一起训练
- 多类分类



R-CNN

■ 两阶段检测器：R-CNN



□ 第三步：对Region Proposal进行分类+边框校准

■ 分类：线性SVM分类器，Softmax

■ 边框校准

□ 让检测框的位置更加准确，同时更加紧致（包含更少的背景区域）

□ 线性回归模型

线性变换 \leftarrow Region Proposal (特征)

$$\hat{G}_x = P_w d_x(P) + P_x$$

$$\hat{G}_y = P_h d_y(P) + P_y$$

$$\hat{G}_w = P_w \exp(d_w(P))$$

$$\hat{G}_h = P_h \exp(d_h(P)).$$

Training image regions

Cached region features

Regression targets
(dx, dy, dw, dh)
Normalized coordinates



(0, 0, 0, 0)
Proposal is good



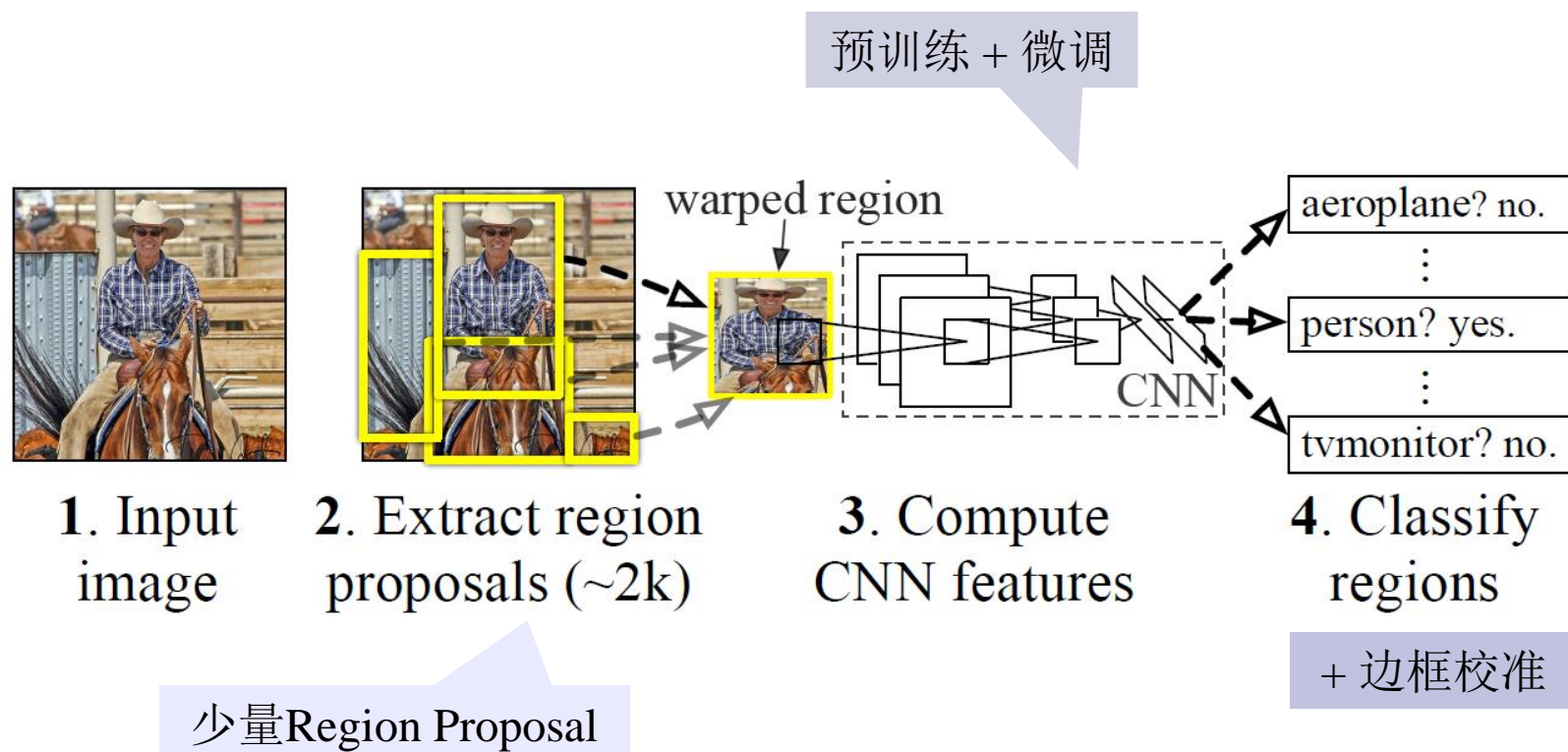
(.25, 0, 0, 0)
Proposal too far to left



(0, 0, -0.125, 0)
Proposal too wide

R-CNN

■ 两阶段检测器：R-CNN

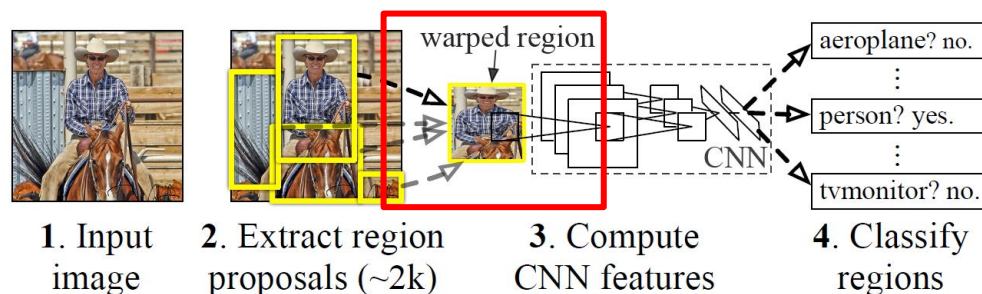


R-CNN → SPPNet

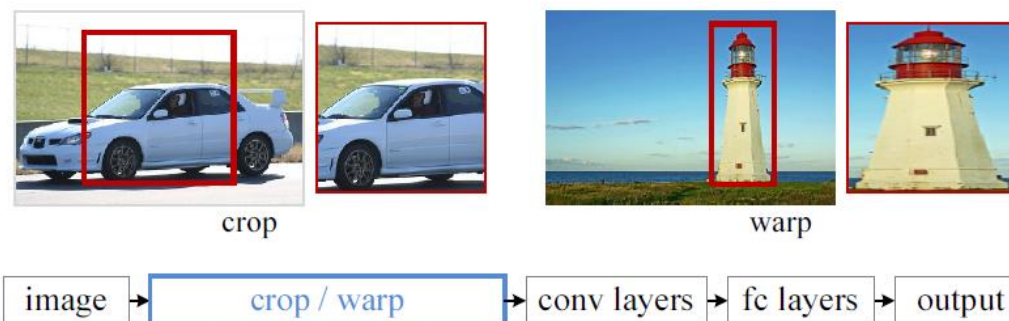
■ 两阶段检测器：SPPNet Spatial Pyramid Pooling

□ R-CNN要求输入图像的尺寸相同

- 不同尺度和长宽比的区域被变换到相同大小(否则没法接后面的分类)



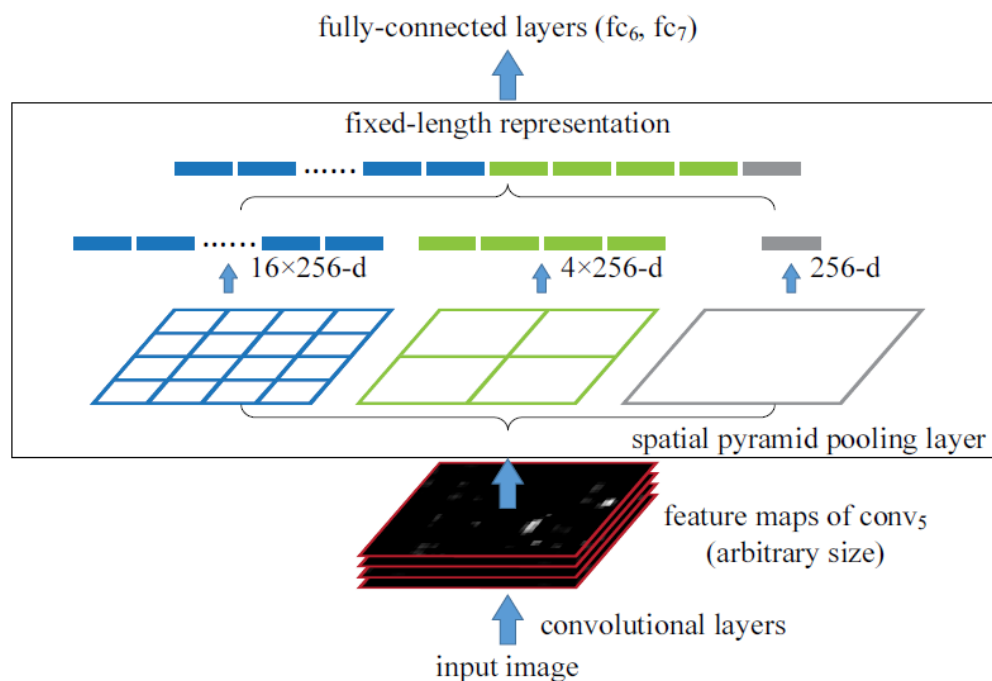
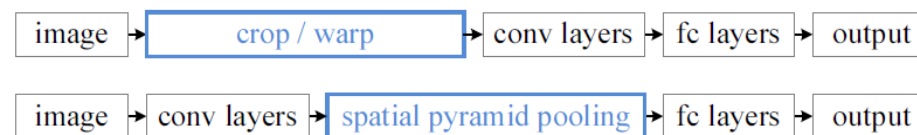
- 裁剪会使信息丢失（或引入过多背景），缩放会使物体变形



R-CNN → SPPNet

■ 两阶段检测器：SPPNet

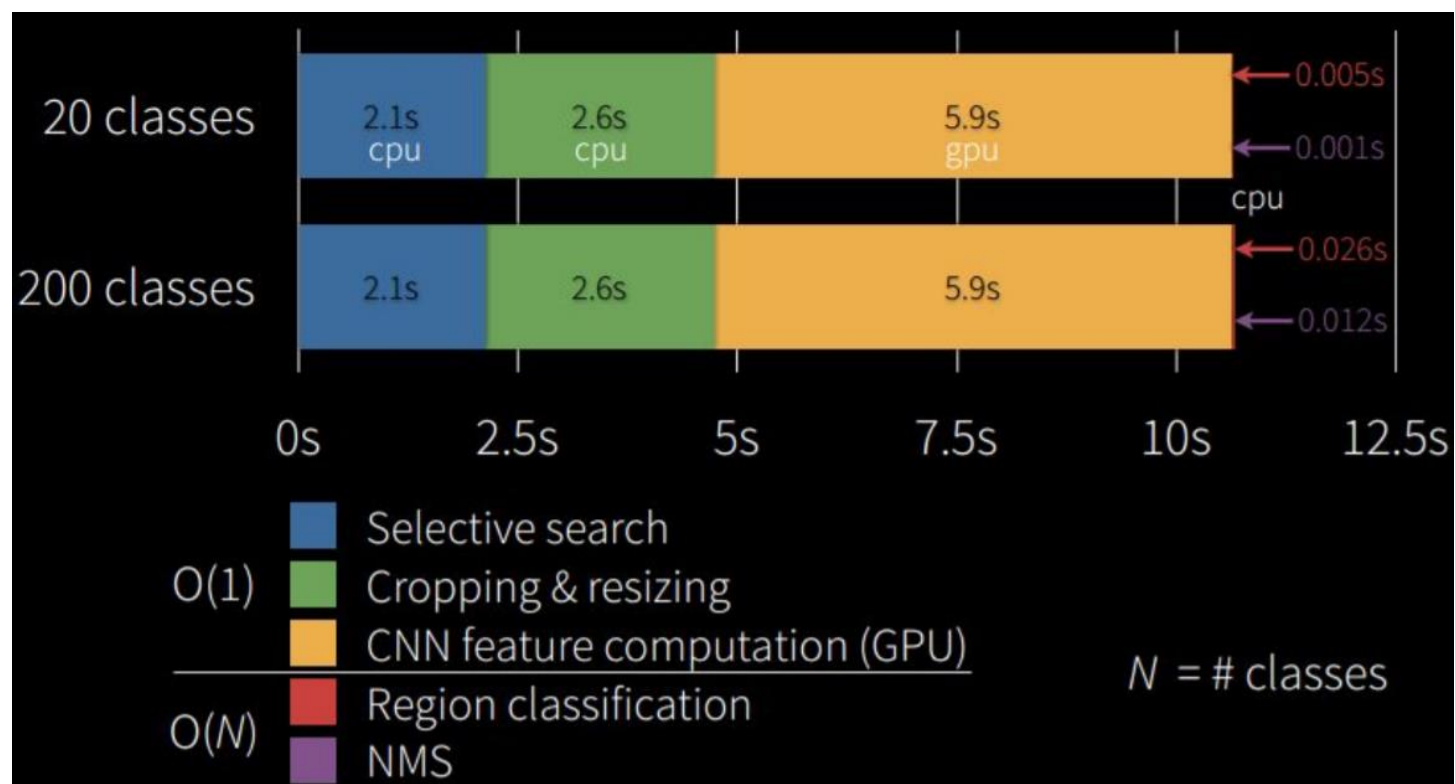
- 卷积：允许任意大小的图像输入网络
- SPP：将不同大小的输入归一化到相同大小
- 全连接：接受固定维度的输入



R-CNN → SPPNet

■ 两阶段检测器：SPPNet

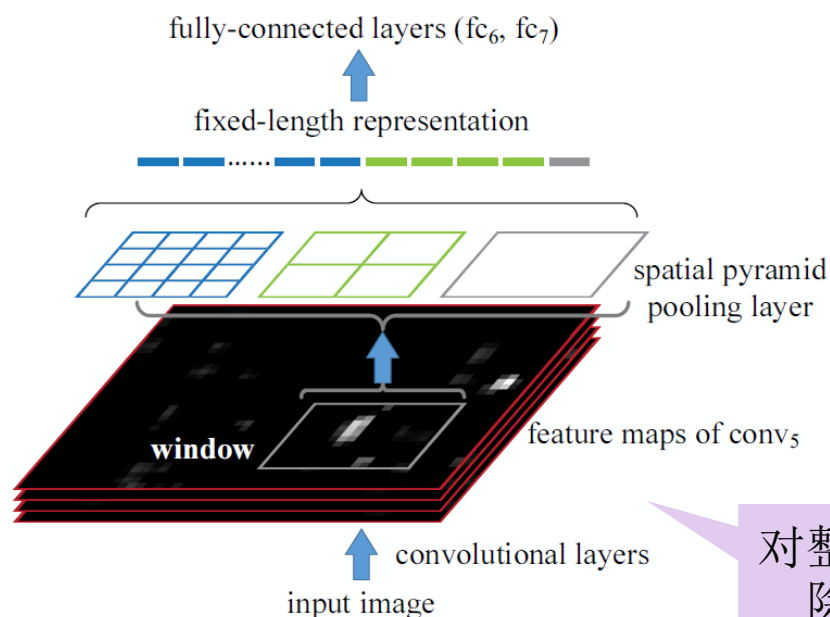
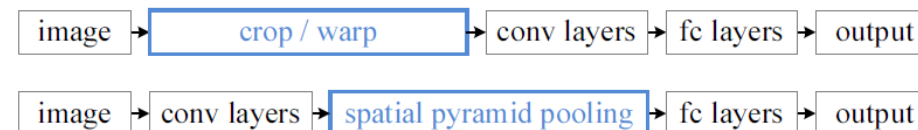
□ 回顾：R-CNN（检测速度）



R-CNN → SPPNet

■ 两阶段检测器：SPPNet

- 卷积：允许任意大小的图像输入网络
- SPP：将不同大小的输入归一化到相同大小
- 全连接：接受固定维度的输入



对整张图计算卷积特征，去除各个区域的重复计算

	SPP (1-sc) (ZF-5)	SPP (5-sc) (ZF-5)	R-CNN (ZF-5)
ftfc ₇	54.5	<u>55.2</u>	55.1
ftfc ₇ bb	58.0	59.2	59.2
conv time (GPU)	0.053s	0.293s	14.37s
fc time (GPU)	0.089s	0.089s	0.089s
total time (GPU)	0.142s	0.382s	14.46s
speedup (vs. RCNN)	102×	38×	-

R-CNN → SPPNet → Fast R-CNN

■ 两阶段检测器：Fast R-CNN

□ R-CNN和SPPNet的训练都包含多个单独的步骤

■ (1) 对网络进行微调

- R-CNN对整个CNN进行微调

- SPP-net只对SPP之后的（全连接）层进行微调

■ (2) 训练SVM & (3) 训练边框回归模型

- 时间长：需要用CNN提取所有训练样本的特征

- 占用存储空间大：所有样本的特征需要存储到磁盘上

□ 检测速度慢，尤其是R-CNN

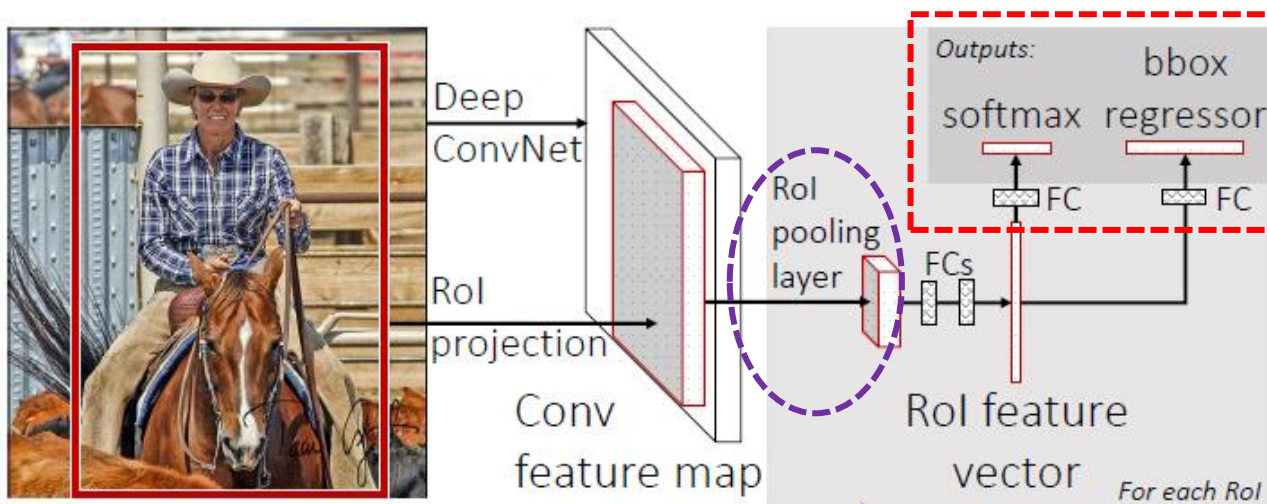
- R-CNN + VGG16：检测一张图需要47s

R-CNN → SPPNet → Fast R-CNN

■ 两阶段检测器：Fast R-CNN

Region of Interest pooling

- 保留SPPNet的优势 ⇒ 简化SPP为单尺度：**RoI pooling**
- 引入**多任务学习**，将多个步骤整合到一个模型中



多任务学习

整张图像进行一次卷积层的计算

整个网络一起训练/微调

R-CNN → SPPNet → Fast R-CNN

- 两阶段检测器：Fast R-CNN
 - 改进边框校准：Smooth L_1 Loss

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise,} \end{cases}$$

- 全连接层加速：Truncated SVD

$$W \approx U \Sigma_t V^T$$

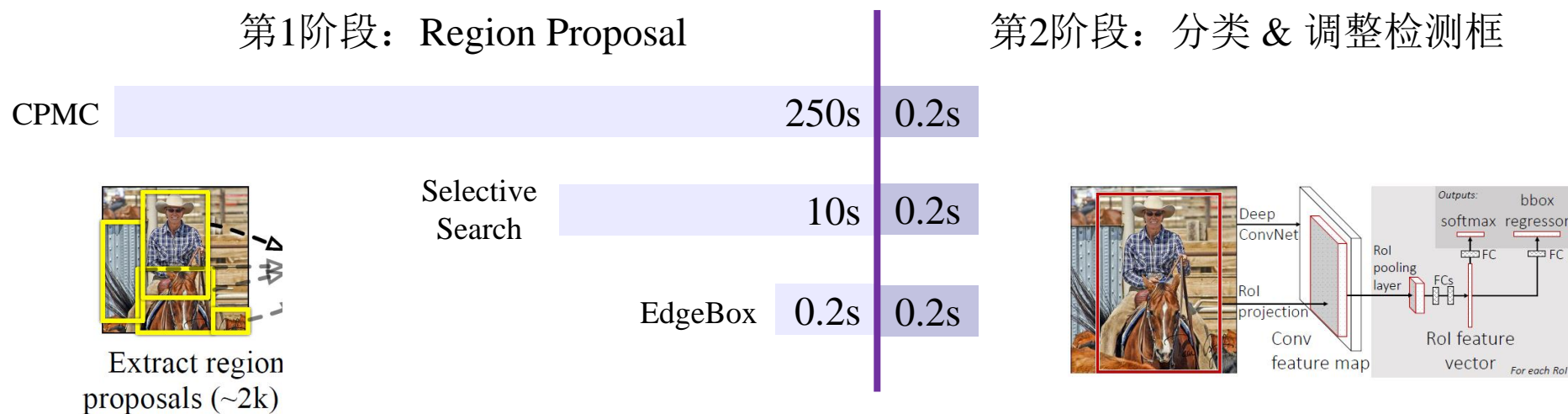
- 一个大全连接层 \Rightarrow 两个小全连接层
- 时间复杂度： $O(uv) \Rightarrow O(t(u+v))$

	Fast R-CNN			R-CNN			SPPnet $^{\dagger}L$
	S	M	L	S	M	L	
train time (h)	1.2	2.0	9.5	22	28	84	25
train speedup	18.3×	14.0×	8.8×	1×	1×	1×	3.4×
test rate (s/im)	0.10	0.15	0.32	9.8	12.1	47.0	2.3
▷ with SVD	0.06	0.08	0.22	-	-	-	-
test speedup	98×	80×	146×	1×	1×	1×	20×
▷ with SVD	169×	150×	213×	-	-	-	-
VOC07 mAP	57.1	59.2	66.9	58.5	60.2	66.0	63.1
▷ with SVD	56.5	58.7	66.6	-	-	-	-

Fast R-CNN → Faster R-CNN

■ 两阶段检测器：Faster R-CNN

□ 专门的Region Proposal模块是速度瓶颈



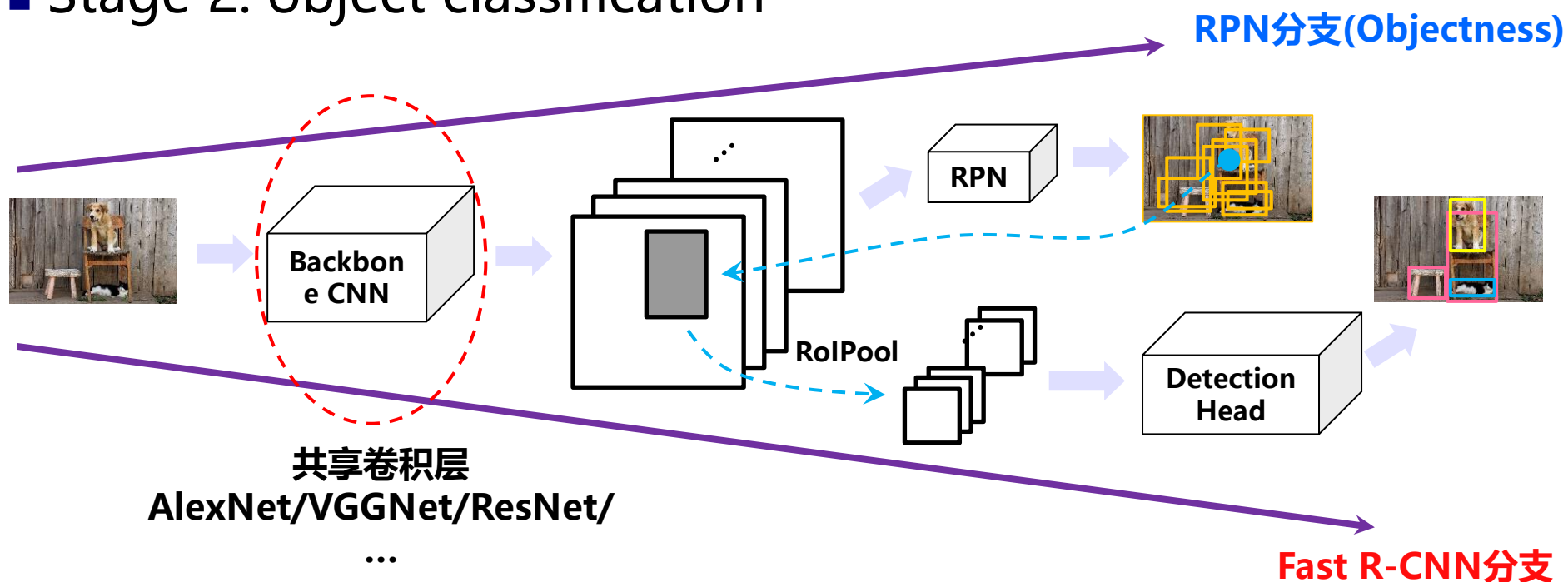
□ 改进：直接用CNN来生成Region Proposal，并且和第二阶段的CNN共享卷积层

Fast R-CNN → Faster R-CNN

■ 两阶段检测器：Faster R-CNN

□ 整体框架

- Stage 1: objectness
- Stage 2: object classification

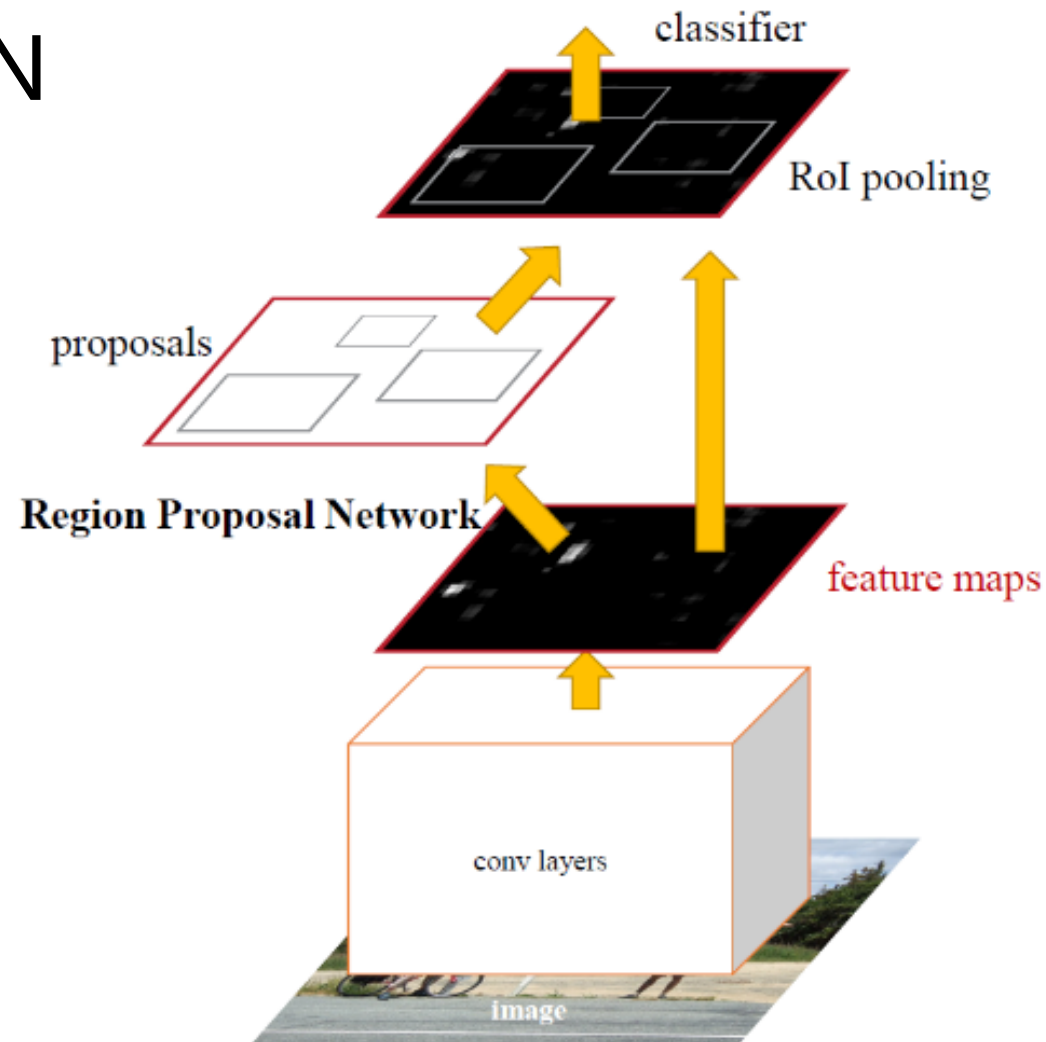


Fast R-CNN → Faster R-CNN

■ 两阶段检测器：Faster R-CNN

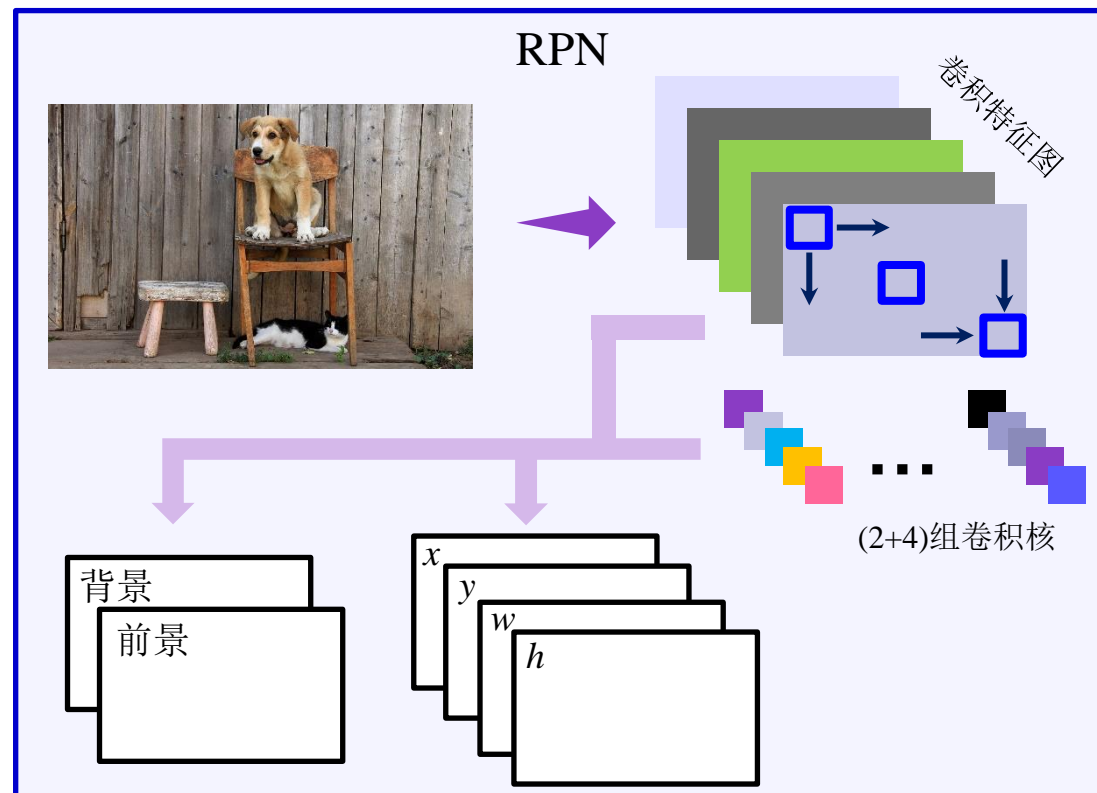
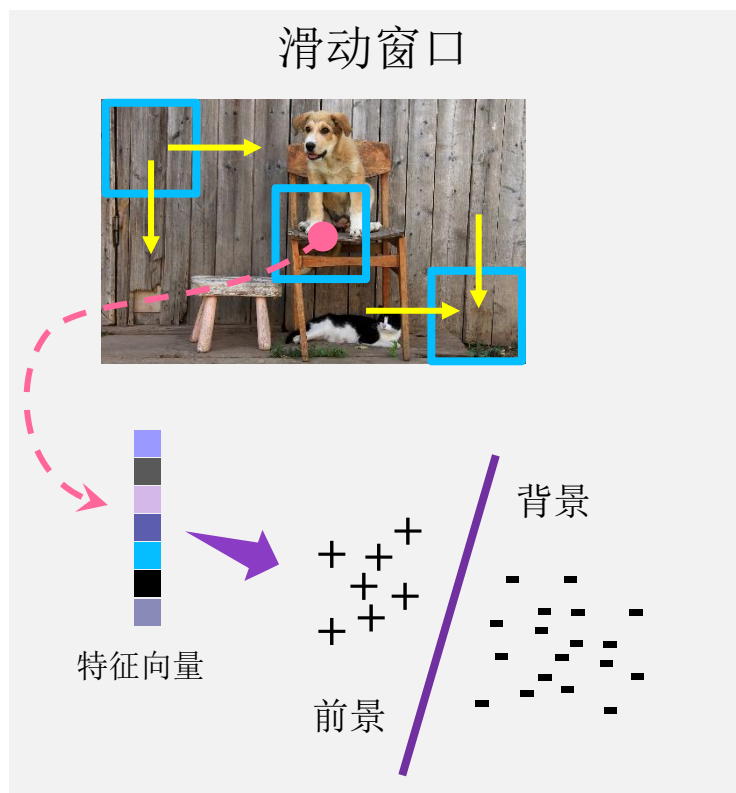
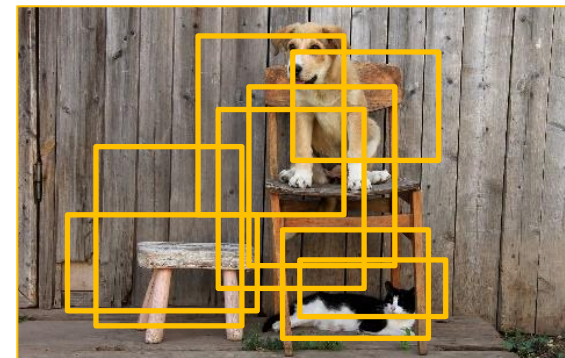
□ 整体框架

- Stage 1: objectness
- Stage 2: object classification



Fast R-CNN → Faster R-CNN

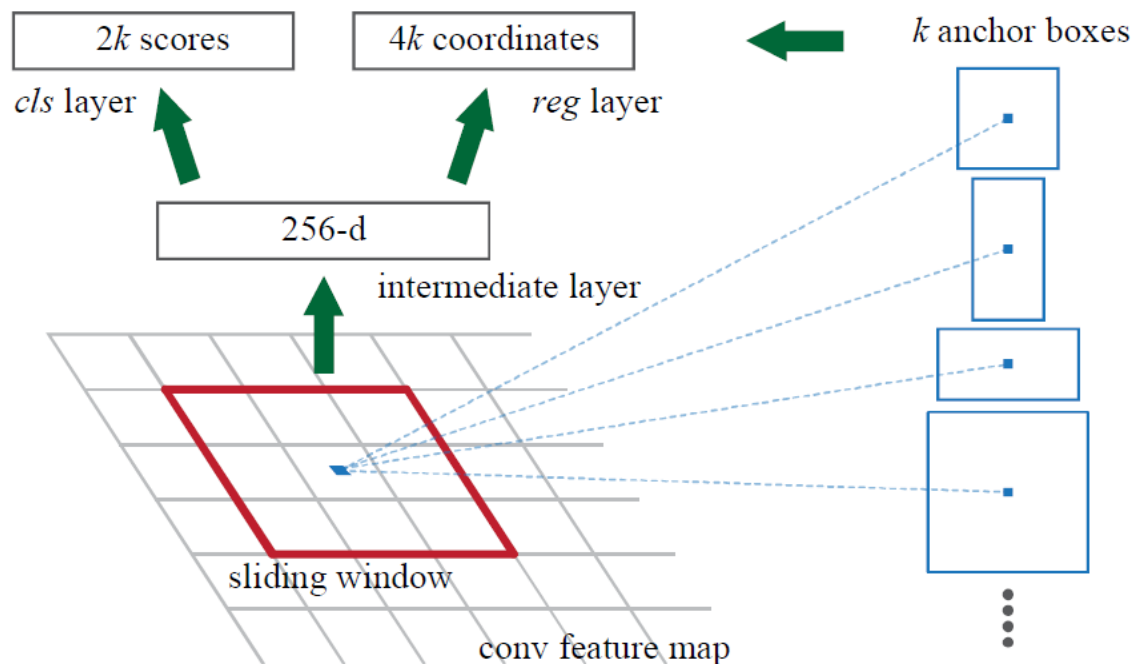
- 如何检测objectness?
 - Region proposal network
 - 本质：在特征图上滑动窗口



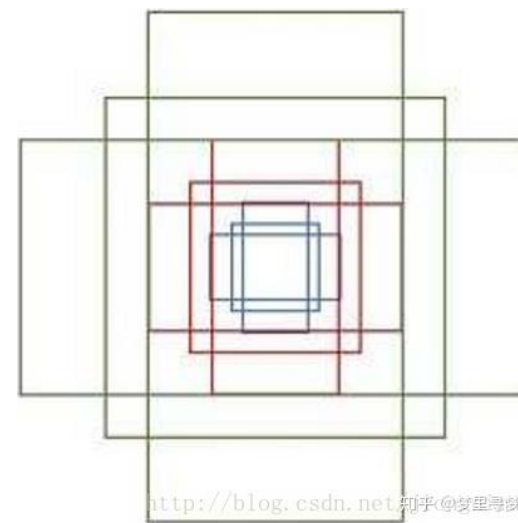
Fast R-CNN → Faster R-CNN

■ Region Proposal Network (RPN)

- 在**特征图**上**滑动 $n \times n$** 大小窗口，每个窗口的特征输入一个**全连接小网络**预测 **k 个**Anchors**是前景or背景($2k$)，同时修正每个Anchor坐标($4k$)**



Anchor: 一组不同大小
($128 \times 128, 256 \times 256, 512 \times 512$)
和长宽比(1:1, 1:2, 2:1)的窗口



Fast R-CNN → Faster R-CNN

■ Region Proposal Network (RPN)

- 在**特征图**上**滑动** $n \times n$ 大小窗口，每个窗口的特征输入一个**全连接小网络**预测 k 个**Anchors**是前景or背景($2k$)，同时修正每个Anchor坐标($4k$)

- $n=3$

- $k=9$

- 特征图大小: $40 \times 60 = 2400$

- 会产生约 $9 \times 2400 = 20000$ 个anchors (前景, 背景, x, y, w, h)

Fast R-CNN → Faster R-CNN

■ Region Proposal Network (RPN)

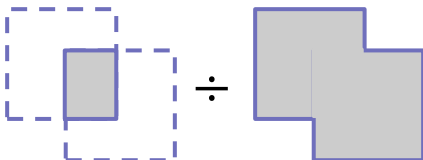
- 在**特征图**上**滑动** $n \times n$ 大小窗口，每个窗口的特征输入一个**全连接小网络**预测 k 个**Anchors**是前景or背景($2k$)，同时修正每个Anchor坐标($4k$)

■ 如何训练RPN网络？

- **前景Anchors**：与Ground Truth (GT) 的IoU >0.7
- **背景Anchors**：与Ground Truth (GT) 的IoU <0.3
- **不使用**IoU在 $[0.3, 0.7]$ 之间的Anchors
- 在训练anchor属于前景与背景的网络时，在一张图中，随机抽取了128个前景anchor，128个背景anchor

$$\text{IoU} = \frac{\text{Intersection}}{\text{Union}}$$

(交并比 Intersection-over-Union)



Fast R-CNN → Faster R-CNN

■ 两阶段检测器：Faster R-CNN

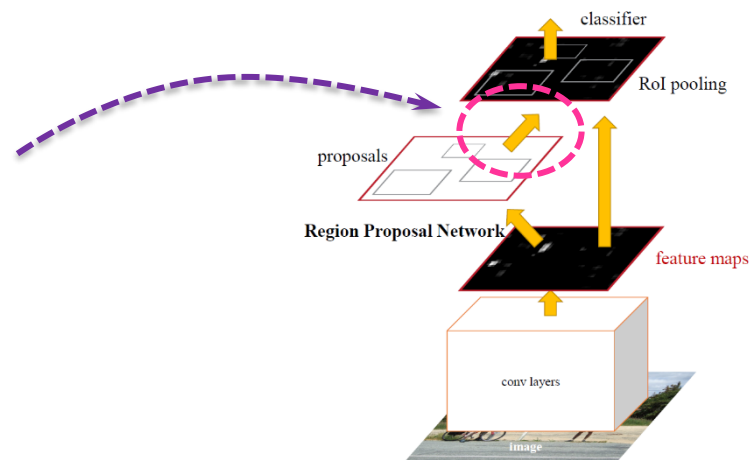
□ 模型训练

■ 交替式4步法训练

- 基于预训练模型训练RPN
- 基于预训练模型，以及上一步得到的RPN，训练Fast R-CNN
- 固定共享的卷积层，训练RPN
- 固定共享的卷积层，基于上一步得到的RPN，训练Fast R-CNN

■ 端到端训练

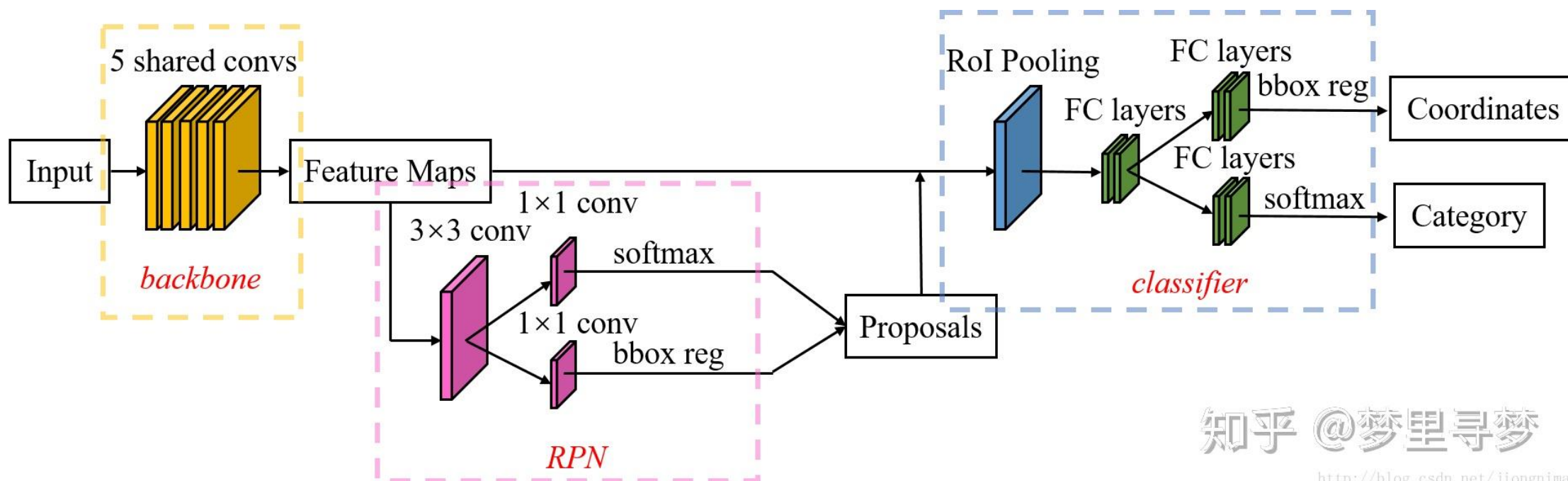
- 同时学习RPN和Fast R-CNN
- Fast R-CNN的梯度不向RPN回传



Fast R-CNN → Faster R-CNN

■ 两阶段检测器：Faster R-CNN

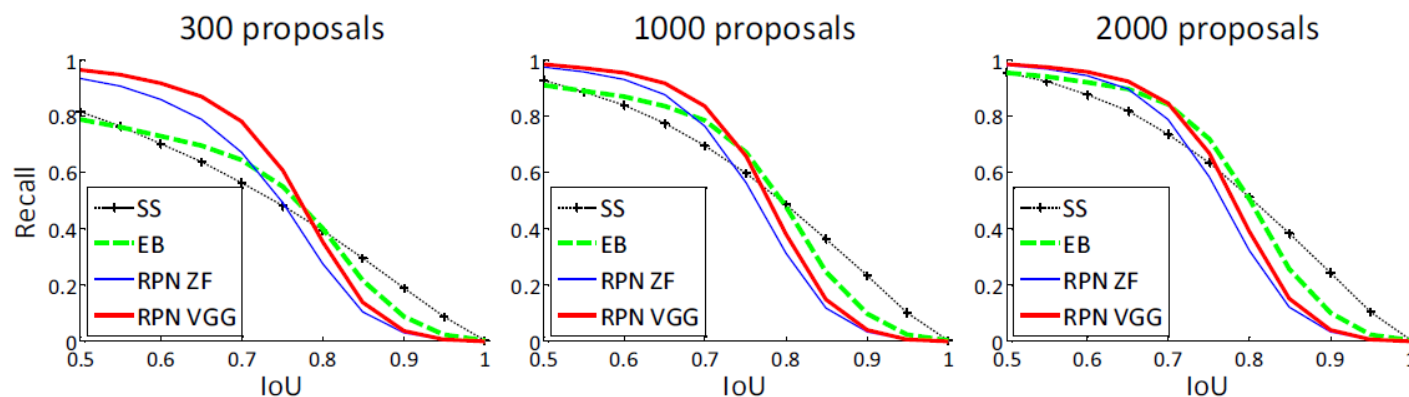
- Stage 1: objectness (RPN)
- Stage 2: object classification (Fast R-CNN)



Fast R-CNN → Faster R-CNN

■ 两阶段检测器：Faster R-CNN

□ RPN的召回率



□ 卷积层：共享 vs 不共享

method	# proposals	data	mAP (%)
SS	2000	07	66.9 [†]
SS	2000	07+12	70.0
RPN+VGG, unshared	300	07	68.5
RPN+VGG, shared	300	07	69.9
RPN+VGG, shared	300	07+12	73.2
RPN+VGG, shared	300	COCO+07+12	78.8

Fast R-CNN → Faster R-CNN

■ 两阶段检测器：不同R-CNN检测器的比较

□ 化零为整：多任务学习，参数/计算共享

	Region Proposal	提取特征	分类	边框校准
R-CNN, SPPNet	Selective Search	CNN	SVM	线性模型
Fast R-CNN	Selective Search	CNN		
Faster R-CNN	CNN			

□ 由慢变快：SPP, RoI pooling, Truncated SVD

	Region Proposal	提取特征	分类	边框校准
R-CNN	$10^2 \sim 10^4$ ms	$10^3 \sim 10^4$ ms	< 10ms	
SPPNet		$10^2 \sim 10^3$ ms		
Fast R-CNN		10^2 ms		
Faster R-CNN	10^2 ms			

Faster R-CNN → Mask R-CNN

■ 两阶段检测器：Mask R-CNN

□ 新的任务：实例分割

- 对于检测到的每个物体（实例），精确地标记出其每个像素

目标定位 *Localization*



CAT

目标检测



DOG, DOG, CAT

实例分割

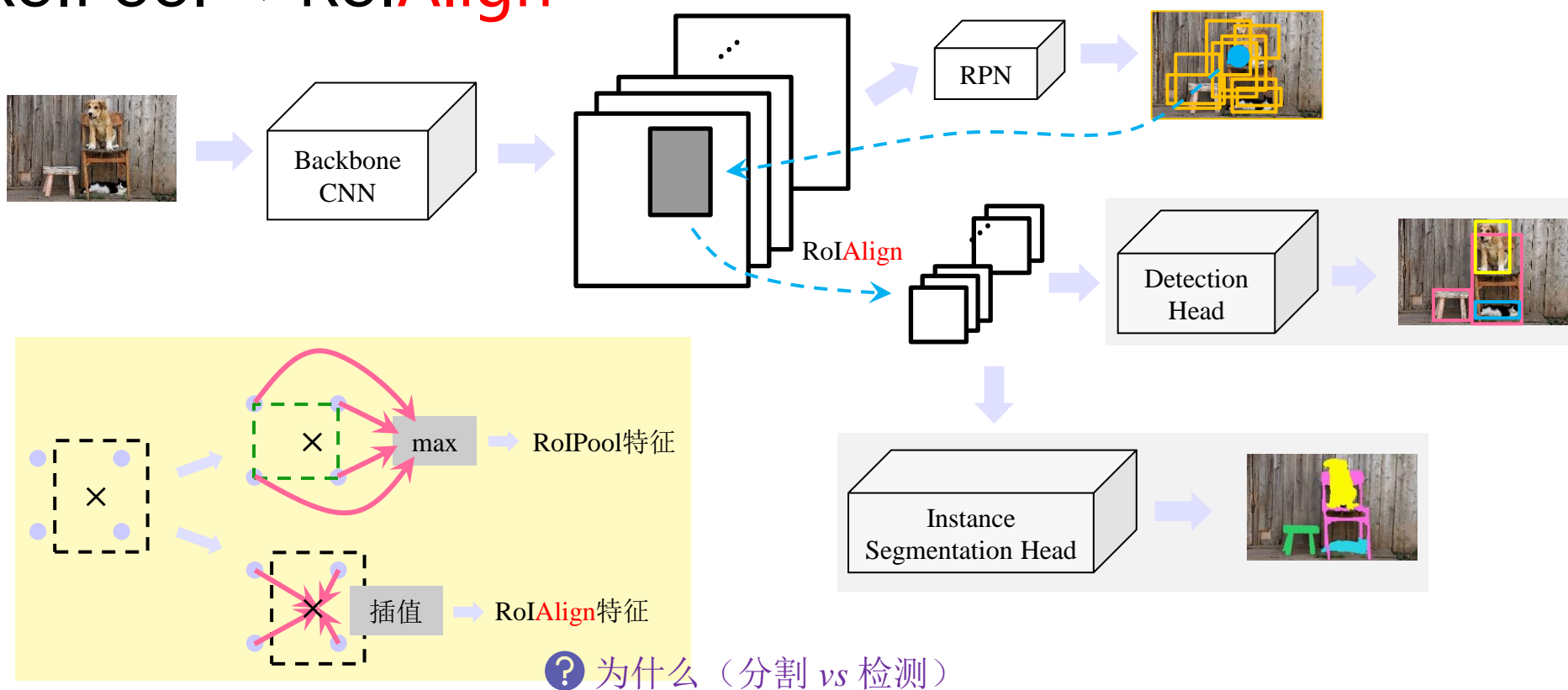


DOG, DOG, CAT

Faster R-CNN → Mask R-CNN

■ 两阶段检测器：Mask R-CNN

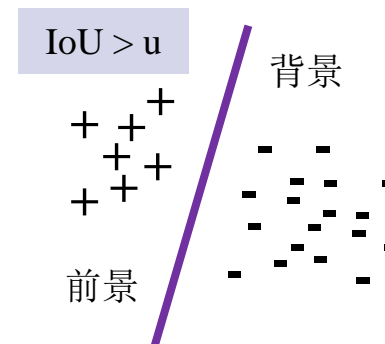
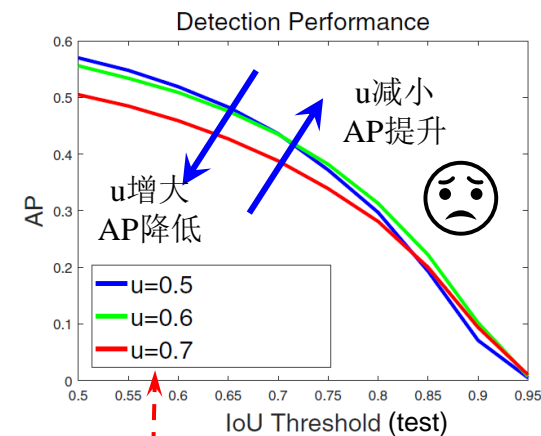
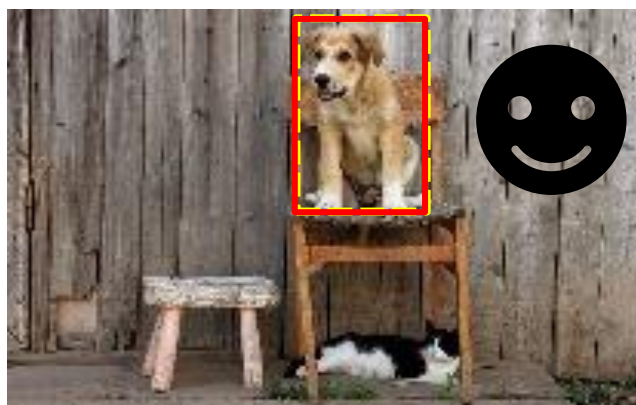
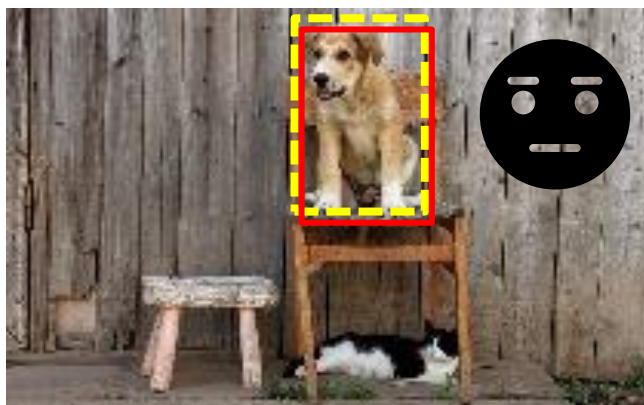
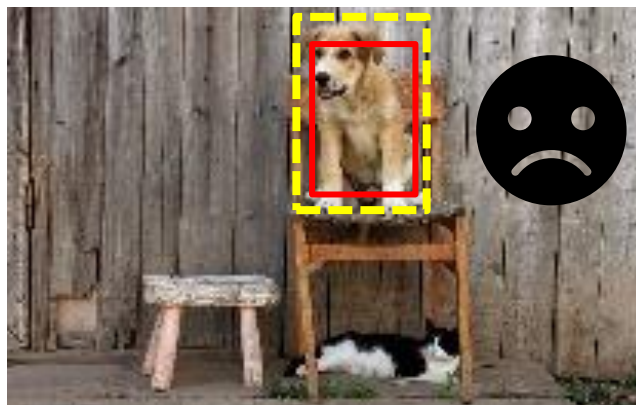
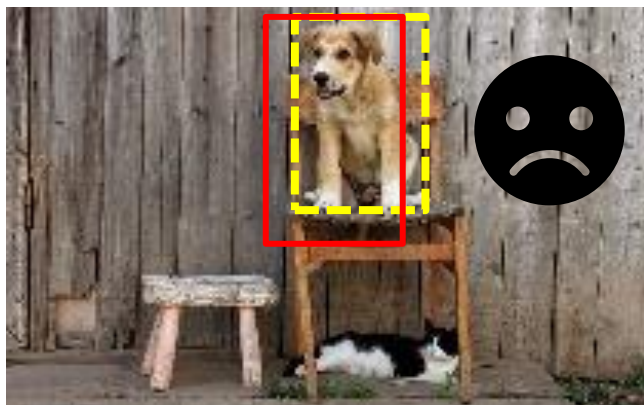
- 在Faster R-CNN中增加**实例分割**模块
- RoIPool → RoI**Align**



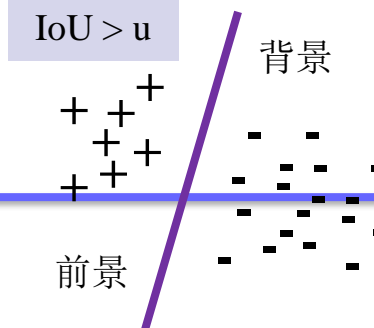
Cascade R-CNN

■ 两阶段检测器：Cascade R-CNN

□ 检测框的位置准确率：和标注框交并比越高越好

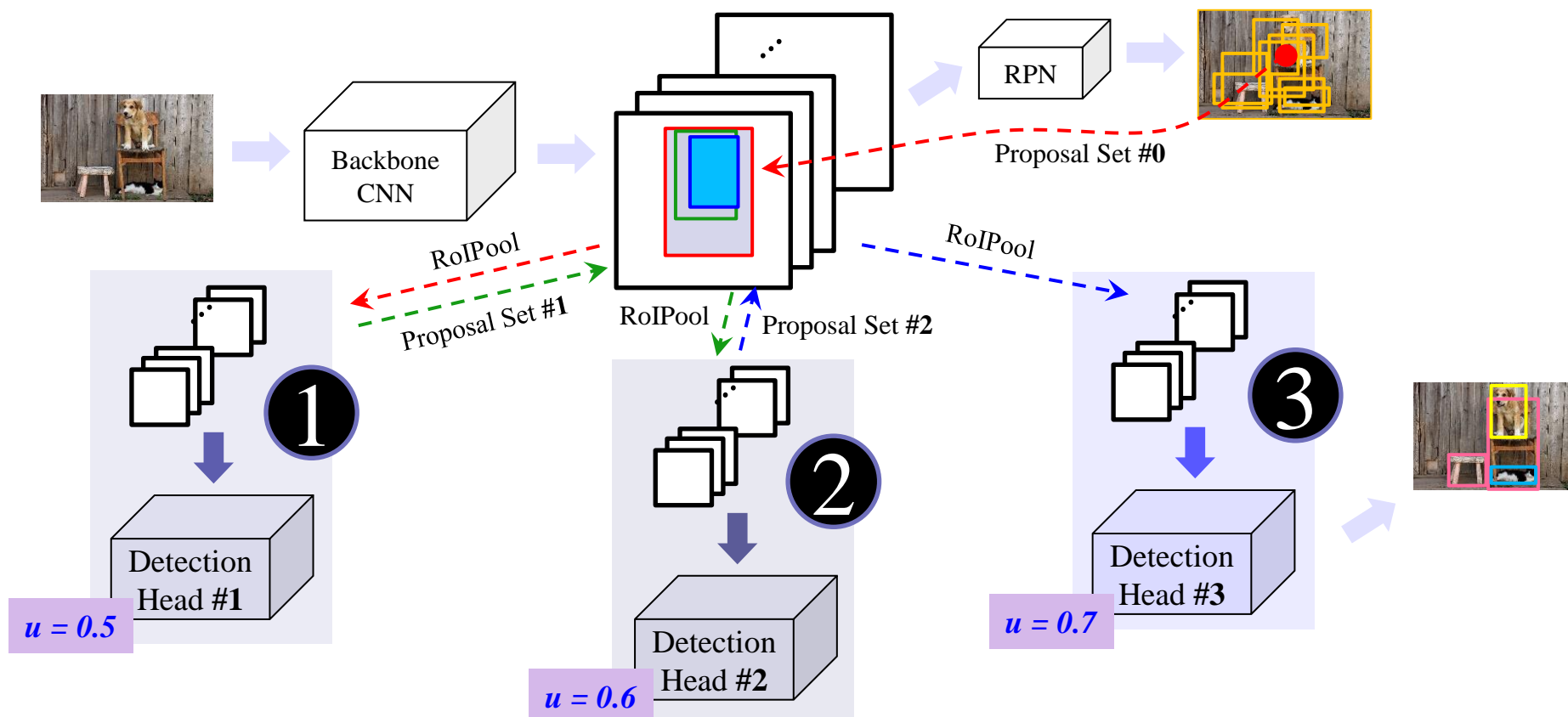


Cascade R-CNN



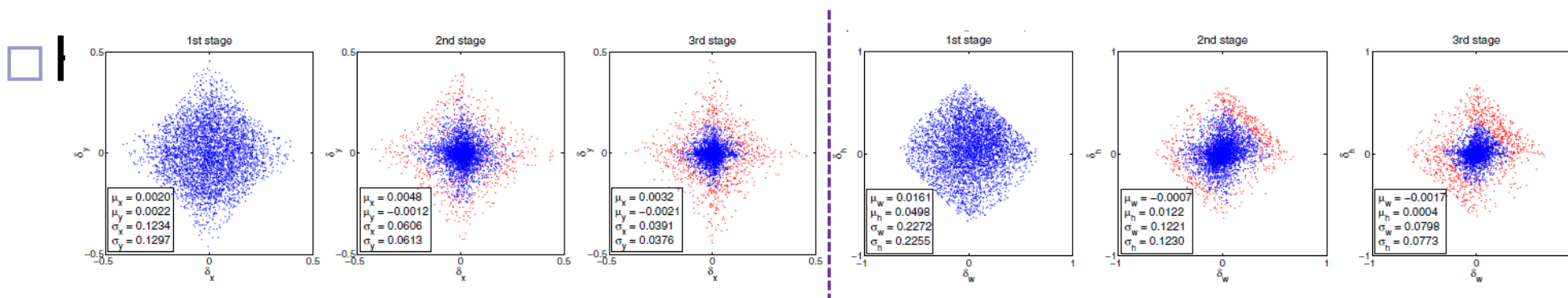
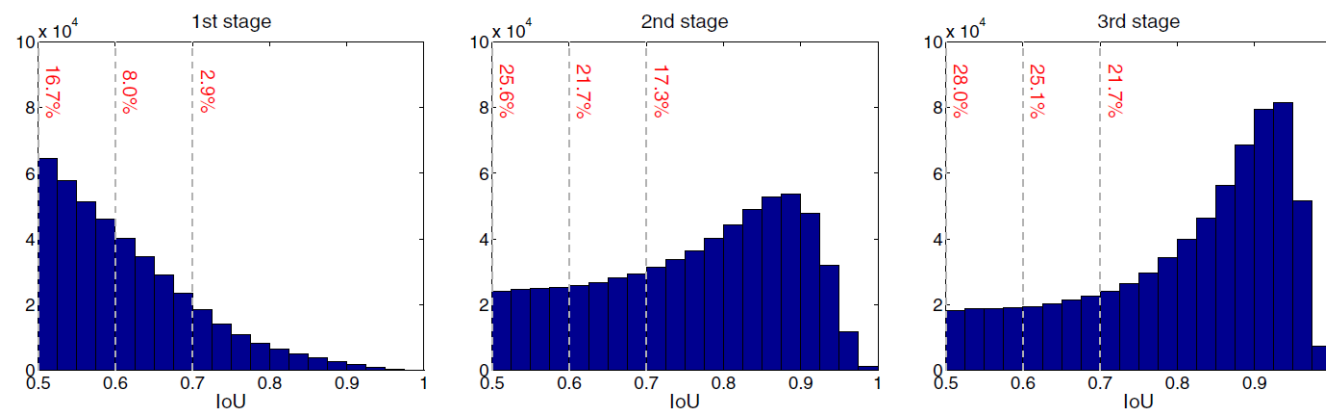
■ 两阶段检测器：Cascade R-CNN

□ 级联多个Detection Head：逐步调整检测框，慢慢升高IoU



Cascade R-CNN

- 两阶段检测器：Cascade R-CNN
 - Head #1 → #3: 高IoU的框占比逐步增大



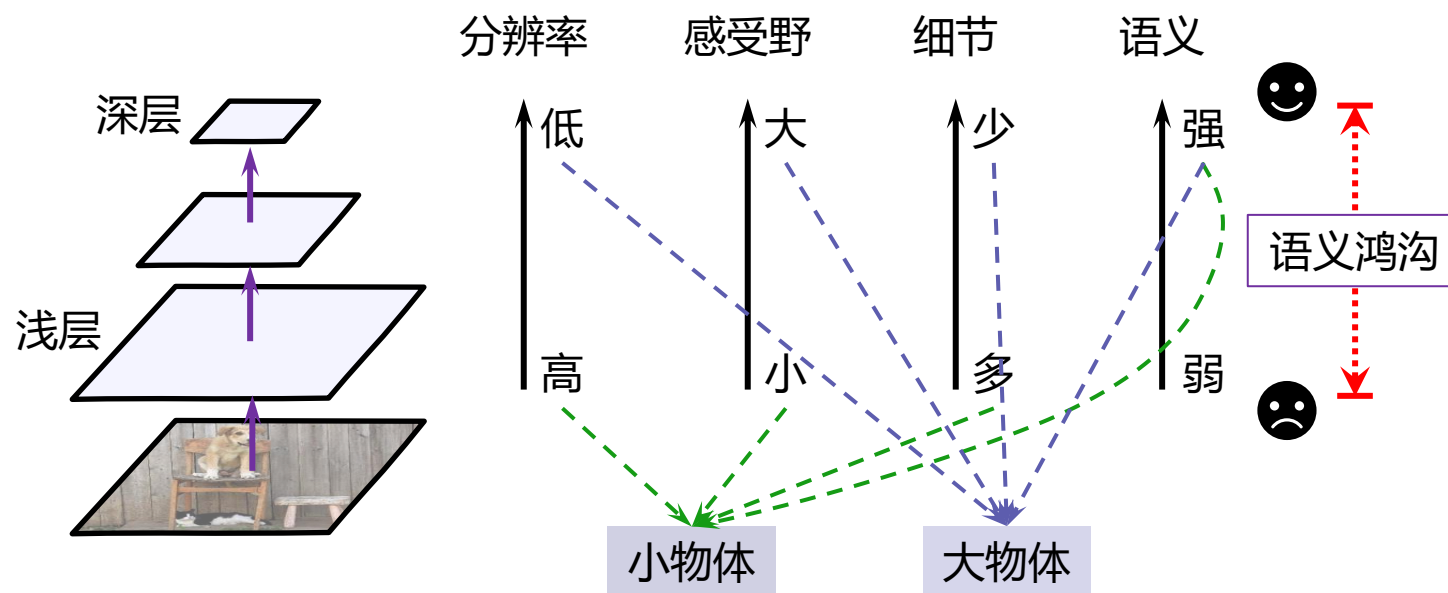
特征金字塔网络Feature Pyramid Network

■ 两阶段检测器：FPN *Feature Pyramid Network*

□ 物体的尺度变化具有很大的范围

- 在同一层上使用多尺度Anchor Box没有考虑尺度对特征的影响
- **图像金字塔**会带来很大的计算代价

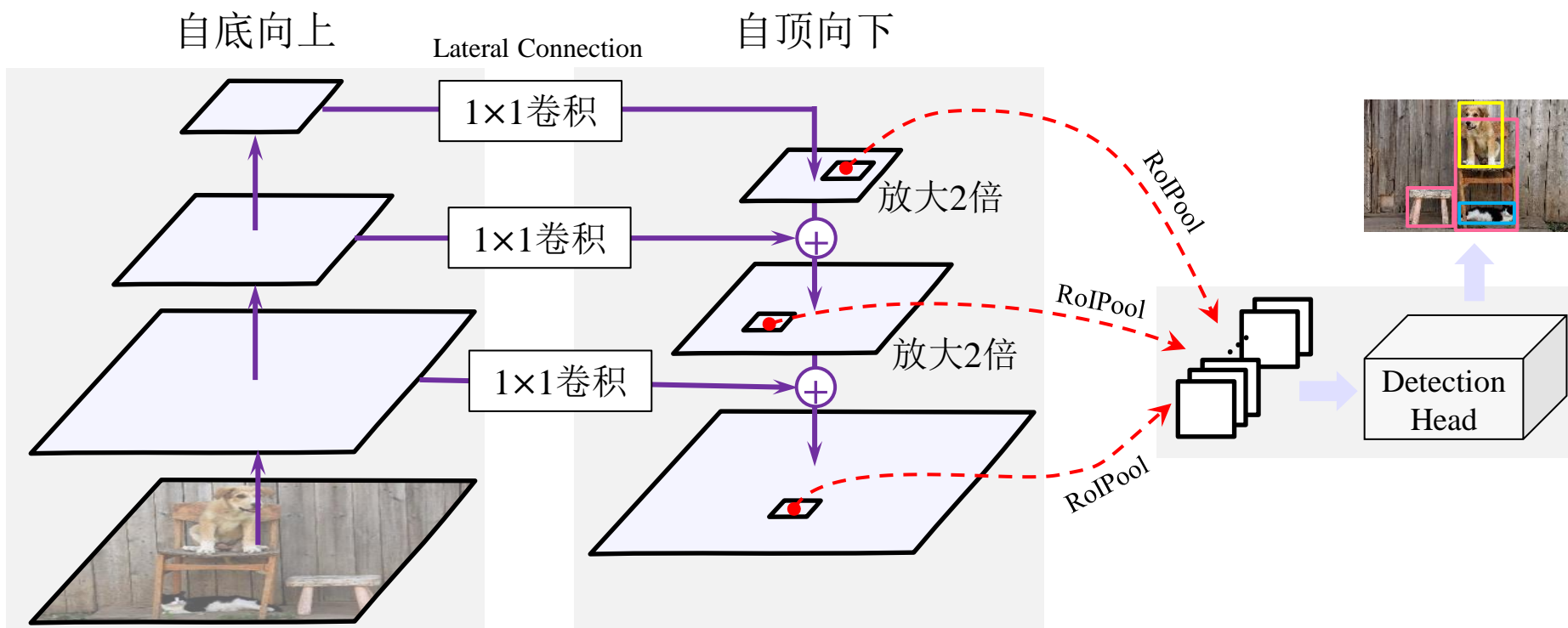
CNN本身的分层结构相当于一个特征金字塔



特征金字塔网络Feature Pyramid Network

■ 两阶段检测器：FPN

- 利用好CNN本身的结构，在此基础上**对浅层进行语义补偿**



深度学习时代的目标检测

■ 两阶段检测器：SNIP/SNIPER/AutoFocus

Scale Normalization
for Image Pyramids

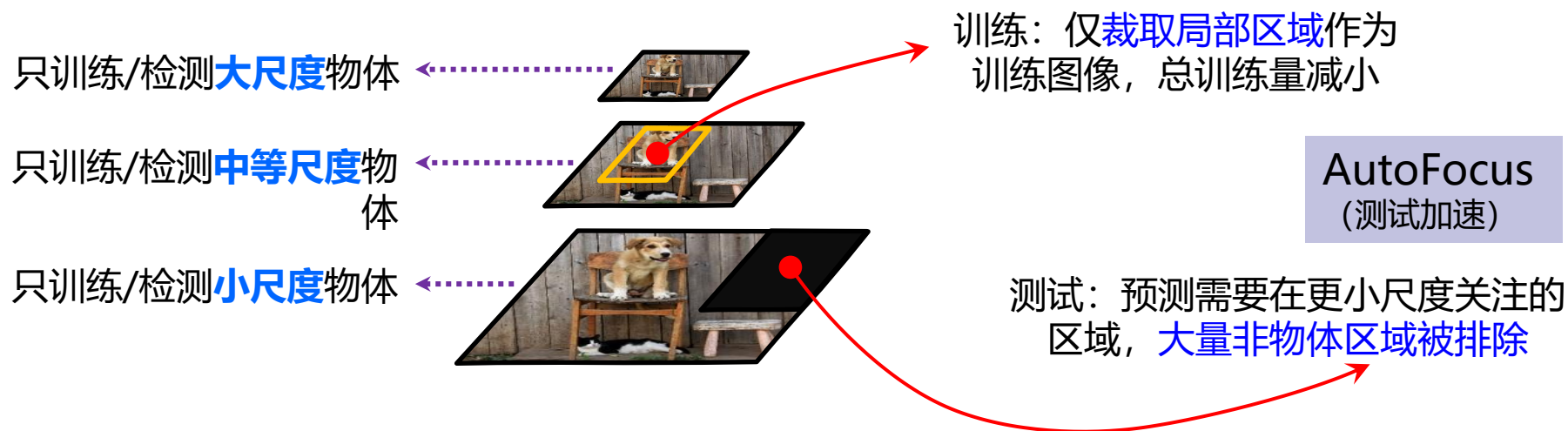
□ CNN只具有有限的尺度不变性→回归到**图像金字塔**

- 多尺度输入 + “单” 尺度模型
- 在不同的输入尺度上处理不同尺度的物体
- 保证训练和测试时输入尺度的一致性

SNIP
(分析问题)

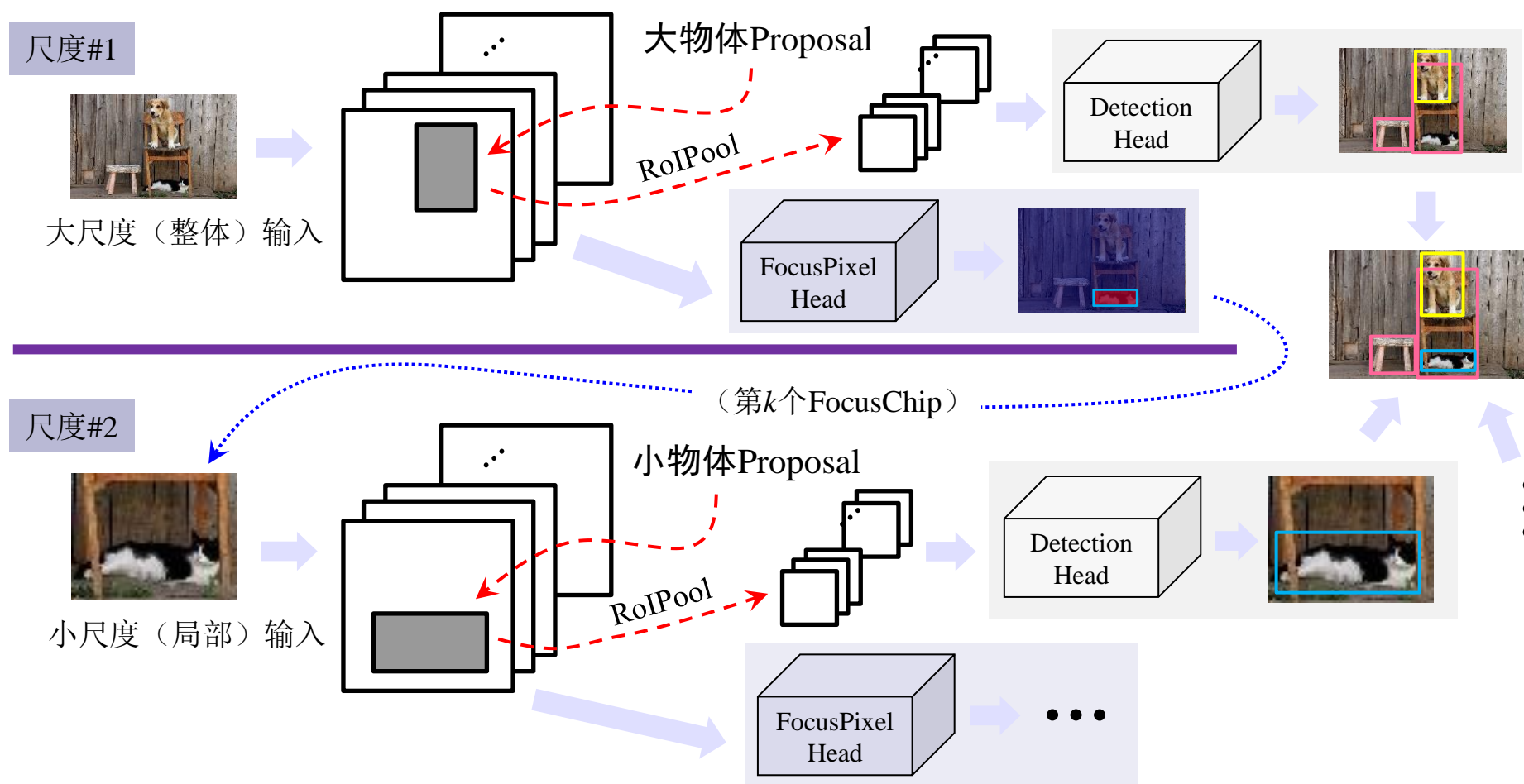
SNIPER
(训练加速)

AutoFocus
(测试加速)



深度学习时代的目标检测

■ 两阶段检测器：SNIP/SNIPER/AutoFocus



深度学习时代的目标检测

■ 两阶段检测器：延伸阅读

□ 关于目标检测的其它问题

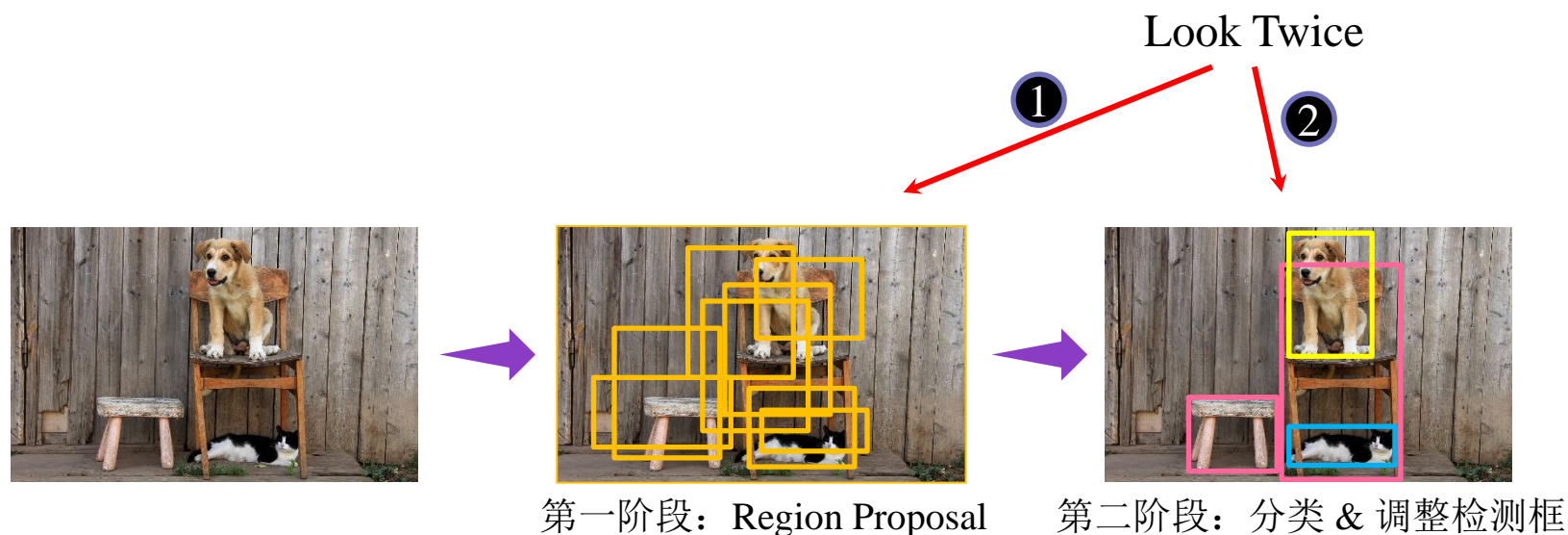
- SMN(Spatial Memory Network): 引入记忆和推理机制对上下文建模
- DeNet(Directed Sparse Sampling)
 - 将目标检测定义为概率分布估计问题
- DCN(Deformable ConvNet): 可变形卷积建模物体形变

□ 模型结构、训练等方面的优化

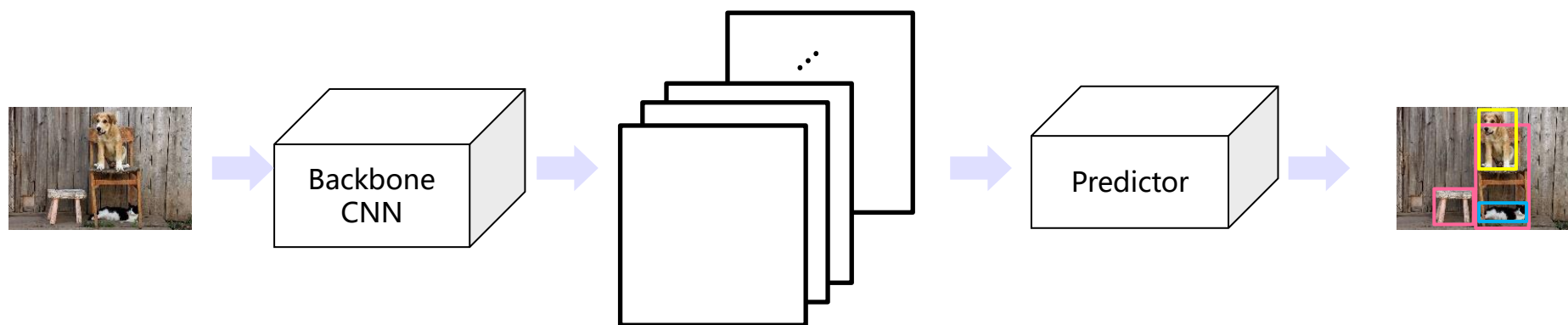
- R-FCN: 针对Faster R-CNN + ResNet进行结构优化
- Light-Head R-CNN: 轻量级头部网络
- DetNet: 面向检测的主干网络
- OHEM(Online Hard Example Mining): 难例挖掘
- MegDet: 大Batch Size情况下检测模型的训练

深度学习时代的目标检测

■ 回顾：两阶段检测器



- 问题形式化：目标检测问题被形式化成分类/回归问题
- 模型训练：两个阶段的多个任务同时进行优化并不容易
- 速度：第二阶段所需时间随着Region Proposal的增多呈线性增长



Overfeat, DenseBox → YOLO & SSD → ...

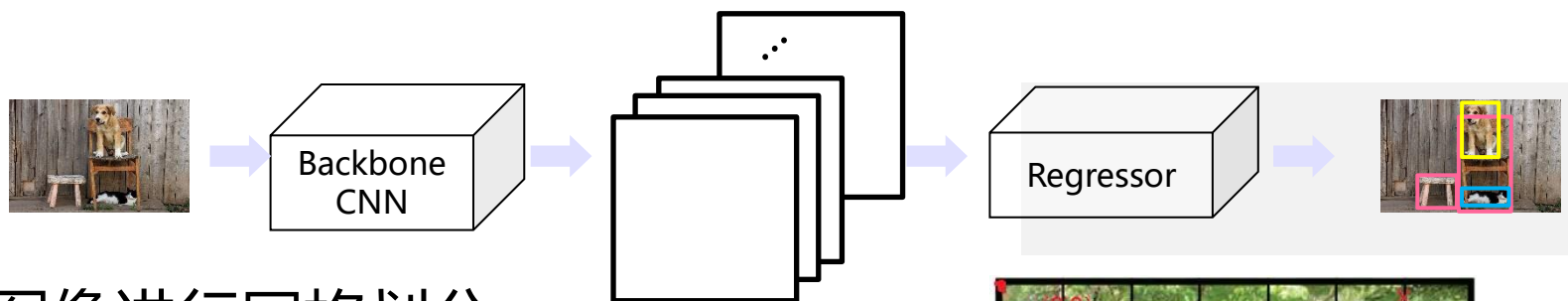
单阶段检测器

单阶段检测器——YOLO

■ 单阶段检测器：YOLO (**You Only Look Once**)

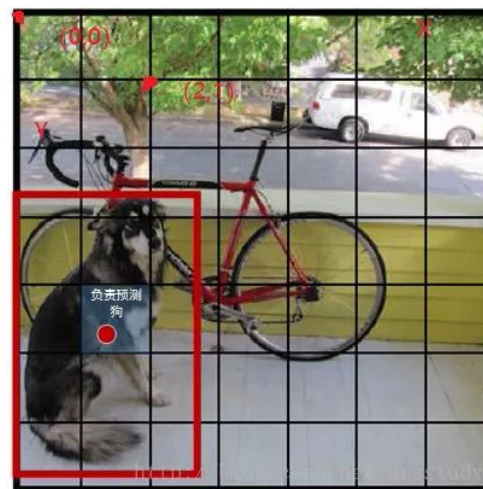
□ 将**目标检测形式化为回归问题**

- 从整张图像直接回归类别概率和检测框



■ 对图像进行网格划分

- 目标中心所在cell负责该物体的检测
- 只需物体中心落在该cell中

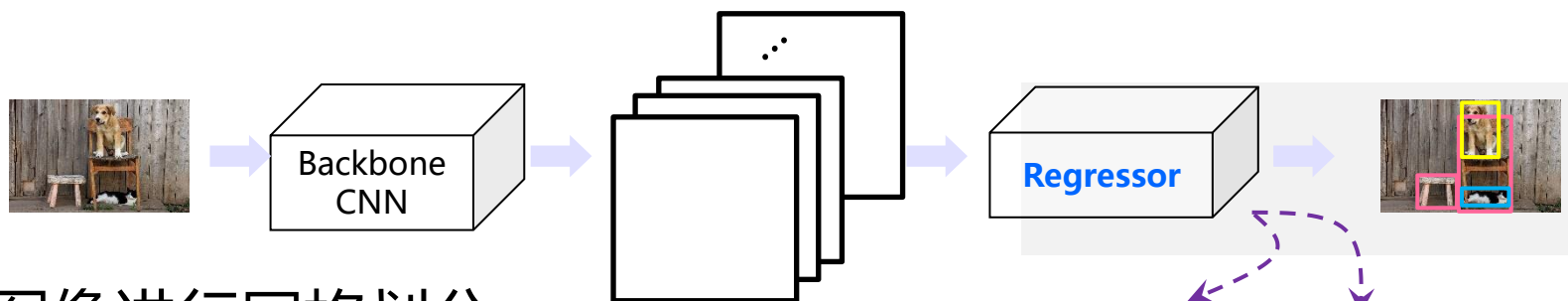


单阶段检测器——YOLO

■ 单阶段检测器：YOLO (You Only Look Once)

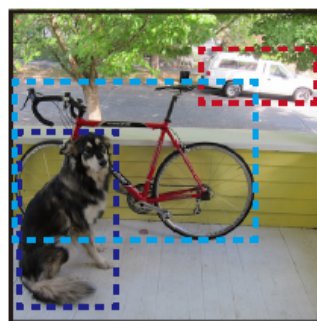
□ 将目标检测形式化为回归问题

- 从整张图像直接回归类别概率和检测框

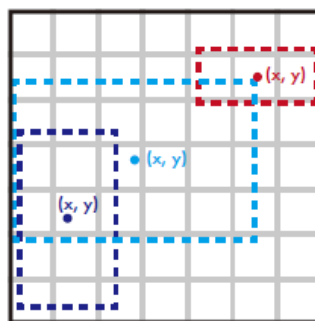


■ 对图像进行网格划分

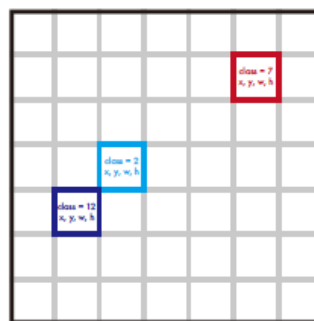
固定
输入尺寸
448×448



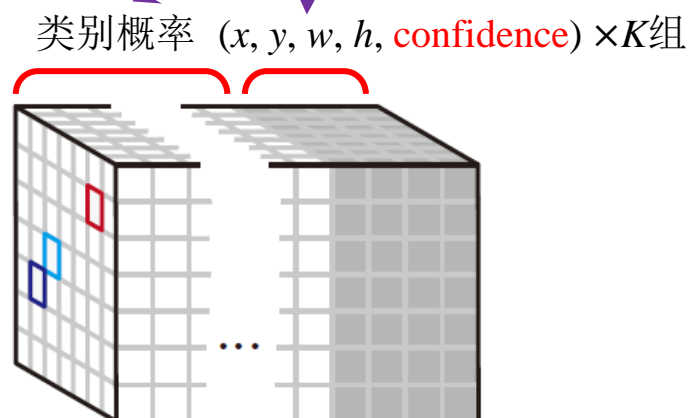
Resize The Image
And bounding boxes to 448 x 448.



Divide The Image
Into a 7 x 7 grid. Assign detections to
grid cells based on their centers.



Train The Network
To predict this grid of class probabilities
and bounding box coordinates.



单阶段检测器——YOLO

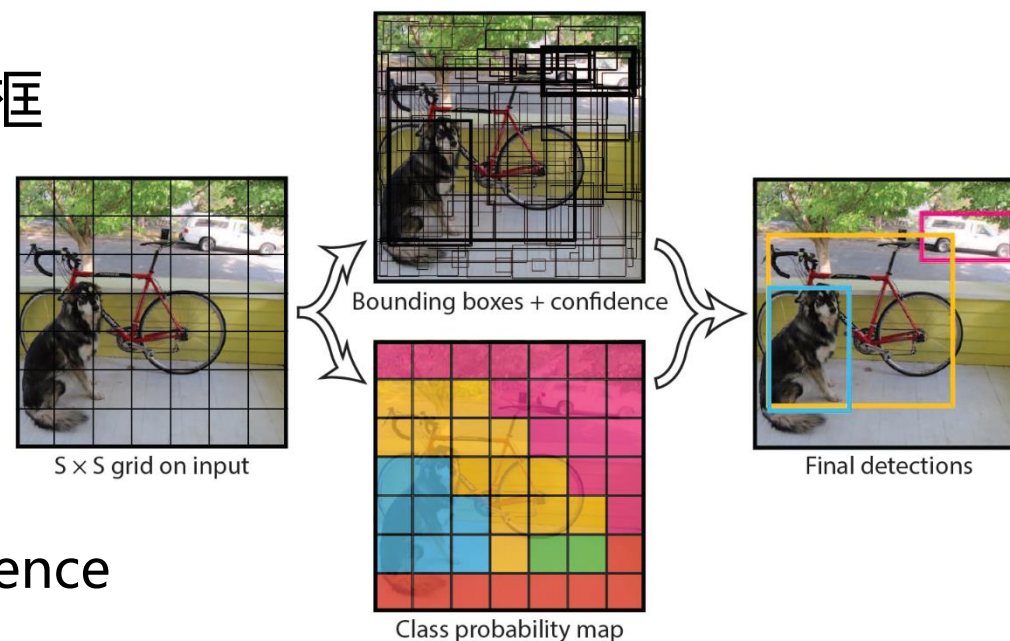
■ 单阶段检测器：YOLO

□ 将目标检测形式化为回归问题

- 从整张图像直接回归类别概率和检测框

□ 步骤如下

- 将全图缩放为固定大小(e.g.448*448)
- 划分为网格 $S \times S$ 个cell (e.g. $S=7$)
- 对每个cell, 用全图特征预测
 - B个BBox及其每个BBox是目标的Confidence
 - 每个cell属于C类中每一类的概率
- 得到 $S \times S \times (B \times 5 + C)$ Tensor【B取值为2】
 - 5: x, y, w, h, confidence; C: 每个cell属于每个类别的概率

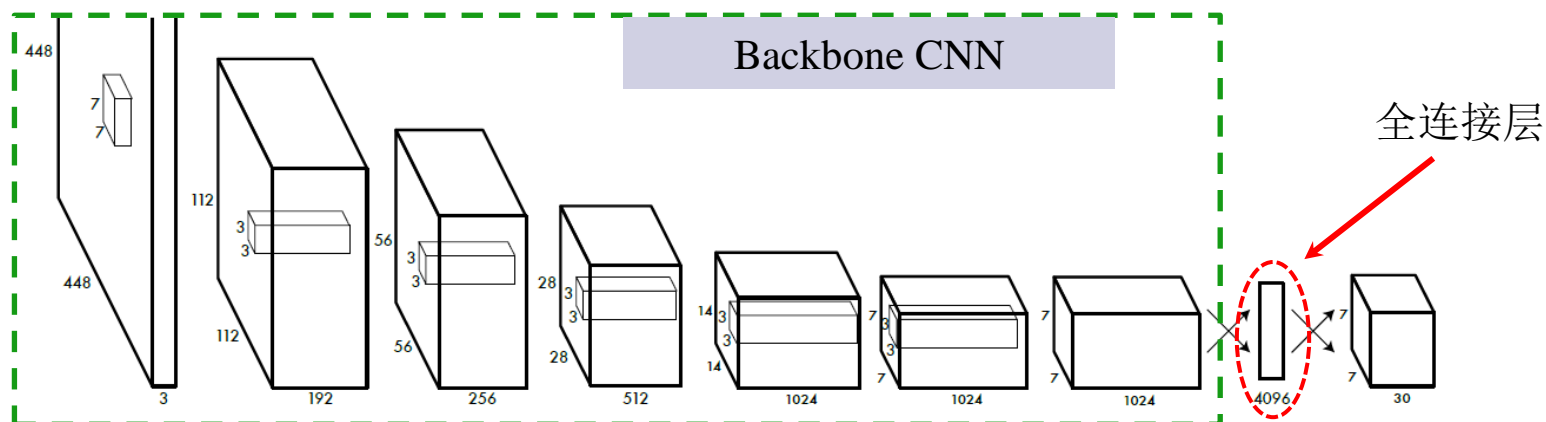
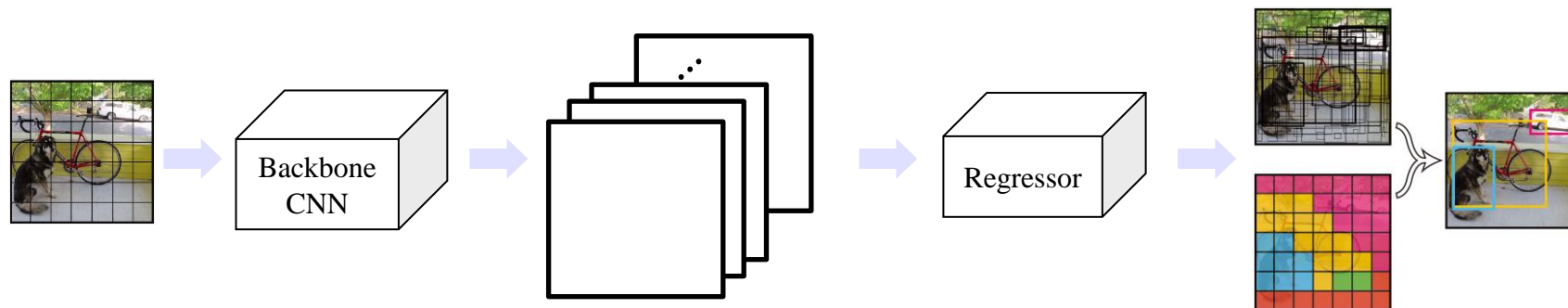


单阶段检测器——YOLO

■ 单阶段检测器：YOLO

□ 将目标检测形式化为回归问题

- 从整张图像直接回归类别概率和检测框



单阶段检测器——YOLO

■ 单阶段检测器：YOLO

□ 将目标检测形式化为回归问题

- 从整张图像直接回归类别概率和检测框

□ 缺点

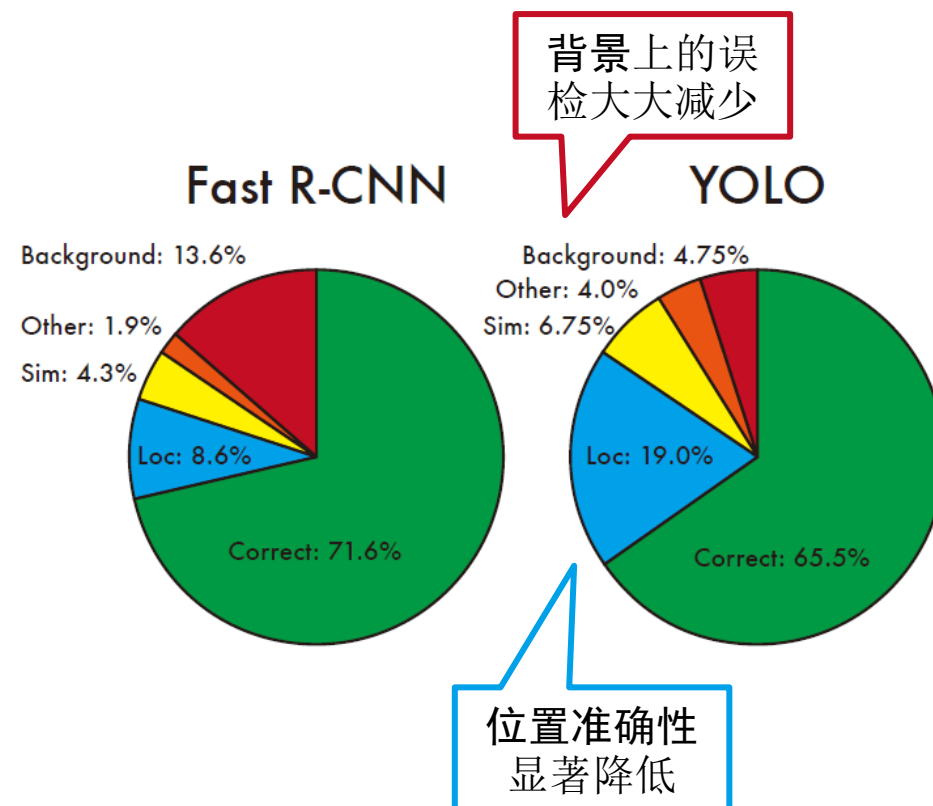
- 虽然一个cell可以预测多个bounding boxes，但是只能识别出一个物体，因此每个cell需要预测物体的类别，而bounding box不需要
 - 这是Yolo v1的缺点，在后来的改进版本中，Yolo9000是把类别概率预测值与BBBox绑定的
- 另一个缺点：最多只能检测出 $S \times S$ 个物体，所以对小目标（一个cell中可能有多个目标）的检测不理想。

单阶段检测器——YOLO

■ 单阶段检测器：YOLO

□ YOLO的检测速度和精度

Real-Time Detectors	Train	mAP	FPS
100Hz DPM [30]	2007	16.0	100
30Hz DPM [30]	2007	26.1	30
Fast YOLO	2007+2012	52.7	155
YOLO	2007+2012	63.4	45
Less Than Real-Time			
Fastest DPM [37]	2007	30.4	15
R-CNN Minus R [20]	2007	53.5	6
Fast R-CNN [14]	2007+2012	70.0	0.5
Faster R-CNN VGG-16[27]	2007+2012	73.2	7
Faster R-CNN ZF [27]	2007+2012	62.1	18

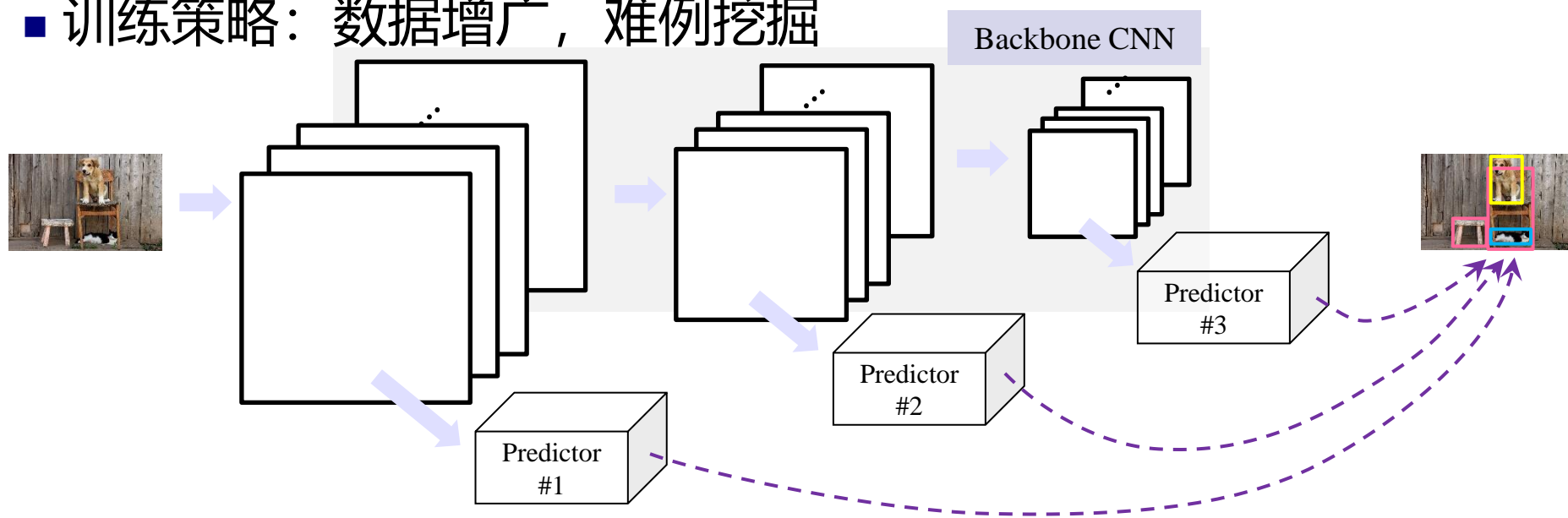


单阶段检测器——SSD

■ 单阶段检测器：SSD(Single-Shot MultiBox Detector)

□ 首次给出了“Single-Shot”的说法

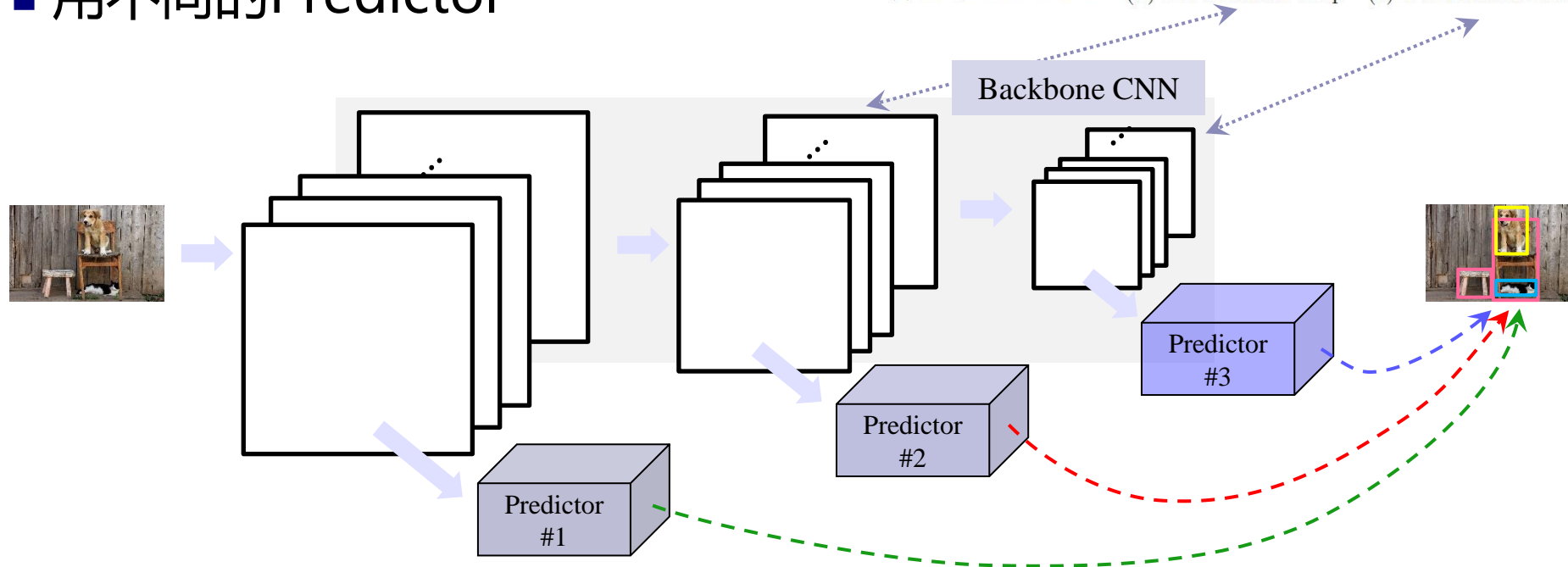
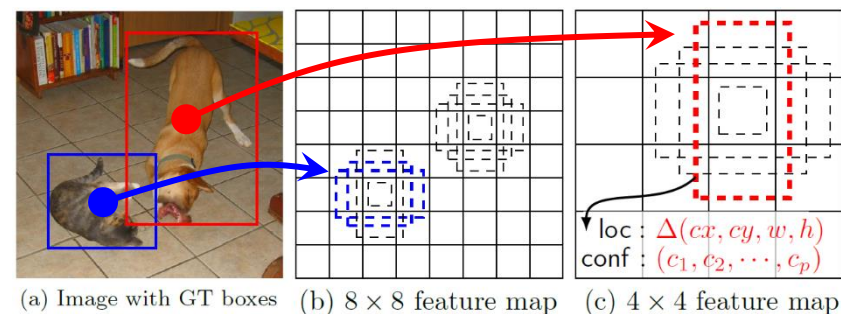
- 类似YOLO的出发点：不生成Region Proposal，直接输出检测结果
- 借鉴RPN的设计：全卷积，Anchor Box → Default Box
- 引入新的设计：多尺度
- 训练策略：数据增广，难例挖掘



单阶段检测器——SSD

■ 单阶段检测器：SSD

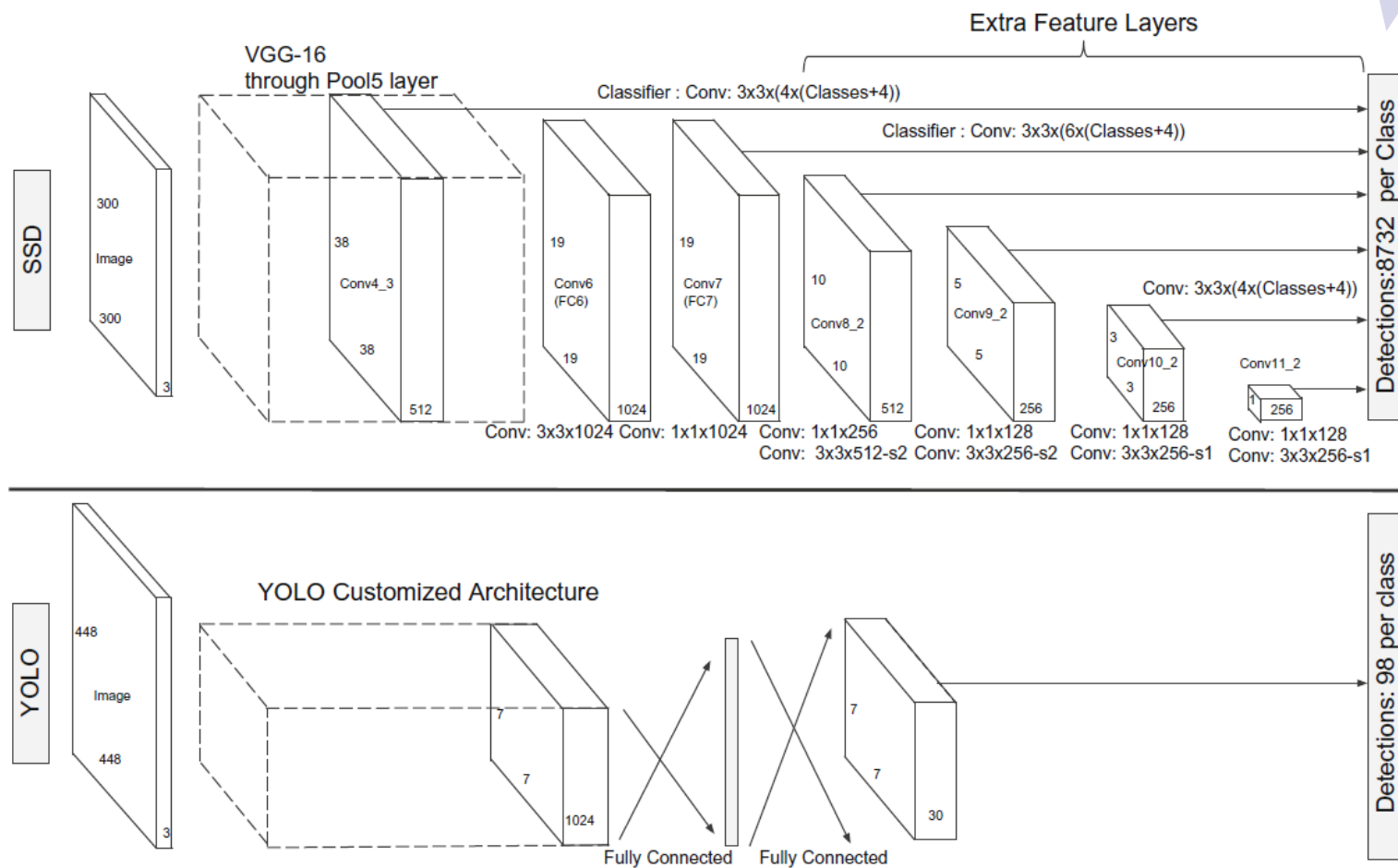
- 多尺度：对于**不同大小**的物体
 - 在**不同尺度**的特征图上预测
 - 用不同的Predictor



单阶段检测器——SSD

■ 单阶段检测器：SSD vs YOLO

框的数量
大大增加！



深度学习时代的目标检测

■ 单阶段检测器：SSD

□ 实现细节的影响

- 数据增广非常关键

	SSD300					
more data augmentation?	✓	✓	✓	✓	✓	✓
use conv4_3?	✓	✓	✓	✓	✓	✓
include $\{\frac{1}{2}, 2\}$ box?	✓	✓	✓	✓	✓	✓
include $\{\frac{1}{3}, 3\}$ box?	✓	✓	✓	✓	✓	✓
use atrous?	✓	✓	✓	✓	✓	✓
VOC2007 test mAP	65.4	68.1	69.2	71.2	71.4	72.1

□ 精度和速度比较

Method	data	Average Precision		
		0.5	0.75	0.5:0.95
Fast R-CNN [6]	train	35.9	-	19.7
Faster R-CNN [2]	train	42.1	-	21.5
Faster R-CNN [2]	trainval	42.7	-	21.9
ION [21]	train	42.0	23.0	23.0
SSD300	trainval35k	38.0	20.5	20.8
SSD500	trainval35k	43.7	24.7	24.4

MS COCO test-dev2015

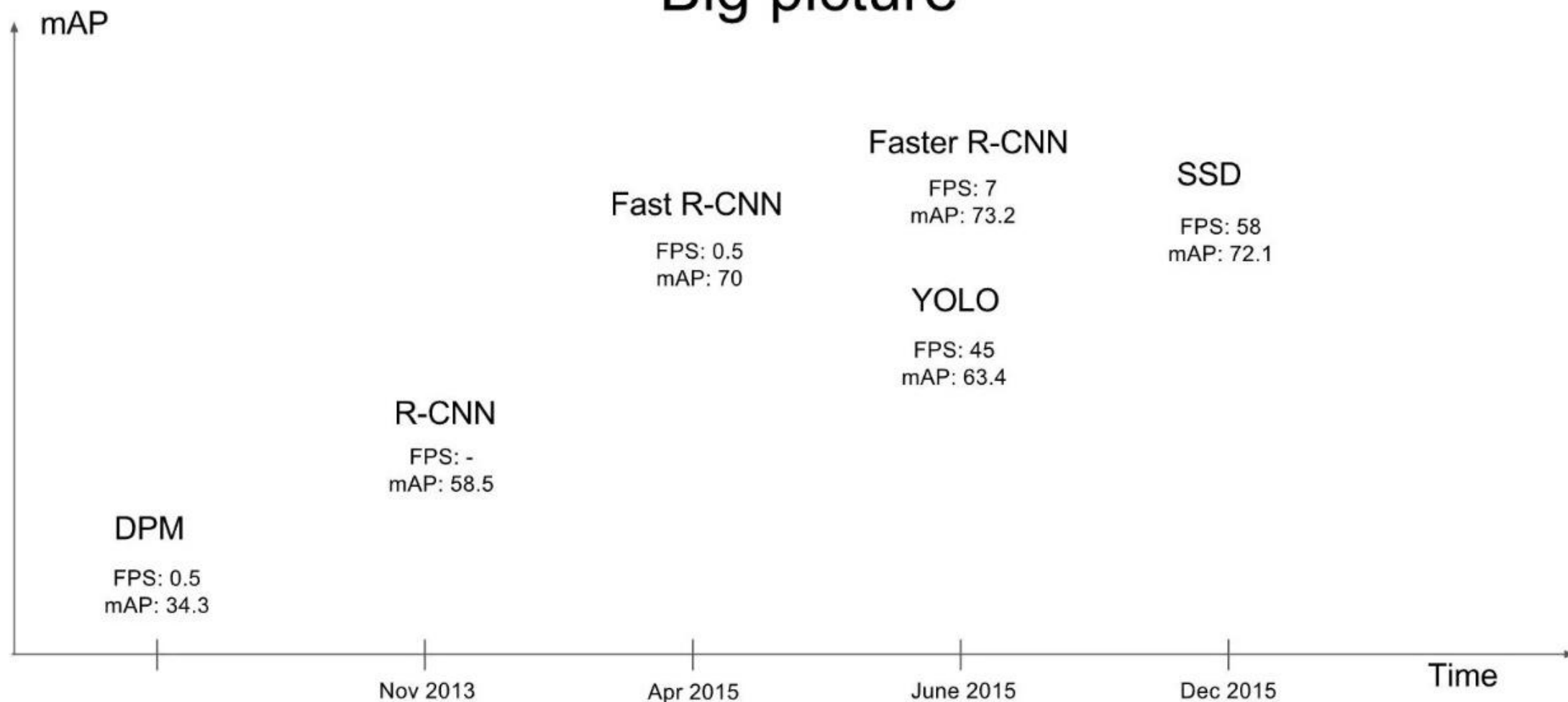
Method	mAP	FPS	# Boxes
Faster R-CNN [2] (VGG16)	73.2	7	300
Faster R-CNN [2] (ZF)	62.1	17	300
YOLO [5]	63.4	45	98
Fast YOLO [5]	52.7	155	98
SSD300	72.1	58	7308
SSD500	75.1	23	20097

Pascal VOC 2007

若干早期目标检测器的性能比较



Big picture

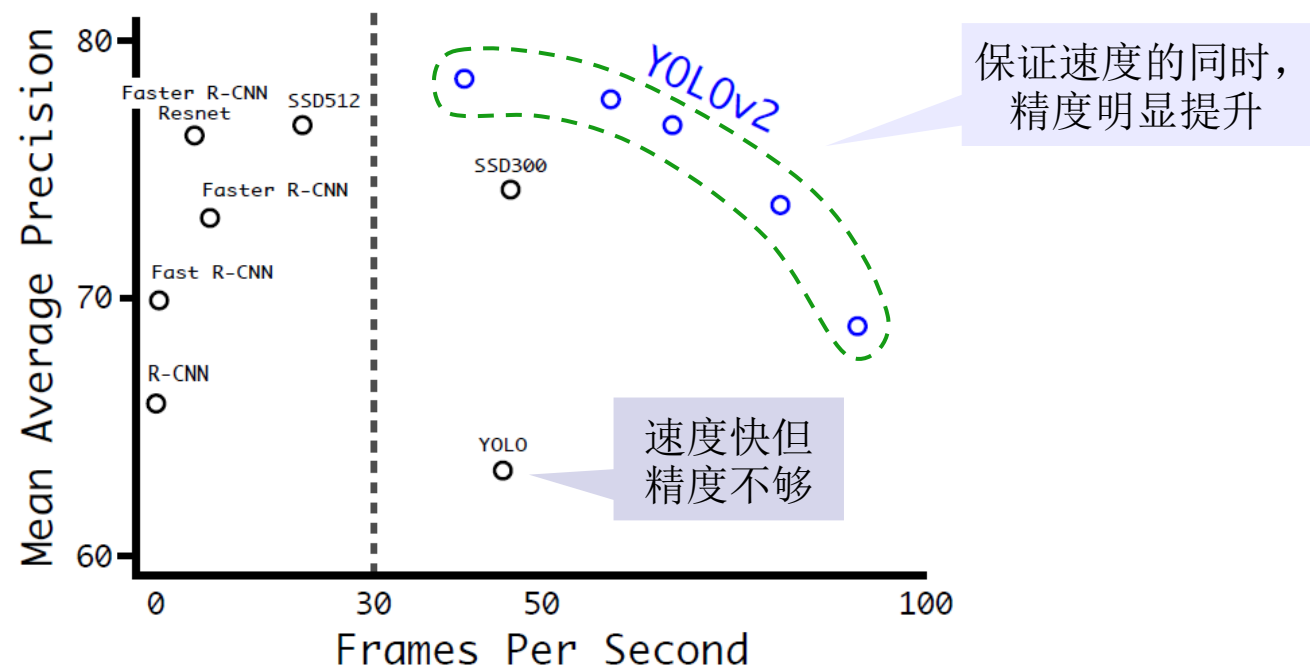


YOLO的持续改进——YOLO v2

■ 单阶段检测器：YOLOv2

□ 借鉴两阶段检测器中RPN的设计

- 全连接替换为卷积（全卷积网络）
- 引入**Anchor Box**：采用聚类的方式确定其尺度和长宽比设置



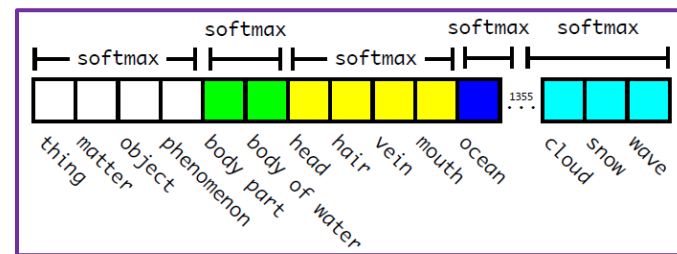
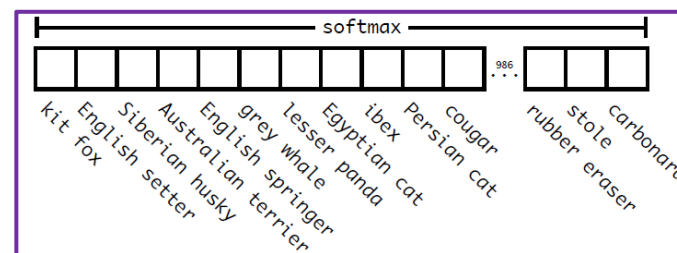
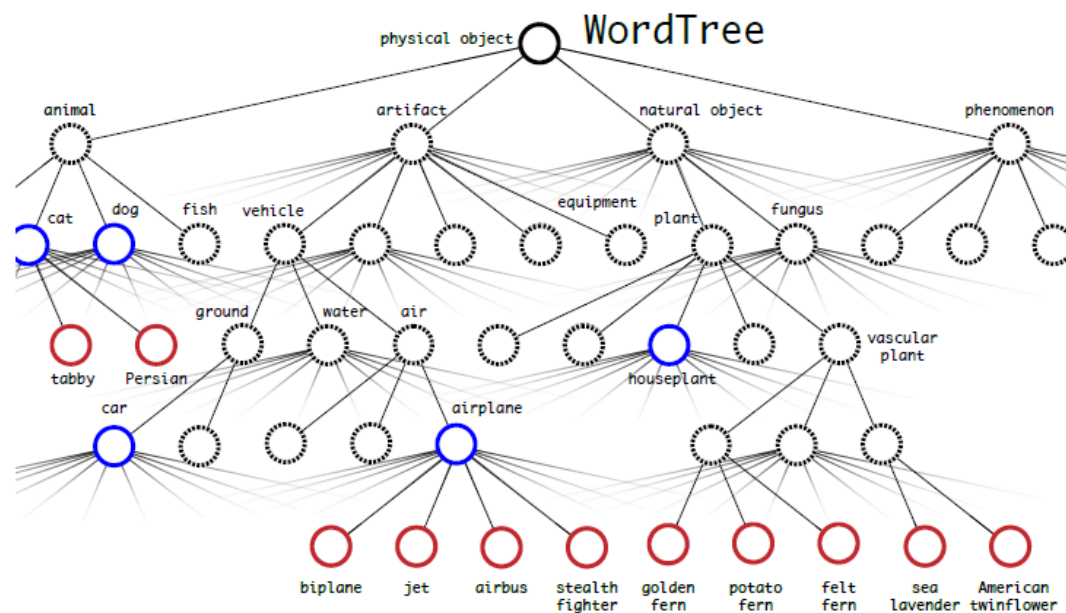
YOLO的持续改进——YOLO v2

■ 单阶段检测器：YOLOv2/YOLO9000 第一个针对如此大规模类别的目标检测器

□ 学习9000个物体类别的检测器

- 层次化分类：从根结点到当前结点的概率相乘
- 没有框标注的类别（弱监督）在当前概率最大的位置学习分类

相当于猜了一个伪标签

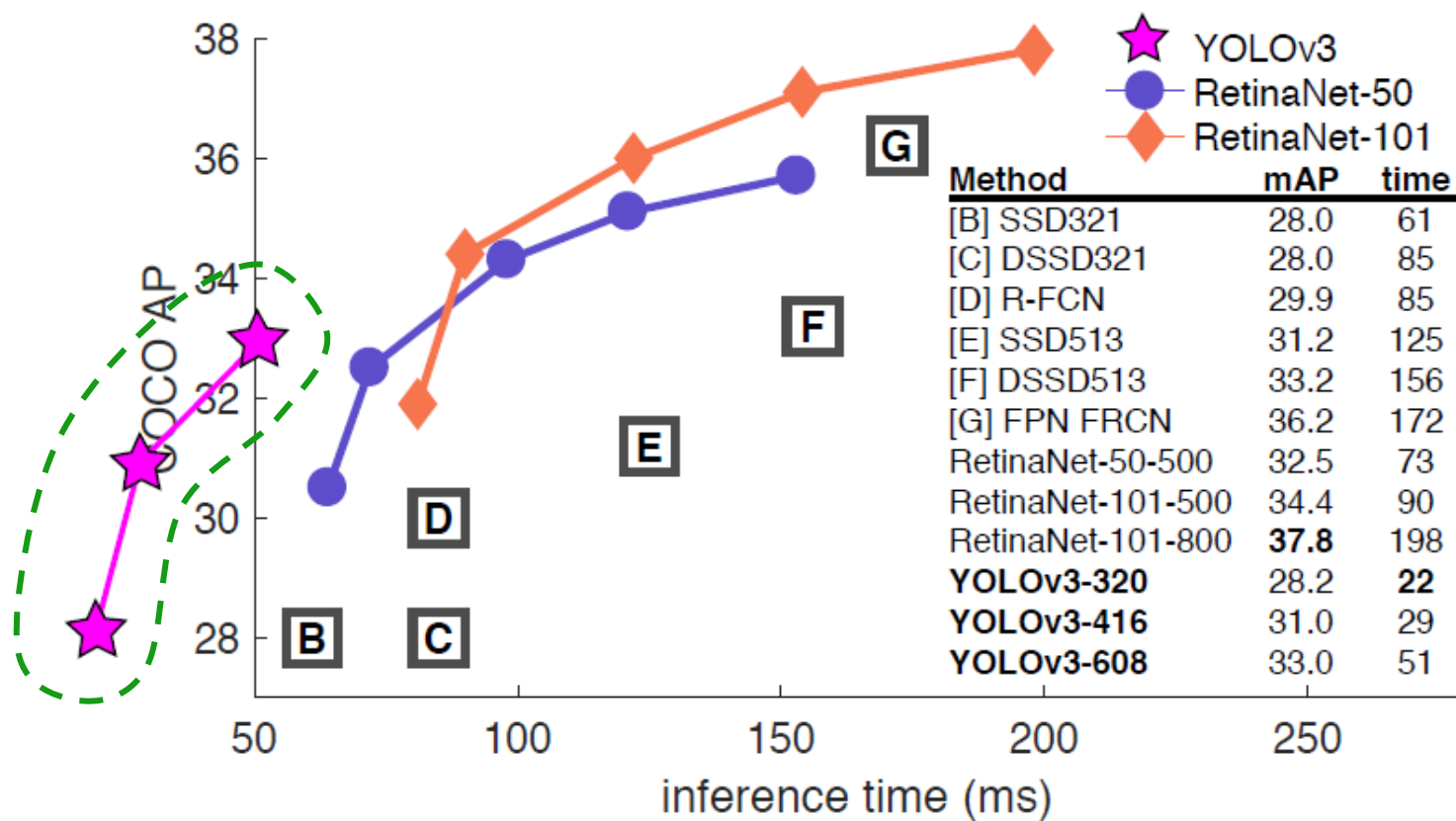


YOLO的持续改进——YOLO v3

■ 单阶段检测器：YOLOv3

The devil is in the detail

□ 工程化调优：尝试目标检测领域的各项最新成果



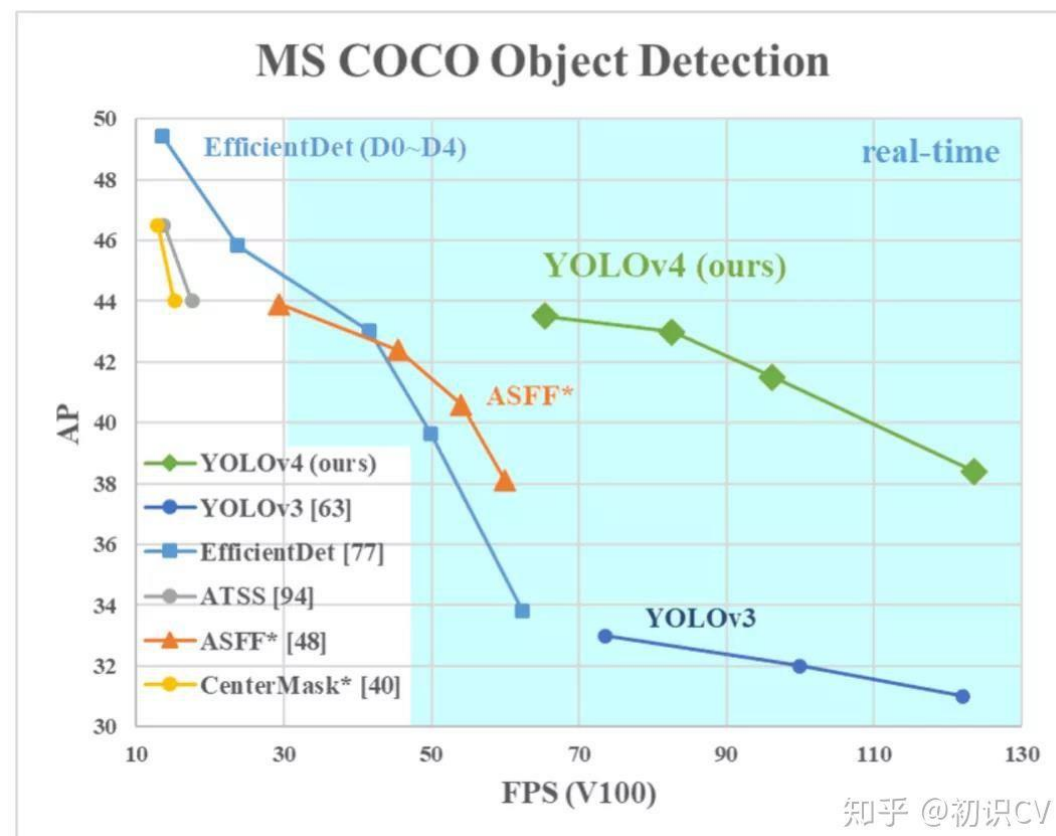
YOLO的持续改进——YOLO v4

■ 单阶段检测器：YOLOv4

The devil is in the detail

□ 工程化调优：尝试目标检测领域的各项最新成果

- Backbone: CSPDarknet53
- Neck: SPP, PAN
- Head: YOLOv3



深度学习时代的目标检测

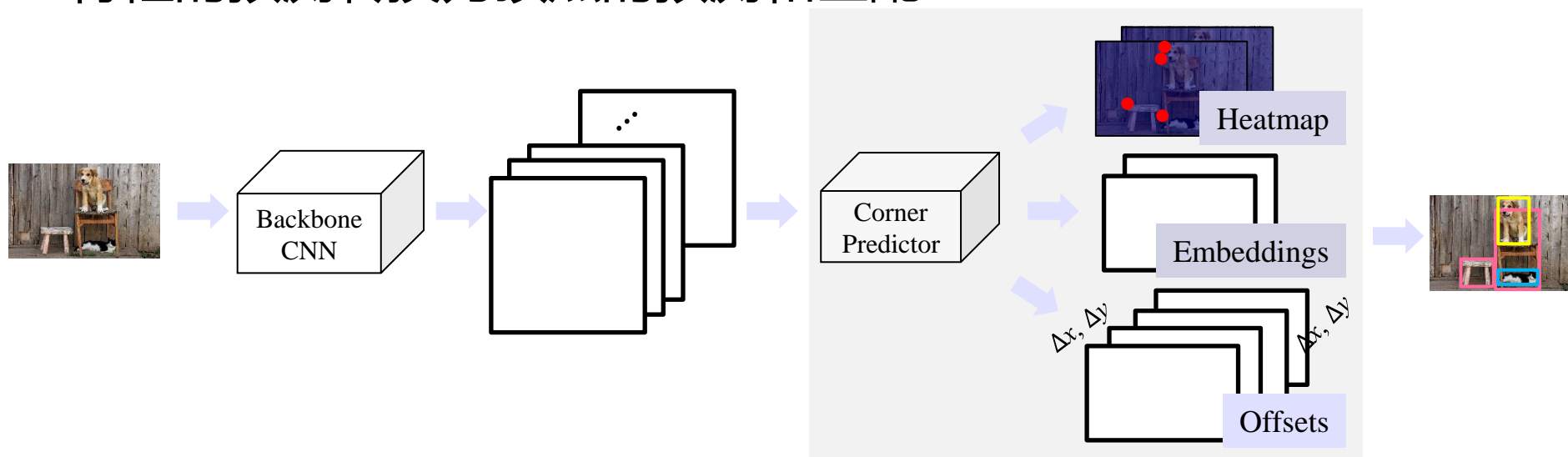
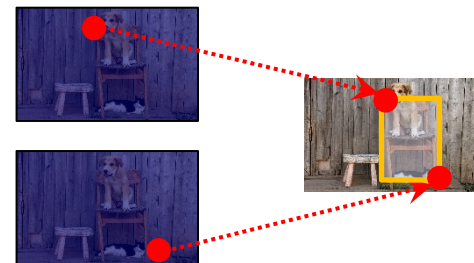
■ 单阶段检测器：CornerNet

□ 使用Anchor Box的问题

- 数量多，导致正负样例极不平衡
- 需要人工定义，且引入了大量超参数，这些选择严重依赖于数据

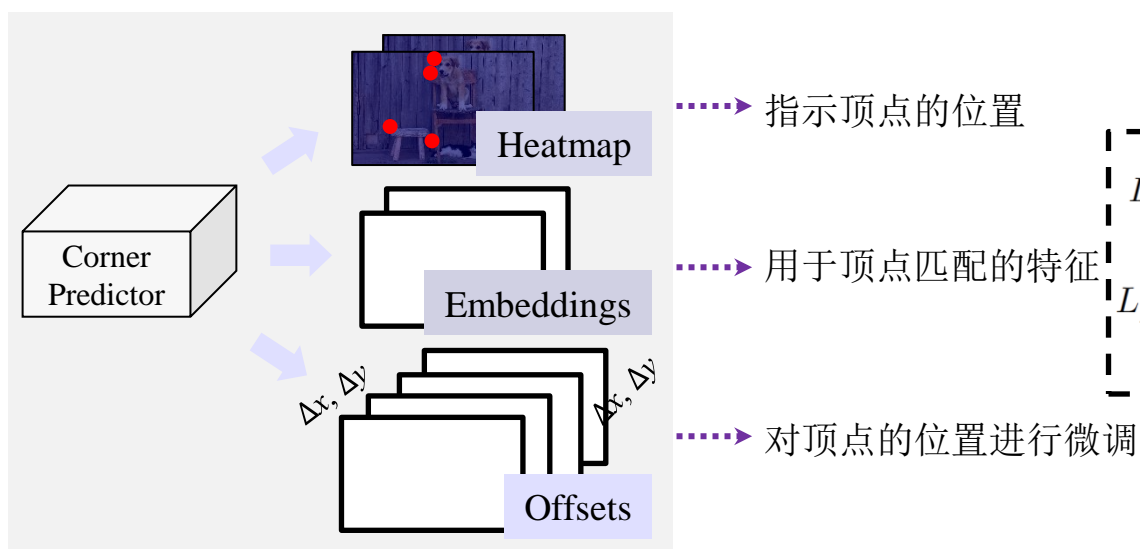
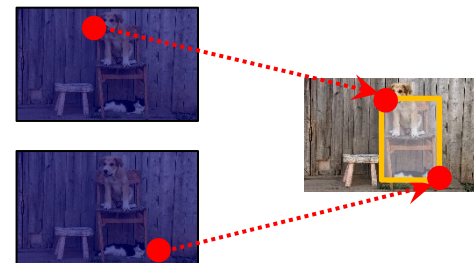
□ Anchor Box → Corner

- 将框的预测转换为顶点的预测和匹配



深度学习时代的目标检测

- 单阶段检测器：CornerNet
 - Anchor Box → Corner
 - 将框的预测转换为顶点的预测和匹配



同框顶点相吸，
异框顶点相斥

$$L_{pull} = \frac{1}{N} \sum_{k=1}^N \left[(e_{t_k} - e_k)^2 + (e_{b_k} - e_k)^2 \right]$$

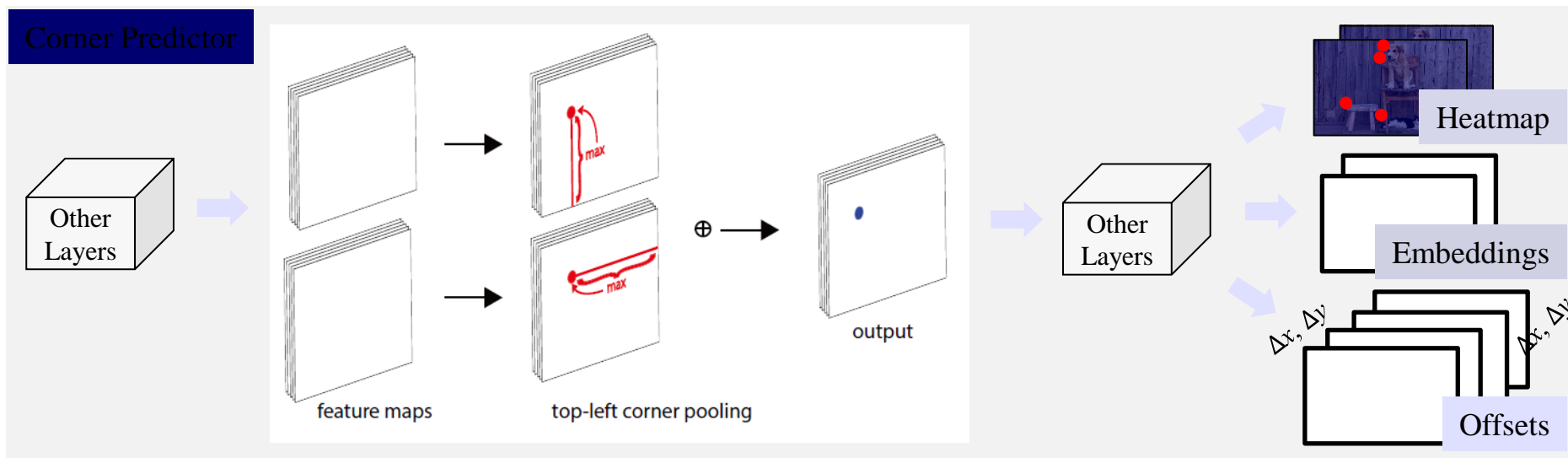
$$L_{push} = \frac{1}{N(N-1)} \sum_{k=1}^N \sum_{\substack{j=1 \\ j \neq k}}^N \max(0, \Delta - |e_k - e_j|)$$

深度学习时代的目标检测

■ 单阶段检测器：CornerNet

□ 提取适合Corner预测的特征：Corner Pooling

- 框的顶点位置可能不在物体上
- 局部区域Pooling → 在一个方向上进行Pooling

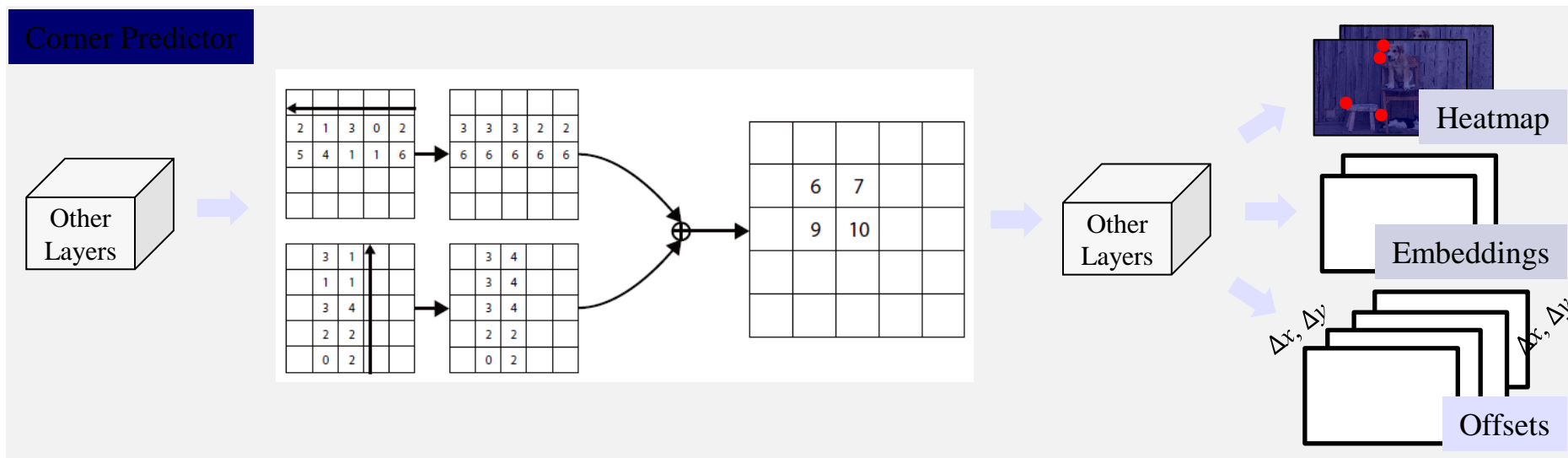


深度学习时代的目标检测

■ 单阶段检测器：CornerNet

□ 提取适合Corner预测的特征：Corner Pooling

- 框的顶点位置可能不在物体上
- 局部区域Pooling → 在一个方向上进行Pooling



深度学习时代的目标检测

■ 单阶段检测器：延伸阅读

□ 关于目标检测的其它问题

- RFB：模拟人的视觉感受野设计特征提取模块

Receptive Field Block

□ 模型结构、训练等方面的优化

- OHEM ~* ■ Focal Loss (RetinaNet)：平衡简单和困难样例对训练的作用

- FPN ~* ■ DSSD：融合浅层和深层信息

Deconvolutional SSD

- Cascade R-CNN ~* ■ RefineDet：学习更好的Default Box

Deeply Supervised Object Detector

- DSOD：不用预训练，从零训练检测模型

深度学习时代的目标检测

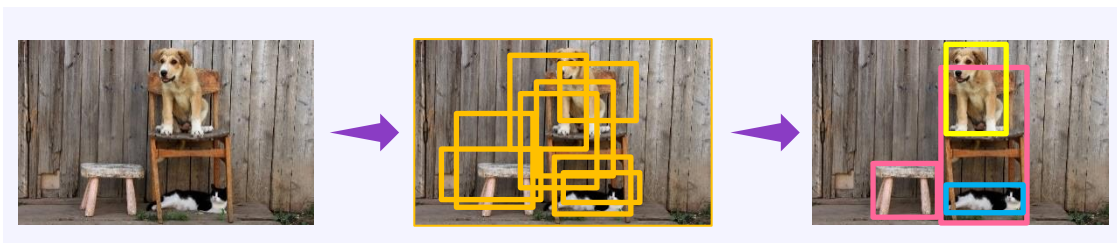
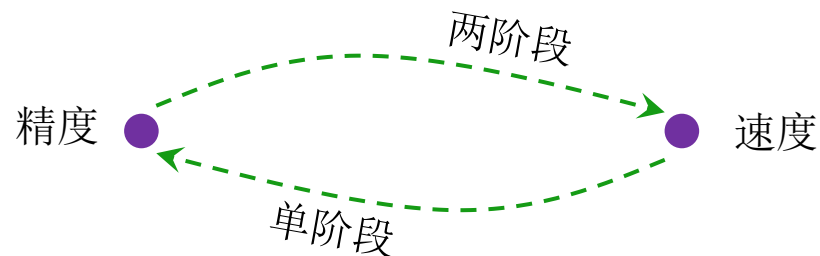
■ 小结

□ 两阶段检测器

- R-CNN → SPPNet, Fast R-CNN → Faster R-CNN
- Mask R-CNN, Cascade R-CNN, FPN, SNIP/SNIPER/AutoFocus

□ 单阶段检测器

- YOLO, SSD
- CornerNet, EfficientDet



深度学习时代的目标检测

■ 更进一步的话题

- 弱监督：如何用更简单的标注学习一个好的检测器？
- 小数据：如何用更少数据学习一个好的检测器？
- 知识迁移：如何利用已经学到的知识辅助新的检测任务？
- 增量学习：如何让现有的检测器学习新的类别？
- 领域适配：如何让检测器在目标场景下表现得更好？
- 视频目标检测：如何实时地检测视频中的物体？
- 3D目标检测：如何以点云作为输入预测3D物体边框？

谢谢!



中国科学院计算技术研究所

Institute of Computing Technology, Chinese Academy of Sciences