



基于R-CNN的系列目标检测算法

R-CNN, SPP NET, Fast R-CNN, Faster R-CNN

CONTENTS 目录

- 01 背景知识
- 02 RCNN
- 03 SPP NET
- 04 Fast R-CNN
- 05 Faster R-CNN
- 06 实例展示

01

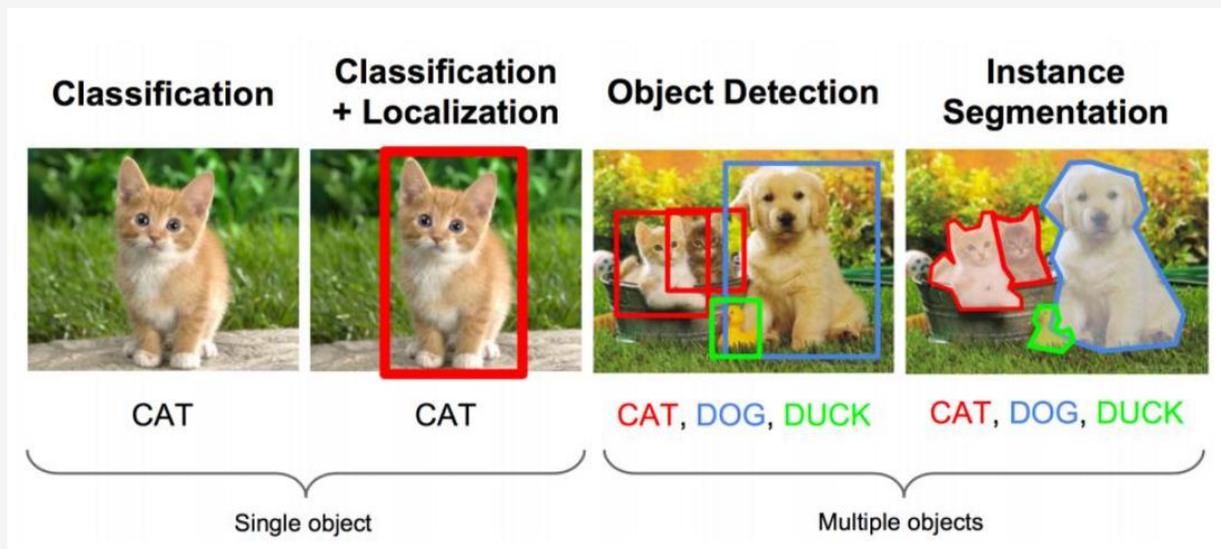
PART ONE

背景知识

背景知识

目标检测

目标检测是在给定的图片中精确找到物体所在位置，并标注出物体的类别。物体的尺寸变化范围很大，摆放物体的角度，姿态不定，而且可以出现在图片的任何地方，并且物体还可以是多个类别。



背景知识

图像识别 (classification) :

输入: 图片

输出: 物体的类别

评估方法: 准确率。



定位 (localization) :

输入: 图片

输出: 方框在图片中的位置
(x, y, w, h)

评估方法: 检测评价函数
intersection-over-union



背景知识

目标检测算法

1. 传统的目标检测算法：Cascade + HOG/DPM + Haar/SVM以及上述方法的诸多改进、优化；
2. 候选区域/窗 + 深度学习分类：通过提取候选区域，并对相应区域进行以深度学习方法为主的分类的方案，如：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等系列方法；
3. 基于深度学习的回归方法：YOLO/SSD/DenseBox 等方法；以及最近出现的结合RNN算法的RRC detection；结合DPM的Deformable CNN等。

02

PART TWO

R-CNN



R-CNN

传统目标检测方法

传统目标检测的算法基本流程如下：

- 使用不同尺度的滑动窗口选定图像的某一区域为候选区域；
- 从对应的候选区域提取如Harr HOG LBP LTP等一类或者多类特征；
- 使用Adaboost、SVM 等分类算法对对应的候选区域进行分类，判断是否属于待检测的目标。

传统目标检测方法的缺点

- 1、基于滑动窗口的区域选择策略没有针对性，时间复杂度高，窗口冗余；
- 2、手工设计的特征对于多样性的变化没有很好的鲁棒性。



R-CNN

R-CNN是Region-based Convolutional Neural Networks的缩写，中文翻译是基于区域的卷积神经网络，是一种结合区域提名（Region Proposal）和卷积神经网络（CNN）的目标检测方法。

区域提名（Region Proposal）：利用图像中的纹理、边缘、颜色等信息，预先找出图中目标可能出现的位置。需要解决的问题：（1）适应不同尺度 （2）多类别图像的适应性 （3）速度。

R-CNN的主要贡献：

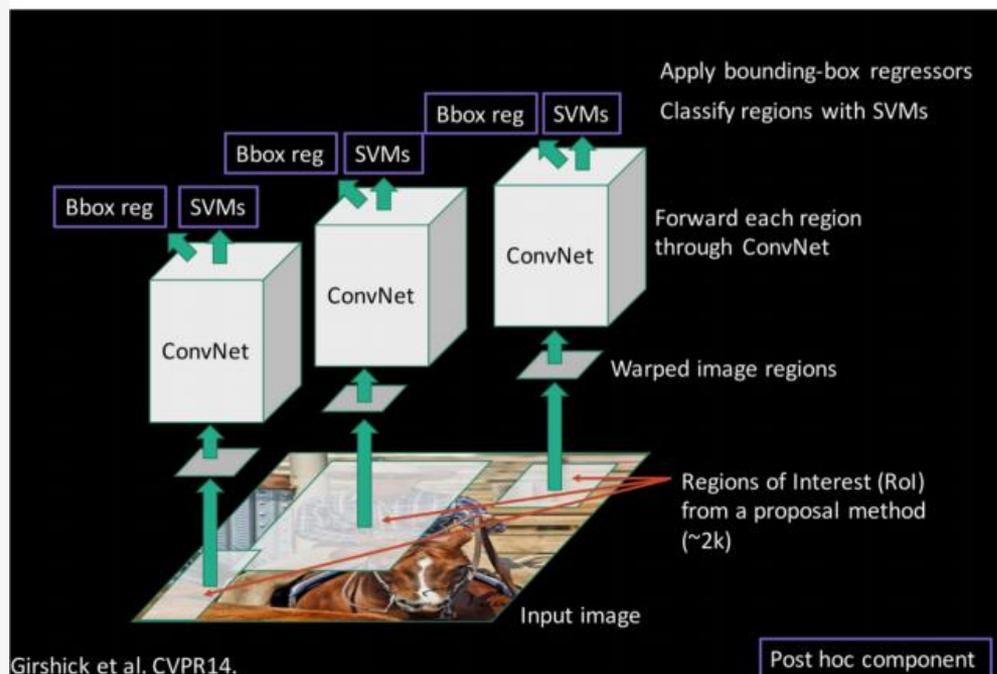
- 1) 传统目标检测算法一般使用滑动窗口扫描所有可能区间，同时需要考虑变化窗口尺寸以适应不同大小的目标，这种方法效率太低。R-CNN使用Selective Search方法预先提取所有候选区域；
- 2) 传统目标检测算法依赖人工设计的特征，R-CNN使用深度学习自动提取和学习特征。

R-CNN

R-CNN的总体思路

R-CNN的简要步骤如下：

- (1) 输入测试图像；
- (2) 利用选择性搜索（ 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；
- (3) 因为取出的区域大小各自不同，所以需要将每个候选区域缩放（warp）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；
- (4) 将每个候选区域提取到的CNN特征输入到SVM进行分类。



R-CNN

Selective Search算法

1. 使用 Efficient Graph-Based Image Segmentation的方法获取原始分割区域

$$R = \{r_1, r_2, \dots, r_n\}.$$

区域内间距区域为对应最小生成树中权重最大的边的权重值。区域间间距即在所有分别属于两个区域且有边连接的点对中，寻找权重最小的那对（若两个区域内的点没有边相连，则定义间距为正无穷大）。

2. 初始化相似度集合 $S = \emptyset$ 。
3. 计算两两相邻区域之间的相似度将其添加到相似度集合 S 中。

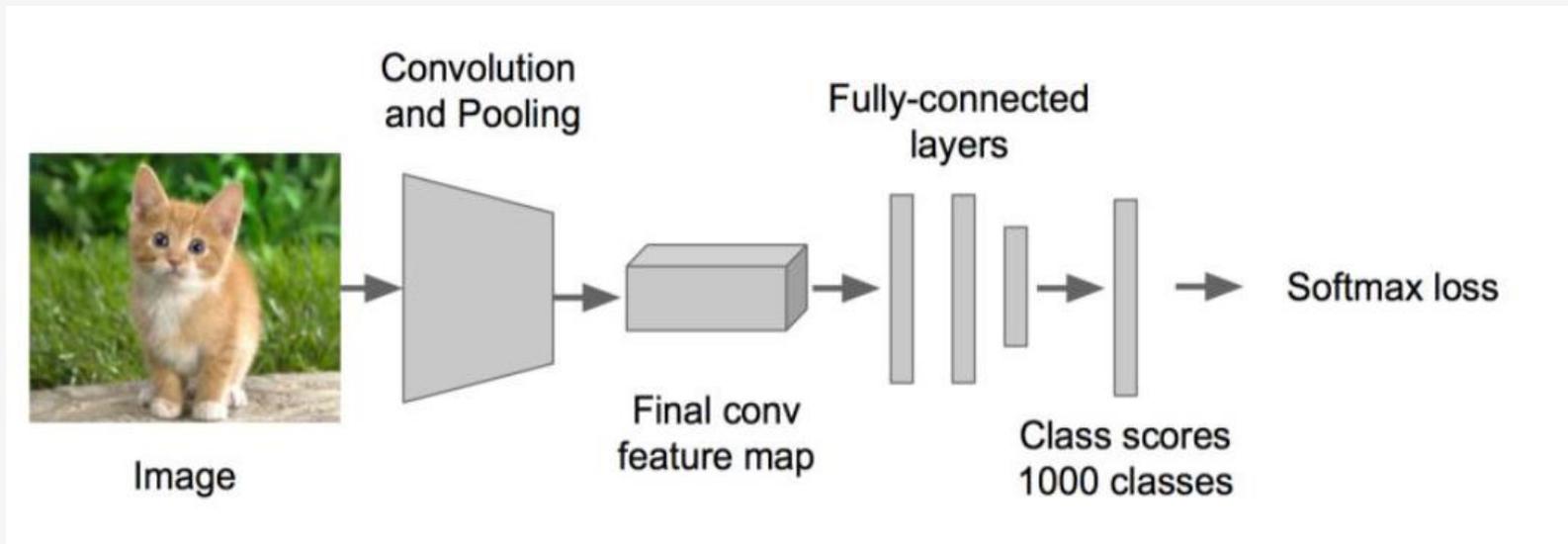
$$s(r_i, r_j) = a_1 S_{colour}(r_i, r_j) + a_2 S_{texture}(r_i, r_j) + a_3 S_{size}(r_i, r_j) + a_4 S_{fill}(r_i, r_j)$$

4. 从相似度集合 S 中找出，相似度最大的两个区域 r_i 和 r_j ，将其合并成为一个区域 r_t ，从相似度集合中除去原先与 r_i 和 r_j 相邻区域之间计算的相似度，计算 r_t 与其相邻区域（原先与 r_i 或 r_j 相邻的区域）的相似度，将其结果添加的到相似度集合 S 中。同时将新区域 r_t 添加到区域集合 R 中。
5. 获取每个区域的 Bounding Boxes，这个结果就是物体位置的可能结果 L 。

R-CNN

RCNN的具体步骤

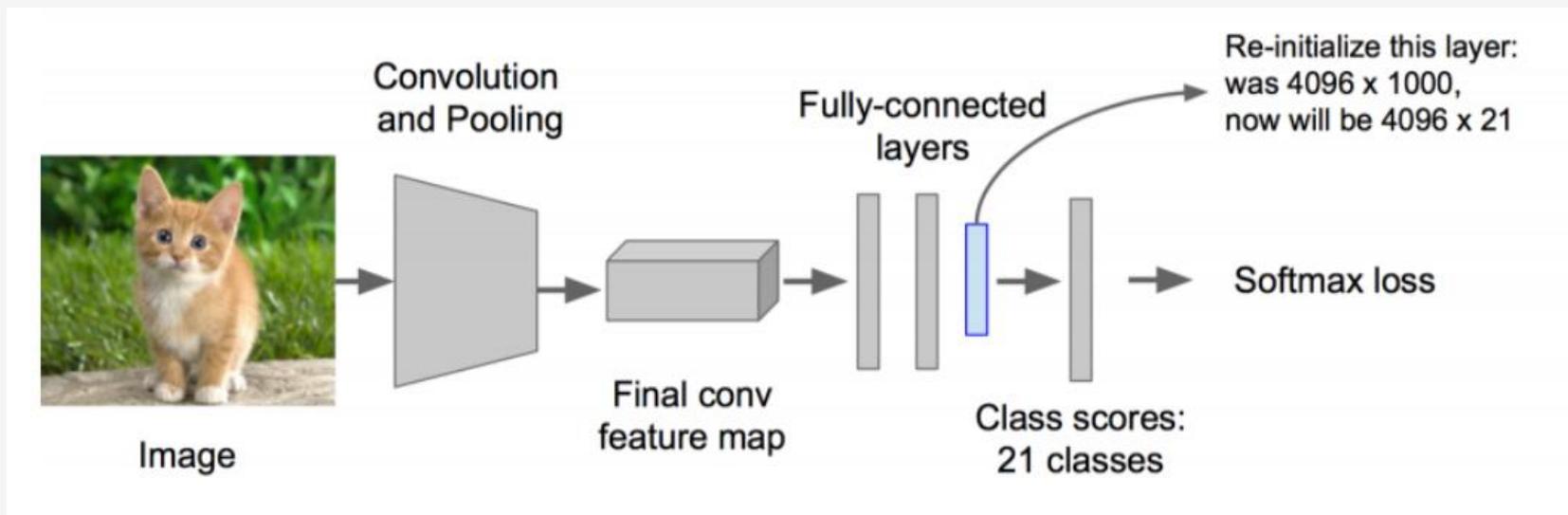
步骤一：训练（或者下载）一个分类模型（比如AlexNet）



R-CNN

步骤二：对该模型做fine-tuning

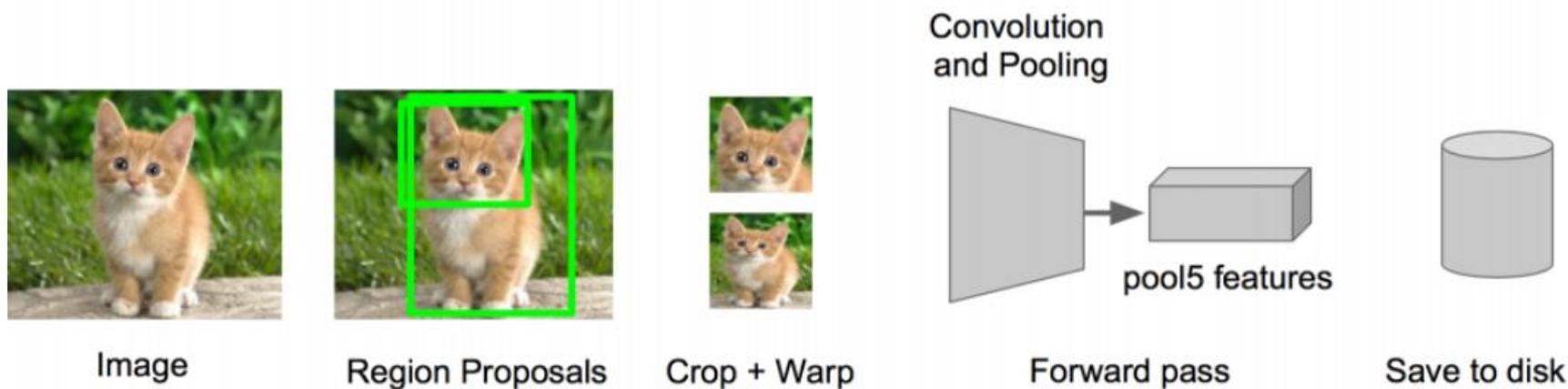
- 将分类数从1000改为20，比如20个物体类别 + 1个背景
- 去掉最后一个全连接层



R-CNN

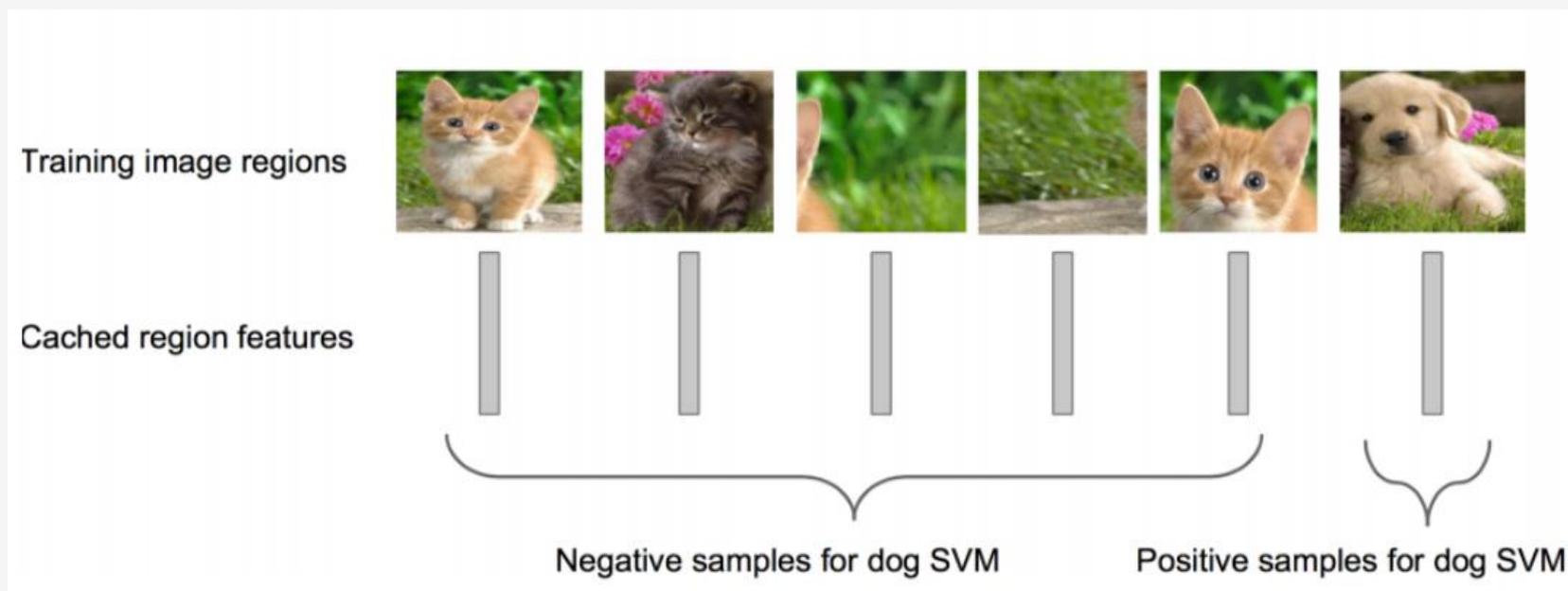
步骤三：特征提取

- 提取图像的所有候选框（选择性搜索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（就是对候选框提取到的特征）存到硬盘。



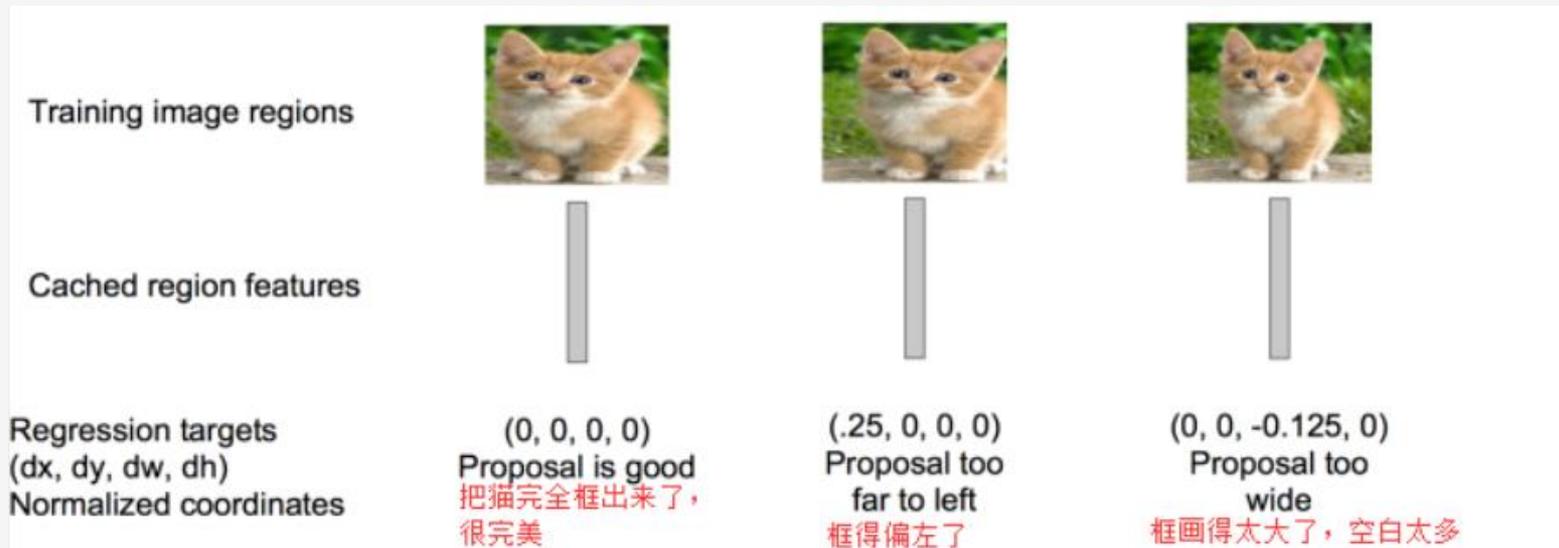
R-CNN

步骤四：训练一个SVM分类器（二分类）来判断这个候选框里物体的类别
每个类别对应一个SVM，判断是不是属于这个类别，是就是positive，反之negative。
比如下图，就是对狗分类的SVM



R-CNN

步骤五：使用回归器精细修正候选框位置：对于每一个类，训练一个线性回归模型去判定这个框是否框得完美



R-CNN

R-CNN存在的问题

- 1、基于R-CNN目标检测算法只能输入固定尺寸的图片，样本输入受限
- 2、经人工处理过的图片，易降低网络识别检测精度
- 3、R-CNN需对各候选区域进行一次卷积操作，计算量大，耗时长



人工图片处理样例

03

PART THREE

SPP NET



SPP NET

在R-CNN的第一步中，对原始图片通过Selective Search提取的候选框多达2000个左右，而这2000个候选框每个框都需要进行CNN提特征+SVM分类，计算量很大，导致R-CNN检测速度很慢，一张图都需要47s。

而且，基于R-CNN目标检测算法只能输入固定尺寸的图片，样本输入受限，使用很不方便。那么如何改进呢？SPP-NET的出现恰好解决了这些问题。

SPP-Net (Spatial Pyramid Pooling) 是何凯明2014年提出的方法，通过解决传统CNN无法处理不同尺寸输入的问题对同年的R-CNN算法做改进，实验结果表明SPP方法比R-CNN快了近100倍

从算法架构上，SPP-Net与R-CNN相似：通过Selective Search获取候选区域，最后也是使用SVM做分类。

但不再将每个候选区域过一次CNN，而是将原始图过一次CNN，在CNN的全连接层前添加新提出的SPP层，根据候选区域位置crop的图像卷积结果通过SPP层来确保输入全连接层的尺寸满足要求。最后在全连接层的输出一次性获得所有候选区域的特征向量。



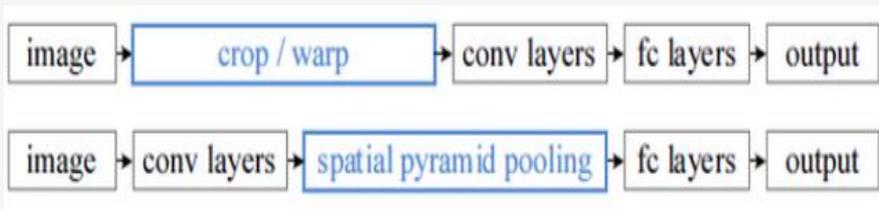
SPP NET

SPP NET的原理

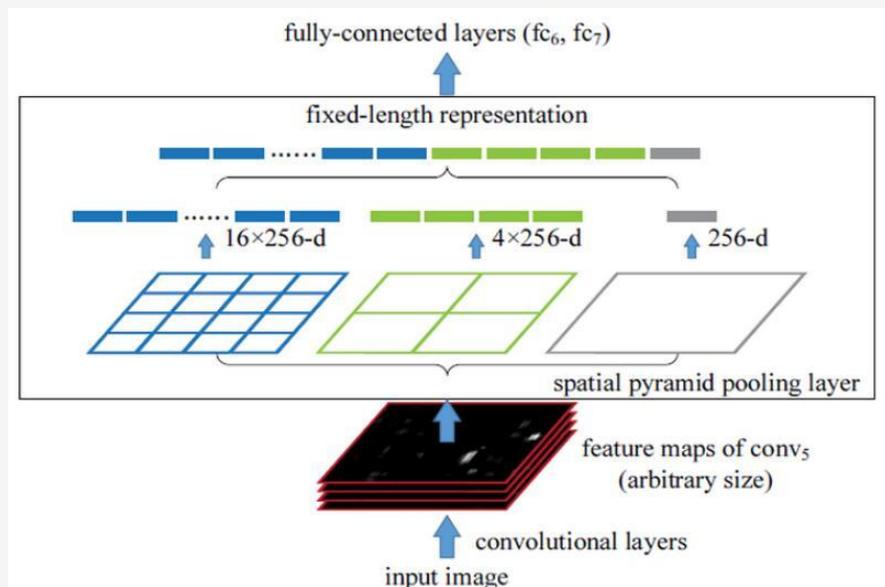
可允许不同尺寸图片输入，将R-CNN最后一个卷积层后的池化层替换为SPP层，生成长度固定的特征，并输入到最后的全连接层中。

特点:

- 1、可实现CNN多尺度图像的输入;
- 2、只对原图进行一次卷积特征提取。



SPP Net vs R-CNN



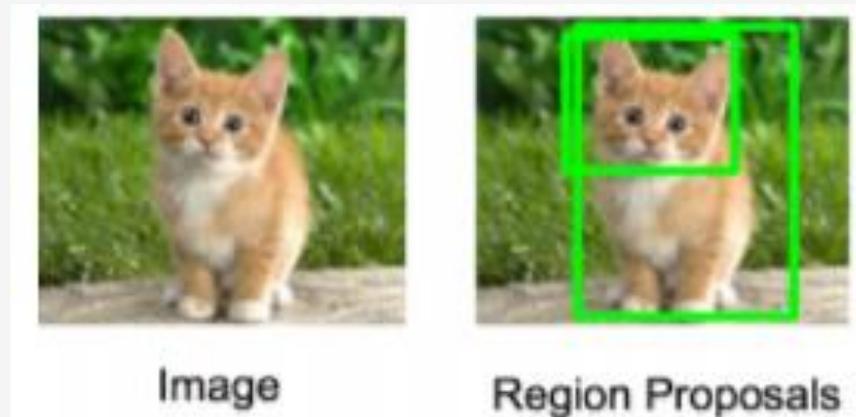
SPP Net结构

SPP NET

SPP NET的具体步骤

步骤一：选择性搜索

对待检测图片，选择性搜索出2000个候选框

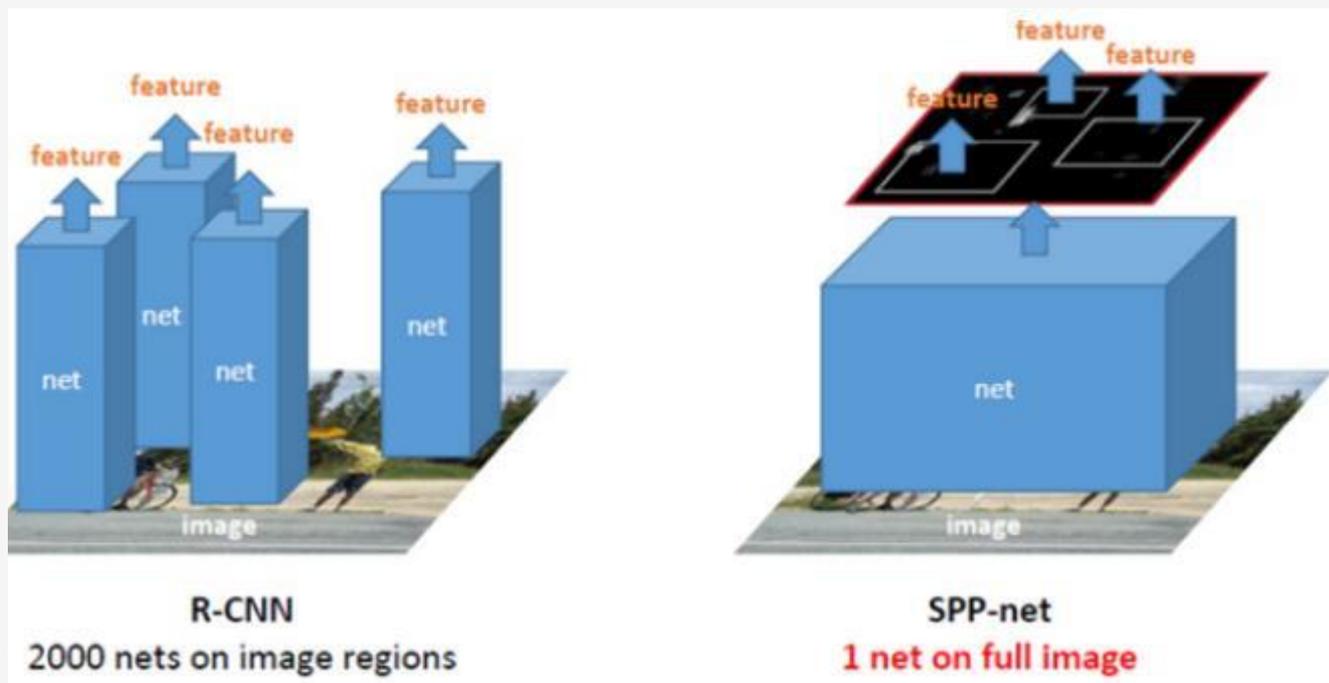


候选区域提取

SPP NET

步骤二：特征提取

将整张待检测图片输入CNN中，进行一次特征提取，得到feature maps。在各feature map中找到各候选框区域。



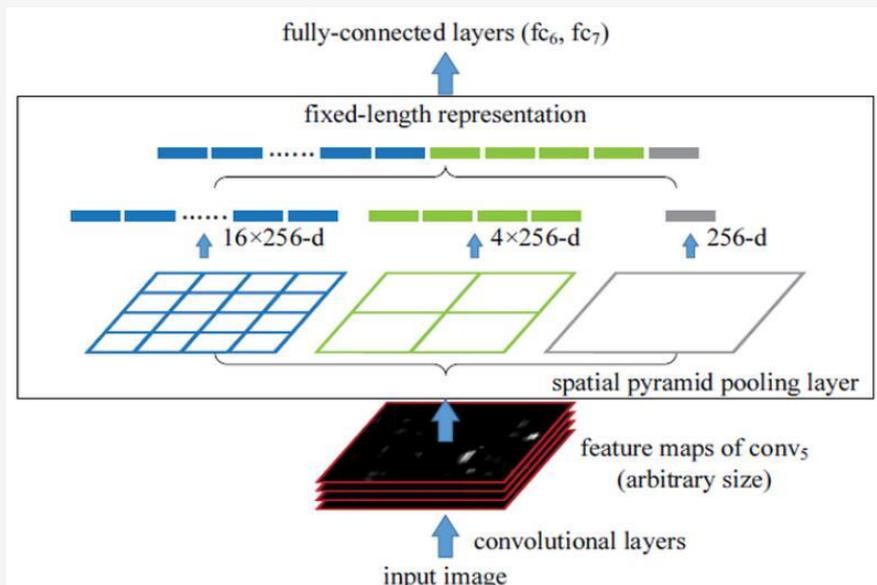
SPP Net特征提取



SPP NET

步骤三：空间金字塔池化

对各候选框区域进行空间金字塔池化，提取出固定长度的特征向量。



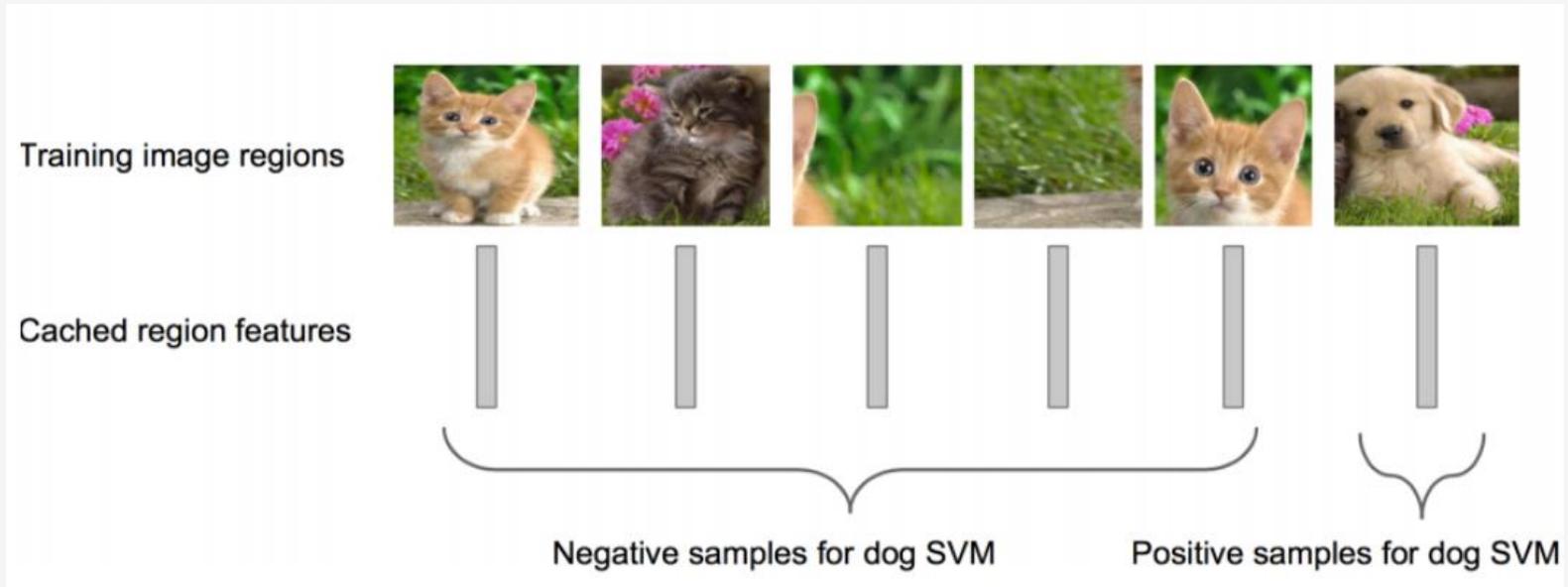
空间金字塔池化



SPP NET

步骤四：训练SVM分类器

利用SVM算法，对各候选区域对应的特征向量进行分类识别。



SVM分类识别



SPP NET

SPP NET存在的问题

经SPP NET改进后的R-CNN虽能有效提高检测速度，但仍存在局限性：

在SPP NET采用selective search对原始图片进行候选区域提取时，由于候选区域数量较多，存在候选区域特征重复提取计算问题，限制了SPP NET的检测速度。

另外，对SPP NET，虽然ROI特征在最后一个卷积层才提取，省去了多次前向CNN。但由于SVM，ROI特征仍需存储。此外，SPP NET中的tuning无法更新SPP层之前的所有权重，因此对于比较深的网络无能为力。

04 PART FOUR

Fast R-CNN



Fast R-CNN

R-CNN和SPP NET的不足：

- 1) R-CNN和SPP NET的训练都需要经过多个阶段：fine-tuning得到网络卷积层的特征输出、SVM对每组特征向量的学习、位置bounding box的回归
- 2) 对R-CNN，训练和测试的时间空间开销大。每个图像提取的大量ROI特征需要存储和通过CNN
- 3) 对SPP NET，虽然ROI特征在最后一个卷积层才提取，省去了多次前向CNN。但由于SVM，ROI特征仍需存储。此外，SPP NET中的tunning无法更新SPP层之前的所有权重，因此对于比较深得网络无能为力

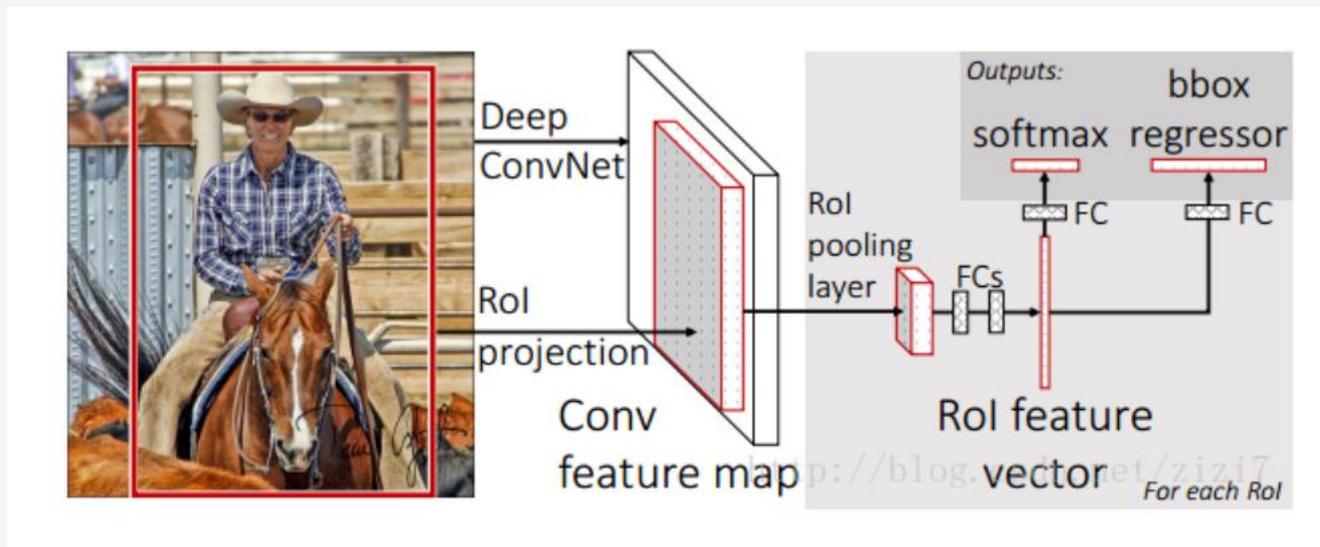
Fast R-CNN是Ross Girshick在2015年对上一年度的SPP Net算法做的改进。作者在VGG16网络的测试表明：Fast R-CNN在训练阶段比R-CNN快了9倍，比SPP NET快了3倍；在测试阶段比R-CNN快了213倍，比SPP NET快了10倍；同时精度也有一定提升。



Fast R-CNN

Fast R-CNN算法思想

Fast R-CNN算法首先通过 selective search方法生成约2K个ROI，连同图像一起输入到CNN网络；在最后一个卷积层后求取ROI位置的映射关系，使用1层的SPP池化层将每个ROI统一到相同大小；最后通过两个全连接层，一个FC层后接softmax实现分类，一个FC层后接 bounding box回归得到类别修正后的位置。



Fast R-CNN

(1) ROI pooling layer

实际上是SPP-NET的一个精简版，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层的输入。换言之，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。

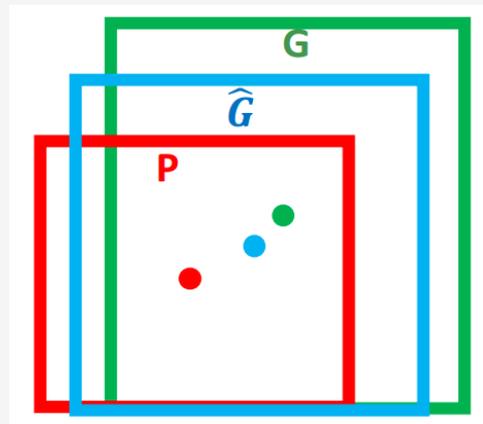
(2) 边框回归

对于窗口一般使用四维向量 (x,y,w,h) ，分别表示窗口的中心点坐标和宽高。红色的框 P 代表原始的Proposal，绿色的框 G 代表目标的Ground Truth，

我们的目标是寻找一种关系使得输入原始的窗口 P 经过映射得到一个跟真实窗口 G 更接近的回归窗口 \hat{G} 。所以，边框回归的目的即是：给定 (P_x, P_y, P_w, P_h) 寻找一种映射 f ，使得

$$f(P_x, P_y, P_w, P_h) = (G_x^{\wedge}, G_y^{\wedge}, G_w^{\wedge}, G_h^{\wedge})$$

$$\text{并且 } (G_x^{\wedge}, G_y^{\wedge}, G_w^{\wedge}, G_h^{\wedge}) \approx (G_x, G_y, G_w, G_h)$$





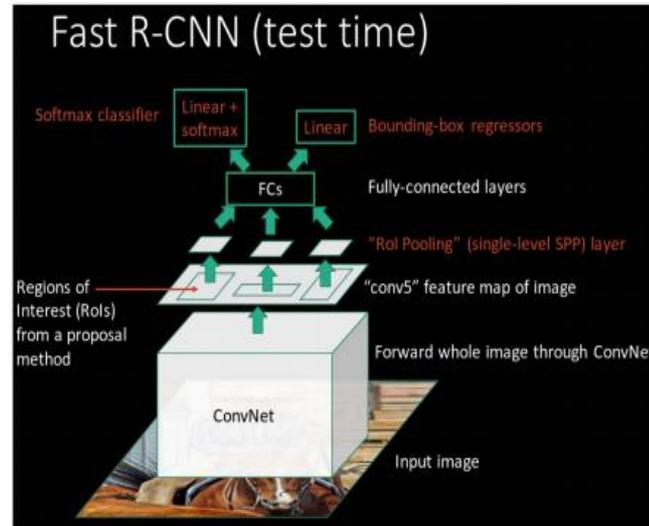
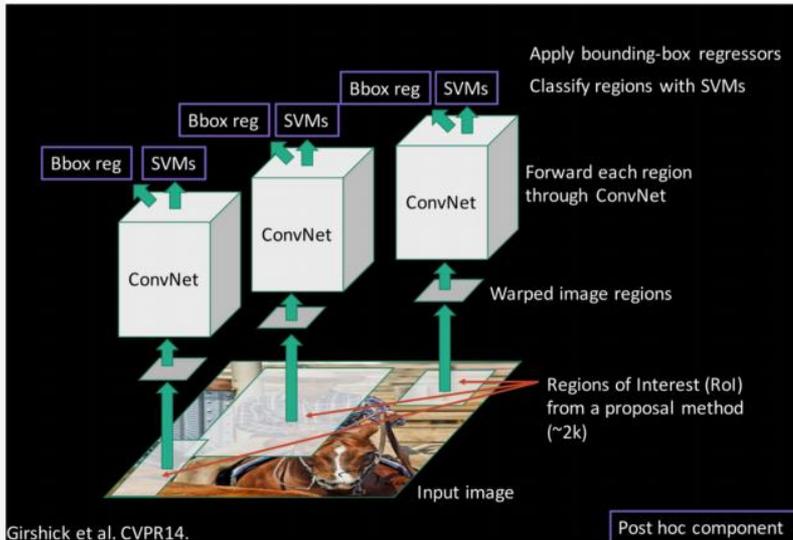
Fast R-CNN

Fast R-CNN vs R-CNN

与R-CNN框架图对比，可以发现主要有两处不同：

一是最后一个卷积层后加了一个ROI pooling layer；

二是损失函数使用了多任务损失函数，将边框回归Bounding Box Regression 直接加入到CNN网络中训练。



R-CNN Problem #1:
Slow at test-time due to independent forward passes of the CNN

Solution:
Share computation of convolutional layers between proposals for an image

Fast R-CNN

R-CNN: 许多候选框 -> CNN -> 得到每个候选框的特征 -> 分类+回归

Fast R-CNN: 一张图片 -> CNN -> 得到每张候选框的特征 -> 分类+回归

所以, Fast R-CNN相对于R-CNN的提速原因就在于: 不像R-CNN把每个候选区域给深度网络提特征, 而是整张图提一次特征, 再把候选框映射到第五个卷积层上, 而Fast R-CNN只需要计算一次特征, 剩下的只需要在第五个卷积层上操作就可以了。

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

然而, Fast R-CNN在进行选择性搜索时, 需要找出所有的候选框, 这个过程也非常耗时。

05

PART FIVE

Faster R-CNN



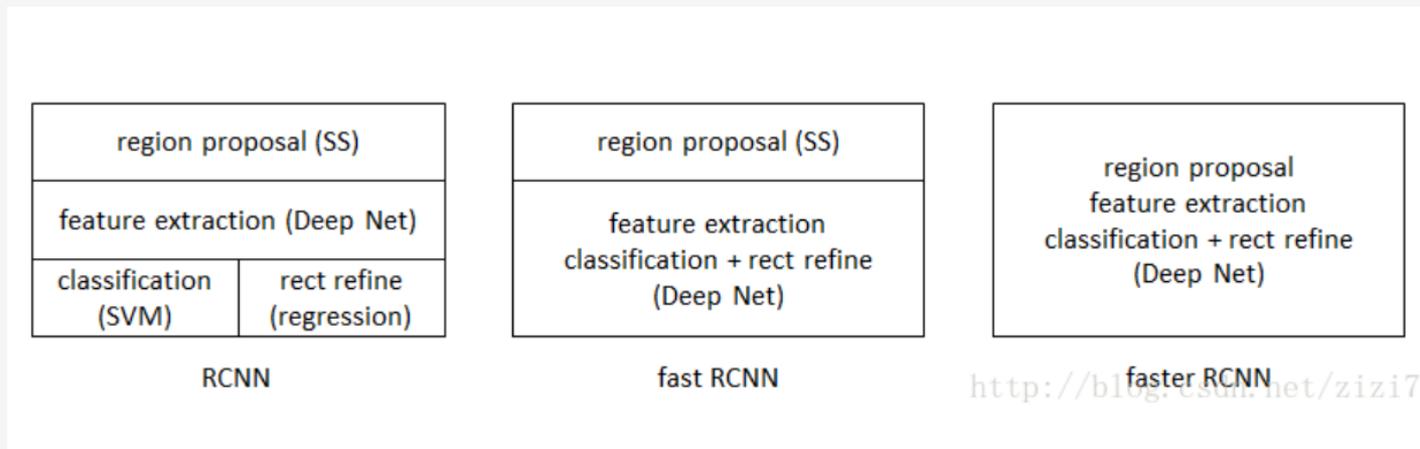
Faster R-CNN

由于Fast R-CNN在进行选择性搜索时，需要找出所有的候选框，这大大限制了其检测的速度。Faster R-CNN是Ross Girshick对Fast R-CNN算法的改进。简单网络（ZF）目标检测速度达到17fps，在PASCAL VOC上准确率为59.9%；复杂网络（VGG-16）达到5fps，准确率78.8%。

Faster R-CNN算法思想

从R-CNN到Fast R-CNN，再到Faster R-CNN，目标检测的四个基本步骤（候选区域生成，特征提取，分类，位置精修）终于被统一到一个深度网络框架之内。所有计算没有重复，完全在GPU中完成，大大提高了运行速度。

Faster R-CNN可以简单地看做“区域生成网络RPN+ Fast RCNN”的系统，用RPN代替fast RCNN中的Selective Search方法。



Faster R-CNN

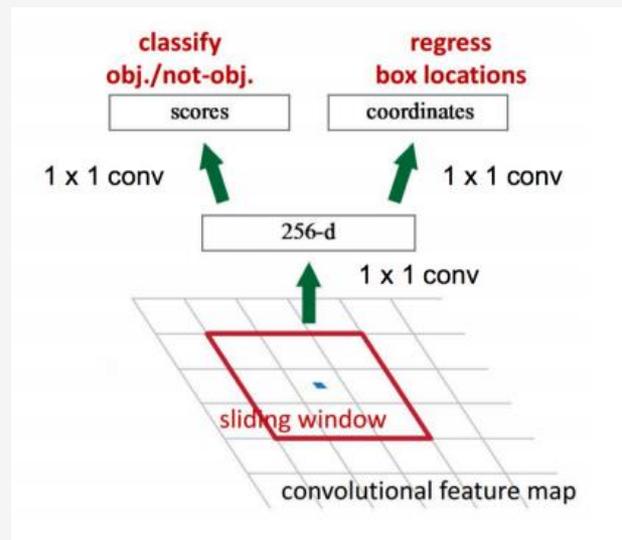
RPN

为了提高候选区域的寻找速度，加入一个提取边缘的神经网络，也就是说，寻找候选框的任务也交给神经网络来完成。在Faster 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就是位置和大小固定的box，可以理解成事先设置好的固定的proposal）。

具体做法：

将RPN放在提取整幅图片特征的CNN后面，RPN直接训练得到候选区域。

- 1.在feature map上滑动窗口。
- 2.建一个神经网络用于物体分类+框位置的回归。
- 3.滑动窗口的位置提供了物体的大体位置信息。
- 4.框的回归提供了框更精确的位置。





Faster R-CNN

R-CNN VS Fast R-CNN VS Faster R-CNN

RCNN:

Selective Search -> 每个候选框CNN特征提取 -> SVM分类 -> 边框回归

Fast R-CNN:

Selective Search -> 整张图片输入CNN, 得到feature map -> 每个候选框在feature map上的映射patch作为卷积特征输入到SPP layer和之后的层 -> softmax分类 + 边框回归

Faster R-CNN:

整张图片输入CNN, 得到feature map -> 输入到RPN, 得到候选框 -> 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 -> 分类 + 边框回归

	R-CNN	Fast R-CNN	Faster R-CNN
提取候选框	Selective Search	Selective Search	RPN网络
提取特征	CNN	CNN+ROI Pooling	
特征分类	SVM		

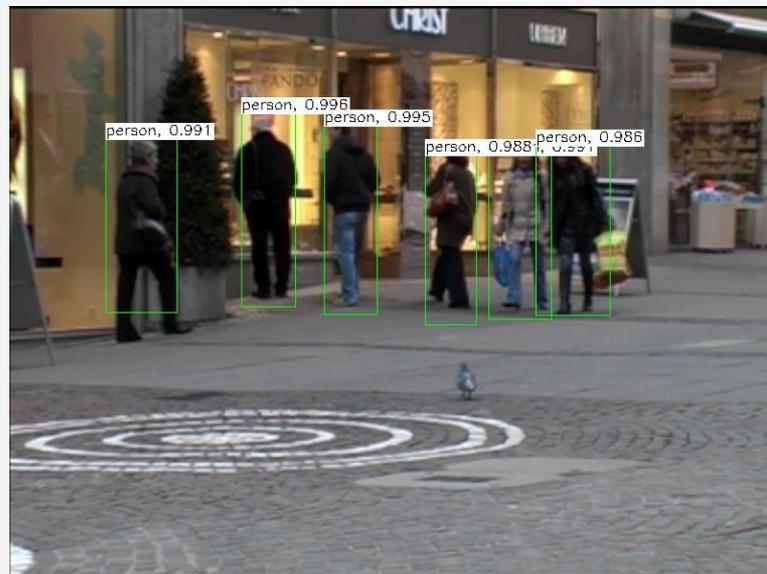
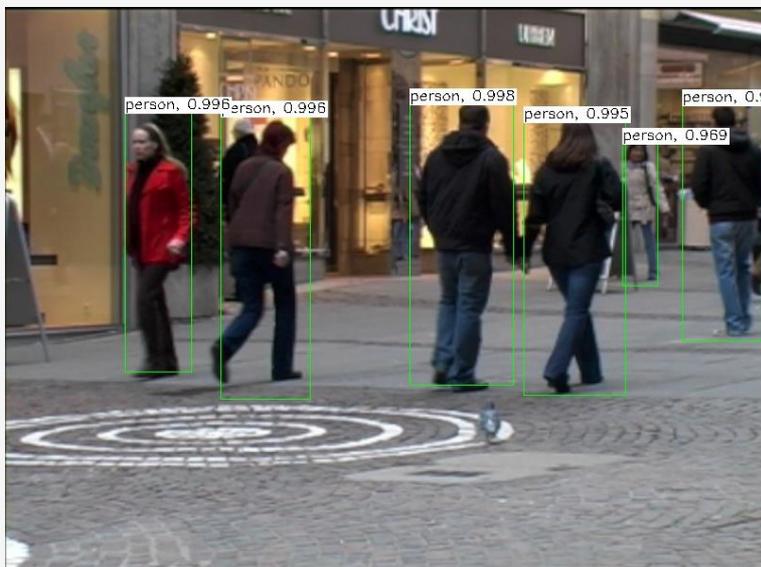
	R-CNN	Fast R-CNN	Faster R-CNN
Test time per image (with proposals)	50 seconds	2 seconds	0.2 seconds
(Speedup)	1x	25x	250x
mAP (VOC 2007)	66.0	66.9	66.9

PART SIX

实验展示

实验展示

采用MIT pedestrian detection database 基于VGG16的Faster RCNN实验结果



Faster RCNN行人检测结果

由测试结果可以看到，Faster RCNN 对目标具有较好的检测效果，且对于人群间的重叠，仍能较好的进行检测。



感谢各位聆听

Thanks for Listening