shartoo +

RCNN,Fast RCNN,Faster RCNN 总结

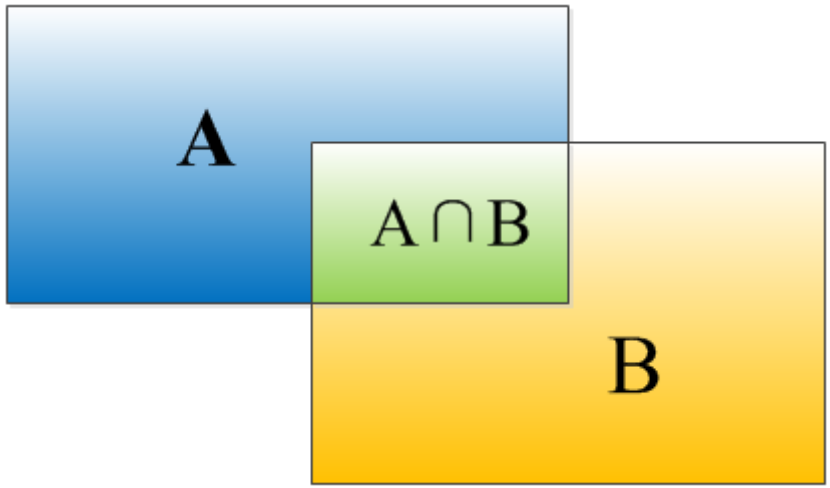
2017-01-13

一 背景知识

1.1 IOU的定义

物体检测需要定位出物体的bounding box，就像下面的图片一样，我们不仅要定位出车辆的bounding box 我们还要识别出bounding box 里面的物体就是车辆。对于bounding box的定位精度，有一个很重要的概念，因为我们算法不可能百分百跟人工标注的数据完全匹配，因此就存在一个定位精度评价公式：IOU。

IOU定义了两个bounding box的重叠度，如下图所示:



矩形框A、B的一个重合度IOU计算公式为：

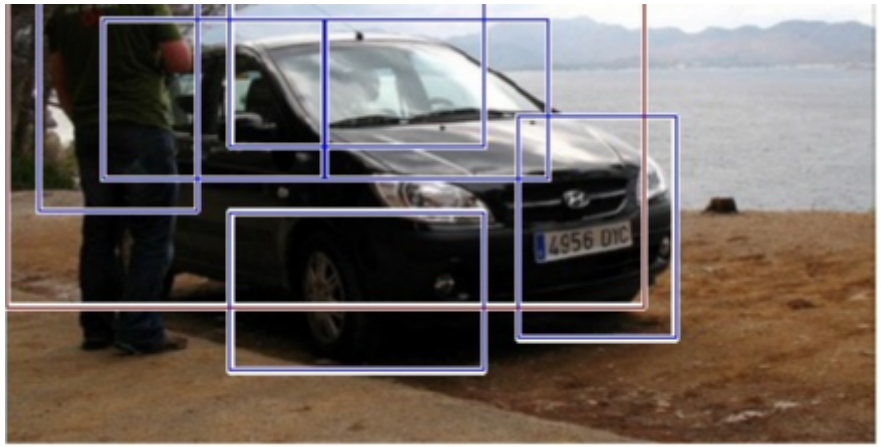
IOU=(A∩B)/(A∪B)

就是矩形框A、B的重叠面积占A、B并集的面积比例:

IOU=SI/(SA+SB-SI)

1.2 非极大值抑制

RCNN算法，会从一张图片中找出n多个可能是物体的矩形框，然后为每个矩形框为做类别分类概率：

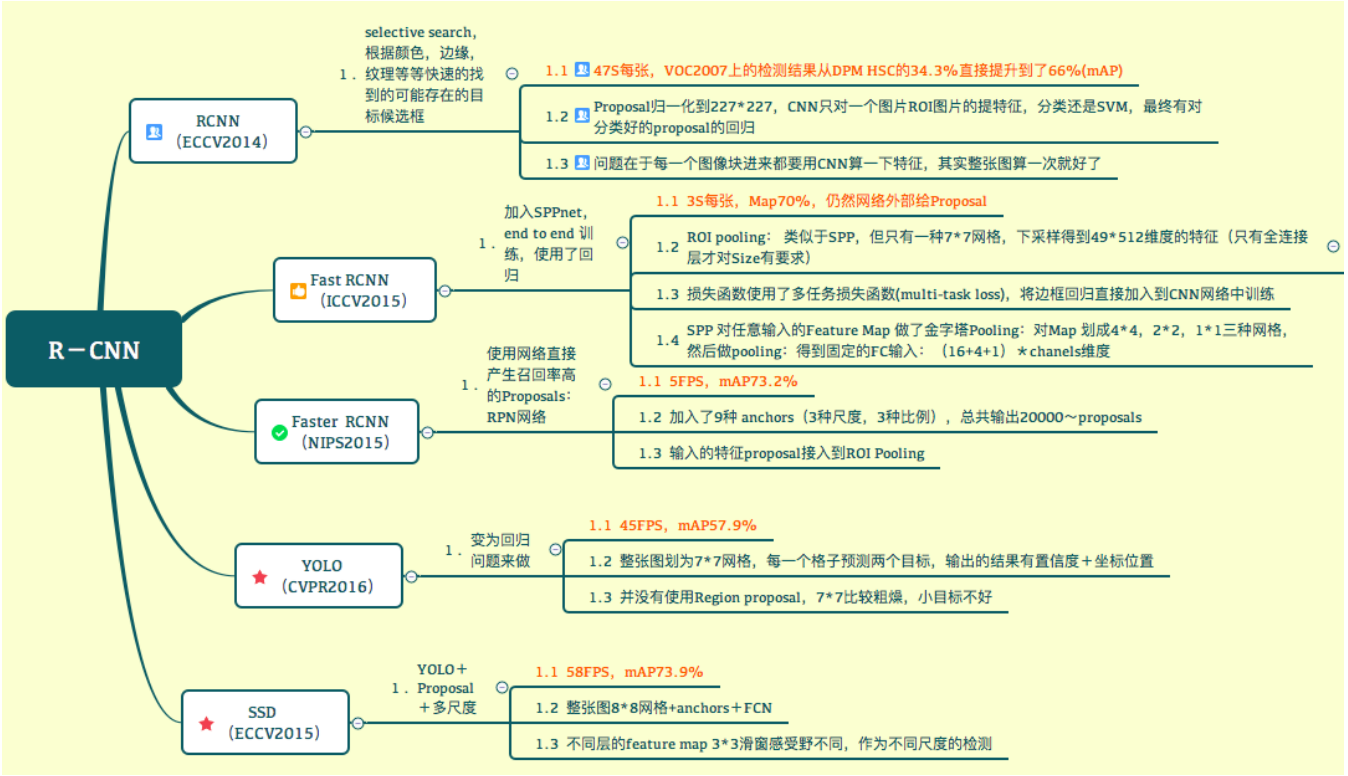


就像上面的图片一样，定位一个车辆，最后算法就找出了一堆的方框，我们需要判别哪些矩形框是没用的。非极大值抑制：先假设有6个矩形框，根据分类器类别分类概率做排序，从小到大分别属于车辆的概率分别为A、B、C、D、E、F。

1. 从最大概率矩形框F开始，分别判断A~E与F的重叠度IOU是否大于某个设定的阈值;
2. 假设B、D与F的重叠度超过阈值，那么就扔掉B、D；并标记第一个矩形框F，是我们保留下来的。
3. 从剩下的矩形框A、C、E中，选择概率最大的E，然后判断E与A、C的重叠度，重叠度大于一定的阈值，那么就扔掉；并标记E是我们保留下来的第二个矩形框。

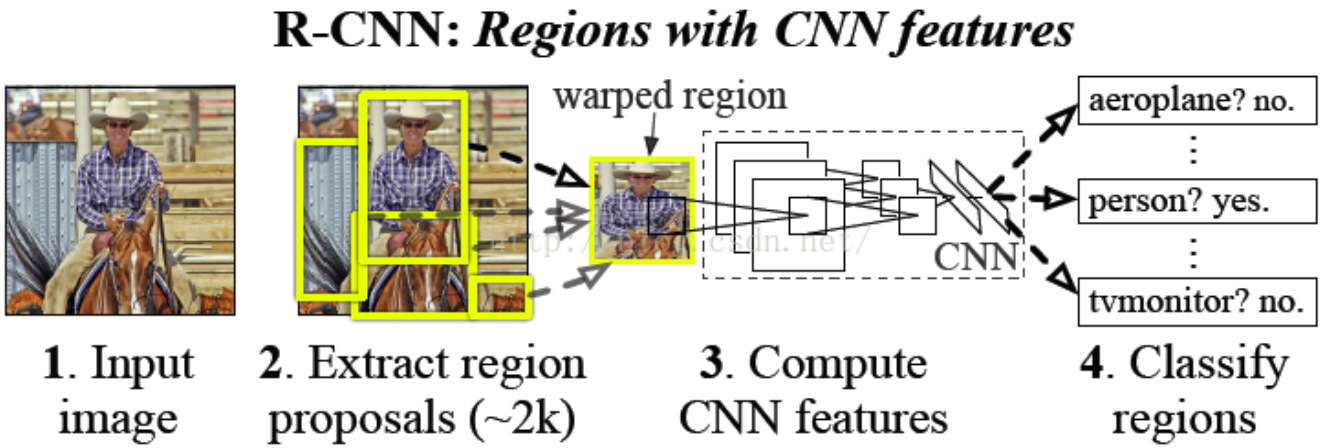
就这样一直重复，找到所有被保留下来的矩形框。

1.3 一张图概览RCNN



二 RCNN

算法概要：首先输入一张图片，我们先定位出2000个物体候选框，然后采用CNN提取每个候选框中图片的特征向量，特征向量的维度为4096维，接着采用svm算法对各个候选框中的物体进行分类识别。也就是总个过程分为三个程序：a、找出候选框；b、利用CNN提取特征向量；c、利用SVM进行特征向量分类。具体的流程如下图片所示：



下面分别讲解各个步骤。

2.1 候选框搜索

当我们输入一张图片时，我们要搜索出所有可能是物体的区域，这个采用的方法是传统文献的算法selective search (github上有源码)，通过这个算法我们搜索出2000个候选框。然后从上面的总流程图中可以看到，搜出的候选框是矩形的，而且是大小各不相同。然而CNN对输入图片的大小是有固定的，如果把搜索到的矩形选框不做处理，就扔进CNN中，肯定不行。因此对于每个输入的候选框都需要缩放到固定的大小。下面我们讲解要怎么进行缩放处理，为了简单起见我们假设下一阶段CNN所需要的输入图片大小是个正方形图片227*227。因为我们经过selective search 得到的是矩形框，paper试验了两种不同的处理方法：

(1)各向异性缩放

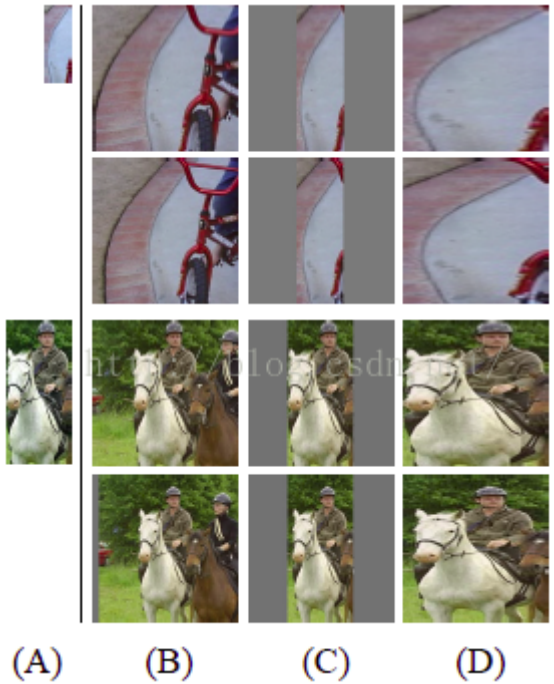
这种方法很简单，就是不管图片的长宽比例，管它是否扭曲，进行缩放就是了，全部缩放到CNN输入的大小227*227，如下图(D)所示；

(2)各向同性缩放

因为图片扭曲后，估计会对后续CNN的训练精度有影响，于是作者也测试了“各向同性缩放”方案。这个有两种办法

A. 直接在原始图片中，把bounding box的边界进行扩展延伸成正方形，然后再进行裁剪；如果已经延伸到了原始图片的外边界，那么就用bounding box中的颜色均值填充；如下图(B)所示;

B. 先把bounding box图片裁剪出来，然后用固定的背景颜色填充成正方形图片(背景颜色也是采用bounding box的像素颜色均值),如下图(C)所示;



对于上面的异性、同性缩放，文献还有个padding处理，上面的示意图中第1、3行就是结合了padding=0,第2、4行结果图采用padding=16的结果。经过最后的试验，作者发现采用各向异性缩放、padding=16的精度最高。

上面处理完后，可以得到指定大小的图片，因为我们后面还要继续用这2000个候选框图片，继续训练CNN、SVM。然而人工标注的数据一张图片中就只标注了正确的bounding box，我们搜索出来的2000个矩形框也不可能会出现一个与人工标注完全匹配的候选框。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。

2.2 网络设计

网络架构我们有两个可选方案：第一选择经典的Alexnet；第二选择VGG16。经过测试Alexnet精度为58.5%，VGG16精度为66%。VGG这个模型的特点是选择比较小的卷积核、选择较小的跨步，这个网络的精度高，不过计算量是Alexnet的7倍。后面为了简单起见，我们就直接选用Alexnet，并进行讲解；Alexnet特征提取部分包含了5个卷积层、2个全连接层，在Alexnet中p5层神经元个数为9216、 f6、f7的神经元个数都是4096，通过这个网络训练完毕后，最后提取特征每个输入候选框图片都能得到一个4096维的特征向量。

2.2.1 网络初始化

直接用Alexnet的网络，然后连参数也是直接采用它的参数，作为初始的参数值，然后再fine-tuning训练。

网络优化求解：采用随机梯度下降法，学习速率大小为0.001；

2.2.2 fine-tuning阶段

我们接着采用selective search 搜索出来的候选框，然后处理到指定大小图片，继续对上面预训练的cnn模型进行fine-tuning训练。假设要检测的物体类别有N类，那么我们就需要把上面预训练阶段的CNN模型的最后一层给替换掉，替换成N+1个输出的神经元(加1，表示还有一个背景)，然后这一层直接采用参数随机初始化的方法，其它网络层的参数不变；接着就可以开始继续SGD训练了。开始的时候，SGD学习率选择0.001，在每次训练的时候，我们batch size大小选择128，其中32个正样本、96个负样本。

三 Fast RCNN

3.1 引入原因

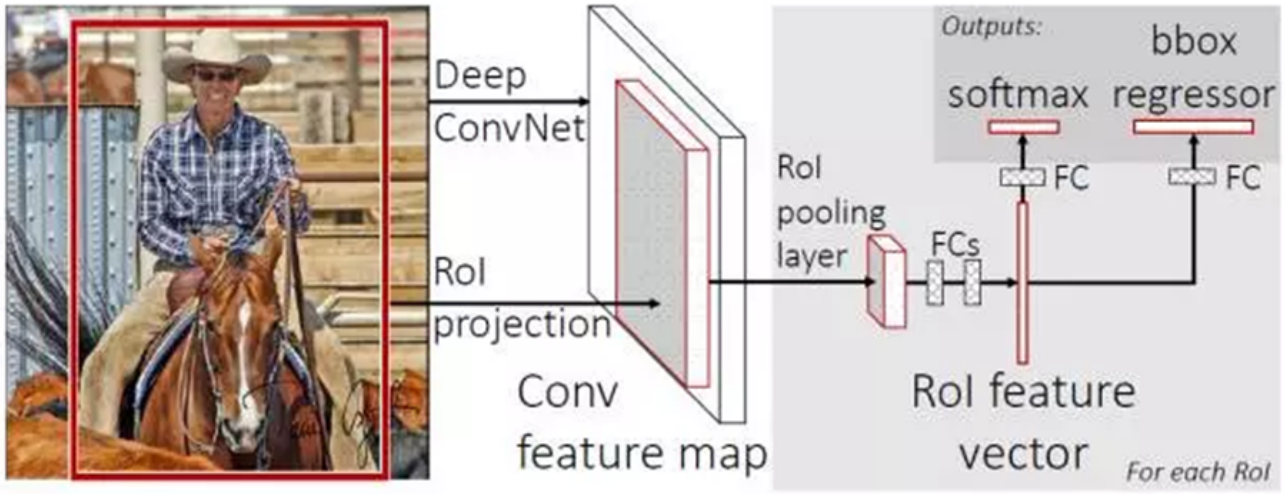
FRCNN针对RCNN在训练时是multi-stage pipeline和训练的过程中很耗费时间空间的问题进行改进。它主要是将深度网络和后面的SVM分类两个阶段整合到一起，使用一个新的网络直接做分类和回归。主要做以下改进:

1. 最后一个卷积层后加了一个ROI pooling layer。ROI pooling layer首先可以将image中的ROI定位到feature map，然后是用一个单层的SPP layer将这个feature map patch池化为固定大小的feature之后再传入全连接层。

2. 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。

3.2 模型

fast rcnn 的结构如下



图中省略了通过ss获得proposal的过程，第一张图中红框里的内容即为通过ss提取到的proposal，中间的一块是经过深度卷积之后得到的conv feature map，图中灰色的部分就是我们红框中的proposal对应于conv feature map中的位置，之后对这个特征经过ROI pooling layer处理，之后进行全连接。在这里得到的ROI feature vector最终被分享，一个进行全连接之后用来做softmax回归，用来进行分类，另一个经过全连接之后用来做bbox回归。

注意： 对中间的Conv feature map进行特征提取。每一个区域经过RoI pooling layer和FC layers得到一个 **固定长度** 的feature vector(这里需要注意的是，输入到后面RoI pooling layer的feature map是在Conv feature map上提取的，故整个特征提取过程，只

计算了一次卷积。虽然在最开始也提取出了大量的RoI，但他们还是作为整体输入进卷积网络的，最开始提取出的RoI区域只是为了最后的Bounding box 回归时使用，用来输出原图中的位置)。

3.3 SPP网络

何恺明研究员于14年撰写的论文，主要是把经典的Spatial Pyramid Pooling结构引入CNN中，从而使CNN可以处理任意size和scale的图片；这中方法不仅提升了分类的准确率，而且还非常适合Detection，比经典的RNN快速准确。

本文不打算详细解释SPP网络，只介绍其中的SPP-layer，由于fast rcnn会使用到SPP-layer。

SPP layer

根据pooling规则，每个pooling bin（window）对应一个输出，所以最终pooling后特征输出由bin的个数来决定。本文就是分级固定bin的个数，调整bin的尺寸来实现多级pooling固定输出。

如图所示，layer-5的unpooled FM维数为16*24，按照图中所示分为3级，

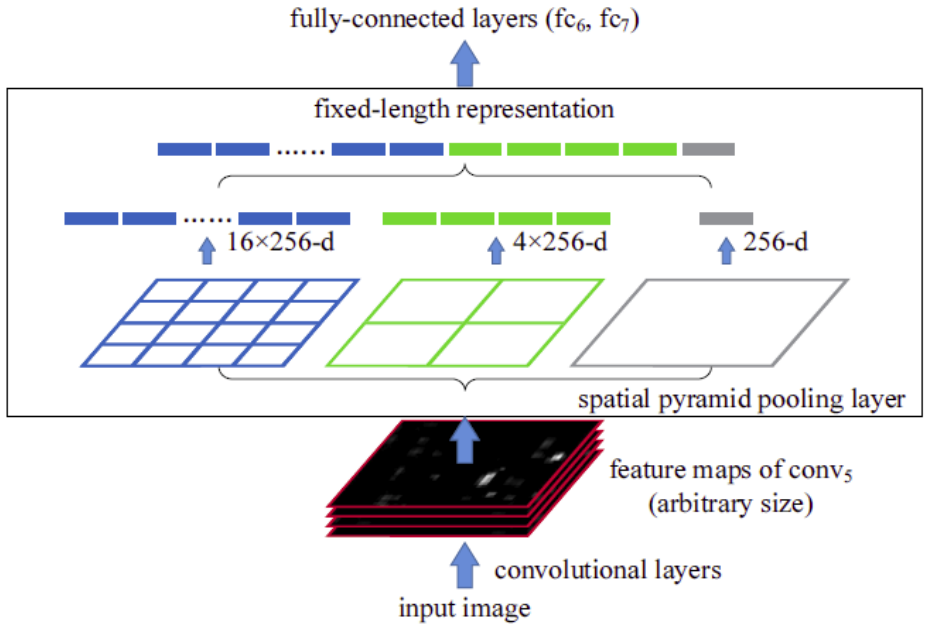


Figure 3. The network structure with a **spatial pyramid pooling layer**. Here 256 is the filter number of the $conv_5$ layer.

第一级bin个数为1，最终对应的window大小为16*24；

第二级bin个数为4个，最终对应的window大小为4*8

第三级bin个数为16个，最终对应的window大小为1*1.5（小数需要舍入处理）

通过融合各级bin的输出，最终每一个unpooled FM经过SPP处理后，得到了1+4+16维的SPPed FM输出特征，经过融合后输入分类器。

这样就可以在任意输入size和scale下获得固定的输出；不同scale下网络可以提取不同尺度的特征，有利于分类。

3.4 RoI pooling layer

每一个RoI都有一个四元组（r,c,h,w）表示，其中（r，c）表示左上角，而（h，w）则代表高度和宽度。这一层使用最大池化（max pooling）来将RoI区域转化成固定大小的HW的特征图。假设一个RoI的窗口大小为hw,则转换成HW之后，每一个网格

都是一个h/H * w/W大小的子网，利用最大池化将这个子网中的值映射到HW窗口即可。Pooling对每一个特征图通道都是独立的，这是SPP layer的特例，即只有一层的空间金字塔。

3.5 从预训练的网络中初始化数据

有三种预训练的网络：CaffeNet，VGG_CNN_M_1024，VGG-16，他们都有5个最大池化层和5到13个不等的卷积层。用他们来初始化Fast R-CNN时，需要修改三处：

①最后一个池化层被RoI pooling layer取代

②最后一个全连接层和softmax被替换成之前介绍过的两个兄弟并列层

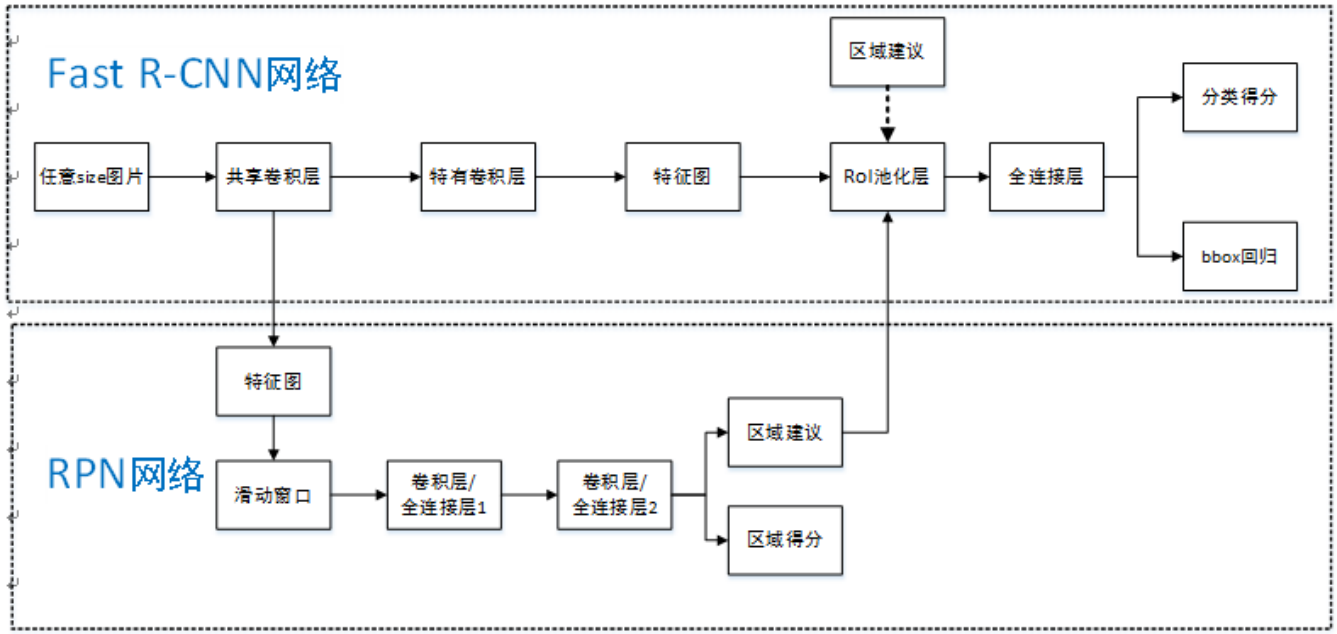
③网络输入两组数据：一组图片和那些图片的一组RoIs

3.6 检测中的微调

使用BP算法训练网络是Fast R-CNN的重要能力，前面已经说过，SPP-net不能微调spp层之前的层，主要是因为当每一个训练样本来自于不同的图片时，经过SPP层的BP算法是很低效的（感受野太大）. Fast R-CNN提出SGD mini_batch分层取样的方法：首先随机取样N张图片，然后每张图片取样R/N个RoIs e.g. N=2 and R=128 除了分层取样，还有一个就是FRCN在一次微调中联合优化softmax分类器和bbox回归，看似一步，实际包含了多任务损失（multi-task loss）、小批量取样（mini-batch sampling）、RoI pooling层的反向传播（backpropagation through RoI pooling layers）、SGD超参数（SGD hyperparameters）。

4 Faster RCNN

Faster R-CNN统一的网络结构如下图所示，可以简单看作RPN网络+Fast R-CNN网络。



原理步骤如下:

1. 首先向CNN网络【ZF或VGG-16】输入任意大小图片；

2. 经过CNN网络前向传播至最后共享的卷积层，一方面得到供RPN网络输入的特征图，另一方面继续前向传播至特有卷积层，产生更高维特征图；

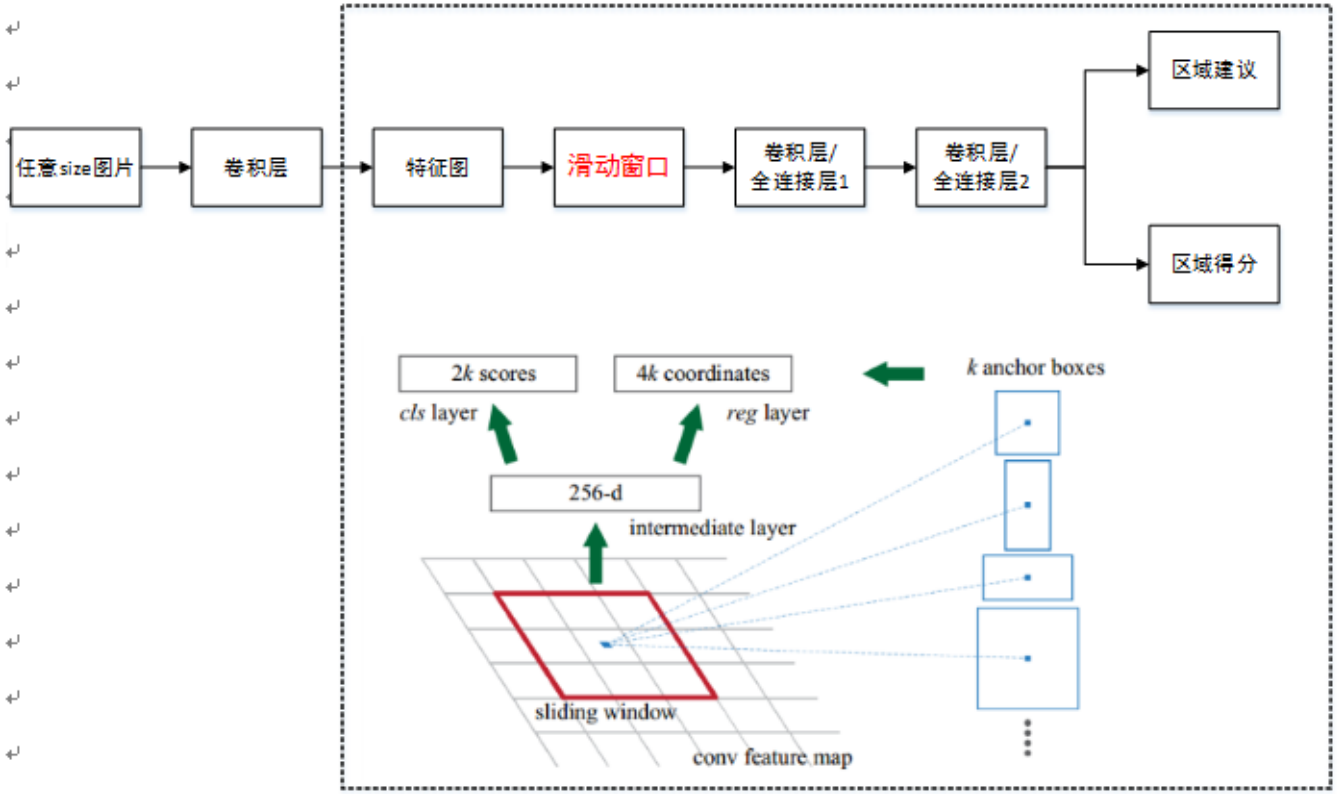
3. 供RPN网络输入的特征图经过RPN网络得到区域建议和区域得分，并对区域得分采用非极大值抑制【阈值为0.7】，输出其Top-N【文中为300】得分的区域建议给RoI池化层；

4. 第2步得到的高维特征图和第3步输出的区域建议同时输入RoI池化层，提取对应区域建议的特征；

5. 第4步得到的区域建议特征通过全连接层后，输出该区域的分类得分以及回归后的bounding-box。

4.1 单个RPN网络结构

单个RPN网络结构如下:



注意： 上图中卷积层/全连接层表示卷积层或者全连接层，作者在论文中表示这两层实际上是全连接层，但是网络在所有滑窗位置共享全连接层，可以很自然地用n×n卷积核【论文中设计为3×3】跟随两个并行的1×1卷积核实现

RPN的作用：RPN在CNN卷积层后增加滑动窗口操作以及两个卷积层完成区域建议功能，第一个卷积层将特征图每个滑窗位置编码成一个特征向量，第二个卷积层对应每个滑窗位置输出k个区域得分和k个回归后的区域建议，并对得分区域进行非极大值抑制后输出得分Top-N【文中为300】区域，告诉检测网络应该注意哪些区域，本质上实现了Selective Search、EdgeBoxes等方法的功能。

4.2 RPN层的具体流程

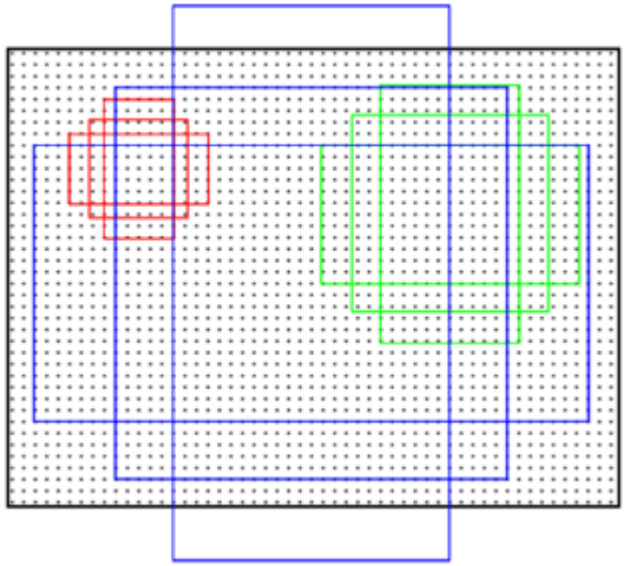
1. 首先套用ImageNet上常用的图像分类网络，本文中试验了两种网络：ZF或VGG-16，利用这两种网络的部分卷积层产生原始图像的特征图；

2. 对于1中特征图，用n×n**【论文中设计为3×3，n=3看起来很小，但是要考虑到这是非常高层的feature map，其size本身也没有多大，因此9个矩形中，每个矩形窗框都是可以感知到很大范围的】的滑动窗口在特征图上滑动扫描【代替了从原始图滑窗获取特征】**，每个滑窗位置通过卷积层1映射到一个低维的特征向量**【ZF网络：256维；VGG-16网络：512维，低维是相对于特征图大小W×H，typically~60×40=2400】**后采用ReLU，并为每个滑窗位置考虑k种**【论文中k=9】**可能的参考窗口**【论文中称为anchors，见下解释】**，这就意味着每个滑窗位置会同时预测最多9个区域建议**【超出边界的不考虑】**，对于一个W×H的特征图，就会产生W×H×k个区域建议；

3. 步骤2中的低维特征向量输入两个并行连接的卷积层2：reg窗口回归层**【位置精修】**和cls窗口分类层，分别用于回归区域建议产生bounding-box**【超出图像边界的裁剪到图像边缘位置】**和对区域建议是否为前景或背景打分，这里由于每个滑窗位置产生k个区域建议，所以reg层有4k个输出来编码【平移缩放参数】k个区域建议的坐标，cls层有2k个得分估计k个区域建议为前景或者背景的概率。

4.3 Anchor

Anchors是一组大小固定的参考窗口：三种尺度{ 128^2 ，256^2 ，512^2 }×三种长宽比{1:1，1:2，2:1}，如下图所示，表示RPN网络中对特征图滑窗时每个滑窗位置所对应的原图区域中9种可能的大小，相当于模板，对任意图像任意滑窗位置都是这9中模板。继而根据图像大小计算滑窗中心点对应原图区域的中心点，通过中心点和size就可以得到滑窗位置和原图位置的映射关系，由此原图位置并根据与Ground Truth重复率贴上正负标签，让RPN学习该Anchors是否有物体即可。对于每个滑窗位置，产生k=9个anchor对于一个大小为W*H的卷积feature map，总共会产生WHk个anchor。



平移不变性

Anchors这种方法具有平移不变性，就是说在图像中平移了物体，窗口建议也会跟着平移。同时这种方式也减少了整个模型的size，输出层 $512 \times (4+2) \times 9 = 2.8 \times 10^4$ 个参数【512是前一层特征维度，(4+2)×9是9个Anchors的前景背景得分和平移缩放参数】，而MultiBox有 $1536 \times （4+1） \times 800 = 6.1 \times 10^6$个 参数，而较小的参数可以在小数据集上减少过拟合风险。

当然，在RPN网络中我们只需要找到大致的地方，无论是位置还是尺寸，后面的工作都可以完成，这样的话采用小网络进行简单的学习【估计和猜差不多，反正有50%概率】，还不如用深度网络【还可以实现卷积共享】，固定尺度变化，固定长宽比变化，固定采样方式来大致判断是否是物体以及所对应的位置并降低任务复杂度。

4.4 多尺度多长宽比率

有两种方法解决多尺度多长宽比问题:

1. **图像金字塔**:对伸缩到不同size的输入图像进行特征提取，虽然有效但是费时.

2. **feature map上使用多尺度（和/或长宽比）的滑窗**:例如，DPM分别使用不同大小的filter来训练不同长宽比的模型。若这种方法用来解决多尺度问题，可以认为是“filter金字塔(pyramid of filters)”

4.5 训练过程

4.5.1 RPN网络训练过程

RPN网络被ImageNet网络【ZF或VGG-16】进行了有监督预训练，利用其训练好的网络参数初始化； 用标准差0.01均值为0的高斯分布对新增的层随机初始化。

4.5.2 Fast R-CNN网络预训练

同样使用mageNet网络【ZF或VGG-16】进行了有监督预训练，利用其训练好的网络参数初始化。

4.5.3 RPN网络微调训练

PASCAL VOC 数据集中既有物体类别标签，也有物体位置标签； 正样本仅表示前景，负样本仅表示背景； 回归操作仅针对正样本进行； 训练时弃用所有超出图像边界的anchors，否则在训练过程中会产生较大难以处理的修正误差项，导致训练过程无法收敛； 对去掉超出边界后的anchors集采用非极大值抑制，最终一张图有2000个anchors用于训练【详细见下】； 对于ZF网络微调所有层，对VGG-16网络仅微调conv3_1及conv3_1以上的层，以便节省内存。

SGD mini-batch采样方式： 同Fast R-CNN网络，采取 `image-centric` 方式采样，即采用层次采样，先对图像取样，再对anchors取样，同一图像的anchors共享计算和内存。 每个mini-batch包含从一张图中随机提取的256个anchors，正负样本比例为1:1【当然可以对一张图所有anchors进行优化，但由于负样本过多最终模型会对正样本预测准确率很低】来计算一个mini-batch的损失函数，如果一张图中不够128个正样本，拿负样本补凑齐。

训练超参数选择： 在PASCAL VOC数据集上前60k次迭代学习率为0.001，后20k次迭代学习率为0.0001；动量设置为0.9，权重衰减设置为0.0005。

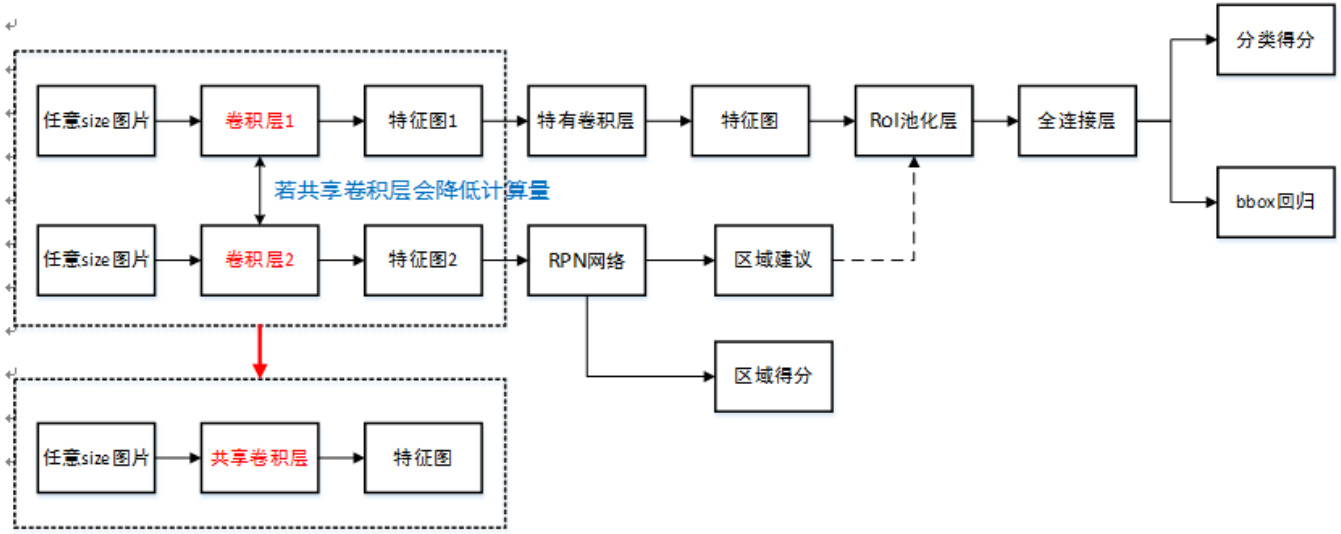
多任务目标函数【 `分类损失+回归损失` 】具体如下：

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}}\sum_i L_{cls}(p_i, p_i^*) + \lambda\frac{1}{N_{reg}}\sum_i p_i^* L_{reg}(t_i, t_i^*)$$

- `i` 为一个anchor在一个mini-batch中的下标
- p_i 是anchor i为一个object的预测可能性
- $p_i^\star$ 为ground-truth标签。如果这个anchor是positive的，则ground-truth标签 $p_i^\star$ 为1，否则为0。
- t_i 表示表示正样本anchor到预测区域bounding box的4个参数化坐标，**【以anchor为基准的变换】**
- $t_i^\star$ 是这个positive anchor对应的ground-truth box。**【以anchor为基准的变换】**
- L_{cls} 分类的损失（classification loss），是一个二值分类器（是object或者不是）的softmax loss。其公式为 $L_{cls}(p_i, p_i^\star) = -log[p_i * p_i^\star + (1-p_i^\star)(1-p_i)]$
- L_{reg} 回归损失（regression loss），$L_{reg}(t_i, t_i^\star) = R(t_i - t_i^\star)$【两种变换之差越小越好】，其中R是Fast R-CNN中定义的robust ross function (smooth L1)。$p_i^\star L_{reg}$ 表示回归损失只有在positive anchor（ $p_i^\star = 1$)的时候才会被激活。cls与reg层的输出分别包含{p_i}和{ t_i }。R函数的定义为: $smooth_{L1}(x) = 0.5x^2 \quad if \mid x \mid < 1 \quad otherwise \quad \mid x \mid -0.5$
- λ参数用来权衡分类损失 L_{cls} 和回归损失 L_reg ，默认值λ=10【文中实验表明 λ从1变化到100对mAP影响不超过1%】；
- N_{cls} 和 N_{reg} 分别用来标准化分类损失项 L_{cls} 和回归损失项 L_{reg}，默认用mini-batch size=256设置 N_{cls}，用anchor位置数目~2400初始化 N_{reg}，文中也说明标准化操作并不是必须的，可以简化省略。

4.5.4 RPN网络、Fast R-CNN网络联合训练

训练网络结构示意图如下所示：



如上图所示，**RPN网络、Fast R-CNN网络联合训练是为了让两个网络共享卷积层，降低计算量**。

文中通过4步训练算法，交替优化学习至共享特征：

1. 进行上面RPN网络预训练，和以区域建议为目的的RPN网络end-to-end微调训练。
2. 进行上面Fast R-CNN网络预训练，用第①步中得到的区域建议进行以检测为目的的Fast R-CNN网络end-to-end微调训练【此时无共享卷积层】。
3. 使用第2步中微调后的Fast R-CNN网络重新初始化RPN网络，固定共享卷积层【即设置学习率为0，不更新】，仅微调RPN网络独有的层【此时共享卷积层】。
4. 固定第3步中共享卷积层，利用第③步中得到的区域建议，仅微调Fast R-CNN独有的层，至此形成统一网络如上图所示。

4.6 相关解释

**RPN网络中bounding-box回归怎么理解？同Fast R-CNN中的bounding-box回归相比有什么区别？ **

对于bounding-box回归，采用以下公式：

- t

$$t_x = \frac{(x - x_a)}{w_a}$$
$$t_y = \frac{(y - y_a)}{h_a}$$
$$t_w = log\frac{w}{w_a}$$
$$t_h = log\frac{h}{h_a}$$

- t^*

$$t_x^* = \frac{(x^* - x_a)}{w_a}$$
$$t_y^* = \frac{(y^* - y_a)}{h_a}$$
$$t_w^* = log\frac{w^*}{w_a}$$
$$t_h^* = log\frac{h^*}{h_a}$$

其中，x，y，w，h表示窗口中心坐标和窗口的宽度和高度，变量x，x_a 和 x^* 分别表示预测窗口、anchor窗口和Ground Truth的坐标【y，w，h同理】，因此这可以被认为是一个从anchor窗口到附近Ground Truth的bounding-box 回归；

RPN网络中bounding-box回归的实质其实就是计算出预测窗口。这里以anchor窗口为基准，计算Ground Truth对其的平移缩放变化参数，以及预测窗口【可能第一次迭代就是anchor】对其的平移缩放参数，因为是以anchor窗口为基准，所以只要使这两组参数越接近，以此构建目标函数求最小值，那预测窗口就越接近Ground Truth，达到回归的目的；

文中提到， Fast R-CNN中基于RoI的bounding-box回归所输入的特征是在特征图上对任意size的RoIs进行Pool操作提取的，所有size RoI共享回归参数，而在Faster R-CNN中，用来bounding-box回归所输入的特征是在特征图上相同的空间size【3×3】上提取的，为了解决不同尺度变化的问题，同时训练和学习了k个不同的回归器，依次对应为上述9种anchors，这k个回归量并不分享权重。因此尽管特征提取上空间是固定的【3×3】，但由于anchors的设计，仍能够预测不同size的窗口。

文中提到了三种共享特征网络的训练方式？

1. **交替训练**,训练RPN，得到的区域建议来训练Fast R-CNN网络进行微调；此时网络用来初始化RPN网络，迭代此过程【文中所有实验采用】；

2. **近似联合训练:** 如上图所示，合并两个网络进行训练，前向计算产生的区域建议被固定以训练Fast R-CNN；反向计算到共享卷积层时RPN网络损失和Fast R-CNN网络损失叠加进行优化，但此时把区域建议【Fast R-CNN输入，需要计算梯度并更新】当成固定值看待，忽视了Fast R-CNN一个输入：区域建议的导数，则无法更新训练，所以称之为近似联合训练。实验发现，这种方法得到和交替训练相近的结果，还能减少20%~25%的训练时间，公开的python代码中使用这种方法；

3. **联合训练** 需要RoI池化层对区域建议可微，需要RoI变形层实现，具体请参考这片paper：Instance-aware Semantic Segmentation via Multi-task Network Cascades。

图像Scale细节问题？

文中提到训练和检测RPN、Fast R-CNN都使用单一尺度，统一缩放图像短边至600像素； 在缩放的图像上，对于ZF网络和VGG-16网络的最后卷积层总共的步长是16像素，因此在缩放前典型的PASCAL图像上大约是10像素【~500×375；600/16=375/10】。

Faster R-CNN中三种尺度怎么解释：

- **原始尺度**：原始输入的大小，不受任何限制，不影响性能；

- **归一化尺度**：输入特征提取网络的大小，在测试时设置，源码中opts.test_scale=600。anchor在这个尺度上设定，这个参数和anchor的相对大小决定了想要检测的目标范围；

- **网络输入尺度**：输入特征检测网络的大小，在训练时设置，源码中为224×224。

理清文中anchors的数目

文中提到对于1000×600的一张图像，大约有20000(~60×40×9)个anchors，忽略超出边界的anchors剩下6000个anchors，利用非极大值抑制去掉重叠区域，剩2000个区域建议用于训练； 测试时在2000个区域建议中选择Top-N【文中为300】个区域建议用于Fast R-CNN检测。

参考博客

RCNN 介绍

Fast RCNN介绍

Faster RCNN论文笔记

Fast RCNN简要笔记

SPP网络

撰写评论

使用社交网站账户登录

或使用来必力便捷评论

邮件

写评论

总评论数 0

按时间正序

还没有评论，快来抢沙发吧！

来必力是？ | 询问