



i-VisionGroup@Tsinghua

# 关于R-CNN的综述

刘弘辉

2017-04-21

- R-CNN及其各种改进版本，是目前主流的基于深度学习的物体检测算法，在大规模的自然场景分析问题上，效果相比于之前的方法有明显优势。
- 我们针对R-CNN及其两种改进，fast R-CNN，faster R-CNN作概述。

# 背景

- 物体检测，是计算机视觉的一个基本问题。它的一般任务是从图像（或视频）中将指定类别的物体用较为精细的bounding box标记出。

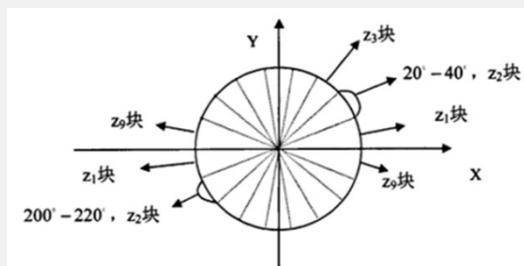
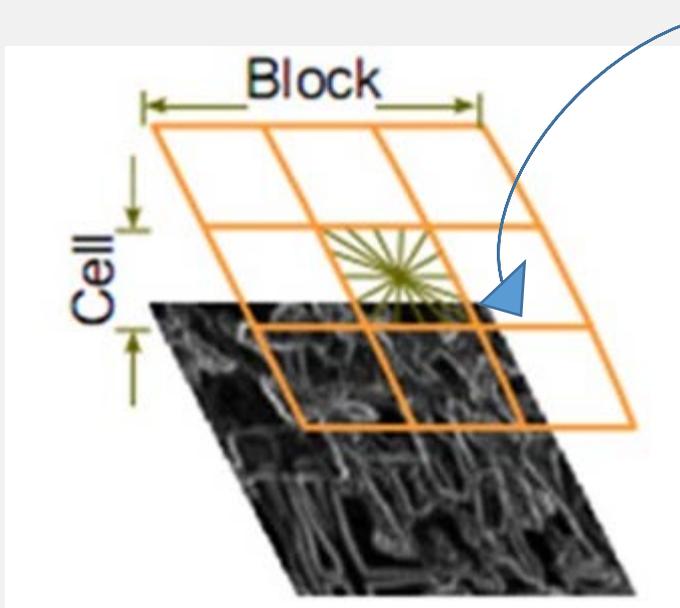
# 相关工作

- 在深度方法兴起之前，对于物体检测问题，主流的方法：

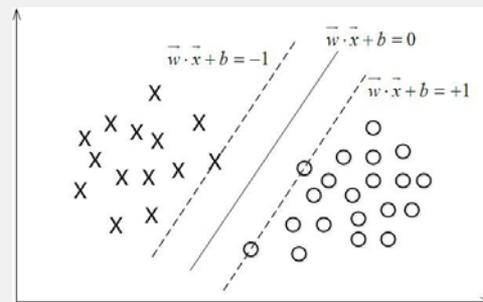
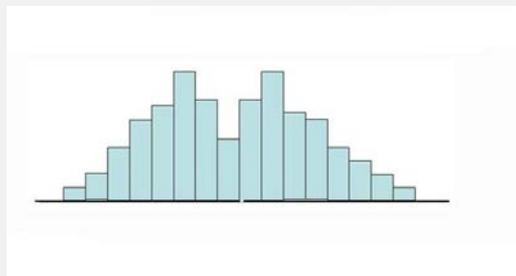


- 其中，HOG和SIFT是当时应用最为广泛的特征；
- 分类器角度，比较成熟的是SVM，以及boosting；

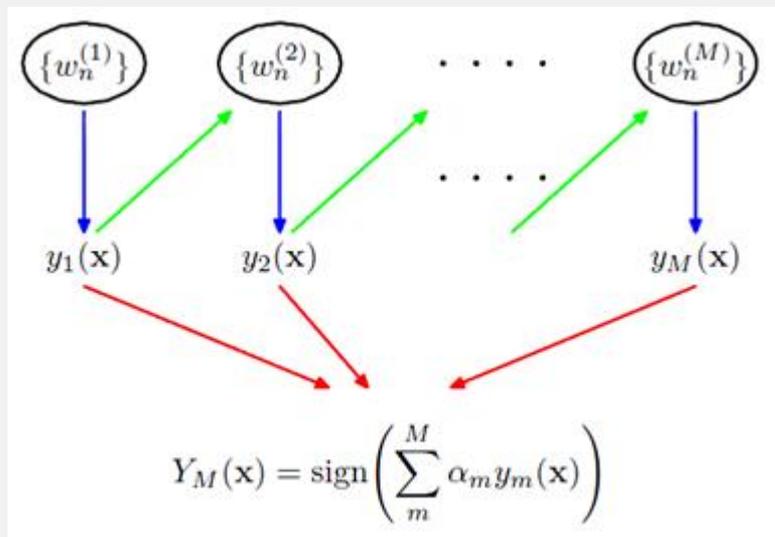
- SIFT和HOG特征对于图像的几何形变，光学影响具有较好的不变性，早期在行人检测问题上取得了很好的效果，并被推广至更复杂的自然场景检测问题。



- 对于经典分类器如SVM，采用SIFT/HOG特征进行训练，相比于直接对原图像（或图像中的patch）进行分类，效果有显著提升。这是由于图像信息有很大冗余，而特征提取的阶段，使得学习到的分类器具有更强的表征能力。

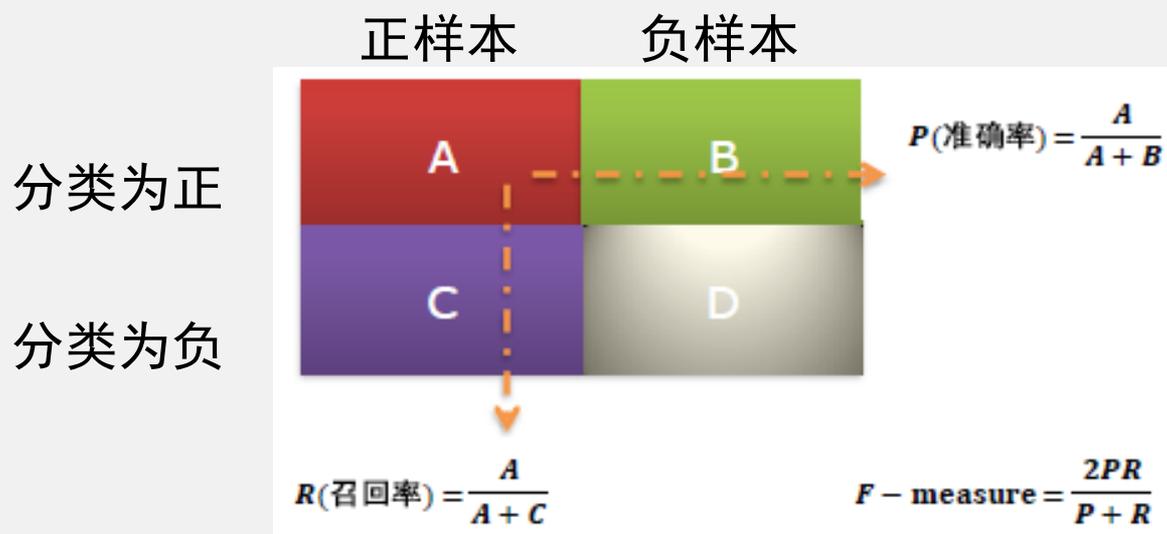


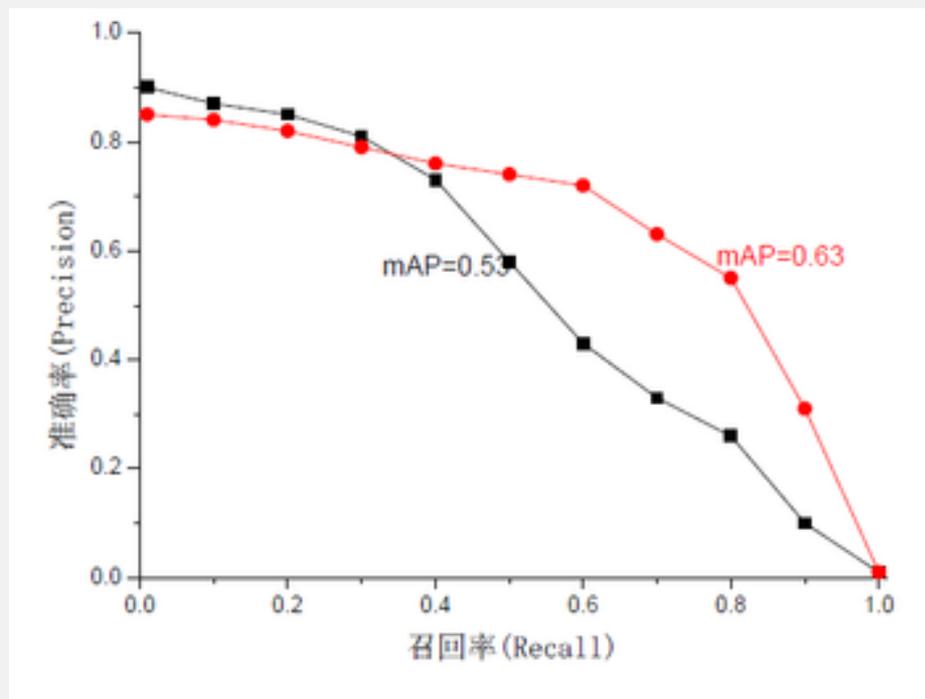
- Boosting是一类将弱分类器加权组合成为强分类器的方法，Viola&Jones采用adaBoost在人脸数据集上实现了高精度的实时检测；boosting方法同样也被推广到更广泛的应用。



- 在VOC数据集上，state-of-the-art方法的效果在一段时间内基本进入了平台期。经典的方法逐渐加入了很多改进，当时最领先的方法是将低层次的图像特征和高层次的纹理相结合，并构建出很复杂的ensemble分类器进行学习，但相比于HOG+SVM也并没有本质的提升。
- 平均准确率均值（mean average precision, mAP）维持在30%左右的水平。

- 这里简要介绍mAP的概念，这是对于物体检测方法通用的评价指标之一。





$$AP = \int P(R)dR$$

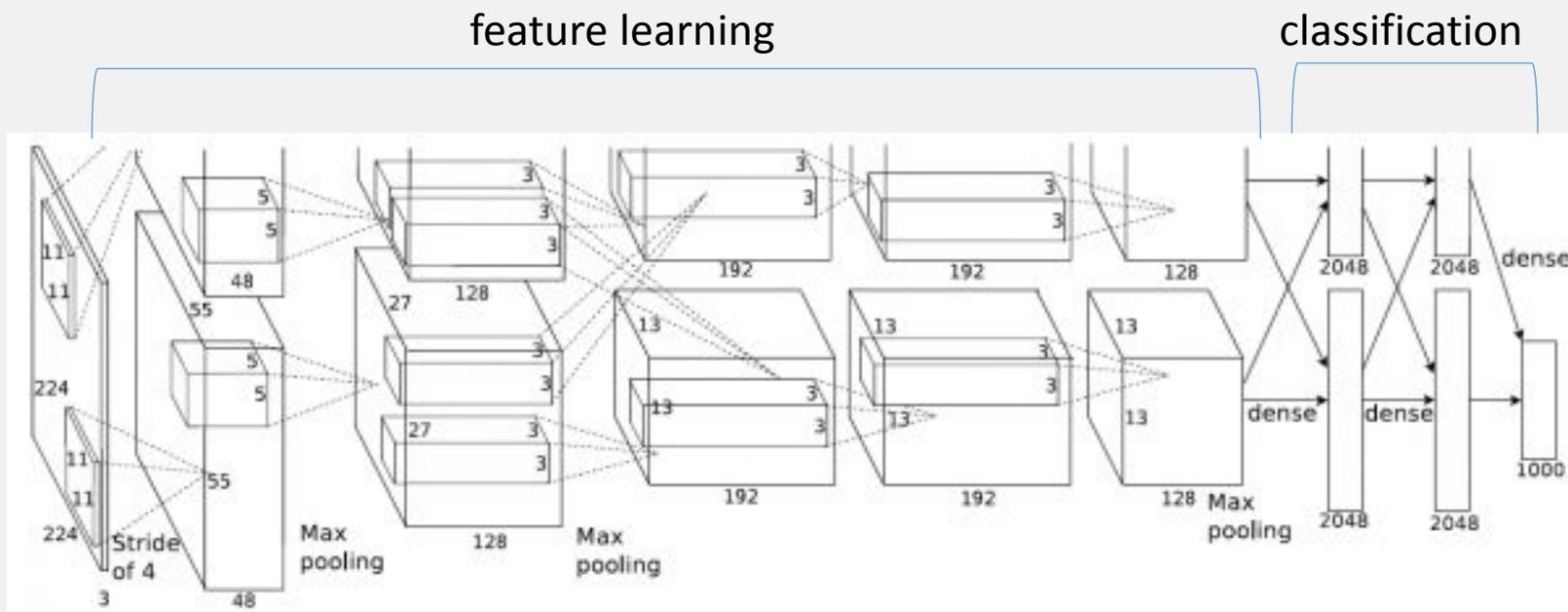
多类问题：对各个类，分别视为二分类求AP，再取均值得mAP

# R-CNN

R Girshick, J Donahue, T Darrell, J Malik. “Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation”. CVPR 2014.

- 人类的认知系统是从浅入深的分层次的，因此分层次，多阶段的特征提取更符合人类的认知习惯。
- 传统的SIFT/HOG特征具有其局限性，尽管具有尺度不变等优点，但本质上仍然是浅层特征。为了能够达到更好的识别效果，需要提取更高层次的特征。
- 深度神经网络就是具有很强的特征学习（feature learning）的能力的结构。

- 卷积神经网络（CNN）对于图像分类问题取得了显著进步。其若干层卷积+池化的操作，本质上是从底层到高层次的特征提取过程，网络的训练同时也是特征的学习。



# 拓展

- Question: 是否能够将CNN的思想和方法应用于物体检测问题?
- 不同于简单的图像分类问题, 物体检测需要准确定位图像中的物体 (可能有多个物体)。一种思路是将检测直接视作对位置的回归问题, 但在VOC数据集上, 这种处理方法效果并没有很好的提升 (mAP 约为30.5%) ;
- 另一种做法是构建一个sliding-window detector, 即遍历扫描图像的滑动窗口监测子。最早应用CNN做物体检测的一些尝试就是这种简单的处理方法, 但这和很早的基于SVM的滑动检测子思想并无本质区别。(特别是处于速度考量采用比较浅的CNN)

- 本文的工作同样是基于sliding-window的，但是与之前的一些尝试主要区别在于：
  - 采用深层次的神经网络结构；
  - 基于区域来考虑对物体的识别；
  - 测试时，不需要用多尺度的滑窗扫描全图，只需要从图像中提取确定数量的区域patch（采用selective search算法，从每张图像采集2000个）。

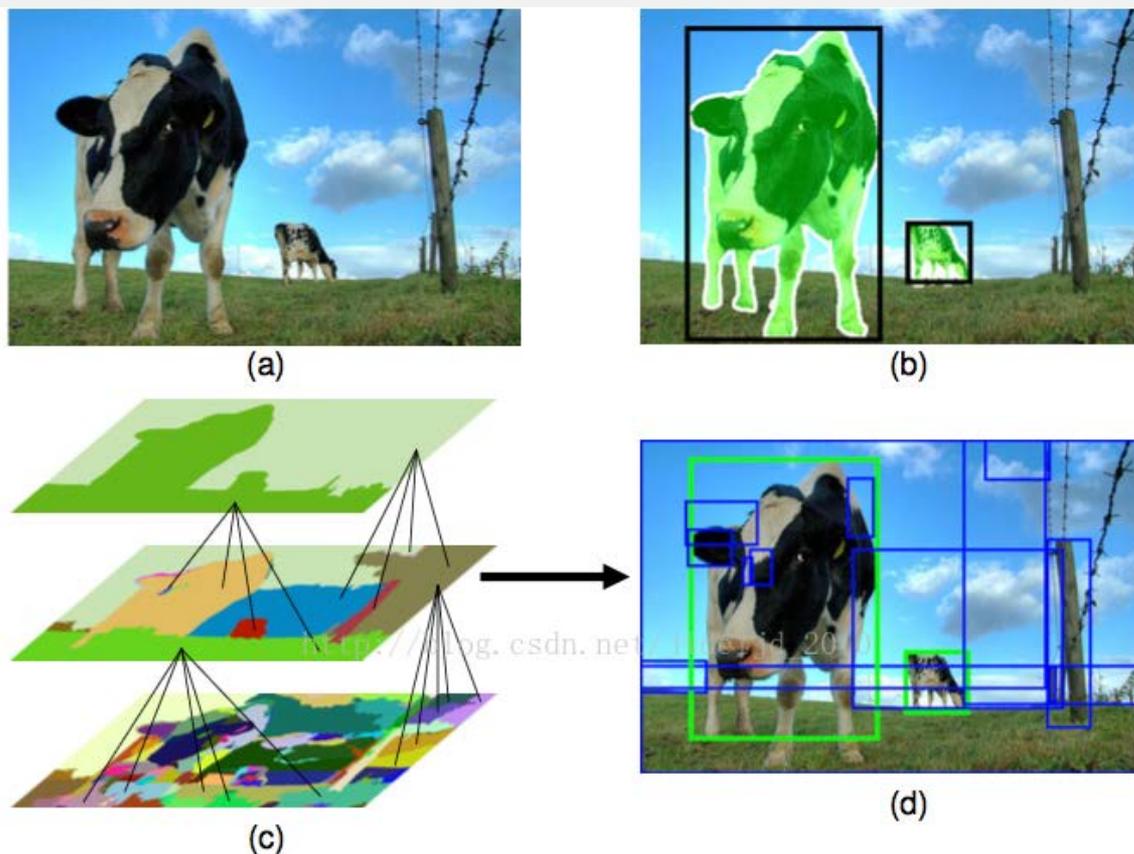
# selective search

- 选择性搜索，是针对物体识别问题，为了替代穷举搜索（exhaustive search），从图像中选择proposal的一种方法。
- 对图像首先进行过分割，方法基于efficient graph-based image segmentation（或者superpixel）。

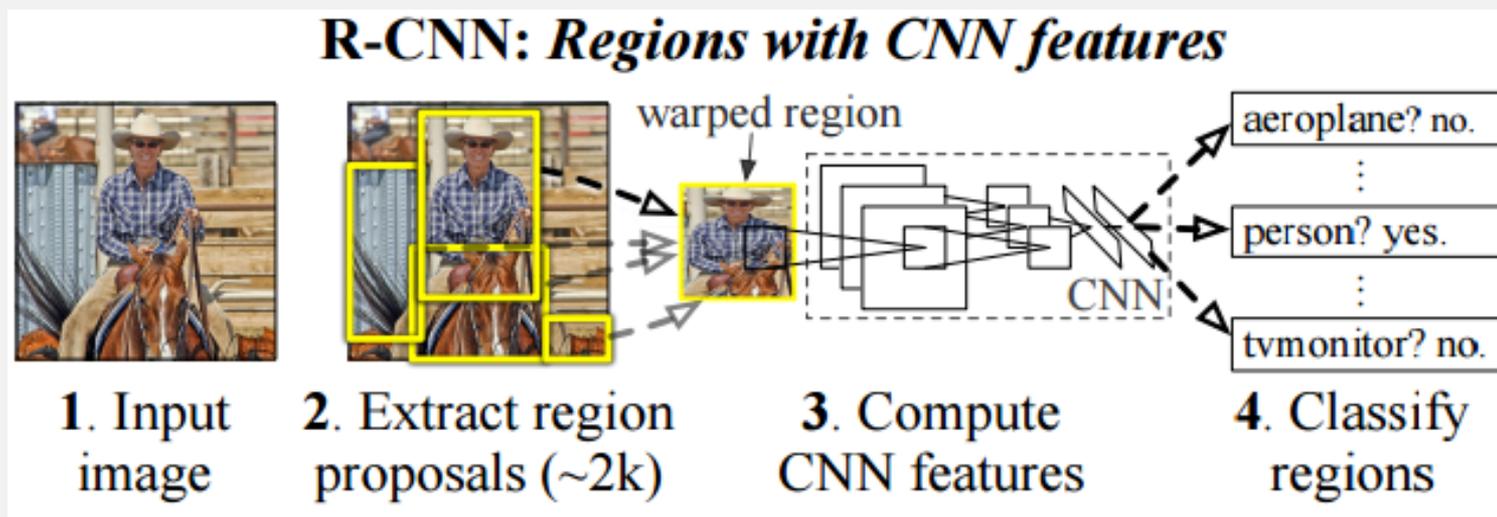


# selective search

- 以过分割结果为初始聚类，根据颜色空间距离（可以在RGB，HSV等空间上采用不同的定义方式），进行聚类的合并，直到聚类数目达到预定的数量。



## □ 算法online test的流程



- 图像中不同的物体对应的区域，提取patch，作为训练集来训练一个CNN网络；
- 但是，CNN要求输入的尺寸是固定的（例如：224\*224），而不同物体对应的patch，尺寸，高宽比例都是不同的，因此需要对它们作resize操作；
- 这里采用的是一种比较简单的做法：“图像的仿射扭曲”（affine image warping）

- 各向异性缩放：即不管图像的初始size是多少，直接resize成 $224*224$ （例如，MATLAB中的`imresize`就是这种操作）；
- 各向同性缩放：现将原始图像作边界扩展，得到一个正方形patch，在上（下）采样至 $227*227$ 的目标尺寸

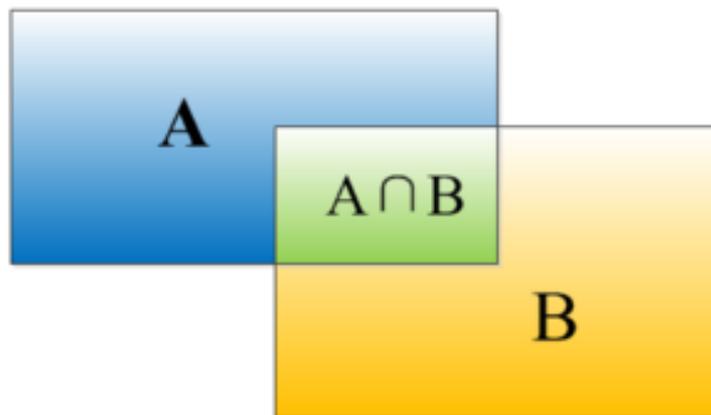


A: 原始patch; B-扩展为正方形; C-将原始patch以外的区域抹为灰色背景; D-直接采样调整到目标尺寸的效果

注: 对每张图像, 上下两行, 分别是padding取0和16的效果; 实验证明保留16的padding效果更好

- 经过上述处理，可以得到指定大小的patch，需要注意的是，初始数据中，人工标注只是标注了图像中的物体所对应的bounding box，因此，如果是从图像中采2000个patch（或者，更简单地，只是随机地采patch），一般不会恰好和bounding box重合。
- 因此我们需要用IoU为2000个bounding box打标签，以便下一步CNN训练使用。在CNN阶段，如果用selective search挑选出来的候选框与物体的人工标注矩形框的重叠区域IoU大于0.5，那么我们就把这个候选框标注成物体类别，否则我们就把它当做背景类别。

□ IoU用于定义两个bounding box的重叠程度：



矩形框A、B的一个重合度IOU计算公式为：

$$IOU=(A \cap B)/(A \cup B)$$

□ 从VOC2007中生成的训练样本集示例：



# R-CNN的训练

□ 网络的架构，这里尝试了两种经典架构：

➤ Alexnet，精度为58.5%；

➤ VGGnet，精度为66%；

□ VGG的特点是卷积核较小，精度高，但计算量也很高（约为Alexnet的7-8倍）；

（下表中BB，表示采用了参考文献<sup>[1]</sup>中的方法进行了后处理）

VOC 2007 test	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
R-CNN T-Net	64.2	69.7	50.0	41.9	32.0	62.6	71.0	60.7	32.7	58.5	46.5	56.1	60.6	66.8	54.2	31.5	52.8	48.9	57.9	64.7	54.2
R-CNN T-Net BB	68.1	72.8	56.8	43.0	36.8	66.3	74.2	67.6	34.4	63.5	54.5	61.2	69.1	68.6	58.7	33.4	62.9	51.1	62.5	64.8	58.5
R-CNN O-Net	71.6	73.5	58.1	42.2	39.4	70.7	76.0	74.5	38.7	71.0	56.9	74.5	67.9	69.6	59.3	35.7	62.1	64.0	66.5	71.2	62.2
R-CNN O-Net BB	73.4	77.0	63.4	45.4	44.6	75.1	78.1	79.8	40.5	73.7	62.2	79.4	78.1	73.1	64.2	35.6	66.8	67.2	70.4	71.1	66.0

[1] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan.

Object detection with discriminatively trained part based models. *TPAMI*, 2010.

- 参数初始化部分：物体检测的一个难点在于，物体标签训练数据少，如果直接采用随机初始化CNN参数的方法，那么训练数据量难以满足要求。
- 这种情况下，要采用特殊方法做参数初始化，然后在进行有监督的参数微调，本文采用的是有监督的预训练。
- 也就是，采用Alexnet的网络，并利用ILSVRC2012数据集训练，作为初始的参数值，然后再改变最后一个softmax分类层，做fine-tuning。
- 网络优化求解：采用SGD（随机梯度下降），学习速率大小为0.001；

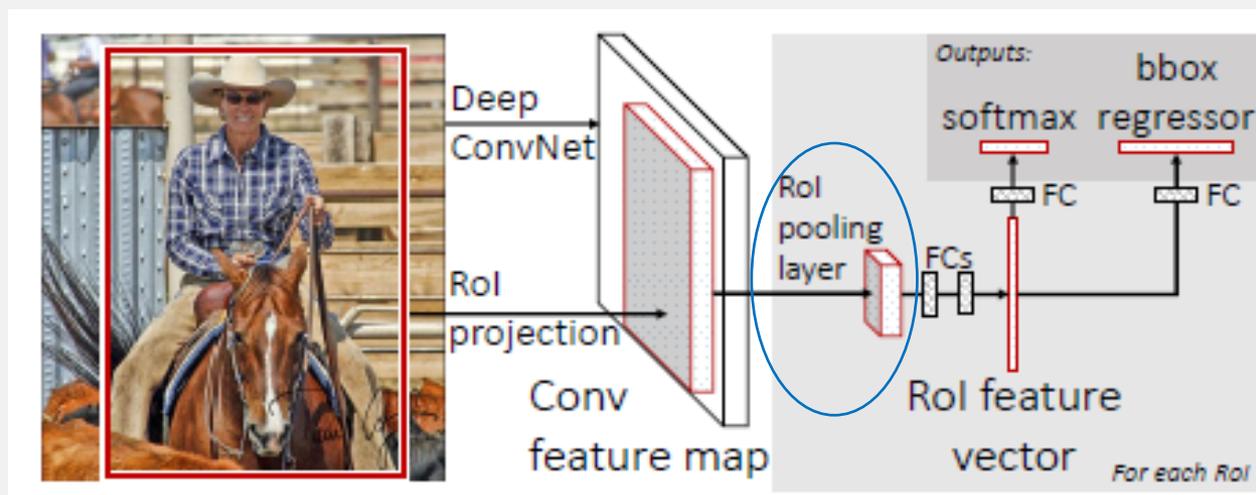
# fast R-CNN

Ross Girshick. “Fast R-CNN”. ICCV 2015.

- fast R-CNN是R-CNN的一个改进版本，大约0.3s处理一张图像，并且对VOC07数据集达到68%的mAP，。
- 为了理解fast R-CNN，首先回顾一下R-CNN的特点：
  - R-CNN的训练，pipeline是隔离的，先提取patch proposal，再训练CNN，再训练SVM，最后作bbox regression；fast R-CNN实现了end-to-end的联合训练；
  - R-CNN采用的selective search，一张图像的候选框之间有大量的重叠，输入CNN，提取特征的操作也非常冗余；fast R-CNN将整张图像归一化后直接送入网络，只在末尾少数几层处理每个候选框；

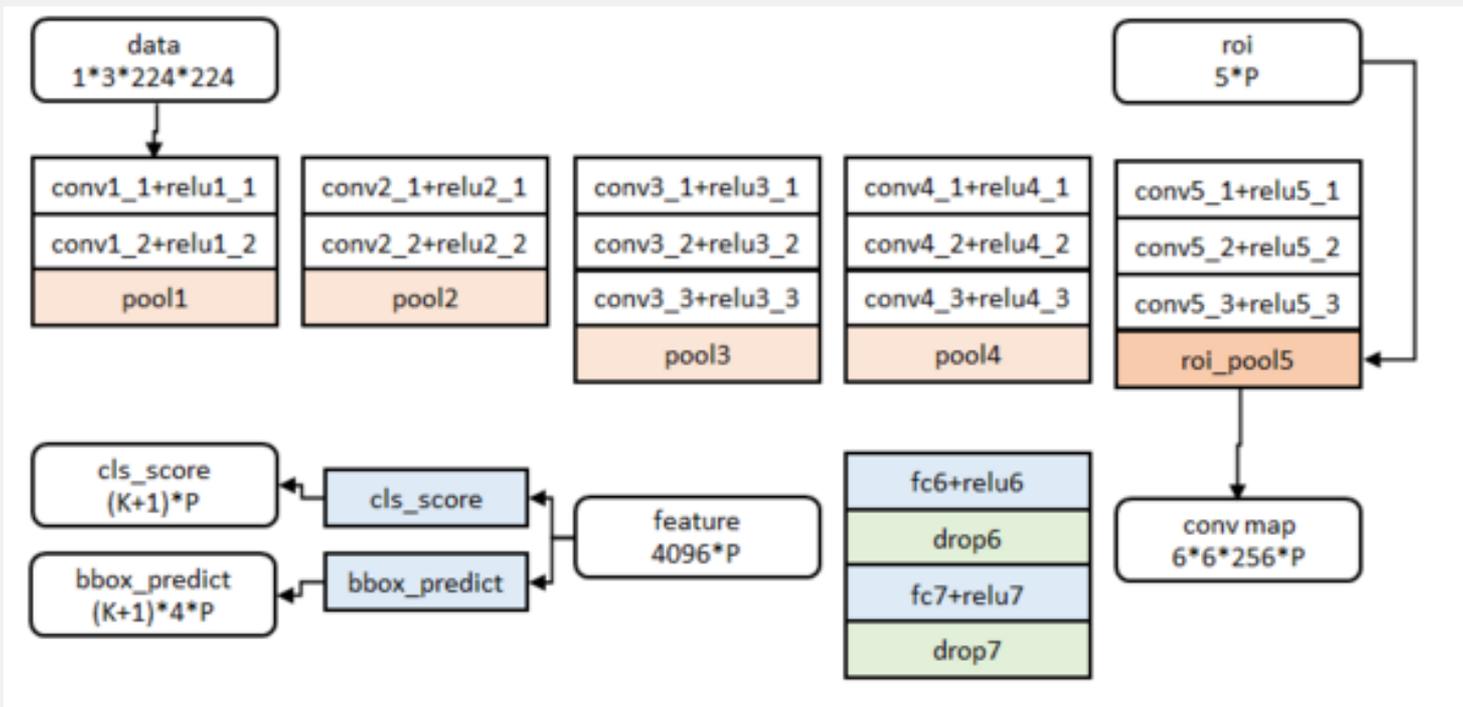
# fast R-CNN

## □ fast R-CNN的基本结构示意图



fast R-CNN的训练，是一个多任务的优化问题，即：同时将分类和回归作为网络的loss，来学习参数

## □ 较细致的网络结构示意图：



# RoI pooling layer

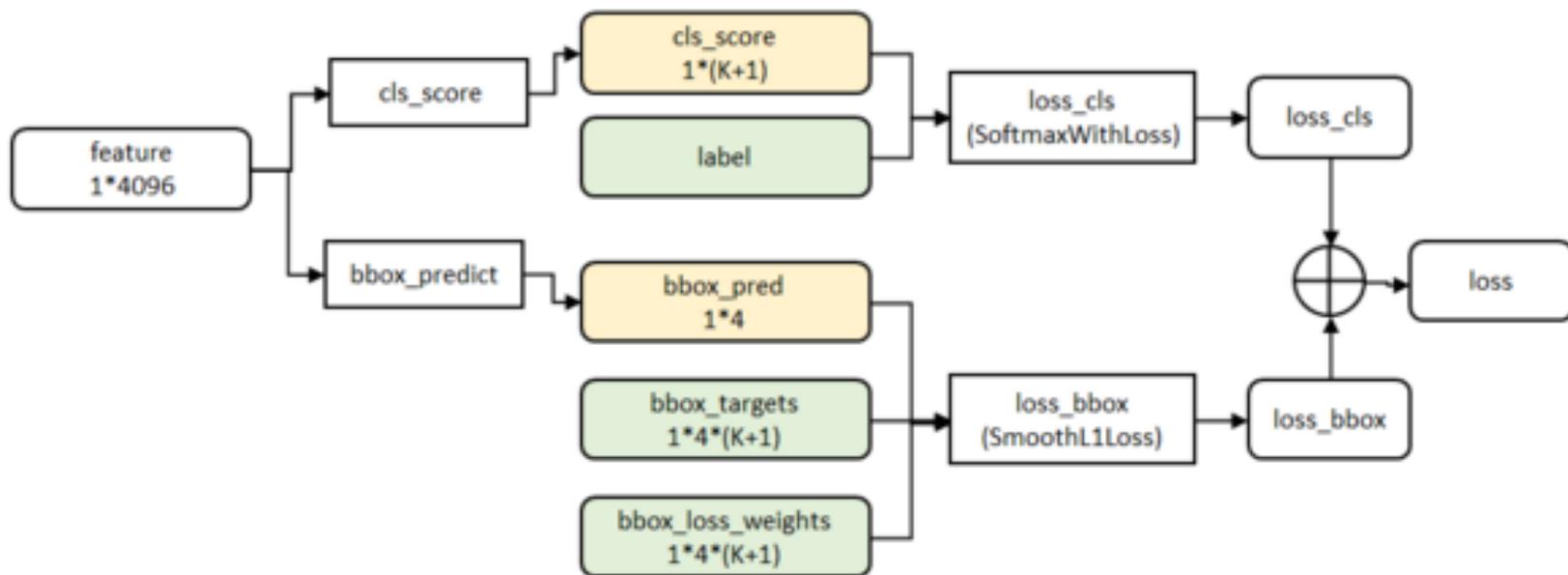
- RoI池化层，是本文工作的一个创新点。
- RoI池化层主要包括两个作用：
  - 首先是将原始image中的ROI区域定位到feature map中，找到对应的patch；
  - 随后采用max pooling操作，将patch采样到固定大小的feature；（例如，对一个 $h \times w$ 的patch，为了将其采样至固定的尺寸 $H \times W$ ，则采用 $\frac{h}{H} \times \frac{w}{W}$ 的最大池化）

# Multi-task loss

网络的输出包含两个部分，一个softmax层输出分类结果，一个regressor层输出bounding box的回归结果。

记softmax的输出为 $p = (p_0, \dots, p_N)$ ，每一个训练的RoI对应一个人工标注的ground truth类别 $u$ ，和一个bounding-box回归的目标向量 $v$ 。总的loss为两个loss之和：

$$L(p, u, t^u, v) = L_{cls}(p, u) + \lambda[u \geq 1]L_{loc}(t^u, v)$$



- 分类的loss，即softmax通用的负对数似然：  

$$L_{cls}(p, u) = -\log p_u$$

# Multi-task loss

- 回归loss, 记训练集中人工标注的bounding box为  $v = (v_x, v_y, v_w, v_h)$ , 而网络的regressor输出为  $t^u = (t_x^u, t_y^u, t_w^u, t_h^u)$

$$L_{loc}(t^u, v) = \sum_{i \in \{x, y, w, h\}} \text{smooth}_{L_1}(t_i^u - v_i)$$

其中,

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

# fast R-CNN: 全连接层的加速

□ 通常对于CNN网络，最后的softmax层是通过全连接层来计算的，R-CNN类网络同样如此。假设全连接层输入  $x \in R^v$ ，输出  $y \in R^u$ ，全连接层的参数对应矩阵  $W \in M_{u \times v}$ ，

$$\text{即: } y = Wx;$$

这个矩阵与向量的乘法复杂度为  $u \times v$ ;

如果将  $W$  作奇异值分解，并且只保留前面  $t$  个特征值近似，则：

$$W = U\Sigma V^T \approx U(:, 1:t)\Sigma(1:t, 1:t)V(:, 1:t)^T$$

那么，可以将原来的乘法分为两步：

$$y = Wx = U(\Sigma V^T)x = Uz$$

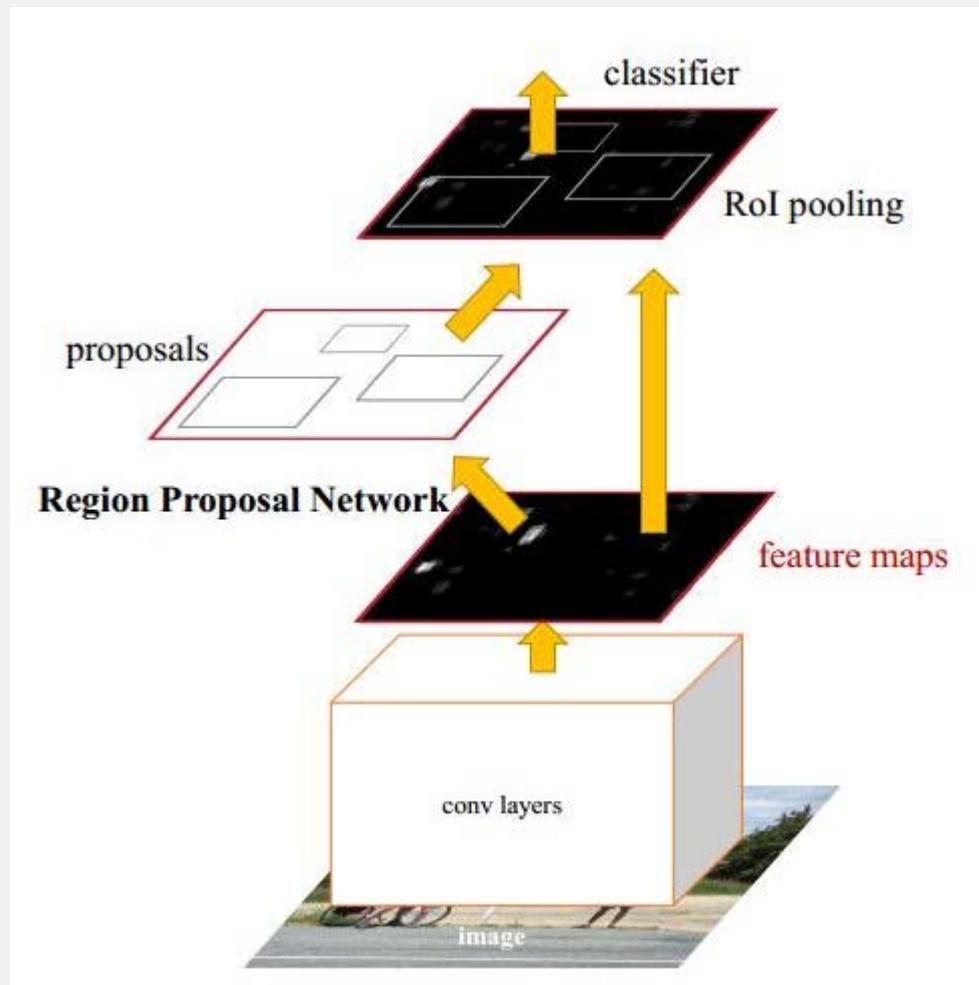
复杂度变为  $t \times v + u \times t$

# faster R-CNN

Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". NIPS 2015.

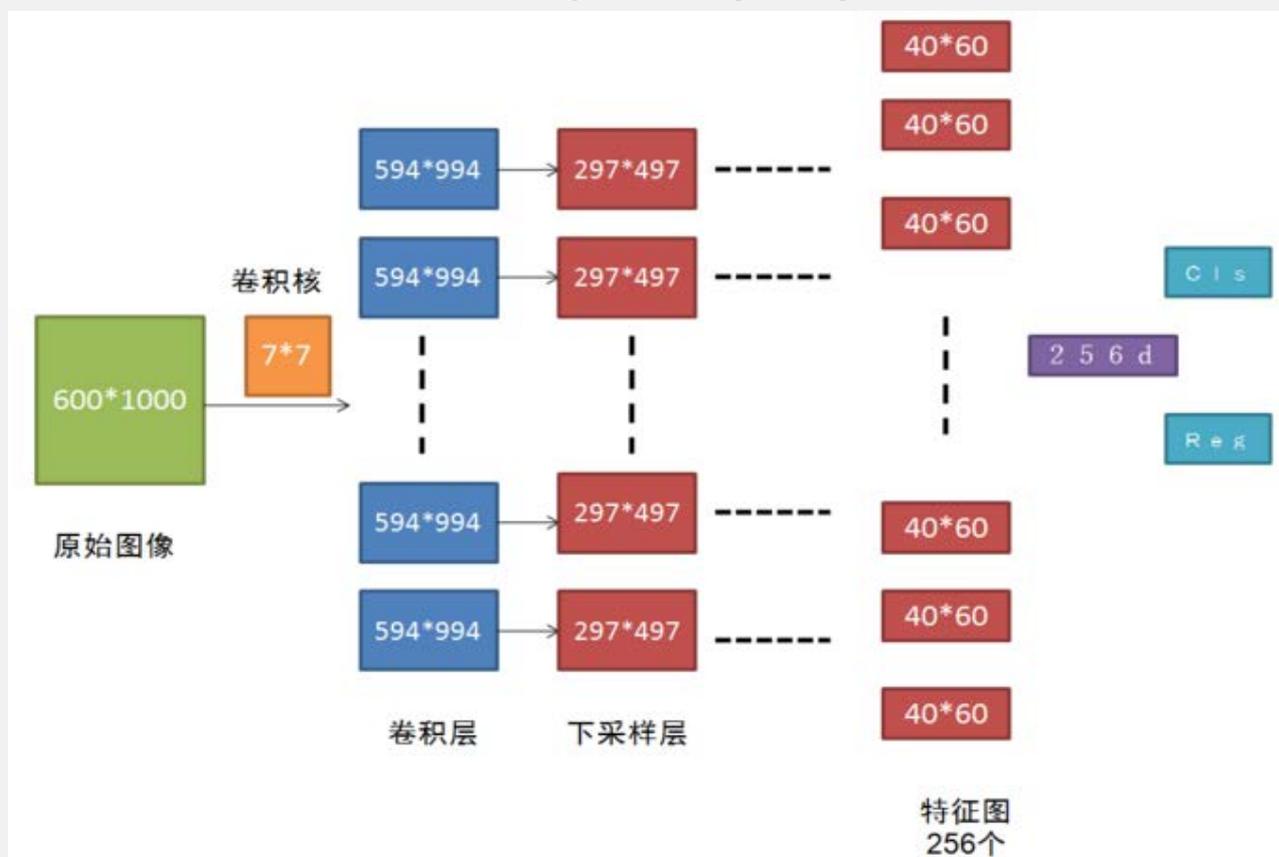
- 回顾fast R-CNN，相比于R-CNN，它的主要优势在于将训练过程整合在一起，而非CNN和SVM分别训练，因此提升了效率；
- 那么fast R-CNN在效率方面还有什么不足？
- fast R-CNN和R-CNN一样，需要预先计算region proposal，（采用的是selective search的方法）再输入到网络中，并且实验表明，这个步骤成为了限制其速度的主要瓶颈。
- 因此，faster R-CNN作为又一个改进版本，网络不但整合训练过程，同时还能够自主生成proposal。

# faster R-CNN的整体框架



# Region Proposal Network

- Region Proposal Network (RPN), 主要思想是利用卷积网络直接产生 region proposal。

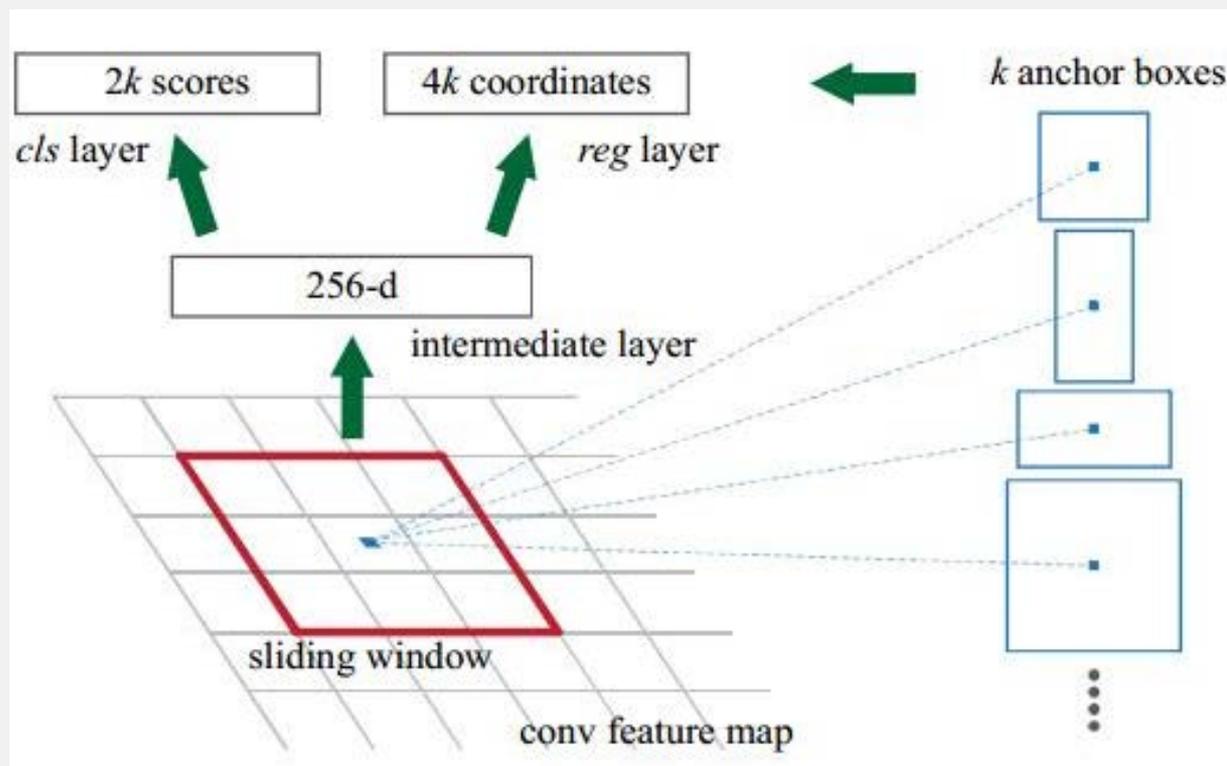


# Region Proposal Network

- RPN本身可以认为是一个fast R-CNN，它的主要作用包括：
  - (1) 输出proposal的位置(坐标)和score
  - (2) 将不同scale和ratio的proposal映射为低维的feature vector
  - (3) 输出是否是前景的classification和进行位置的regression

# Anchor

- Anchor 是本文提出的另一个重要概念，不同的  $k$  个 anchor 对应于不同  $scale$  和  $shape$  的 region proposal



# Region Proposal Network

- RPN的输出，分类输出为前景/背景概率（**注意：**RPN仅对前景背景分类，而不考虑前景的具体分类问题，具体分类仍由后续的fast R-CNN完成）。
- 而对于regression，根据anchor的数量 $k$ ，给出 $k$ 个经过校正的regressor，对应不同的ratio和scale。

# faster R-CNN的训练

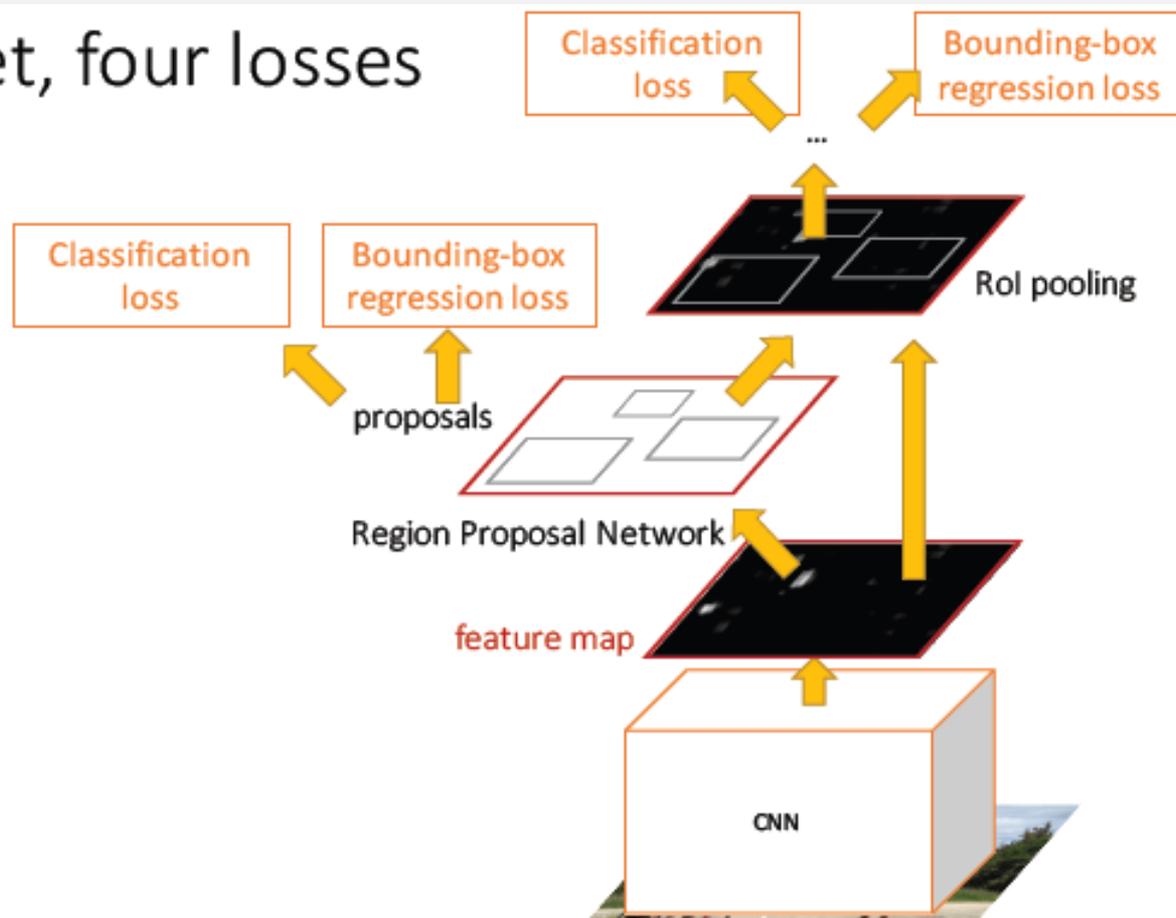
- 由于faster R-CNN是基于RPN + fast R-CNN的结构，因此，可以考虑两种不同的训练策略：
  - Alternating optimization (交替训练)
  - Approx. joint optimization (联合训练)

# Alternating optimization

- 使用在ImageNet上预训练的模型初始化RPN网络参数，微调RPN网络；
- (2) 使用(1)中RPN网络提取region proposal训练Fast R-CNN网络，也用ImageNet上预训练的模型初始化该网络参数；（现在看来两个网络相对独立）
- (3) 使用(2)的Fast R-CNN网络重新初始化RPN，固定卷积层进行微调，微调RPN网络；
- (4) 固定(2)中Fast R-CNN的卷积层，使用(3)中RPN提取的region proposal对Fast R-CNN网络进行微调。

# Joint optimization

One net, four losses



# 影响精度的因素

## R-CNN/fast R-CNN

Selective search得到的proposal尺寸不一，但是经过RoI pooling层的处理，得到了固定尺寸的特征

## faster R-CNN

RPN采用多个anchors处理特征，实际上得到了一组特征的多尺度金字塔，因此相比于selective search，精度有所提升

# R-CNN

RCNN  
(ECCV2014)

- selective search, 根据颜色, 边缘, 纹理等等快速的找到的可能存在的目标候选框

- 1.1 475张, VOC2007上的检测结果从DPM HSC的34.3%直接提升到了66%(mAP)
- 1.2 Proposal归一化到227\*227, CNN只对一个图片ROI图片的提特征, 分类还是SVM, 最终有对分类好的proposal的回归
- 1.3 问题在于每一个图像块进来都要用CNN算一下特征, 其实整张图算一次就好了

Fast RCNN  
(ICCV2015)

- 1. 加入SPPnet, end to end 训练, 使用了回归

- 1.1 3S每张, Map70%, 仍然网络外部给Proposal
- 1.2 ROI pooling: 类似于SPP, 但只有一种7\*7网格, 下采样得到49\*512维度的特征 (只有全连接层才对Size有要求)
- 1.3 损失函数使用了多任务损失函数(multi-task loss), 将边框回归直接加入到CNN网络中训练
- 1.4 SPP对任意输入的Feature Map做了金字塔Pooling: 对Map划成4\*4, 2\*2, 1\*1三种网格, 然后做pooling: 得到固定的FC输入: (16+4+1) \* channels维度

Faster RCNN  
(NIPS2015)

- 1. 使用网络直接产生召回率高的Proposals: RPN网络

- 1.1 5FPS, mAP73.2%
- 1.2 加入了9种 anchors (3种尺度, 3种比例), 总共输出20000~proposals
- 1.3 输入的特征proposal接入到ROI Pooling

YOLO  
(CVPR2016)

- 1. 变为回归问题来做

- 1.1 45FPS, mAP57.9%
- 1.2 整张图划为7\*7网格, 每一个格子预测两个目标, 输出的结果有置信度+坐标位置
- 1.3 并没有使用Region proposal, 7\*7比较粗燥, 小目标不好

SSD  
(ECCV2015)

- 1. YOLO+ Proposal + 多尺度

- 1.1 58FPS, mAP73.9%
- 1.2 整张图8\*8网格+anchors+FCN
- 1.3 不同层的feature map 3\*3滑动感受野不同, 作为不同尺度的检测

# 性能对比

方法	训练总时间	平均测试时间	VOC07 test mAP
R-CNN	84h	49s/img	66.0%
fast R-CNN	8.75h	2.32s/img	68.1%
faster R-CNN (alternating)	26.2h	0.32s/img	69.9%
faster R-CNN (joint)	17.2h	0.32s/img	70.0%