CV：object detection(RCNN)

一、简介

Region CNN(RCNN)可以说是利用深度学习进行目标检测的开山之作。作者Ross Girshick多次在PASCAL VOC的目标检测竞赛中折桂，2010年更带领团队获得终身成就奖，如今供职于Facebook旗下的FAIR。
这篇文章思路简洁，在DPM方法多年平台期后，效果提高显著。包括本文在内的一系列目标检测算法：RCNN,Fast RCNN, Faster RCNN代表当下目标检测的前沿水平，在github都给出了基于Caffe的源码。

思想

本文解决了目标检测中的两个关键问题。

问题一：速度

经典的目标检测算法使用滑动窗法依次判断所有可能的区域。本文则预先提取一系列较可能是物体的候选区域，之后仅在这些候选区域上提取特征，进行判断。

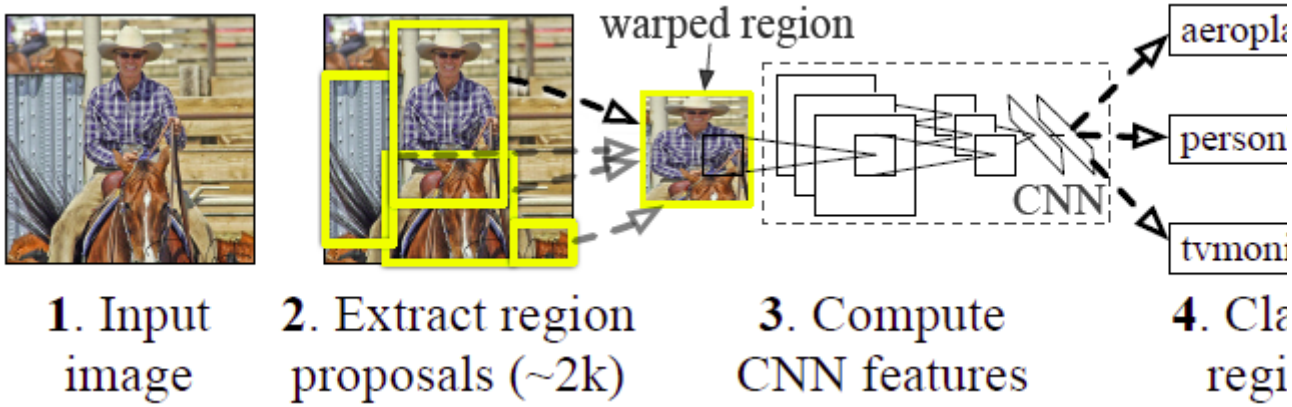
问题二：训练集

经典的目标检测算法在区域中提取人工设定的特征（Haar，HOG）。本文则需要训练深度网络进行特征提取。可供使用的有两个数据库：
一个较大的识别库（ImageNet ILSVC 2012）：标定每张图片中物体的类别。一千万图像，1000类。
一个较小的检测库（PASCAL VOC 2007）：标定每张图片中，物体的类别和位置。一万图像，20类。
本文使用识别库进行预训练，而后用检测库调优参数。最后在检测库上评测。

流程

RCNN算法分为4个步骤
- 一张图像生成1K~2K个候选区域
- 对每个候选区域，使用深度网络提取特征
- 特征送入每一类的SVM 分类器，判别是否属于该类

- 使用回归器精细修正候选框位置



候选区域生成

使用了Selective Search1方法从一张图像生成约2000-3000个候选区域。基本思路如下：
- 使用一种过分割手段，将图像分割成小区域
- 查看现有小区域，合并可能性最高的两个区域。重复直到整张图像合并成一个区域位置
- 输出所有曾经存在过的区域，所谓候选区域

候选区域生成和后续步骤相对独立，实际可以使用任意算法进行。

合并规则

优先合并以下四种区域：
- 颜色（颜色直方图）相近的
- 纹理（梯度直方图）相近的
- 合并后总面积小的
- 合并后，总面积在其BBOX中所占比例大的

第三条，保证合并操作的尺度较为均匀，避免一个大区域陆续“吃掉”其他小区域。

> 例：设有区域a-b-c-d-e-f-g-h。较好的合并方式是：ab-cd-ef-gh -> abcd-efgh -> abcdefgh。
> 不好的合并方法是：ab-c-d-e-f-g-h ->abcd-e-f-g-h ->abcdef-gh -> abcdefgh。

第四条，保证合并后形状规则。

例：左图适于合并，右图不适于合并。

上述四条规则只涉及区域的颜色直方图、纹理直方图、面积和位置。合并后的区域特征可以直接由子区域特征计算而来，速度较快。

多样化与后处理

为尽可能不遗漏候选区域，上述操作在多个颜色空间中同时进行（RGB,HSV,Lab等）。在一个颜色空间中，使用上述四条规则的不同组合进行合并。所有颜色空间与所有规则的全部结果，在去除重复后，都作为候选区域输出。

作者提供了Selective Search的源码，内含较多.p文件和.mex文件，难以细查具体实现。

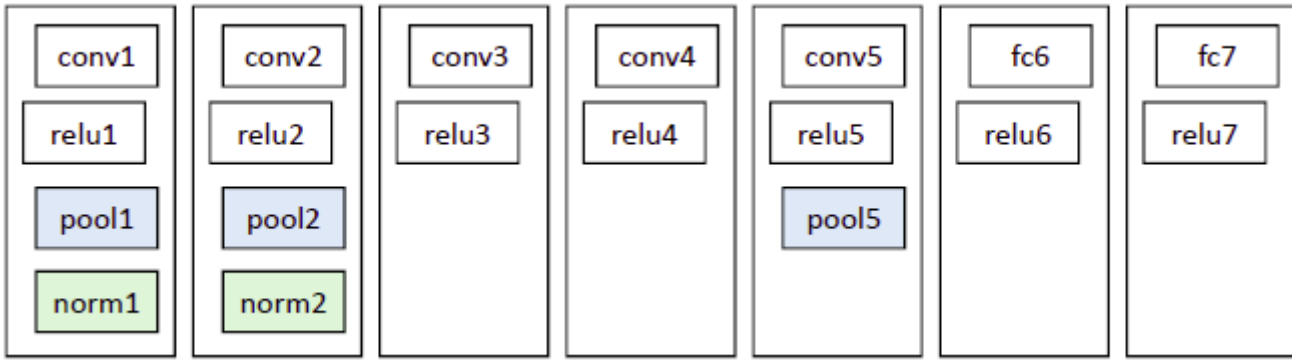
特征提取

预处理

使用深度网络提取特征之前，首先把候选区域归一化成同一尺寸227×227。
此处有一些细节可做变化：外扩的尺寸大小，形变时是否保持原比例，对框外区域直接截取还是补灰。会轻微影响性能。

预训练

网络结构
基本借鉴Hinton 2012年在Image Net上的分类网络2，略作简化3。



此网络提取的特征为4096维，之后送入一个4096->1000的全连接(fc)层进行分类。
学习率0.01。

训练数据
使用ILVCR 2012的全部数据进行训练，输入一张图片，输出1000维的类别标号。

调优训练

网络结构
同样使用上述网络，最后一层换成4096->21的全连接网络。
学习率0.001，每一个batch包含32个正样本（属于20类）和96个背景。

训练数据
使用PASCAL VOC 2007的训练集，输入一张图片，输出21维的类别标号，表示20类+背景。
考察一个候选框和当前图像上所有标定框重叠面积最大的一个。如果重叠比例大于0.5，则认为此候选框为此标定的类别；否则认为此候选框为背景。

类别判断

分类器
对每一类目标，使用一个线性SVM二类分类器进行判别。输入为深度网络输出的4096维特征，输出是否属于此类。
由于负样本很多，使用hard negative mining方法。
正样本
本类的真值标定框。
负样本
考察每一个候选框，如果和本类所有标定框的重叠都小于0.3，认定其为负样本

位置精修

目标检测问题的衡量标准是重叠面积：许多看似准确的检测结果，往往因为候选框不够准确，重叠面积很小。故需要一个位置精修步骤。
回归器
对每一类目标，使用一个线性脊回归器进行精修。正则项λ=10000。
输入为深度网络pool5层的4096维特征，输出为xy方向的缩放和平移。
训练样本
判定为本类的候选框中，和真值重叠面积大于0.6的候选框。

结果

论文发表的2014年，DPM已经进入瓶颈期，即使使用复杂的特征和结构得到的提升也十分有限。本文将深度学习引入检测领域，一举将PASCAL VOC上的检测率从35.1%提升到53.7%。
本文的前两个步骤（候选区域提取+特征提取）与待检测类别无关，可以在不同类之间共用。这两步在GPU上约需13秒。
同时检测多类时，需要倍增的只有后两步骤（判别+精修），都是简单的线性运算，速度很快。这两步对于100K类别只需10秒。

二、RCNN

一、相关理论

本篇博文主要讲解2014年CVPR上的经典paper：《Rich feature hierarchies for Accurate Object Detection and Segmentation》，这篇文章的算法思想又被称之为：R-CNN（Regions with Convolutional Neural Network Features），是物体检测领域曾经获得state-of-art精度的经典文献。

这篇paper的思想，改变了物体检测的总思路，现在好多文献关于深度学习的物体检测的算法，基本上都是继承了这个思想，比如：《Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition》，所以学习经典算法，有助于我们以后搞物体检测的其它paper。

之前刚开始接触物体检测算法的时候，老是分不清deep learning中，物体检测和图片分类算法上的区别，弄得我头好晕，终于在这篇paper上，看到了解释。物体检测和图片分类的区别：图片分类不需要定位，而物体检测需要定位出物体的位置，也就是相当于把物体的bbox检测出来，还有一点物体检测是要把所有图片中的物体都识别定位出来。

(笔记后感by ym：

个人理解testing整个流程即：先将region通过ss检测出来(2k+)，然后根据cnn提取的region特征丢入svm进行分类(compute score)，得到的就是一个region-bbox以及对应的类别,再利用(IoU->nms)得到具体的框，目的防止泛滥，为了精确bbox，再根据pool5 feature做了个bbox regression来decrease location error，training的trick则为hnm + finetuning)

Selective Search

因为研究RCNN的需要，在这里看一下Selective Search的操作流程

原文链接：http://koen.me/research/pub/uijlings-ijcv2013-draft.pdf

选择性搜索遵循如下的原则：

1. 图片中目标的尺寸不一，边缘清晰程度也不一样，选择性搜索应该能够将所有的情况都考虑进去，如下图，最好的办法就是使用分层算法来实现
2. 区域合并的算法应该多元化。初始的小的图像区域（Graph-Based Image Segmentation得到）可能是根据颜色、纹理、部分封闭等原因得到的，一个单一的策略很难能适应所有的情况将小区域合并在一起，因此需要有一个多元化的策略集，能够在不同场合都有效。
3. 能够快速计算。

二、基础知识

1、有监督预训练与无监督预训练

(1)无监督预训练(Unsupervised pre-training)

无监督预训练这个名词我们比较熟悉，栈式自编码、DBM采用的都是采用无监督预训练。因为预训练阶段的样本不需要人工标注数据，所以就叫做无监督预训练。

(2)有监督预训练(Supervised pre-training)

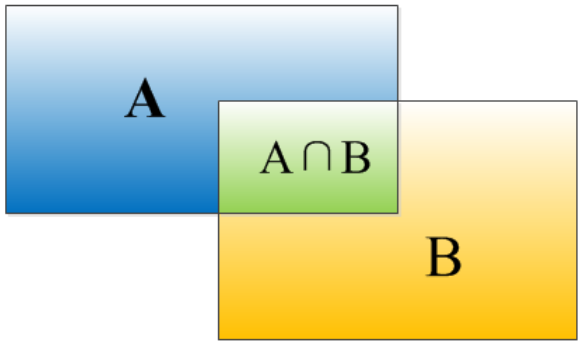
所谓的有监督预训练，我们也可以把它称之为迁移学习。比如你已经有一大堆标注好的人脸年龄分类的图片数据，训练了一个CNN，用于人脸的年龄识别。然后当你遇到新的项目任务是：人脸性别识别，那么这个时候你可以利用已经训练好的年龄识别CNN模型，去掉最后一层，然后其它的网络层参数就直接复制过来，继续进行训练。这就是所谓的迁移学习，说的简单一点就是把一个任务训练好的参数，拿到另外一个任务，作为神经网络的初始参数值,这样相比于你直接采用随机初始化的方法，精度可以有很大的提高。

图片分类标注好的训练数据非常多，但是物体检测的标注数据却很少，如何用少量的标注数据，训练高质量的模型，这就是文献最大的特点，这篇paper采用了迁移学习的思想。文献就先用了ILSVRC2012这个训练数据库（这是一个图片分类训练数据库），先进行网络的图片分类训练。这个数据库有大量的标注数据，共包含了1000种类别物体，因此预训练阶段cnn模型的输出是1000个神经元，或者我们也直接可以采用Alexnet训练好的模型参数。

2、IOU的定义

因为没有搞过物体检测不懂IOU这个概念，所以就简单介绍一下。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：

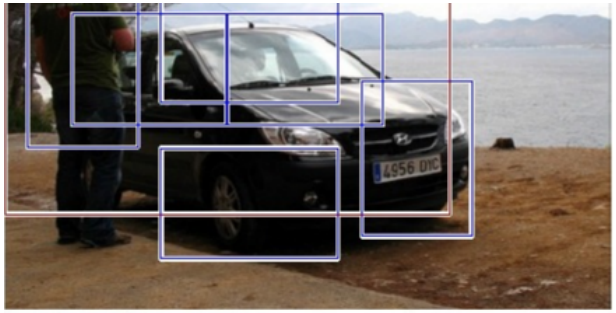
IOU=(A∩B)/(A∪B)

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)

3、非极大值抑制

因为一会儿讲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。

非极大值抑制（NMS）非极大值抑制顾名思义就是抑制不是极大值的元素，搜索局部的极大值。这个局部代表的是一个邻域，邻域有两个参数可变，一是邻域的维数，二是邻域的大小。这里不讨论通用的NMS算法，而是用于在目标检测中用于提取分数最高的窗口的。例如在行人检测中，滑动窗口经提取特征，经分类器分类识别后，每个窗口都会得到一个分数。但是滑动窗口会导致很多窗口与其他窗口存在包含或者大部分交叉的情况。这时就需要用到NMS来选取那些邻域里分数最高（是行人的概率最大），并且抑制那些分数低的窗口。

---

canny detection(canny NMS)：

对梯度幅值进行非极大值抑制

图像梯度幅值矩阵中的元素值越大，说明图像中该点的梯度值越大，但这不不能说明该点就是边缘（这仅仅是属于图像增强的过程）。在Canny算法中，非极大值抑制是进行边缘检测的重要步骤，**通俗意义上是指寻找像素点局部最大值，将非极大值点所对应的灰度值置为0**，这样可以剔除掉一大部分非边缘的点（这是本人的理解）。

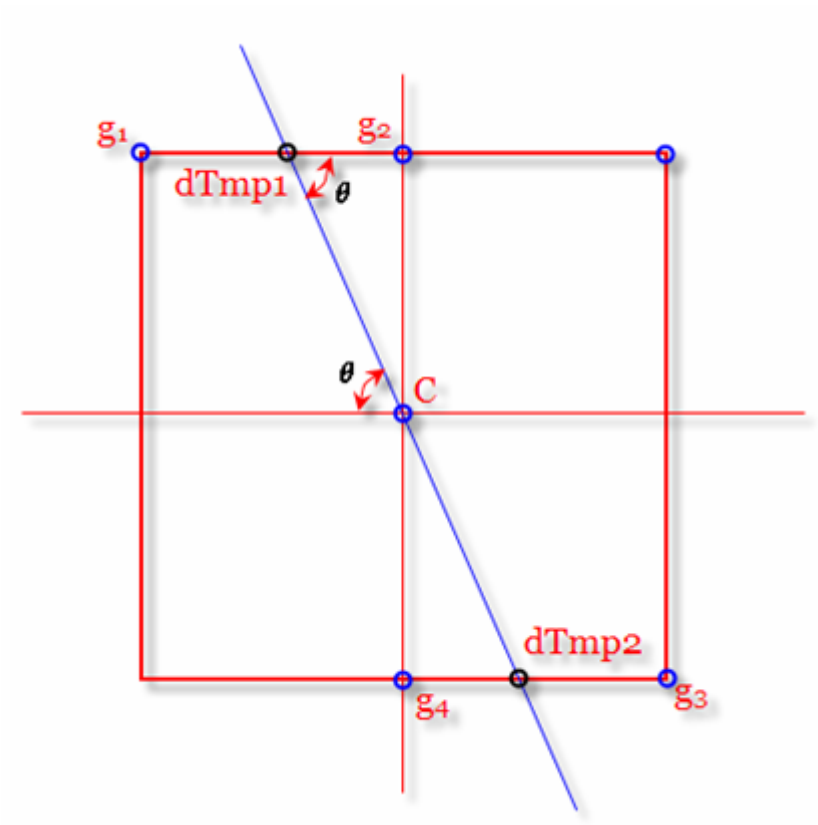


图1 非极大值抑制原理

根据图1 可知，要进行非极大值抑制，就首先要确定像素点C的灰度值在其8值邻域内是否为最大。图1中蓝色的线条方向为C点的梯度方向，这样就可以确定其局部的最大值肯定分布在这条线上，也即出了C点外，梯度方向的交点dTmp1和dTmp2这两个点的值也可能会是局部最大值。因此，判断C点灰度与这两个点灰度大小即可判断C点是否为其邻域内的局部最大灰度点。如果经过判断，C点灰度值小于这两个点中的任一个，那就说明C点不是局部极大值，那么则可以排除C点为边缘。这就是非极大值抑制的工作原理。

作者认为，在理解的过程中需要注意以下两点：

1）中非最大抑制是回答这样一个问题：“当前的梯度值在梯度方向上是一个局部最大值吗？” 所以,要把当前位置的梯度值与梯度方向上两侧的梯度值进行比较；

2）梯度方向垂直于边缘方向。

但实际上，我们只能得到C点邻域的8个点的值，而dTmp1和dTmp2并不在其中，要得到这两个值就需要对该两个点两端的已知灰度进行线性插值，也即根据图1中的g1和g2对dTmp1进行插值，根据g3和g4对dTmp2进行插值，这要用到其梯度方向，这是上文Canny算法中要求解梯度方向矩阵Thita的原因。

完成非极大值抑制后，会得到一个二值图像，非边缘的点灰度值均为0，可能为边缘的局部灰度极大值点可设置其灰度为128。根据下文的具体测试图像可以看出，这样一个检测结果还是包含了很多由噪声及其他原因造成的假边缘。因此还需要进一步的处理。

4、VOC物体检测任务

这个就相当于一个竞赛，里面包含了20个物体类别：http://host.robots.ox.ac.uk/pascal/VOC/voc2012/examples/index.html 还有一个背景，总共就相当于21个类别，因此一会设计fine-tuning CNN的时候，我们softmax分类输出层为21个神经元。

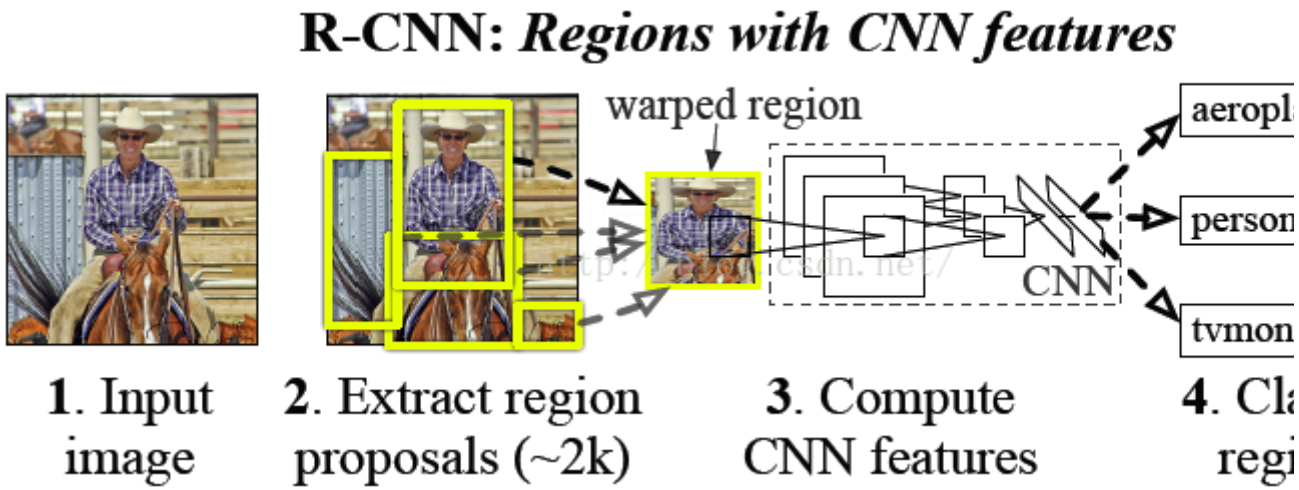
三、算法总体思路

开始讲解paper前，我们需要先把握总体思路，才容易理解paper的算法。

(作者通过在recongnition using regions操作的方法来解决CNN的定位问题，这个方法在目标检测和语义分割中都取得了成功。测试阶段，这个方法对每一个输入的图片产生近2000个不分种类的“region proposals，使用CNNs从每个region proposals”中提取一个固定长度的特征向量，然后对每个region proposal(SS for Detction)提取的特征向量使用特定种类的线性SVM进行分类(CNN+SVM for classification)。)

图片分类与物体检测不同，物体检测需要定位出物体的位置，这种就相当于回归问题，求解一个包含物体的方框。而图片分类其实是逻辑回归。这种方法对于单物体检测还不错，但是对于多物体检测就……

因此paper采用的方法是：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：



后面我们将根据这三个过程，进行每个步骤的详细讲解。

四、候选框搜索阶段

(作者也考虑过使用一个滑动窗口的方法，然而由于更深的网络，更大的输入图片和滑动步长，使得使用滑动窗口来定位的方法充满了挑战)

1、实现方式

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：《search for object recognition》，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，具体不再啰嗦。

OK，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。SVM阶段的正负样本标签问题，等到了svm讲解阶段我再具体讲解。

五、CNN特征提取阶段

1、算法实现

a、网络结构设计阶段

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。

b、网络有监督预训练阶段

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，如果要直接采用随机初始化CNN参数的方法，那么目前的训练数据量是远远不够的。这种情况下，最好的是采用某些方法，把参数初始化了，然后在进行有监督的参数微调，这边文献采用的是有监督的预训练。所以paper在设计网络结构的时候，是直接用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；

C、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) (20 + 1bg)，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个事正样本、96个事负样本（正负样本的定义前面已经提过，不再解释）。

2、问题解答

OK，看完上面的CNN过程后，我们会有一些细节方面的疑问。首先，反正CNN都是用于提取特征，那么我直接用Alexnet做特征提取，省去fine-tuning阶段可以吗？这个是可以的，你可以不需重新训练CNN，直接采用Alexnet模型，提取出p5、或者f6、f7的特征，作为特征向量，然后进行训练svm，只不过这样精度会比较低。那么问题又来了，没有fine-tuning的时候，要选择哪一层的特征作为cnn提取到的特征呢？我们有可以选择p5、f6、f7，这三层的神经元个数分别是9216、4096、4096。从p5到p6这层的参数个数是：4096*9216 ，从f6到f7的参数是4096*4096。那么具体是选择p5、还是f6，又或者是f7呢？

文献paper给我们证明了一个理论，如果你不进行fine-tuning，也就是你直接把Alexnet模型当做万金油使用，类似于HOG、SIFT一样做特征提取，不针对特定的任务。然后把提取的特征用于分类，结果发现p5的精度竟然跟f6、f7差不多，而且f6提取到的特征还比f7的精度略高；如果你进行fine-tuning了，那么f7、f6的提取到的特征最会训练的svm分类器的精度就会飙涨。

据此我们明白了一个道理，如果不针对特定任务进行fine-tuning，而是把CNN当做特征提取器，卷积层所学到的特征其实就是基础的共享特征提取层，就类似于SIFT算法一样，可以用于提取各种图片的特征，而f6、f7所学习到的特征是用于针对特定任务的特征。
打个比方：对于人脸性别识别来说，一个CNN模型前面的卷积层所学习到的特征就类似于学习人脸共性特征，然后全连接层所学习的特征就是针对性别分类的特征了。

还有另外一个疑问：CNN训练的时候，本来就是对bounding box的物体进行识别分类训练，是一个端到端的任务，在训练的时候最后一层softmax就是分类层.

那么为什么作者闲着没事干要先用CNN做特征提取（提取fc7层数据），然后再把提取的特征用于训练svm分类器？

这个是因为svm训练和cnn训练过程的正负样本定义方式各有不同，导致最后采用CNN softmax输出比采用svm精度还低。

事情是这样的，cnn在训练的时候，对训练数据做了比较宽松的标注，比如一个bounding box可能只包含物体的一部分，那么我也把它标注为正样本，用于训练cnn；采用这个方法的主要原因在于因为CNN容易过拟合，所以需要大量的训练数据，所以在CNN训练阶段我们是对Bounding box的位置限制条件限制的比较松(IOU只要大于0.5都被标注为正样本了)；

然而svm训练的时候，因为svm适用于少样本训练，所以对于训练样本数据的IOU要求比较严格，我们只有当bounding box把整个物体都包含进去了，我们才把它标注为物体类别，然后训练svm，具体请看下文。

六、SVM训练、测试阶段

这是一个二分类问题，我么假设我们要检测车辆。我们知道只有当bounding box把整量车都包含在内，那才叫正样本；如果bounding box 没有包含到车辆，那么我们就可以把它当做负样本。但问题是当我们的检测窗口只有部分包好物体，那该怎么定义正负样本呢？作者测试了IOU阈值各种方案数值0,0.1,0.2,0.3,0.4,0.5。最后我们通过训练发现，如果选择IOU阈值为0.3效果最好（选择为0精度下降了4个百分点，选择0.5精度下降了5个百分点）,即当

重叠度小于0.3的时候，我们就把它标注为负样本。一旦CNN f7层特征被提取出来，那么我们将为每个物体累训练一个svm分类器。当我们用CNN提取2000个候选框，可以得到2000*4096这样的特征向量矩阵，然后我们只需要把这样的一个矩阵与svm权值矩阵4096*N点（Therefore，the pool5 need to be set as）乘(N为分类类别数目，因为我们训练的N个svm，每个svm包好了4096个W)，就可以得到结果了。

参考文献：

1、《Rich feature hierarchies for Accurate Object Detection and Segmentation》

2、《Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition》

三、fast-RCNN

SPP已有一定的速度提升，它在ConvNet的最后一个卷积层才提取proposal，但是依然有不足之处。和R-CNN一样，它的训练要经过多个阶段，特征也要存在磁盘中，另外，SPP中的微调只更新spp层后面的全连接层，对很深的网络这样肯定是不行的。

在微调阶段谈及SPP-net只能更新FC层,这是因为卷积特征是线下计算的，从而无法再微调阶段反向传播误差。而在fast-RCNN中则是通过image-centric sampling提高了卷积层特征抽取的速度，从而保证了梯度可以通过SPP层（即ROI pooling层）反向传播。

Fast-Rcnn 改进：

1. 比R-CNN更高的检测质量（mAP）；
2. 把多个任务的损失函数写到一起，实现单级的训练过程；
3. 在训练时可更新所有的层；
4. 不需要在磁盘中存储特征。

解决方式具体即以下几点:
1.训练的时候，pipeline是隔离的，先提proposal，然后CNN提取特征，之后用SVM分类器，最后再做bbox regression。FRCN实现了end-to-end的joint training(提proposal阶段除外)。
2.训练时间和空间开销大。RCNN中ROI-centric的运算开销大，所以FRCN用了image-centric的训练方式来通过卷积的share特性来降低运算开销；RCNN提取特征给SVM训练时候需要中间要大量的磁盘空间存放特征，FRCN去掉了SVM这一步，所有的特征都暂存在显存中，就不需要额外的磁盘空间了。
3.测试时间开销大。依然是因为ROI-centric的原因(whole image as input->ss region映射)，这点SPP-Net已经改

进，然后FRCN进一步通过single scale(pooling->spp just for one scale) testing和SVD(降维)分解全连接来提速。

整体框架

整体框架如Figure 1，如果以AlexNet（5个卷积和3个全连接）为例，大致的训练过程可以理解为：
1.selective search在一张图片中得到约2k个object proposal(这里称为RoI)

2.缩放图片的scale得到图片金字塔，FP得到conv5的特征金字塔。
3.对于每个scale的每个ROI，求取映射关系，在conv5中crop出对应的patch。并用一个单层的SPP layer（这里称为RoI pooling layer）来统一到一样的尺度（对于AlexNet是6x6）。
4.继续经过两个全连接得到特征，这特征有分别share到两个新的全连接，连接上两个优化目标。第一个优化目标是分类，使用softmax，第二个优化目标是bbox regression，使用了一个smooth的L1-loss.
(除了1，上面的2-4是joint training的。测试时候，在4之后做一个NMS即可。)

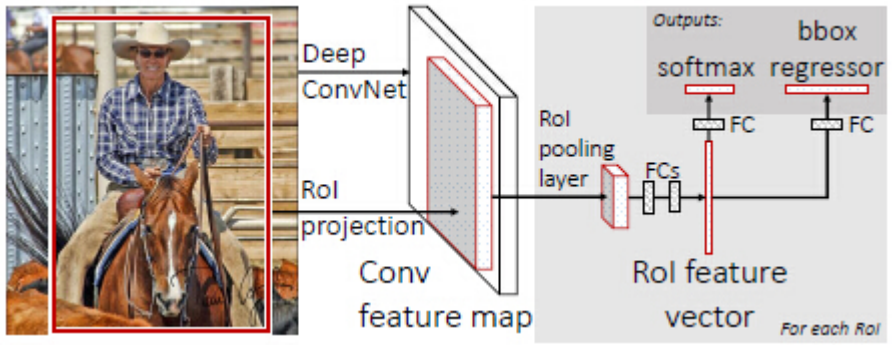


Figure 1. Fast R-CNN architecture. An input image and multiple regions of interest (RoIs) are input into a fully-convolutional network. Each RoI is pooled into a fixed-size feature map and then mapped to a feature vector by fully-connected layers (FCs). The network has two output vectors per RoI: softmax probabilities and per-class bounding-box regression offsets. The architecture is trained end-to-end with a multi-task loss.

整体框架大致如上述所示

再次几句话总结：

1.用selective search在一张图片中生成约2000个object proposal，即RoI。

2.把它们整体输入到全卷积的网络中，在最后一个卷积层上对每个ROI求映射关系，并用一个RoI pooling layer来统一到相同的大小 - > (fc)feature vector 即 - >提取一个固定维度的特征表示。
3.继续经过两个全连接层（FC）得到特征向量。特征向量经由各自的FC层，得到两个输出向量：
第一个是分类，使用softmax，第二个是每一类的bounding box回归。

按照论文所述即：

one that produces softmax probability estimates overKobject classes plus a catch-all “background” class and another layer that outputs four real-valued numbers for each of theKobject classes

对比回来SPP-Net，可以看出**FRCN大致就是一个joint training版本的SPP-Net**，改进如下：

1.改进了SPP-Net在实现上无法同时tuning在SPP layer两边的卷积层和全连接层。

只能更新fc层的原因->按照论文所描述:

the root cause is that back-propagation through the SPPlayer is highly inefficient when each training sample (*i.e*.RoI) comes from a different image, which is exactly howR-CNN and SPPnet networks are trained.
The inefficiency stems from the fact that each RoI mayhave a very large receptive field, often spanning the entire input image. Since the forward pass must process the entire receptive field, the training inputs are large (often the entire image).
2.SPP-Net后面的需要将第二层FC的特征放到硬盘上训练SVM，之后再额外训练bbox regressor。

接下来会介绍FRCN里面的一些细节的motivation和效果。

RoI pooling layer

这是SPP pooling层的一个简化版，只有一级“金字塔”，输入是N个特征映射和一组R个RoI，R>>N。N个特征映射来自于最后一个卷积层，每个特征映射都是H x W⊴ x C的大小。每个RoI是一个元组(n, r, c, h, w)，n是特征映射的索引，n∈{0, ... ,N-1}，(r, c)是RoI左上角的坐标，(h, w)是高与宽。输出是max-pool过的特征映射，H' x W' x C的

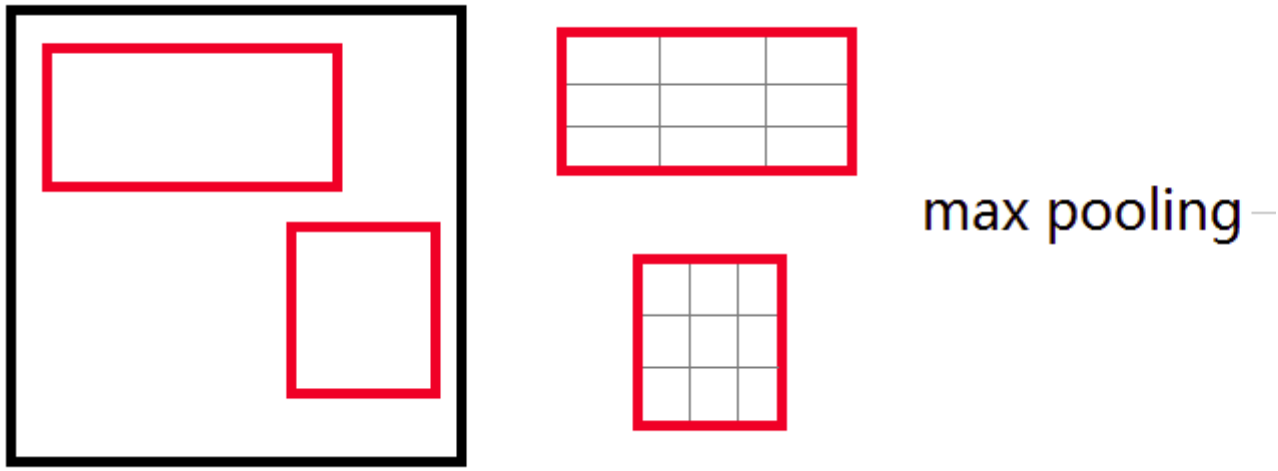
大小，H' ≤ H，W' ≤ W。对于RoI，bin-size ~ h/H' x w/W'，这样就有H'W'个输出bin，bin的大小是自适应的，取决于RoI的大小。

RoI pooling layer的作用主要有两个：
1.是将image中的rol定位到feature map中对应patch
2.是用一个单层的SPP layer将这个feature map patch下采样为大小固定的feature再传入全连接层。即
RoI pooling layer来统一到相同的大小 - > (fc)feature vector 即 - >提取一个固定维度的特征表示。

这里有几个细节:
1.对于某个rol，怎么求取对应的feature map patch？这个论文没有提及，笔者也觉得应该与spp-net的映射关系一致
2.为何只是一层的SPP layer？多层的SPP layer不会更好吗？对于这个问题，笔者认为是因为需要读取pretrain model来finetuning的原因，比如VGG就release了一个19层的model，如果是使用多层的SPP layer就不能够直接使用这个model的parameters，而需要重新训练了。

Roi_pool层的测试(forward)

roi_pool层将每个候选区域均匀分成M×N块，对每块进行max pooling。将特征图上大小不一的候选区域转变为大小统一的数据，送入下一层。



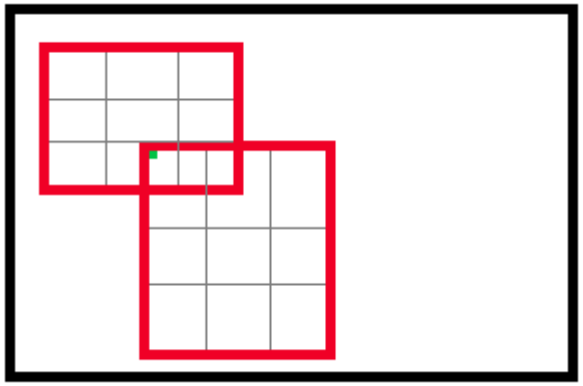
Roi_pool层的训练(backward)

首先考虑普通max pooling层。设xi为输入层的节点，yj为输出层的节点。

$$\frac{\partial L}{\partial x_i}=\begin{cases}0 & \delta(i,j)=\text{false}\\ \frac{\partial L}{\partial y_j} & \delta(i,j)=\text{true}\end{cases}$$

其中判决函数δ(i,j)表示i节点是否被j节点选为最大值输出。不被选中有两种可能：xi不在yj范围内，或者xi不是最大值。

对于roi max pooling，一个输入节点可能和多个输出节点相连。设xi为输入层的节点，yrj为第r个候选区域的第j个输出节点。



$$\frac{\partial L}{\partial x_i}=\sum_{r,j}\delta(i,r,j)\frac{\partial L}{\partial y_{rj}}$$

判决函数δ(i,r,j)表示i节点是否被候选区域r的第j个节点选为最大值输出。代价对于xi的梯度等于所有相关的后一层梯度之和。

Pre-trained networks

用了3个预训练的ImageNet网络（CaffeNet/VGG_CNN_M_1024/VGG16）。
预训练的网络初始化Fast RCNN要经过三次变形：
1. 最后一个max pooling层替换为RoI pooling层，设置H'和W'与第一个全连接层兼容。
(SPPnet for one scale -> arbitrary input image size)
2. 最后一个全连接层和softmax（原本是1000个类）-> 替换为softmax的对K+1个类别的分类层，和bounding box 回归层。
(Cls and Det at same time)
3. 输入修改为两种数据：一组N个图形，R个RoI，batch size和ROI数、图像分辨率都是可变的。

Fine-tuning

前面说过SPPnet有一个缺点是只能微调spp层后面的全连接层，所以SPPnet就可以采用随机梯度下降（SGD）来训练。
RCNN:无法同时tuning在SPP layer两边的卷积层和全连接层
RoI-centric sampling：从所有图片的所有RoI中均匀取样，这样每个SGD的mini-batch中包含了不同图像中的样本。（SPPnet采用）
FRCN想要解决微调的限制,就要反向传播到spp层之前的层->(reason)反向传播需要计算每一个RoI感受野的卷积层，通常会覆盖整个图像，如果一个一个用RoI-centric sampling的话就又慢又耗内存。

Fast RCNN:->改进了SPP-Net在实现上无法同时tuning在SPP layer两边的卷积层和全连接层

image-centric sampling： (solution)mini-batch采用层次取样，先对图像取样，再对RoI取样，同一图像的RoI共享计算和内存。
另外，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）。

Multi-task loss

两个输出层，一个对每个RoI输出离散概率分布：$p = (p_0, \ldots, p_K)$
一个输出bounding box回归的位移：$t^k = \left(t^k_{\mathrm{x}}, t^k_{\mathrm{y}}, t^k_{\mathrm{w}}, t^k_{\mathrm{h}}\right)$
k表示类别的索引，前两个参数是指相对于object proposal尺度不变的平移，后两个参数是指对数空间中相对于object proposal的高与宽。把这两个输出的损失写到一起：

$$L(p, k^*, t, t^*) = L_{\text{cls}}(p, k^*) + \lambda[k^* \geq 1]L_{\text{loc}}(t, t^*), \quad (1)$$

k*是真实类别，式中第一项是分类损失，第二项是定位损失，L由R个输出取均值而来.

以下具体介绍:

1.对于分类loss，是一个N+1路的softmax输出，其中的N是类别个数，1是背景。为何不用SVM做分类器了？在5.4作者讨论了softmax效果比SVM好，因为它引入了类间竞争。（笔者觉得这个理由略牵强，估计还是实验效果验证了softmax的performance好吧 ^_^）
2.对于回归loss，是一个4xN路输出的regressor，也就是说对于每个类别都会训练一个单独的regressor的意思，比较有意思的是，这里regressor的loss不是L2的，而是一个平滑的L1，形式如下：

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

in which

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

作者这样设置的目的是想让loss对于离群点更加鲁棒，控制梯度的量级使得训练时不容易跑飞。
最后在5.1的讨论中，作者说明了Multitask loss是有助于网络的performance的。

Mini-batch sampling

在微调时，每个SGD的mini-batch是随机找两个图片，R为128，因此每个图上取样64个RoI。从object proposal中选25%的RoI，就是和ground-truth交叠至少为0.5的。剩下的作为背景。

分层数据

在调优训练时，每一个mini-batch中首先加入N张完整图片，而后加入从N张图片中选取的R个候选框。这R个候选框可以复用N张图片前5个阶段的网络特征。

实际选择N=2， R=128 - > 每一个mini-batch中首先加入2张完整图片，而后加入从2张图片中选取的128个候选框。这128个候选框可以复用2张图片前5个阶段的网络特征。

训练数据构成

N张完整图片以50%概率水平翻转。
R个候选框的构成方式如下：

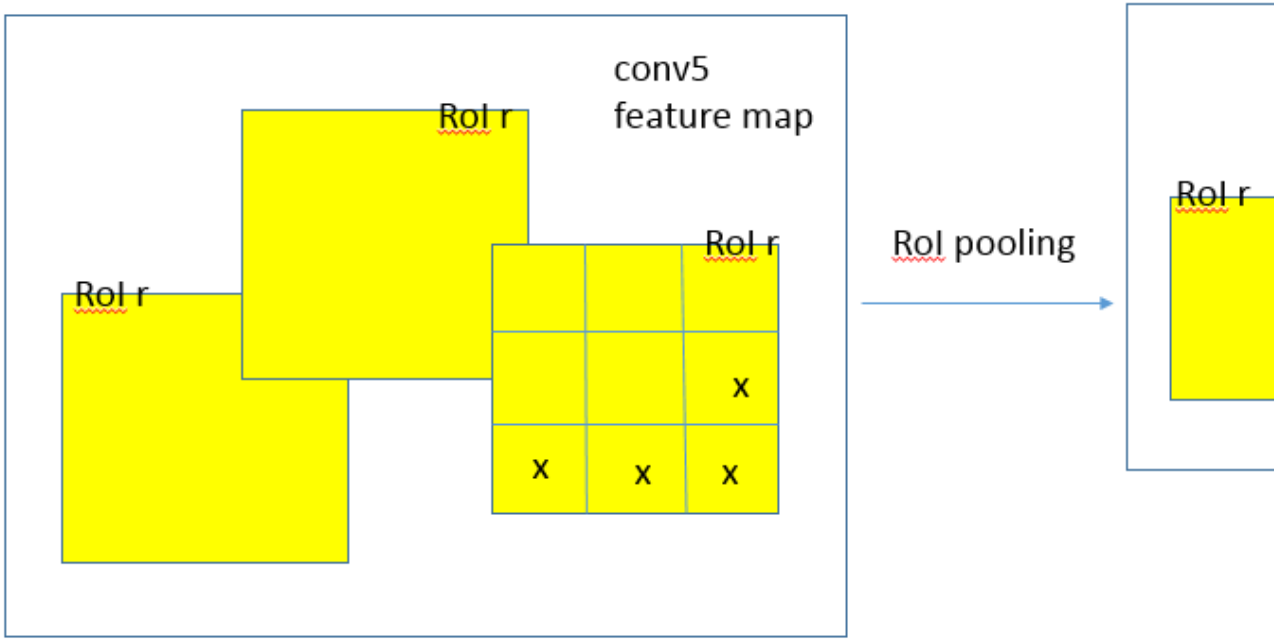
类别	比例	方式
前景	25%	与某个真值重叠在[0.5,1]的候选框
背景	75%	与真值重叠的最大值在[0.1,0.5)的候选框

Backpropagation through RoI pooling layers

RoI pooling层计算损失函数对每个输入变量x的偏导数，如下：

$$\frac{\partial L}{\partial x} = \sum_{r \in R} \sum_{y \in r} [y \text{ pooled } x] \frac{\partial L}{\partial y}.$$

y是pooling后的输出单元，x是pooling前的输入单元，如果y由x pooling而来，则将损失L对y的偏导计入累加值，最后累加完R个RoI中的所有输出单元。下面是我理解的x、y、r的关系：



Scale invariance

SPPnet用了两种实现尺度不变的方法：
1. brute force（single scale），直接将image设置为某种scale，直接输入网络训练，期望网络自己适应这个scale。
2. image pyramids（multi scale），生成一个图像金字塔，在multi-scale训练时，对于要用的RoI，在金字塔上找到一个最接近227x227的尺寸，然后用这个尺寸训练网络。
虽然看起来2比较好，但是非常耗时，而且性能提高也不对，大约只有%1，所以这篇论文在实现中还是用了1。

Which layers to finetune?

对应文中4.5，作者的观察有2点

1. 对于较深的网络，比如VGG，卷积层和全连接层是否一起tuning有很大的差别（66.9 vs 61.4）

2. 有没有必要tuning所有的卷积层？
   答案是没有。如果留着浅层的卷积层不tuning，可以减少训练时间，而且mAP基本没有差别。

Truncated SVD for faster detection

在分类中，计算全连接层比卷积层快，而在检测中由于一个图中要提取2000个RoI，所以大部分时间都用在计算全连接层了。文中采用奇异值分解的方法来减少计算fc层的时间.

具体来说，作者对全连接层的矩阵做了一个SVD分解，mAP几乎不怎么降（0.3%），但速度提速30%

全连接层提速

分类和位置调整都是通过全连接层(fc)实现的，设前一级数据为x后一级为y，全连接层参数为W，尺寸u×v。一次前向传播(forward)即为：

$y=Wx$

计算复杂度为u×v。

将W进行SVD分解，并用前t个特征值近似：

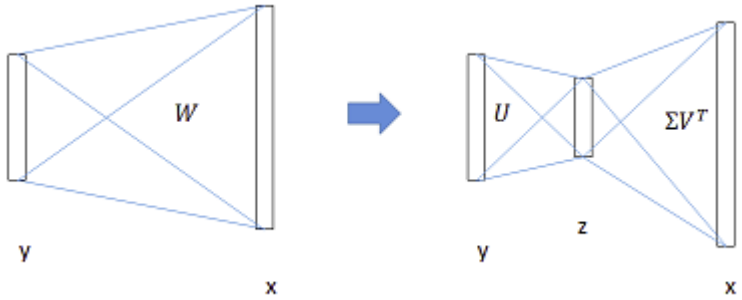
$W=U\Sigma VT\approx U(:,1:t)\cdot\Sigma(1:t,1:t)\cdot V(:,1:t)T$

原来的前向传播分解成两步：

$y=Wx=U\cdot(\Sigma\cdot VT)\cdot x=U\cdot z$

计算复杂度变为u×t+v×t。

在实现时，相当于把一个全连接层拆分成两个，中间以一个低维数据相连。



Data augment

在训练期间，作者做过的唯一一个数据增量的方式是水平翻转。
作者也试过将VOC12的数据也作为拓展数据加入到finetune的数据中，结果VOC07的mAP从66.9到了70.0，说明对于网络来说，数据越多就是越好的。

实验与结论

实验过程不再详述，只记录结论
- 网络末端同步训练的分类和位置调整，提升准确度
- 使用多尺度的图像金字塔，性能几乎没有提高
- 倍增训练数据，能够有2%-3%的准确度提升
- 网络直接输出各类概率(softmax)，比SVM分类器性能略好
- 更多候选窗不能提升性能

results

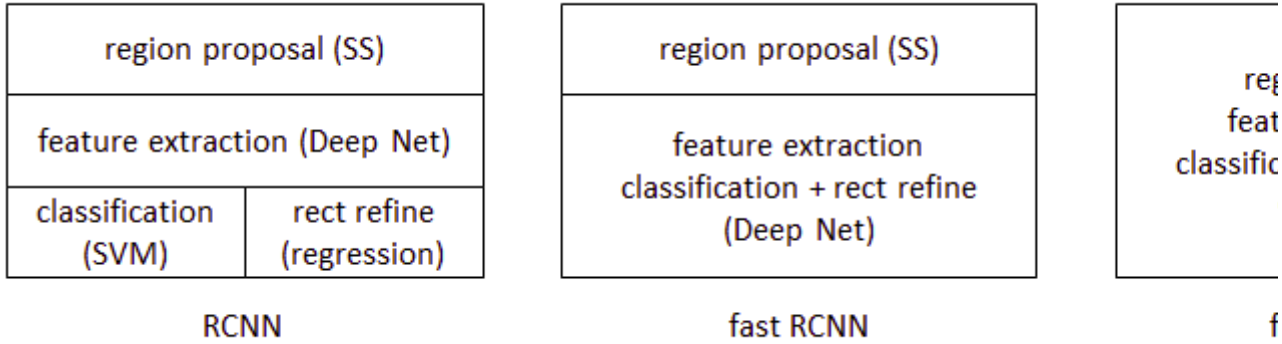
for VOC2007

method	mAP S M L	train time(h) S M L	test rate (s/im) S M L
SPPnet BB	— — 63.1	— — 25	— — 2.3
R-CNN BB	58.5 60.2 66.0	22 28 84	9.8 12.1 47.0
FRCN	57.1 59.2 66.9	1.2 2.0 9.5	0.10 0.15 0.32

四、faster-RCNN

思想

从RCNN到fast RCNN，再到本文的faster RCNN，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。

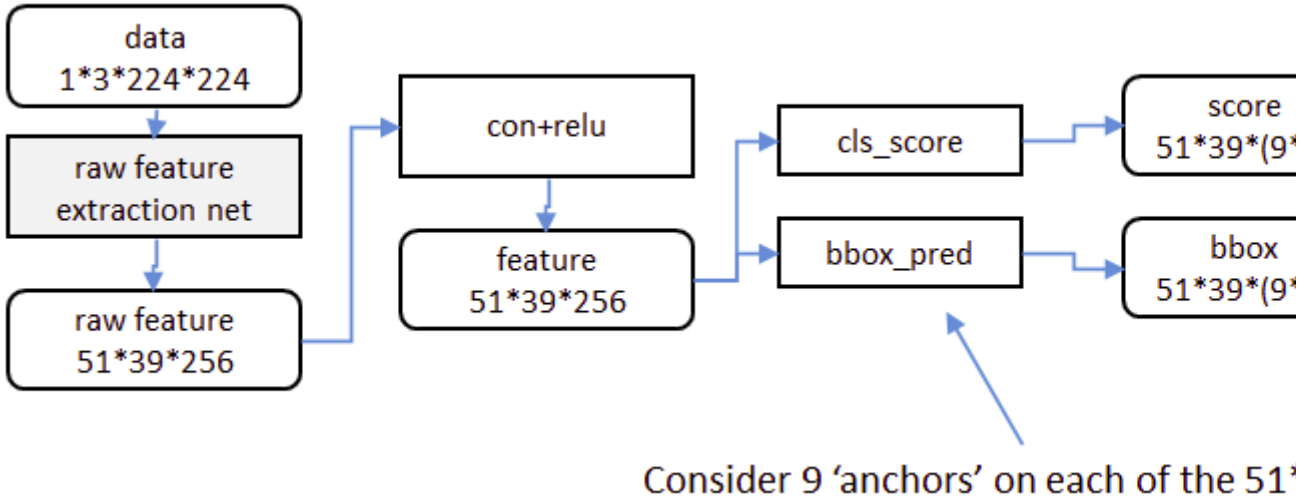


faster RCNN可以简单地看做“区域生成网络+fast RCNN“的系统，用区域生成网络代替fast RCNN中的Selective Search方法。本篇论文着重解决了这个系统中的三个问题：

1. 如何设计区域生成网络
2. 如何训练区域生成网络
3. 如何让区域生成网络和fast RCNN网络共享特征提取网络

区域生成网络：结构

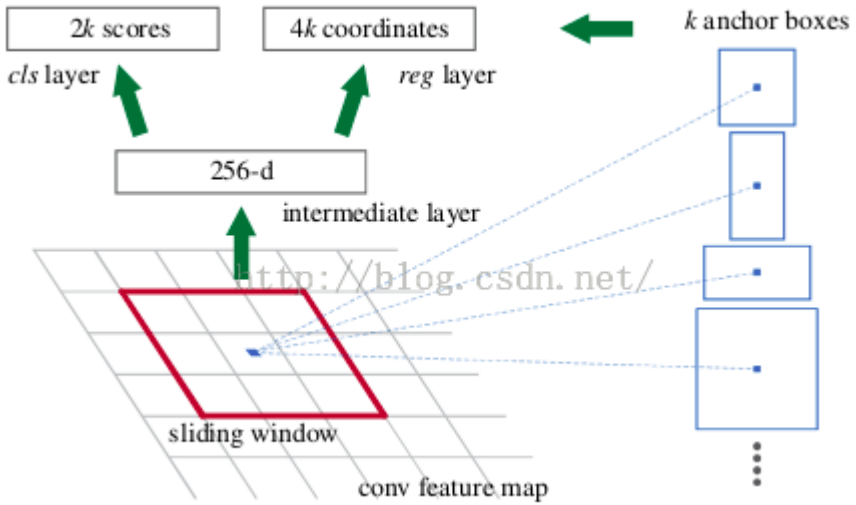
基本设想是：在提取好的特征图上，对所有可能的候选框进行判别。由于后续还有位置精修步骤，所以候选框实际比较稀疏。



特征提取

原始特征提取（上图灰色方框）包含若干层conv+relu，直接套用ImageNet上常见的分类网络即可。本文试验了两种网络：5层的ZF[3]，16层的VGG-16[4]，具体结构不再赘述。
额外添加一个conv+relu层，输出51*39*256维特征（feature）。

Region Proposal Networks的设计和训练思路

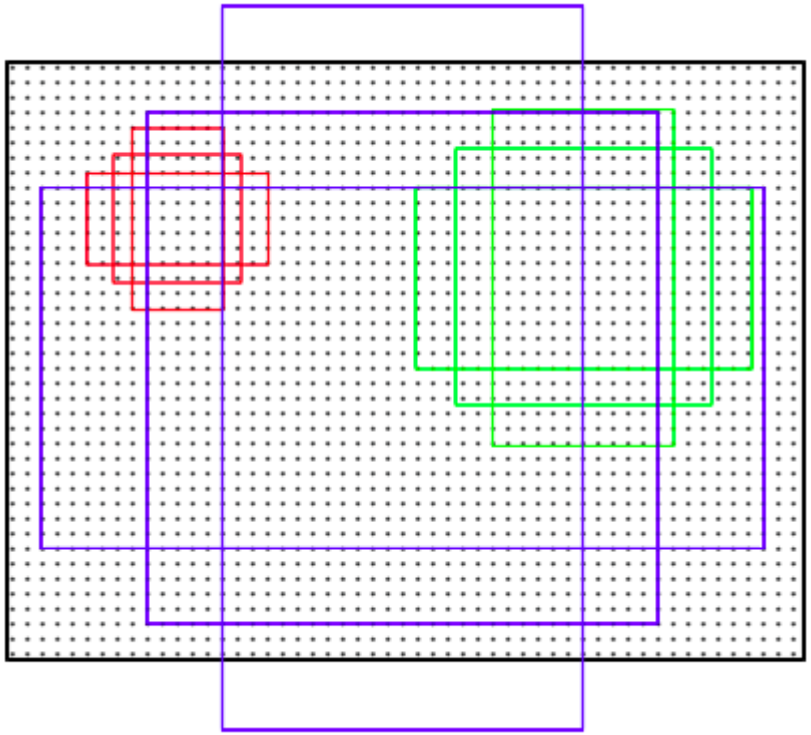


上图是RPN的网络流程图，即也是利用了SPP的映射机制，从conv5上进行滑窗来替代从原图滑窗。
不过，要如何训练出一个网络来替代selective search相类似的功能呢？
实际上思路很简单，就是先通过SPP根据一一对应的点从conv5映射回原图，根据设计不同的固定初始尺度训练一个网络，就是给它大小不同（但设计固定）的region图，然后根据与ground truth的覆盖率给它正负标签，让它学习里面是否有object即可。
这就又变成介绍RCNN之前提出的traditional method，训练出一个能检测物体的网络，然后对整张图片进行滑窗判断，不过这样子的话由于无法判断region的尺度和scale ratio，故需要多次放缩，这样子测试，估计判断一张图片是否有物体就需要很久。(传统hog+svm->dpm)

如何降低这一部分的复杂度？
要知道我们只需要找出大致的地方，无论是精确定位位置还是尺寸，后面的工作都可以完成，这样子的话，与其说用小网络，简单的学习（这样子估计和蒙差不多了，反正有无物体也就50%的概率），还不如用深的网络，固定尺度变化，固定scale ratio变化，固定采样方式（反正后面的工作能进行调整，更何况它本身就可以对box的位置进行调整）这样子来降低任务复杂度呢。
这里有个很不错的地方就是在前面可以共享卷积计算结果，这也算是用深度网络的另一个原因吧。而这三个固定，我估计也就是为什么文章叫这些proposal为anchor的原因了。这个网络的结果就是卷积层的每个点都有有关于k个achor boxes的输出，包括是不是物体，调整box相应的位置。这相当于给了比较死的初始位置（三个固定），然后来大致判断是否是物体以及所对应的位置.
这样子的话RPN所要做的也就完成了，这个网络也就完成了它应该完成的使命，剩下的交给其他部分完成。

候选区域（anchor）

特征可以看做一个尺度51*39的256通道图像，对于该图像的每一个位置，考虑9个可能的候选窗口：三种面积{1282,2562,5122}× 三种比例{1:1,1:2,2:1} 。
这些候选窗口称为anchors。
下图示出51*39个anchor中心，以及9种anchor示例。



关于anchor的问题：
这里在详细解释一下:(1)首先按照尺度和长宽比生成9种anchor,这9个anchor的意思是conv5 feature map 3x3的滑窗对应原图区域的大小.这9个anchor对于任意输入的图像都是一样的，所以只需要计算一次. 既然大小对应关系有了，下一步就是中心点对应关系，接下来(2)对于每张输入图像，根据图像大小计算conv5 3x3滑窗对应原图的中心点. 有了中心点对应关系和大小对应关系，映射就显而易见了.

在整个faster RCNN算法中，有三种尺度。
原图尺度：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。

Region Proposal Networks

RPN的目的是实现"attention"机制,告诉后续的扮演检测\识别\分类角色的Fast-RCNN应该注意哪些区域,它从任意尺寸的图片中得到一系列的带有 objectness score 的 object proposals。
具体流程是：使用一个小的网络在已经进行通过卷积计算得到的feature map上进行滑动扫描，这个小的网络每次在一个feature map上的一个窗口进行滑动(这个窗口大小为n*n----在这里,再次看到神经网络中用于缩减网络训练参数的局部感知策略receptive field,通常n=228在VGG-16,而作者论文使用n=3)，滑动操作后映射到一个低维向量(例如256D或512D,这里说256或512是低维,Q:n=3,n*n=9,为什么256是低维呢?那么解释一下:低维相对不是指窗口大小,窗口是用来滑动的!256相对的是a convolutional feature map of a size W × H (typically ~ 2,400),而2400这个特征数很大,所以说256是低维.另外需要明白的是:这里的256维里的每一个数都是一个Anchor(由2400的特征数滑动后操作后,再进行压缩))最后将这个低维向量送入到两个独立\平行的全连接层:box回归层（a box-regression layer (reg)）和box分类层（a box-classification layer (cls)）

Translation-Invariant Anchors

在计算机视觉中的一个挑战就是平移不变性:比如人脸识别任务中,小的人脸(24*24的分辨率)和大的人脸(1080*720)如何在同一个训练好权值的网络中都能正确识别. 传统有两种主流的解决方式：
第一:对图像或feature map层进行尺度\宽高的采样;
第二,对滤波器进行尺度\宽高的采样(或可以认为是滑动窗口).
但作者的解决该问题的具体实现是:通过卷积核中心(用来生成推荐窗口的Anchor)进行尺度、宽高比的采样。如上图右边，文中使用了3 scales and 3 aspect ratios （1:1,1:2,2:1）, 就产生了 k = 9 anchors at each sliding position.

窗口分类和位置精修

分类层（cls_score）输出每一个位置上，9个anchor属于前景和背景的概率；窗口回归层（bbox_pred）输出每一个位置上，9个anchor对应窗口应该平移缩放的参数。
对于每一个位置来说，分类层从256维特征中输出属于前景和背景的概率；窗口回归层从256维特征中输出4个平移缩放参数。

就局部来说，这两层是全连接网络；就全局来说，由于网络在所有位置（共51*39个）的参数相同，所以实际用尺寸为1×1的卷积网络实现。

需要注意的是：并没有显式地提取任何候选窗口，完全使用网络自身完成判断和修正。

区域生成网络：训练

样本

考察训练集中的每张图像：
a. 对每个标定的真值候选区域，与其重叠比例最大的anchor记为前景样本
b. 对a)剩余的anchor，如果其与某个标定重叠比例大于0.7，记为前景样本；如果其与任意一个标定的重叠比例都小于0.3，记为背景样本
c. 对a),b)剩余的anchor，弃去不用。
d. 跨越图像边界的anchor弃去不用

代价函数

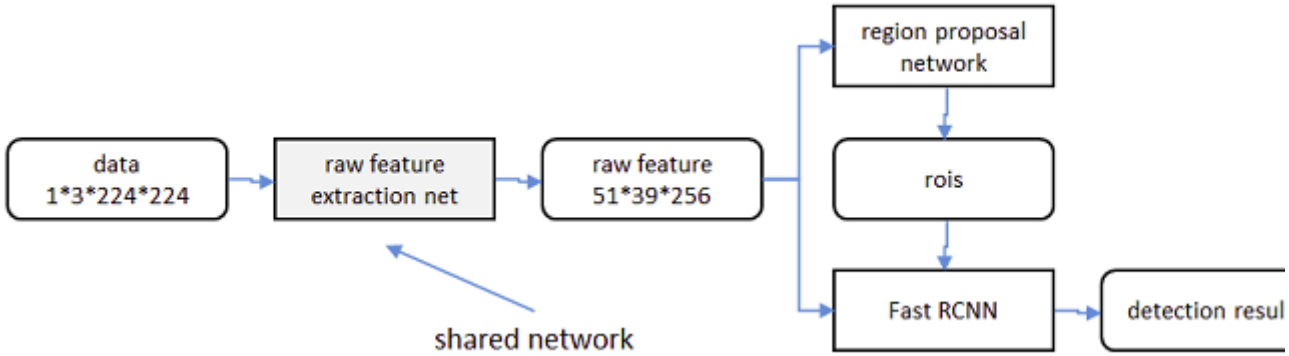
同时最小化两种代价：
a. 分类误差
b. 前景样本的窗口位置偏差

超参数

原始特征提取网络使用ImageNet的分类样本初始化，其余新增层随机初始化。
每个mini-batch包含从一张图像中提取的256个anchor，前景背景样本1:1.
前60K迭代，学习率0.001，后20K迭代，学习率0.0001。
momentum设置为0.9，weight decay设置为0.0005。[5]

共享特征

区域生成网络（RPN）和fast RCNN都需要一个原始特征提取网络（下图灰色方框）。这个网络使用ImageNet的分类库得到初始参数W0，但要如何精调参数，使其同时满足两方的需求呢？本文讲解了三种方法。



轮流训练

a. 从W0开始，训练RPN。用RPN提取训练集上的候选区域
b. 从W0开始，用候选区域训练Fast RCNN，参数记为W1
c. 从W1开始，训练RPN…
具体操作时，仅执行两次迭代，并在训练时冻结了部分层。论文中的实验使用此方法。
如Ross Girshick在ICCV 15年的讲座Training R-CNNs of various velocities中所述，采用此方法没有什么根本原因，主要是因为"实现问题，以及截稿日期"。

近似联合训练

直接在上图结构上训练。在backward计算梯度时，把提取的ROI区域当做固定值看待；在backward更新参数时，来自RPN和来自Fast RCNN的增量合并输入原始特征提取层。
此方法和前方法效果类似，但能将训练时间减少20%-25%。公布的python代码中包含此方法。

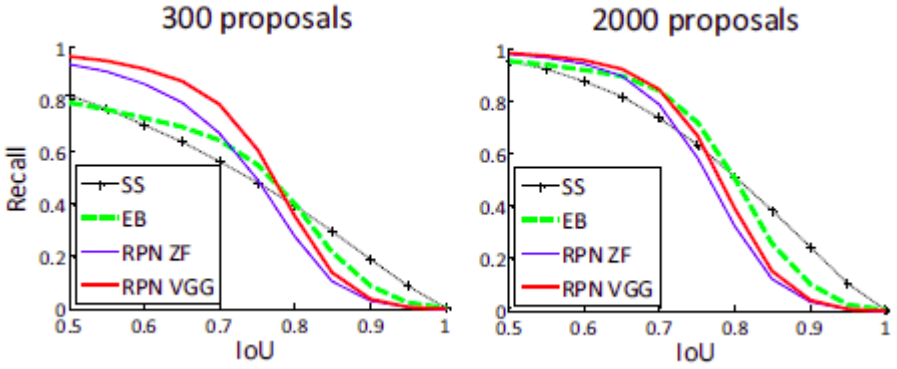
联合训练

直接在上图结构上训练。但在backward计算梯度时，要考虑ROI区域的变化的影响。推导超出本文范畴，请参看15年NIP论文[6]。

实验

除了开篇提到的基本性能外，还有一些值得注意的结论

与Selective Search方法（黑）相比，当每张图生成的候选区域从2000减少到300时，本文RPN方法（红蓝）的召回率下降不大。说明RPN方法的目的性更明确。



使用更大的Microsoft COCO库[7]训练，直接在PASCAL VOC上测试，准确率提升6%。说明faster RCNN迁移性良好，没有over fitting。

training data	2007 test
VOC07	69.9
VOC07+12	73.2
VOC07++12	-
COCO (no VOC)	76.1
COCO+VOC07+12	78.8
COCO+VOC07++12	-

分类: CV

AI-ML-DL
关注 - 0
粉丝 - 15
+加关注

« 上一篇：时间序列分析
» 下一篇：CV：object detection(SPP-Net)

posted @ 2017-02-16 18:21 AI-ML-DL 阅读(516) 评论(0) 编辑 收藏