[一、Introduce 2](#_Toc1055048247)

[二、Fast R-CNN architecture and training 3](#_Toc808123418)

[2.1 RoI pooling layer RoI池化层 6](#_Toc332535716)

[2.2 Initializing from pre-trained networks 7](#_Toc279523202)

[2.3 Fine-tuning for detection 检测中的微调 8](#_Toc637297021)

[（1）Multi-task loss. 9](#_Toc85504662)

[（2）Mini-batch sampling. 11](#_Toc198333540)

[（3）Back-propagation through RoI pooling layers.   11](#_Toc1117397839)

[（4）SGD hyper-parameters. SGD超参数选择 14](#_Toc1413156775)

[2.4 Scale invariance 尺度不变性 14](#_Toc1589497422)

[三、Fast R-CNN detection 16](#_Toc1558987615)

[3.1 Truncated SVD for faster detection 16](#_Toc147264581)

[四、补充 18](#_Toc834556186)

[4.1创新点 18](#_Toc1702654885)

[4.2问题是什么 19](#_Toc1338815258)

[4.3如何解决问题 20](#_Toc1114515786)

[（1）测试过程 20](#_Toc1552361809)

[（2）解释分析 21](#_Toc976014037)

[（3）训练过程 24](#_Toc792870543)

[4.4还存在什么问题 27](#_Toc831506047)

**Fast R-CNN**

# 一、Introduce

相比图像分类、目标检测是一个更有挑战性的任务，因为检测需要对象的准确定位，因此目标检测有两个挑战：

（1）必须处理许多候选对象位置（通常称为“proposals ”）。

（2）这些候选对象只能提供粗糙的定位，但又必须进行完善来实现精确定位。

所以需要更复杂的方法来解决，并且解决这些问题的办法经常需要对速度，准确性及复杂度进行妥协。

在这篇文章中，作者提出一个单阶段训练算法，共同学习分类对象我们提出一个单阶段训练算法，可以同时学习去分类物体建议框并且改善他们的空间位置。

基于区域的卷积网络方法（RCNN）通过使用深层的ConvNet来分类对象建议框，实现了很棒的检测对象精度。然而，R-CNN的缺点也很明显：

（1）Training is a multi-stage pipeline 训练的时候，R-CNN首先使用log loss 在对象建议框（object proposals ）上微调ConvNet，然后提取特征，之后用SVM分类器，最后再进行bounding-box regressors学习。

（2）Training is expensive in space and time 在每张图片中对对象建议框（object proposals）进行提取来对SVM和bounding-box regressor作训练，还要写进磁盘中。训练时间和空间开销很大。

（3）Object detection is slow 对象检测很慢，即测试时间开销大。

RCNN很慢的原因主要是因为没有共享计算。SPP-net就通过共享计算提高了速度，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，因为卷积特征是线下计算的，从而无法在微调阶段反向传播误差。对很深的网络这样肯定是不行的。

Fast-RCNN的优点：

（1）比R-CNN、SPP-net有更高的检测质量（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。

# 二、Fast R-CNN architecture and training

整体框架如 Fig.1 。Fast R-CNN网络输入一整张图片及一系列对象建议框（proposal）。

训练过程如下：

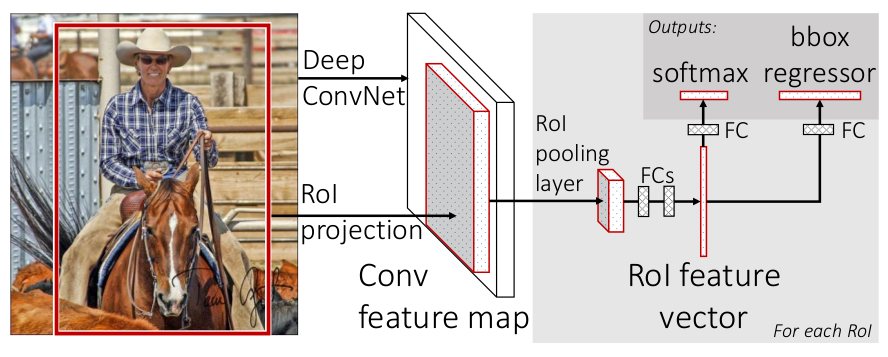
（1）网络首先用几个卷积层（conv）和最大池化层处理整个图像以 产生conv特征图。

（2）然后，对于每个对象建议框（object proposals ），感兴趣区 域（region of interest——RoI）池化层从特征图提取固定长度 的特征向量。

（3）每个特征向量被输送到分支成两个同级输出层的全连接（fc） 层序列中：

其中一层进行分类，对目标关于K个对象类（包括全部“背景 background”类）产生softmax概率估计，即输出每一个RoI 的概率分布；

另一层进行bbox regression，输出K个对象类中每一个类的四 个实数值。每4个值编码K个类中的每个类的精确边界盒 （bounding-box）位置，即输出每一个种类的的边界盒回归偏 差。整个结构是使用多任务损失的端到端训练（trained end-to-end with a multi-task loss） 。



这里再总结来强调一下：

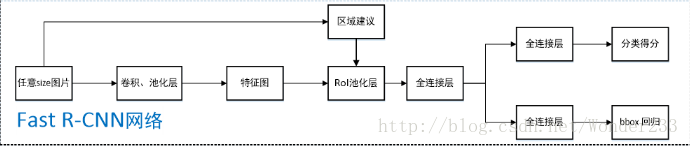
（1）用selective search在一张图片中生成约2000个object proposal，即感兴趣区域RoI。

（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。

另外还有一个关于测试过程的总结也写得不错



（1）任意size图片输入CNN网络，经过若干卷积层与池化层，得到特征图；

（2）在任意size图片上采用selective search算法提取约2k个建议框；

（3）根据原图中建议框到特征图映射关系，在特征图中找到每个建议框对应的特征框【深度和特征图一致】，并在RoI池化层中将每个特征框池化到H×W【VGG-16网络是7×7】的size；

（4）固定H×W【VGG-16网络是7×7】大小的特征框经过全连接层得到固定大小的特征向量；

（5）第4步所得特征向量经由各自的全连接层【由SVD分解实现】，分别得到两个输出向量：一个是softmax的分类得分，一个是Bounding-box窗口回归；

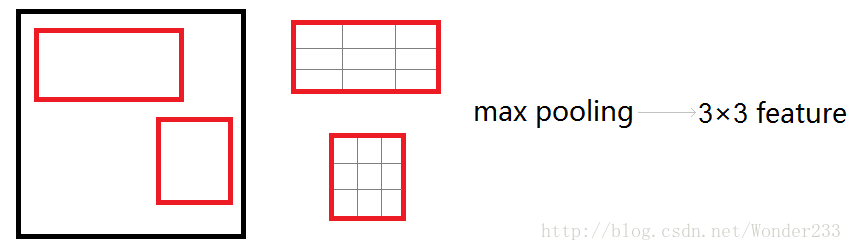
（6）利用窗口得分分别对每一类物体进行非极大值抑制剔除重叠建议框，最终得到每个类别中回归修正后的得分最高的窗口。

## 2.1 RoI pooling layer RoI池化层

每一个RoI都由一个四元组（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）将RoI区域转化成固定大小的 H\*W 的特征图（H和W是取决于任何特定RoI层的超参数）。

RoI 最大池化通过将 h×w RoI窗口划分为 h / H × w / W个子窗口网格，子窗口大小固定为 H × W ，然后将每个子窗口中的值max pooling到相应的输出网格单元 。这是SPP pooling层的一个简化版，即只有一层的“空间金字塔”。

roi\_pool层将每个候选区域均匀分成 H × W 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。



Rol 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。

## 2.2 Initializing from pre-trained networks

作者实验了三个预训练的ImageNet 网络：CaffeNet，VGG\_CNN\_M\_1024，VGG-16，每个网络有五个最大池层和五到十三个转换层。当预训练网络初始化Fast R-CNN网络时，要经历三个transformations：

1、最后一个最大池化层由 RoI 池化层代替，该层将 H 和W 设置为与第一个全连接层兼容；

2、网络最后的全连接层和softmax（被训练用于1000路ImageNet分类）被替换为前面描述的两个同级层：softmax的对K+1个类别的分类层，和bounding box 回归层

3、网络输入修改为两种数据：一组图片和每张图片的一组RoIs

## 2.3 Fine-tuning for detection 检测中的微调

Fast R-CNN用反向传播训练所有网络权重。首先，作者说明了为什么SPPnet无法更新空间金字塔池化层之前的层的权重：当每个训练样本（即RoI）来自不同的图像时，通过SPP层的反向传播非常低效。低效源于每个RoI可能具有非常大的感受野（接收区），通常包括整个输入图像。由于正向传播必须处理整个感受野，训练输入是非常大（通常是整个图像）。

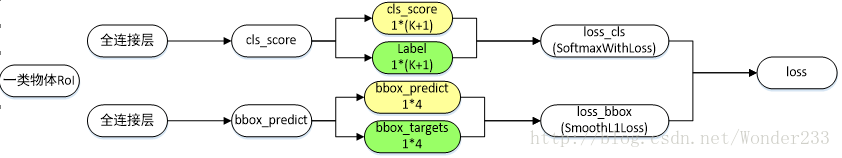
作者提出了一种更有效的训练方法，利用训练期间的特征共享。在Fast RCNN训练中，随机梯度下降（SGD）小批量计算被分级采样，首先随机取样N张图片，然后每张图片取样 R / N 个RoIs 。关键的是，来自相同图像的RoI在向前和向后传播中共享计算和内存。

除了分层采样，Fast R-CNN使用一个精简的训练过程，一次微调中联合优化softmax分类器和bbox回归，而不是在三个独立的阶段训练softmax分类器，SVM和回归因子。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）

### （1）Multi-task loss.

Fast R-CNN网络分类损失和回归损失如下图所示【仅针对一个RoI即一类物体说明】，黄色框表示训练数据，绿色框表示输入目标：



-cls\_score层用于分类，输出K+1维数组p，表示属于K类物体和背景的概率；

-bbox\_predict层用于调整候选区域位置，输出4\*K维数组，也就是说对于每个类别都会训练一个单独的回归器；

Fast RCNN有两个输出层：

dfg

一个对每个RoI输出离散概率分布：

hjik

一个输出bounding box回归的位移：

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。

每个训练的RoI都被标记了ground-truth类别 u 以及ground-truth边界框回归 v 。在每个标记好的RoI上用multi-task loss 函数来级联的训练分类和bbox边界框回归：

gfdg

约定u=0为背景分类，那么[u≥1] 函数表示背景候选区域即负样本不参与回归损失，不需要对候选区域进行回归操作；

第一项是对于 u 类的分类损失（log loss for true class u）。

hty

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。

第二项是回归损失，是在 u 类的真正边界框回归目标的元组 v 上定义的，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，评估回归损失代价就是比较真实分类 u 对应的预测平移缩放参数和真实平移缩放参数的差距：

tfh

其中，

dfgdf

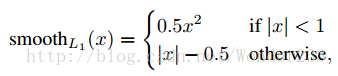
是真实平移缩放参数

对于 u 重新预测bbox回归平移缩放参数：

fhgd

这里的损失不是L2损失函数，而是smooth L1损失函数，对于离群点不敏感，因为有L2损失的训练可能需要仔细调整学习率，以防止爆炸梯度（控制梯度的量级使得训练时不容易跑飞）。

公式如下：



超参数 λ 是用来控制两个损失函数的平衡的。作者对回归目标vi进行归一化使其具有零均值及单位权方差（zero mean and unit variance）。所有的函数都设置超参数 λ = 1。

### （2）Mini-batch sampling.

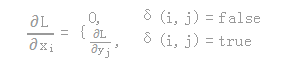
作者从对象建议框（object proposal）中选择25%的RoI，这些RoI与ground-truth bbox边界框至少有0.5的部分交叉重叠，也就是正样本，即 u >= 1。其余的RoI选那些IoU重叠区间在[0.1,0.5)的，作为负样本，即 u = 0，大约为75%。之所以选择负样本需要大于0.1的阈值是因为使用启发式的hard example mining（低于0.1的IoU作为难例挖掘的启发式）。在训练期间，图像有0.5的概率水平翻转。？？？

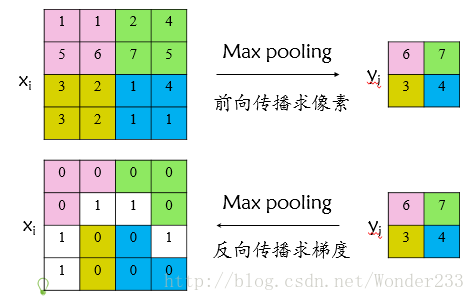


### （3）Back-propagation through RoI pooling layers.

通过RoI池化层的反向传播。

首先看普通max pooling层如何求导，设xi为输入层节点，yi为输出层节点，那么损失函数L对输入层节点xi的梯度为：



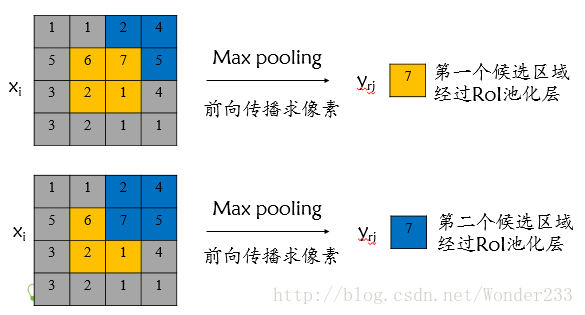


其中判决函数δ(i,j)表示输入i节点是否被输出j节点选为最大值输出。

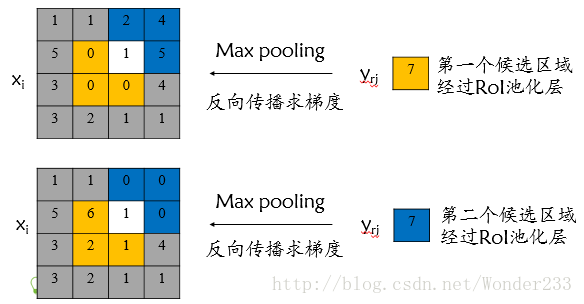
不被选中【δ(i,j)=false】有两种可能：xi不在yi范围内，或者xi不是最大值。

若选中【δ(i,j)=true 】则由链式规则可知损失函数L相对xi的梯度等于损失函数L相对yi的梯度×（yi对xi的梯度->恒等于1），故可得上述所示公式；

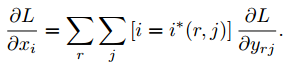
对于RoI max pooling层，设xi为输入层的节点，yri 为第r个候选区域的第j个输出节点，一个输入节点可能和多个输出节点相关连，如下图所示，输入节点7和两个候选区域输出节点相关连；



该输入节点7的反向传播如下图所示。



对于不同候选区域，节点7都存在梯度，所以反向传播中损失函数 L 对输入层节点 xi 的梯度为损失函数 L 对各个有可能的候选区域 r 【 xi 被候选区域r的第j个输出节点选为最大值 】输出 yri 梯度的累加，具体如下公式所示：



bdfgds

判决函数 [i=i∗(r,j)] 表示 i 节点是否被候选区域r 的第j 个输出节点选为最大值输出，若是，则由链式规则可知损失函数L相对 xi 的梯度等于损失函数 L 相对yri 的梯度×（ yrj 对xi 的梯度->恒等于1)，上图已然解释该输入节点可能会和不同的yrj j有关系，故损失函数L相对xi 的梯度为求和形式。

### （4）SGD hyper-parameters. SGD超参数选择

除了修改增加的层，原有的层参数已经通过预训练方式初始化：

用于分类的全连接层以均值为0、标准差为0.01的高斯分布初始化；

用于回归的全连接层以均值为0、标准差为0.001的高斯分布初始化，偏置都初始化为0；

针对PASCAL VOC 2007和2012训练集，前30k次迭代全局学习率为0.001，每层权重学习率为1倍，偏置学习率为2倍，后10k次迭代全局学习率更新为0.0001；

动量设置为0.9，权重衰减设置为0.0005。

## 2.4 Scale invariance 尺度不变性

作者提出了使用两种方式对规模不变的对象进行检测：brute-force（单一尺度）和image pyramids（多尺度，图像金字塔）。

单一尺度直接在训练和测试阶段将image预先固定好像素大小，直接输入网络训练就好，然后期望在训练过程中网络自己能够学习到尺度不变性scale-invariance；

多尺度在训练阶段随机从图像金字塔【缩放图片的scale得到，得到多尺度图片，相当于扩充数据集】中采样训练，通过一个图像金字塔向网络提供一个近似的尺度不变，在测试阶段图像金字塔用来对每个object proposal近似尺度归一化，训练阶段每次采样一个图像就随机采样一个金字塔尺度。

作者在5.2节对单一尺度和多尺度分别进行了实验，不管哪种方式下都定义图像短边像素为s，单一尺度下s=600【维持长宽比进行缩放】，长边限制为1000像素；多尺度s={480,576,688,864,1200}【维持长宽比进行缩放】，长边限制为2000像素，生成图像金字塔进行训练测试；实验结果表明AlexNet【S for small】、VGG\_CNN\_M\_1024【M for medium】下单一尺度比多尺度mAP差1.2%~1.5%，但测试时间上却快不少，VGG-16【L for large】下仅单一尺度就达到了66.9%的mAP【由于GPU显存限制多尺度无法实现】，该实验证明了深度神经网络善于直接学习尺度不变形，对目标的scale不敏感。

第2中方法的表现确实比1好，但是好的不算太多，大概是1个mAP左右，但是时间要慢不少，所以作者实际采用的是第一个策略，也就是single scale。

# 三、Fast R-CNN detection

一旦Fast R-CNN网络被微调，检测相当于运行正向传播（假设对象建议框object proposal是预先计算的）。

网络将图像（或图像金字塔，编码为图像列表）和待给得分的 R 对象建议框（object proposal）列表作为输入。

在测试阶段，R 大约为2K个，当使用图像金字塔的时候，每个RoI被指定尺度使得接近224\*224。对于每个测试RoI r ，网络输出关于 r 的一个后验概率分布 p 和一系列预测bbox偏移（每个类 [共k个类] 获得自己的精确bbox预测）。然后使用估计概率

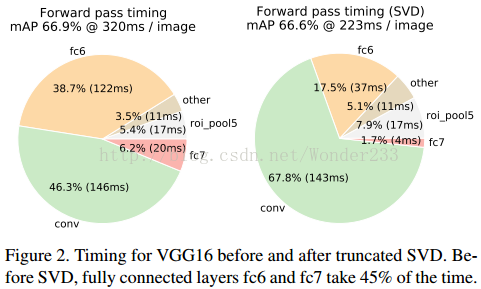
dsg

给 r 赋予关于 k 个对象类的检测置信度。最后给每个类都实施一个非极大值抑制。

## 3.1 Truncated SVD for faster detection

截断SVD ——>减少检测时间

图像分类任务中，用于卷积层计算的时间比用于全连接层计算的时间多；



而在目标检测任务中，要处理的RoI数量比较多，几乎有一半的前向计算时间被用于全连接层（Fig . 2）。就Fast R-CNN而言，RoI池化层后的全连接层需要进行约2k次【每个RoI都要计算】，因此在Fast R-CNN中可以采用SVD分解加速全连接层计算；

具体如何实现的呢？

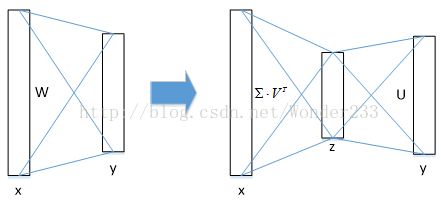
①物体分类和bbox回归都是通过全连接层实现的，假设全连接层输入数据为X，输出数据为Y，全连接层权值矩阵为W，尺寸为 u × v ，那么该层全连接计算为:

Y = W × X

②若将W进行SVD分解（奇异值分解），并用前t个特征值近似代替，即:

dgd

U是 u × t 的左奇异矩阵，Σt 是 t × t 的对角矩阵 ，V是 v × t 的右奇异矩阵。



截断SVD将参数量由原来的 u × v 减少到 t × (u + v) ，当 t 远小于 min(u,v) 的时候降低了很大的计算量。

在实现时，相当于把一个全连接层拆分为两个全连接层，第一个全连接层使用权值矩阵jhtj（不含偏置），第二个全连接层使用矩阵U（含偏置）；

当RoI的数量大时，这种简单的压缩方法有很好的加速。

实验表明（Fig . 2），SVD分解全连接层能使mAP只下降0.3%的情况下提升30%的速度，同时该方法也不必再执行额外的微调操作。

# 四、补充

## 4.1创新点

（1）规避R-CNN中冗余的特征提取操作，只对整张图像全区域进 行一次特征提取；

（2）用RoI pooling层取代最后一层max pooling层，同时引入 建议框信息，提取相应建议框特征；

（3）Fast R-CNN网络末尾采用并行的不同的全连接层，可同时输 出分类结果和窗口回归结果，实现了end-to-end的多任务训练 【建议框提取除外】，也不需要额外的特征存储空间【R-CNN中 这部分特征是供SVM和Bounding-box regression进行训练 的】；

（4）采用SVD对Fast R-CNN网络末尾并行的全连接层进行分解， 减少计算复杂度，加快检测速度。

## 4.2问题是什么

（1）R-CNN网络训练、测试速度都很慢：R-CNN网络中，一张图 经由selective search算法提取约2k个建议框【这2k个建议 框大量重叠】，而所有建议框变形后都要输入AlexNet CNN网络 提取特征【即约2k次特征提取】，会出现上述重叠区域多次重复 提取特征，提取特征操作冗余；

（2） R-CNN网络训练、测试繁琐：R-CNN网络训练过程分为 ILSVRC 2012样本下有监督预训练、PASCAL VOC 2007该特 定样本下的微调、20类即20个SVM分类器训练、20类即20 个 Bounding-box 回归器训练，该训练流程繁琐复杂；同理测 试过程也包括提取建议框、提取CNN特征、SVM分类和 Bounding-box 回归等步骤，过于繁琐；

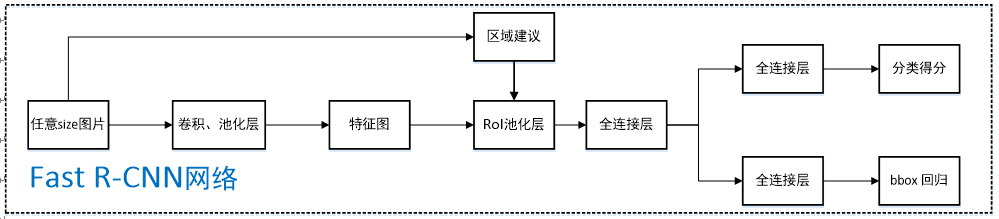
（3）R-CNN网络训练需要大量存储空间：20类即20个SVM分类 器和20类即20个Bounding-box 回归器在训练过程中需要大 量特征作为训练样本，这部分从CNN提取的特征会占用大量存 储空间；

（4）R-CNN网络需要对建议框进行形变操作后【形变为227×227 size】再输入CNN网络提取特征，其实像AlexNet CNN等网络 在提取特征过程中对图像的大小并无要求，只是在提取完特征进 行全连接操作的时候才需要固定特征尺寸【R-CNN中将输入图像 形变为227×227可正好满足AlexNet CNN网络最后的特征尺 寸要求】，然后才使用SVM分类器分类，R-CNN需要进行形变操 作的问题在Fast R-CNN已经不存在，具体见下。

## 4.3如何解决问题

### （1）测试过程

Fast R-CNN的网络结构如下图所示



（1）任意size图片输入CNN网络，经过若干卷积层与池化层，得 到特征图；

（2）在任意size图片上采用selective search算法提取约2k个 建议框；

（3）根据原图中建议框到特征图映射关系，在特征图中找到每个建 议框对应的特征框【深度和特征图一致】，并在RoI池化层中将每 个特征框池化到H×W【VGG-16网络是7×7】的size；

（4）固定H×W【VGG-16网络是7×7】大小的特征框经过全连接 层得到固定大小的特征向量；

（5）第4步所得特征向量经由各自的全连接层【由SVD分解实现】， 分别得到两个输出向量：一个是softmax的分类得分，一个是 Bounding-box窗口回归；

（6）利用窗口得分分别对每一类物体进行非极大值抑制剔除重叠建 议框，最终得到每个类别中回归修正后的得分最高的窗口。

### （2）解释分析

A）整个测试过程为什么可以只进行一次CNN特征提取操作？

先看R-CNN网络，它首先采用selective search算法提取约2k个建议框，并对所有建议框都进行了CNN特征提取操作，会出现重叠区域多次重复提取特征，这些操作非常耗时、耗空间。事实上我们并不需要对每个建议框都进行CNN特征提取操作，只需要对原始的整张图片进行1次CNN特征提取操作即可，因为selective search算法提取的建议框属于整张图片，因此对整张图片提取出特征图后，再找出相应建议框在特征图中对应的区域，这样就可以避免冗余的特征提取操作，节省大量时间。

B）为什么要将每个建议框对应的特征框池化到H×W 的size？如何实现？

问题4中已经指出像AlexNet CNN等网络在提取特征过程中对图像的大小并无要求，只是在提取完特征进行全连接操作的时候才需要固定特征尺寸，利用这一点，Fast R-CNN可输入任意size图片，并在全连接操作前加入RoI池化层，将建议框对应特征图中的特征框池化到H×W 的size，以便满足后续操作对size的要求；

具体如何实现呢?

首先假设建议框对应特征图中的特征框大小为h×w，将其划分H×W个子窗口，每个子窗口大小为h/H×w/W，然后对每个子窗口采用max pooling下采样操作，每个子窗口只取一个最大值，则特征框最终池化为H×W的size【特征框各深度同理】，这将各个大小不一的特征框转化为大小统一的数据输入下一层。

C）为什么要采用SVD分解实现Fast R-CNN网络中最后的全连接层？具体如何实现？

图像分类任务中，用于卷积层计算的时间比用于全连接层计算的时间多，而在目标检测任务中，selective search算法提取的建议框比较多【约2k】，几乎有一半的前向计算时间被花费于全连接层，就Fast R-CNN而言，RoI池化层后的全连接层需要进行约2k次【每个建议框都要计算】，因此在Fast R-CNN中可以采用SVD分解加速全连接层计算；

具体如何实现呢?

请看上文的3.1节

C）文中仅采用selective search算法提取约2k个候选区域，那候选区域越多越好吗？

文中利用selective search算法提取1k~10k中10种数目【1k，2k…】的候选区域进行训练测试，发现随着候选区域个数的增加，mAP成先增加后缓慢下滑的趋势，这表明更多的候选区域会有损精度；与此同时，作者也做了召回率【所谓**召回率**即候选区域为真的窗口与Ground Truth的比值【IoU大于阈值即为真】】分析实验，发现随着候选区域个数的增加，召回率并没有和mAP成很好的相关性，而是一直不断增加，也就是说更高的召回率并不意味着更高的mAP；

文中也以selective search算法提取的2k个候选区域为基础，每次增加1000 × {2, 4, 6, 8, 10, 32, 45}个密集box【滑动窗口方法】进行训练测试，发现mAP比只有selective search方法的2k候选区域下降幅度更大，最终达到53%。

D）如何处理尺度不变性问题？即如何使24×24和1080×720的车辆同时在一个训练好的网络中都能正确识别？

文中提及两种方式处理：brute-force（单一尺度）和image pyramids（多尺度）。单一尺度直接在训练和测试阶段将image定死为某种scale，直接输入网络训练就好，然后期望网络自己能够学习到scale-invariance的表达；多尺度在训练阶段随机从图像金字塔【缩放图片的scale得到，相当于扩充数据集】中采样训练，测试阶段将图像缩放为金字塔中最为相似的尺寸进行测试；

可以看出，多尺度应该比单一尺度效果好。作者在5.2节对单一尺度和多尺度分别进行了实验，不管哪种方式下都定义图像短边像素为s，单一尺度下s=600【维持长宽比进行缩放】，长边限制为1000像素；多尺度s={480,576,688,864,1200}【维持长宽比进行缩放】，长边限制为2000像素，生成图像金字塔进行训练测试；实验结果表明AlexNet【S for small】、VGG\_CNN\_M\_1024【M for medium】下单一尺度比多尺度mAP差1.2%~1.5%，但测试时间上却快不少，VGG-16【L for large】下仅单一尺度就达到了66.9%的mAP【由于GPU显存限制多尺度无法实现】，该实验证明了深度神经网络善于直接学习尺度不变形，对目标的scale不敏感。

E）为什么不沿用R-CNN中的形式继续采用SVM进行分类？

为什么R-CNN中采用SVM分类而不直接用CNN网络输出端进行分类已经在R-CNN博客中说明，针对Fast R-CNN，文中分别进行实验并对比了采用SVM和采用softmax的mAP结果，不管AlexNet【S for small】、VGG\_CNN\_M\_1024【M for medium】、VGG-16【L fo训练过程r large】中任意网络，采用softmax的mAP都比采用SVM的mAP高0.1%~0.8%，这是由于softmax在分类过程中引入了类间竞争，分类效果更好；

Fast R-CNN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。

### （3）训练过程

A）有监督预训练



（4）解释分析

A）Fast R-CNN如何采样进行SGD训练，和R-CNN、SPPnet中SGD采样方式有什么区别和优势？

R-CNN和SPPnet中采用RoI-centric sampling：从所有图片的所有候选区域中均匀取样，这样每个SGD的mini-batch中包含了不同图像的样本，不同图像之间不能共享卷积计算和内存，运算开销大；

Fast R-CNN中采用image-centric sampling：mini-batch采用层次采样，即先对图像采样【N个】，再在采样到的图像中对候选区域采样【每个图像中采样R/N个，一个mini-batch共计R个候选区域样本】，同一图像的候选区域卷积共享计算和内存，降低了运算开销；

image-centric sampling方式采样的候选区域来自于同一图像，相互之间存在相关性，可能会减慢训练收敛的速度，但是作者在实际实验中并没有出现这样的担忧，反而使用N=2，R=128的RoI-centric sampling方式比R-CNN收敛更快。

这里解释一下为什么SPPnet不能更新spatial pyramid pooling层前面的卷积层，而只能更新后面的全连接层？

博主没有看过SPPnet的论文，有网友解释说卷积特征是线下计算的，从而无法在微调阶段反向传播误差；另一种解释是，反向传播需要计算每一个RoI感受野的卷积层梯度，通常所有RoI会覆盖整个图像，如果用RoI-centric sampling方式会由于计算too much整幅图像梯度而变得又慢又耗内存。

B）哪些层参数需要被微调？

SPPnet论文中采用ZFnet【AlexNet的改进版】这样的小网络，其在微调阶段仅对全连接层进行微调，就足以保证较高的精度，作者文中采用VGG-16【L for large】网路，若仅仅只对全连接层进行微调，mAP会从66.9%降低到61.4%， 所以文中也需要对RoI池化层之前的卷积层进行微调；

那么问题来了？向前微调多少层呢？所有的卷积层都需要微调吗？

作者经过实验发现仅需要对conv3\_1及以后卷积层【即9-13号卷积层】进行微调，才使得mAP、训练速度、训练时GPU占用显存三个量得以权衡；

作者说明所有AlexNet【S for small】、VGG\_CNN\_M\_1024【M for medium】的实验结果都是从conv2往后微调，所有VGG-16【L for large】的实验结果都是从conv3\_1往后微调。

## 4.4还存在什么问题

Fast R-CNN中采用selective search算法提取候选区域，而目标检测大多数时间都消耗在这里【selective search算法候选区域提取需要2~3s，而提特征分类只需要0.32s】，这无法满足实时应用需求，而且Fast R-CNN并没有实现真正意义上的端到端训练模式【候选区域是使用selective search算法先提取出来的】；