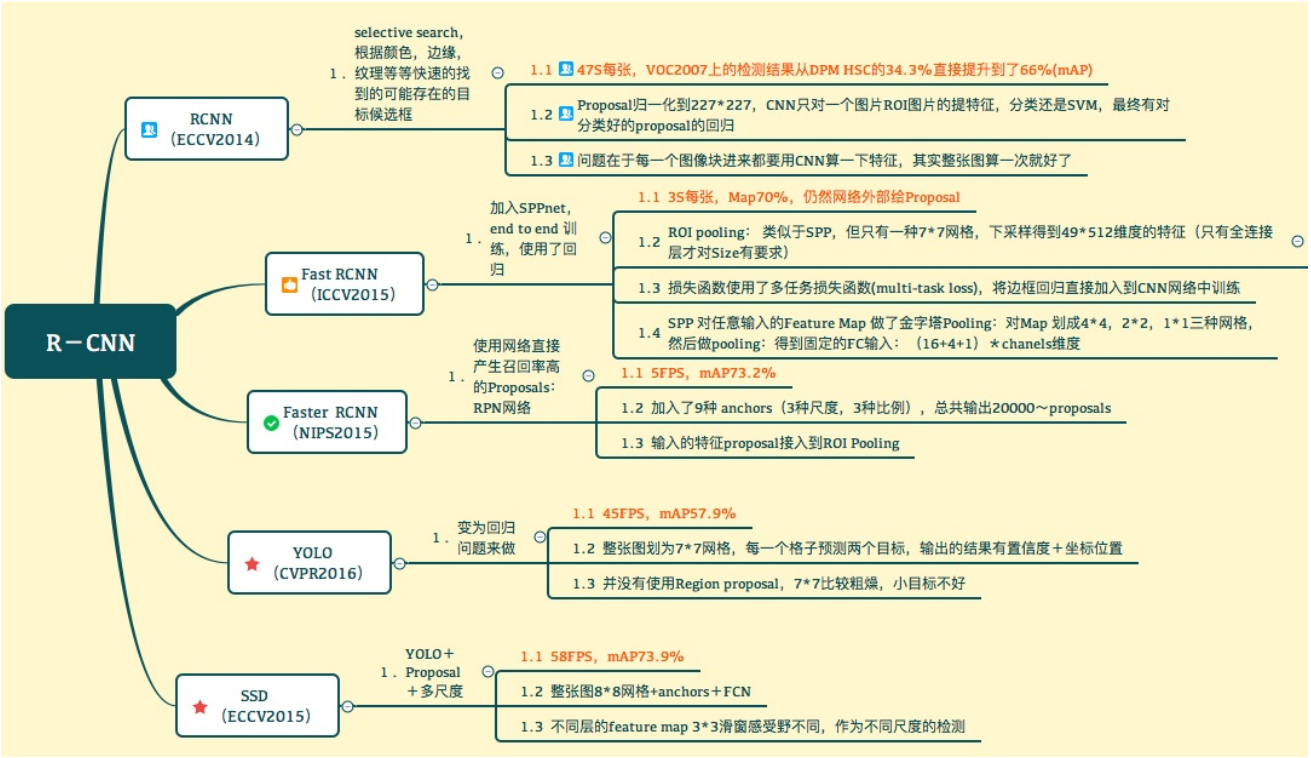


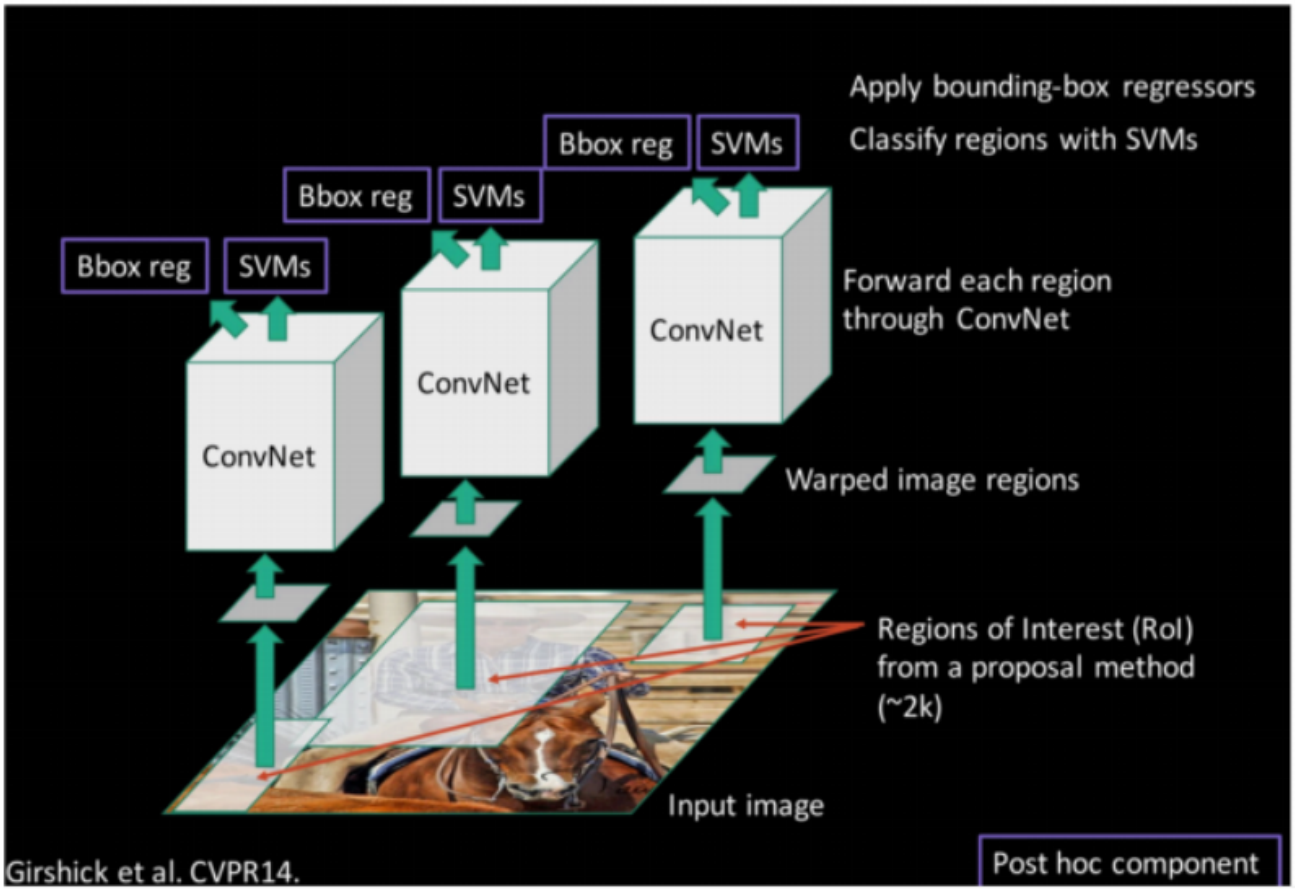
目标检测常见算法

- 传统的目标检测算法：cascade + HOG/DPM + Haar/SVM
- 候选区域 + 深度学习分类：通过提取候选区域，并对相应区域进行深度学习方法为主的分类方案。
  - R-CNN(Selective Search + CNN +SVM)
  - SPP-Net(ROI Pooling)
  - Fast R-CNN(Selective Search + CNN + ROI)
  - Faster R-CNN(RPN + CNN +ROI)
  - R-FCN
- 基于深度学习的回归方法：YOLO/SSD/DenseBox
- 传统目标检测流程：
  - 区域选择（穷举策略：采用滑动窗口，且设置不同的大小，不同的长宽比对图像进行遍历，时间复杂度高）
  - 特征提取（SIFT、HOG等；形态多样性、光照变化多样性、背景多样性使得特征鲁棒性差）
  - 分类器分类（SVM、Adaboost）

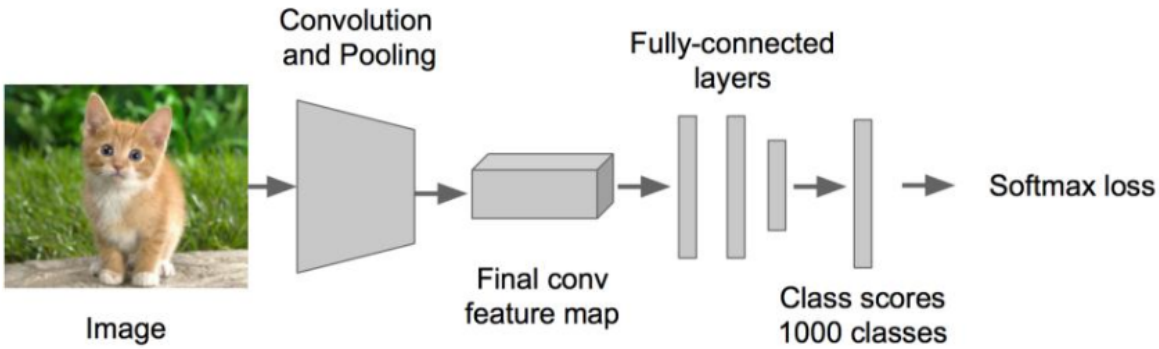
RCNN解决的就是预先找出图中目标可能出现的位置，即候选区域，再对这些区域进行识别分类。

R-CNN

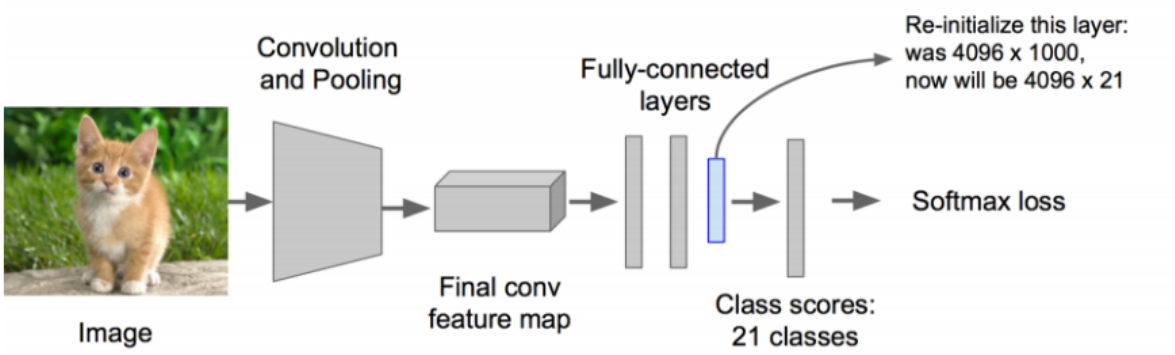
- 5个卷积层，2个全连接层



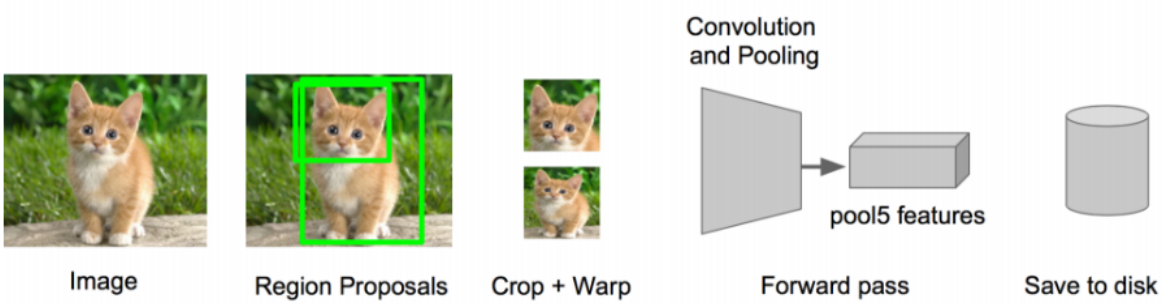
- 步骤如下
  - 输入测试图像
  - 利用Selective Search算法在图像中从下到上提取2000个左右的可能包含物体的候选区域Region Proposal。
  - 需要将取出的区域缩放成统一的227x227的大小并输入到CNN，将CNN的FC7层的输出作为特征。
  - 将每个Region Proposal提取到的CNN特征输入到SVM进行分类。
- 具体步骤如下
  - 步骤一：训练（或下载）一个分类模型(AlexNet)



  - 步骤二：对该模型做fine-tuning（微调）
    - 将分类数从1000改为20，比如20个物体类别 + 1个背景。
    - 去掉最后一个全连接层。



- 步骤三：特征提取
  - 提取图像的所有候选框（Selective Search）
  - 对于每一个区域：修正区域大小以适合CNN的输入，做一次前向运算，将第五个池化层的输出（特征）存到硬盘。

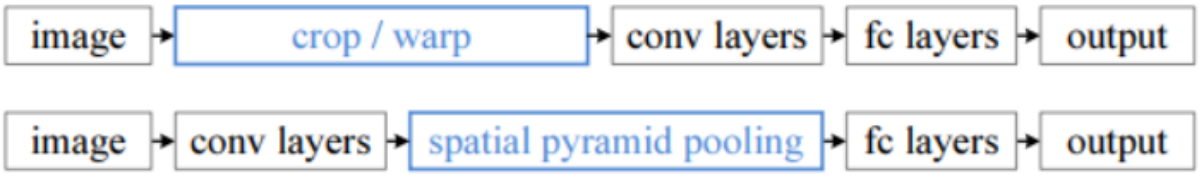


- 步骤四：训练一个SVM分类器来判断这个候选框里物体的类别。每个类别对于一个SVM，判断是不是属于这个类别。
- 步骤五：使用回归器精细修正候选框位置：对于每个类，训练一个线性回归模型取判定这个框是否框的完美。

- 为什么微调时和训练SVM时所采用的正负样本阈值【0.5和0.3】不一致?
  - 微调阶段是由于CNN对小样本容易过拟合，需要大量训练数据，故对IoU限制宽松：Ground Truth+与Ground Truth相交IoU>0.5的建议框为正样本，否则为负样本；
  - SVM这种机制是由于其适用于小样本训练，故对样本IoU限制严格：Ground Truth为正样本，与Ground Truth相交IoU < 0.3的建议框为负样本。
- 为什么不直接采用微调后的AlexNet CNN网络最后一层SoftMax进行21分类而是用了SVM?
  - 发现有时候，SVM分类的精度确实比softmax高，但是加入了SVM就导致了失去了端到端的训练，得不偿失。
- R-CNN缺点：
  - 对每个区域进行卷积计算，计算时间长。（Fast R-CNN解决）
    - 解决方法：共享卷积层，输入一张完整图片，在第五给我卷积层再得到每个候选框的特征。
  - 将所有图片缩放成统一大小，会对图片造成失真。（SPP-Net解决）

SPP Net

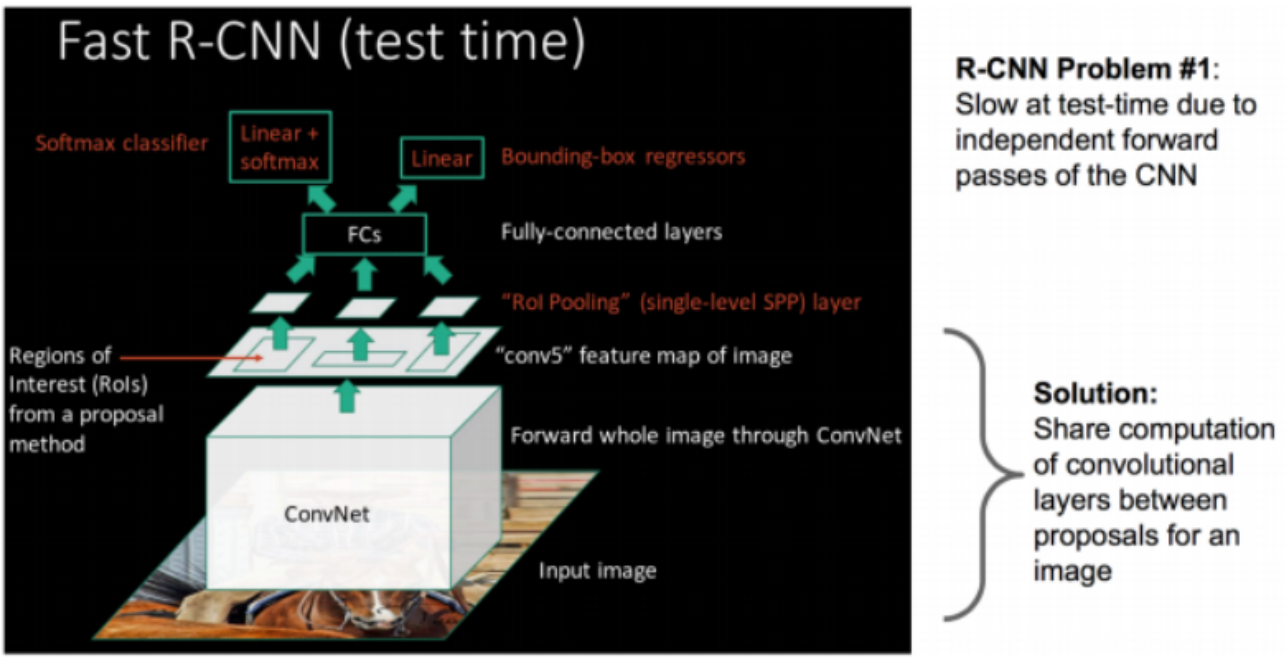
- SPP：空间金字塔池化
- conv1->pool->conv1->pool->.....->conv5->ROI Pooling->Fc->softmax
- R-CNN和SPP-Net检测流程比较：



- R-CNN在卷积层前面就对图像进行了缩放。
- SPP-Net是在卷积完成之后对图像进行缩放。
- 特点
  - 结合空间金字塔方法实现CNNs的多尺度输入。在CNN结构中最后一个卷积层后加入了ROI池化层，使得网络的输入图像可以是任意尺寸的，输出则不变。
  - 只对原图提取一次卷积特征，便得到整张图的卷积特征feature map，然后找到每个候选框在feature map上的映射patch，将此patch作为每个候选框的卷积特征输入到SPP layer和之后的层，完成特征提取工作。

Fast R-CNN

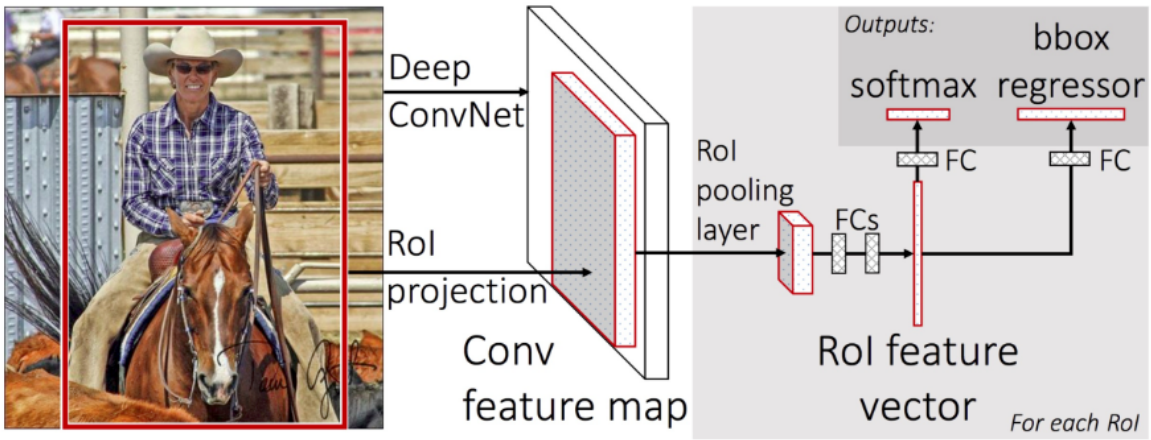
- R-CNN对所有region进行特征提取时会有重复计算。



- 与R-CNN不同点
  - 一是在最后一个卷积层后加入了一个ROI pooling layer。
  - 损失函数使用多任务损失函数（multi-task loss），将边框回归Bounding Box Regression直接加入到CNN网络中训练。
    - （1）ROI pooling layer实际是SPP-Net的精简版，SPP-Net对每个proposal使用了不同大小的金字塔映射，而ROI pooling layer只需要下采样到一个7x7的特征图。对于VGG16网络conv5_3有512个特征图，这样所有region proposal对应了一个7 * 7 * 512维度的特征向量作为全连接的输入。

      也就是说，这个网络层可以把不同大小的输入映射到一个固定尺度的特征向量。

- （2）R-CNN训练过程分为三个阶段，而Fast R-CNN直接使用softmax替代SVM分类，同时利用多任务损失函数边界框回归也加入到网络中，这样整个训练过程是端到端的。分类和回归合并为一个多任务模型。

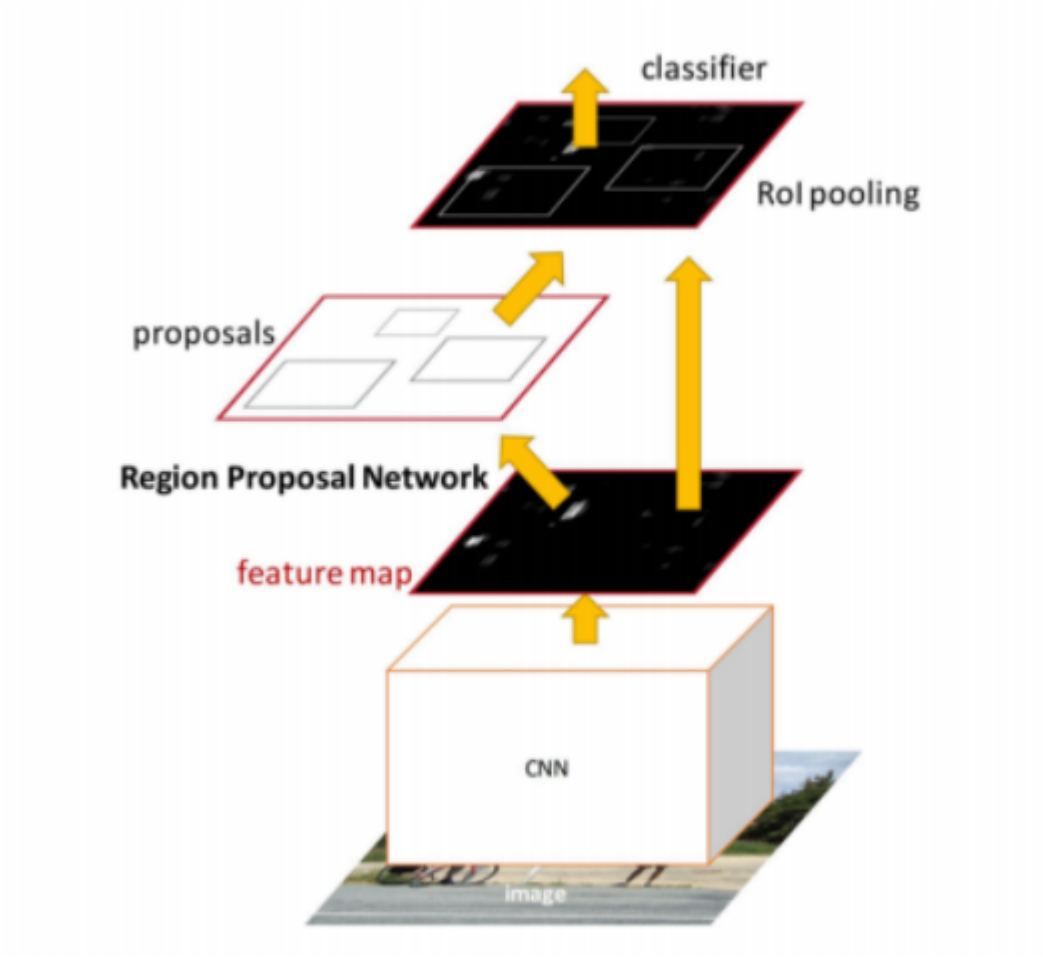


- 性能比较：

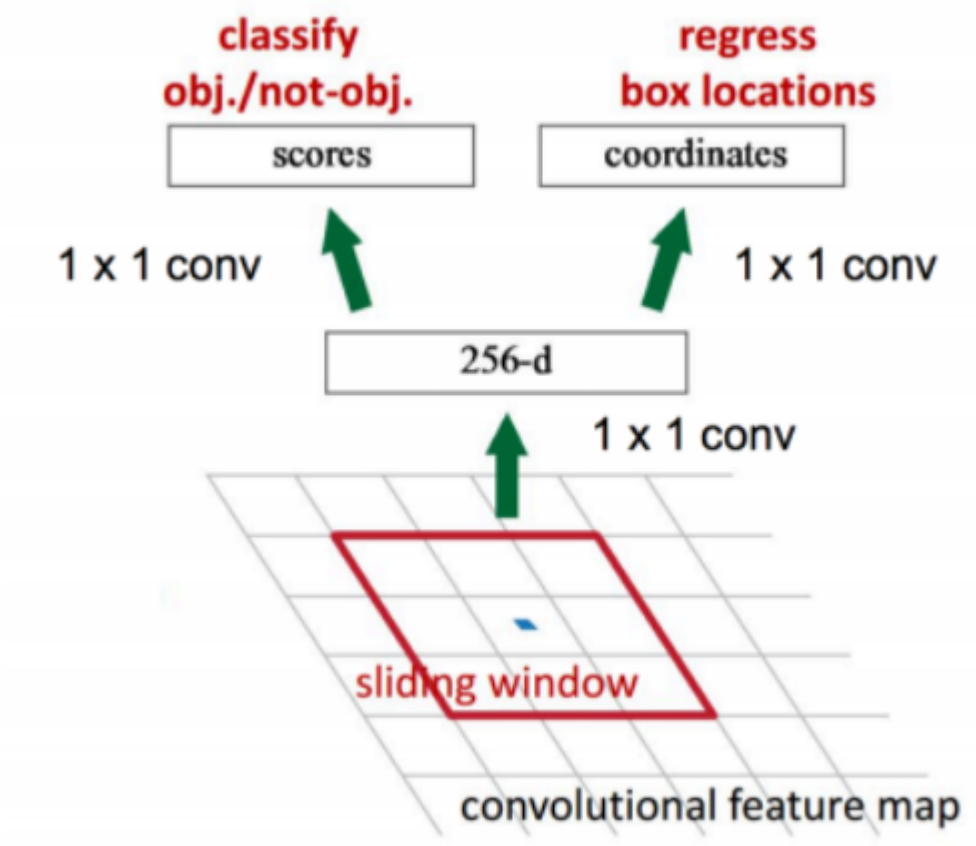
	R-CNN	Fast R-CNN
Faster!	Training Time:	84 hours	**9.5 hours**
	(Speedup)	1x	**8.8x**
FASTER!	Test time per image	47 seconds	**0.32 seconds**
	(Speedup)	1x	**146x**

Faster R-CNN

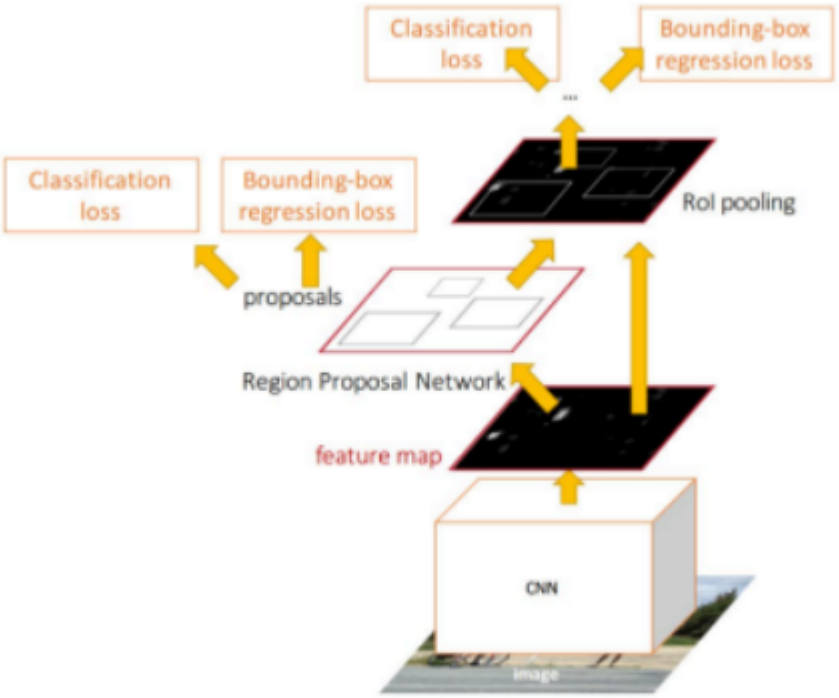
- Fast R-CNN存在的问题：
  - 选择性搜索，找出所有的候选框，非常耗时。
  - 解决：加入一个提取边缘的神经网络。
  - 在Fast R-CNN中引入Region Proposal Network（RPN）替代Selective Search，同时引入anchor box应对目标形状的变化问题（anchor是位置和大小固定的box，可以理解为事先设置好的固定的proposal）。
- 具体做法：
  - 将RPN放在最后一个卷积层后面
  - RPN直接训练得到候选区域



- RPN简介：
  - 在feature map上滑动窗口。
  - 建一个神经网络用于物体分类 + 框位置的回归。
  - 滑动窗口的位置提供了物体的大致位置信息。
  - 框的回归提供了框更精确的位置。



- 一种网络，四个损失函数：
  - RPN classification（anchor good/bad）候选区域分类
  - RPN regression（anchor->proposal）候选区域box回归
  - Fast R-CNN classification（over classes）整个对象分类
  - Fast R-CNN regression（proposal->box）回归



- 速度对比

	R-CNN	Fast R-CNN	Faster R-CNN
Test time per image (with proposals)	50 seconds	2 seconds	**0.2 seconds**
(Speedup)	1x	25x	**250x**
mAP (VOC 2007)	66.0	**66.9**	**66.9**

- Faster R-CNN的主要贡献是设计了提取候选区域的网络RPN。

各种算法的步骤

R-CNN

- 在图像中确定约1000~2000个候选框（使用Selective Search）。
- 每个候选框内图像块缩放至相同大小，并输入到CNN内进行特征提取。
- 对候选框中提取出的特征，使用分类器判别是否属于一个特定类。
- 对属于某一类别的候选框，用回归器进一步调整其位置。

Fast R-CNN

- 在图像中确定约1000~2000个候选框（使用Selective Search）。
- 对整张图片输入CNN进行卷积，得到feature map。
- 找到每个候选框在feature map上的映射patch，将此patch作为每个候选框的卷积特征输入到SPP layer（统一图片大小）和之后的层。

- 对候选框中提取的特征，使用分类器判别是否属于一个特定类。
- 对属于某一类别的候选框，用回归器进一步调整其位置。

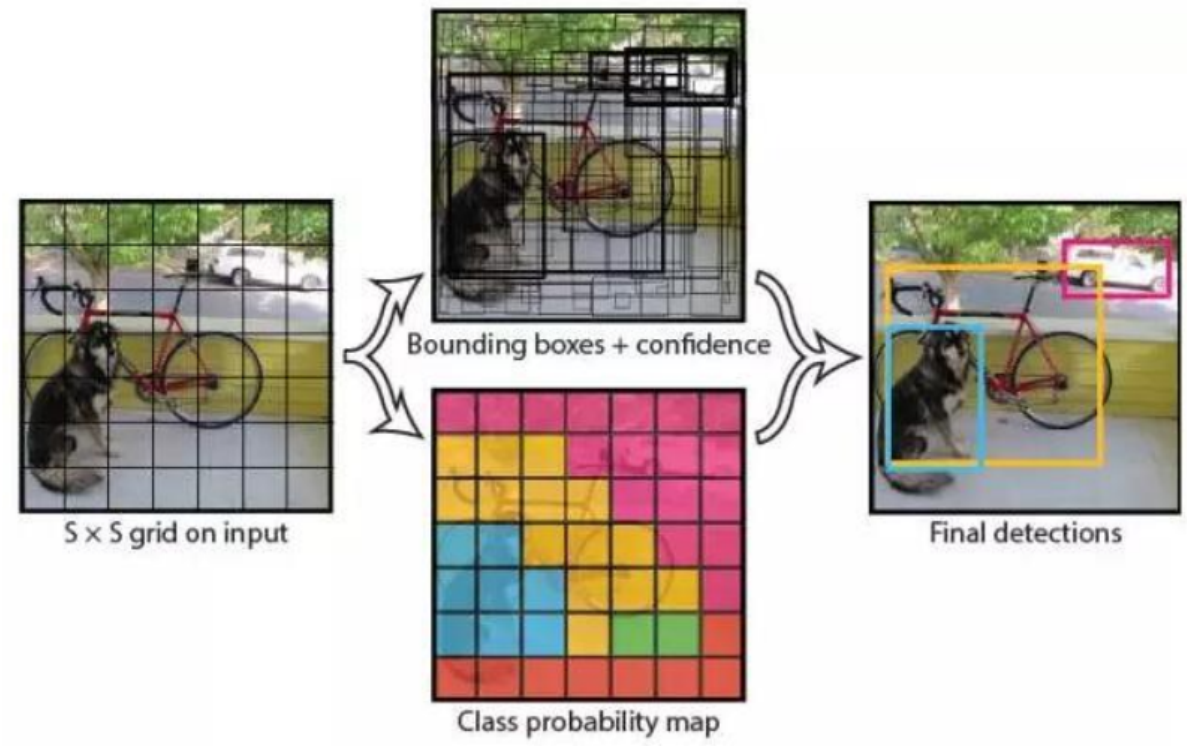
Faster R-CNN

- 对整张图片输入CNN，得到feature map。
- 卷积特征输入到RPN，得到候选框的特征信息。
- 对候选框中提取的特征，使用分类器判别是否属于一个特定类
- 对属于某一类别的候选框，用回归器进一步调整其位置。

项目	R-CNN	Fast R-CNN	Faster R-CNN
提取候选框	Selective Search	Selective Search	RPN网络
提取特征	卷积神经网络(CNN)	CNN + ROI池化	CNN + ROI池化
特征分类	SVM	CNN + ROI池化	CNN + ROI池化

YOLO

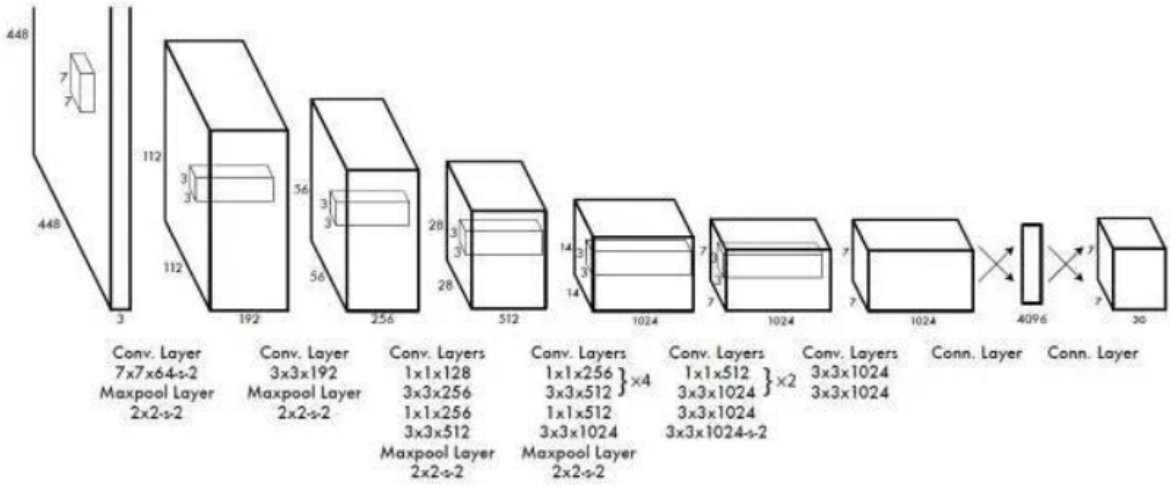
- 方法使用了回归的思想，利用整张图作为网络的输入，直接在图像的多个位置上回归出这个位置的目标边框，以及目标所属的类别。



YOLO目标检测流程图

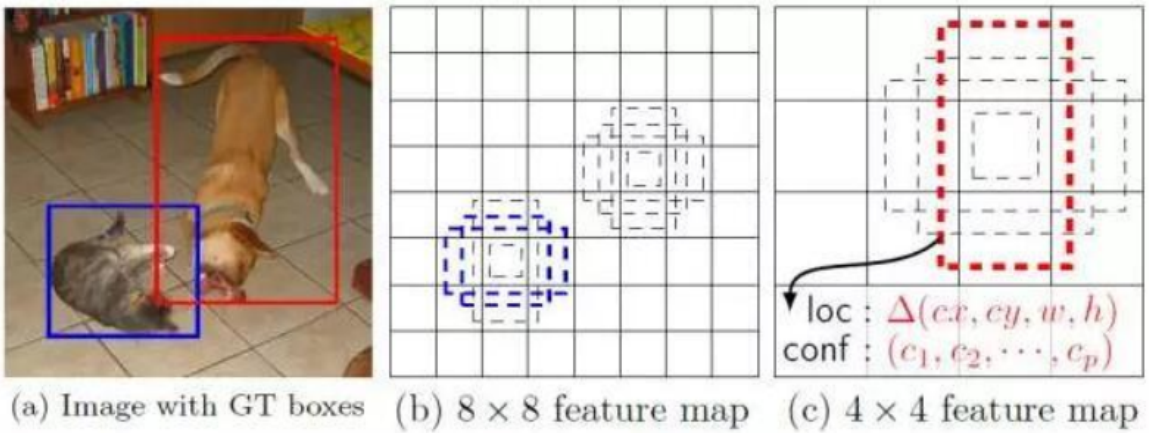
- 流程
  - 给定一个输入图像，首先将图像划分成7*7的网格。
  - 对于每个网格，我们都预测2个边框（包括每个边框是目标的置信度以及每个边框区域在多个类别上的概率）。

- 根据上一步可以预测出`7*7*2`个目标窗口，然后根据阈值去除可能性比较低的目标窗口，最后非极大值抑制（NMS）去除冗余窗口。



- 存在问题：没有Region Proposal机制，只使用7*7的网格回归会使目标不能非常精准定位。

SSD



- 首先获取目标位置和类别的方法跟YOLO一样，都是使用回归。但是YOLO预测某个位置使用的是全图特征，SSD预测某个位置使用的是这个位置位置周围的特征。使用3*3的滑动窗口提取每个位置特征，这个特征回归得到目标的坐标信息和类别信息。
- 不同于Faster R-CNN，这个anchor是在多个feature map上，这样可以利用多层的特征并且自然的达到多尺度。