**Fast RCNN**

**1 引言**

RCNN在目标检测任务中取得了很不错的结果，但作者认为RCNN仍然存在三点缺陷：1、训练过程是多阶段的，而非端到端的；2、训练过程中非常消耗空间和时间；3、目标检测非常缓慢。

此外，当时横空出世的SSPnet也存在上述问题，且SSPnet中的微调算法不能更新Pyramid Pooling layer（金字塔池化层）之前的卷积层，这种局限性（固定的卷积层）限制了深层网络的精度。

于是作者对RCNN中存在的缺陷原因进行分析。

由于SVM和BB Regression的存在，使得在训练过程中需要从每个图像中的每个候选框提取特征，并写入磁盘；对于像VGG16这样的深度卷积神经网络，整个RCNN的训练过程在单个GPU上需要2.5天。所以作者对SVM和BB Regression整合到训练的网络模型的训练过程中，实现了端到端的网络架构。

此外，由于RCNN架构中每次训练的候选框来自于不同的原始图像，所以无法共享参数，导致目标检测过程非常缓慢，所以作者引入了SPPnet对训练过程进行加速。SPPnet计算整个输入图像的卷积特征图，然后使用从关公像特征图提取的特征向量来对每个候选框进行分类。这使得RCNN的测试速度提高了10到100倍。

在RCNN中，作者对提取的候选框进行了简单的上下文填充以及扭曲操作，使得候选框的尺寸适合AlexNet的输入图像尺寸要求。但扭曲操作会使得图像产生几何方面的失真，这也是提高准确率的一个方向。所以作者引入了Spatial pyramid pooling的精简版本，即ROI Pooling，对feature map进行ROI池化，既保留了目标特征，也实现了尺寸的调整。

作者通过对RCNN进行以上操作以及其余的细节优化，实现了更好的性能，并命名为Fast-RCNN。

**2 Fast R-CNN架构以及训练**

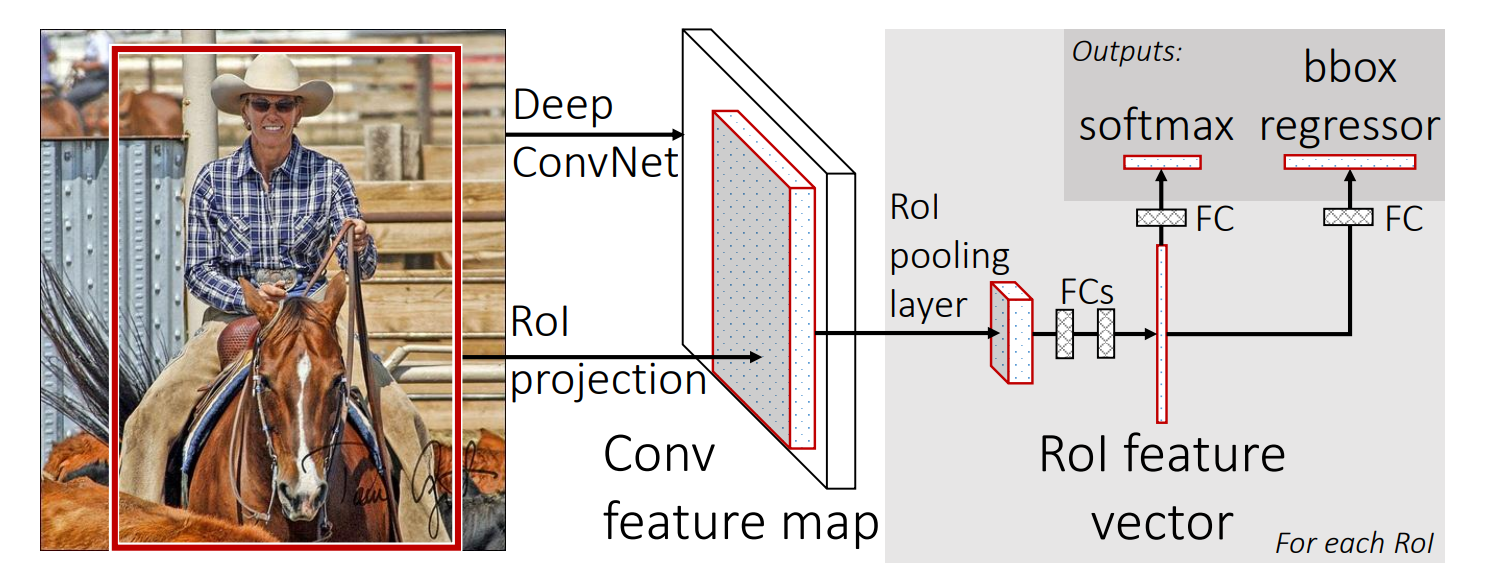


图2。1 Fast RCNN架构

图2.1阐述了Fast R-CNN的整体架构。可以看到与R-CNN架构不同的是，Fast R-CNN并不是将RoI区域单独作为CNN的输入，而是先将整张图像作为CNN的输入，计算对应的feature map，然后根据原始图像和feature map的对应比例将ROI区域映射到feature map中。这样的好处是对于一张图像中的ROI能够实现参数共享，所有的特征也可以暂存在显存中，不占用磁盘空间，这也是RCNN计算缓慢以及耗费空间的原因。

**2.1 ROI池化层**

R-CNN架构中对候选框进行扭曲（warping）以适应AlexNet的输入尺寸要求，所产生的问题在引言中有所论述。作者在Fast R-CNN中摈弃了扭曲这一操作，借鉴了SPPnet，提出了一个ROI池化层。ROI池化层通过最大池化操作来调整feature map的尺寸。假设原始图像的尺寸为h\*w，需要调整为H\*W，那么就设置池化窗口为h/H\*w/W，并进行最大池化操作。

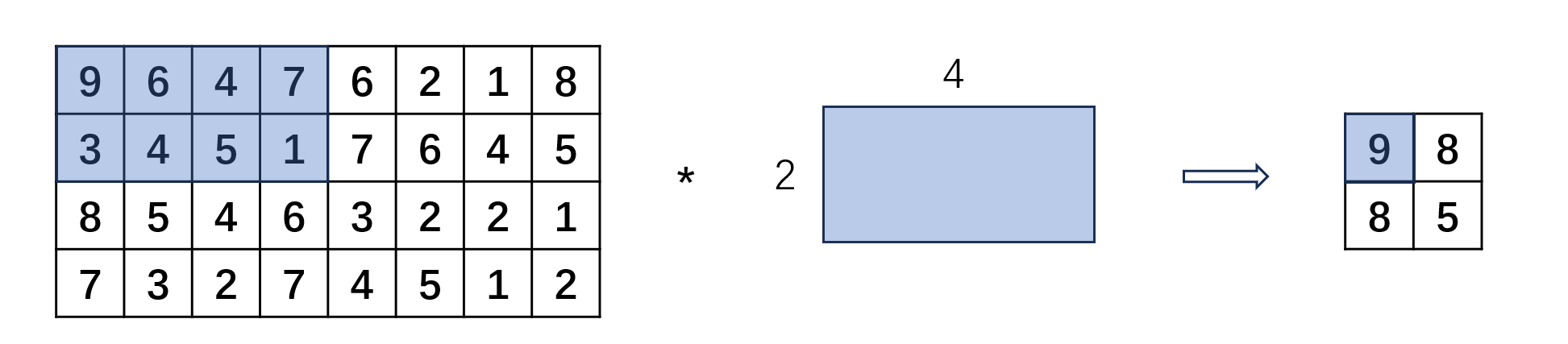


图2-2 ROI Pooling示意图

**2.2 预训练网络的初始化**

本文中作者使用三个在ImageNet数据集上预训练的网络模型（Alexnet，VGG\_CNN\_M\_1024，VGG16），并进行如下调整：

1、最后的最大池化层替换为ROI池化层。

2、最后的全连结层和softmax替换为两个兄弟层，即FC+softmax以及FC+bbox regressor。

3、修改网络以获取两个数据输入：一组图像和一组这些图像的ROI区域。

**2.3 针对目标检测的微调**

在论述具体的微调之前，作者先阐述了为什么在SPPnet中，微调算法无法更新卷积层。最主要的原因是在SPPnet中训练图像（ROI）都来自于不同的原始图像，这导致反向传播过程十分低效。所以，本文提出了一种更有效的训练方法，即每次输入的数据为特定数量的完整图像以及对应图像的一组ROI区域，这使得训练期间实现了特征共享，在前向和后向传播中共享计算和存储器，既提高了计算效率，也降低了存储空间的消耗。

此外，Fast-RCNN对RCNN松散的结构进行了整合，在训练过程中共同实现softmax分类器和bbox regression。其中的细节在下面几节中进行介绍。

**2.4 多任务损失**

Fast-RCNN中存在两个兄弟输出层。其一输出每个ROI的离散概率分布，即对k+1个目标的概率分布p=(p0, p1, …, pk)。其二输出bbox regressor偏移，tk=(x, y w, h)。

由于Fast-RCNN反向传播的端到端特性，我们需要一个共同的损失函数对softmax即bbox regressor进行误差计算。本文提出了如下公式所示的多任务损失函数。



其中，是真实类u的对数损失。是针对类，的真实边界框回归目标的元组定义的，预测元组为。为Iverson bracket indicator function，当时，值为1；反之则为0。这样做的原因是纯背景ROI是没有真实边界框标签的，所以被忽略。对于边界框回归，，其中。

通过该公式，可以实现每个ROI的联合训练分布和边界框回归的拟合。

**2.5 小批量抽样**

在2.3节中提到，微调期间的SGD mini-batch是由N=2个图像构成的，且mini-batch数量设为128，即从每个图像中采样64个ROI。

在Fast R-CNN中，对于每个候选区域，将其与真实目标框的IoU进行比较来判断其是都为正样本或负样本。如果IoU大于等于0.5则将其视为正样本；如果IoU小于0.5且没有与任何真实目标框的IoU大于等于0.1，则将其视为负样本。其余策略和RCNN基本一致。

**2.6通过ROI池化层的反向传播**

ROI池化层的反向传播函数通过遵循argmax激活计算损失函数相对于每个输入变量xi的导数：



**2.7截断奇异值分解**

不同于图像分类任务，目标检测任务中，由于ROI的数量很大，几乎一半的时间用于计算全连结层。通过截断奇异值分解能够很容易地加速上述计算过程。效果如图2-3所示。

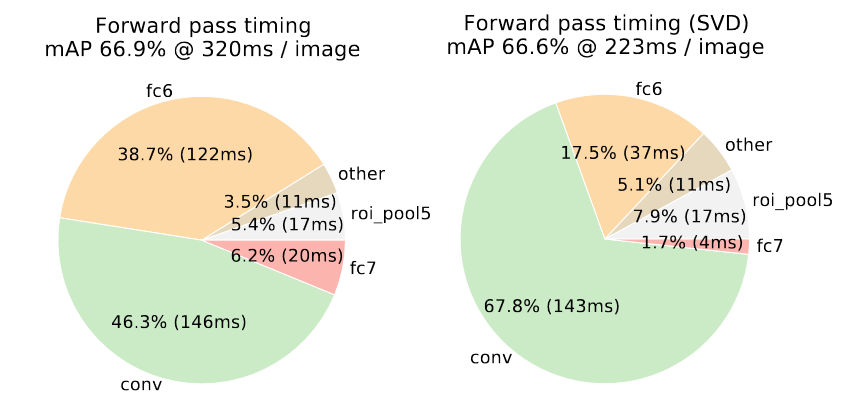
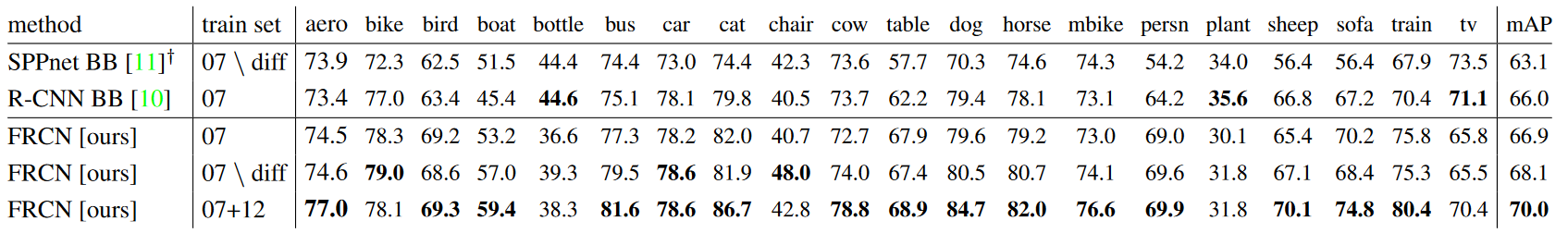
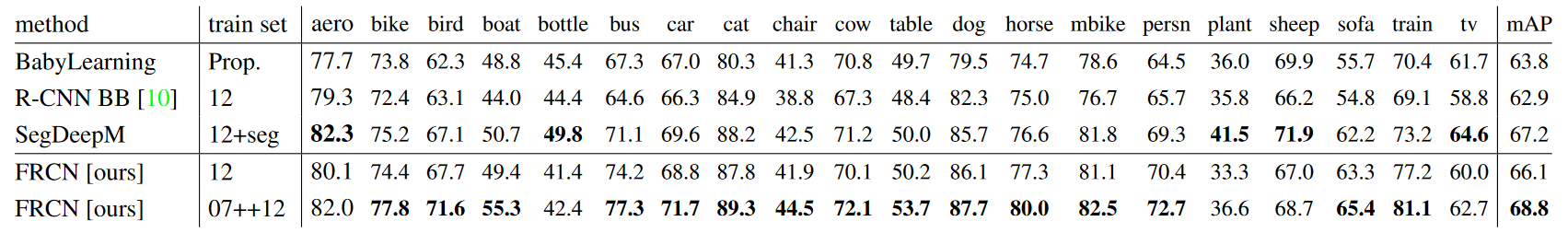


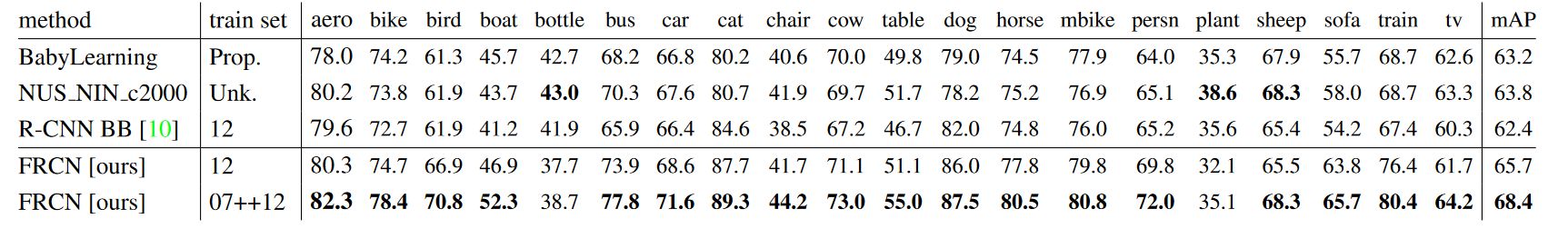
图2-3 无SVD与SVD的训练时间对比图

3 实验结果及分析

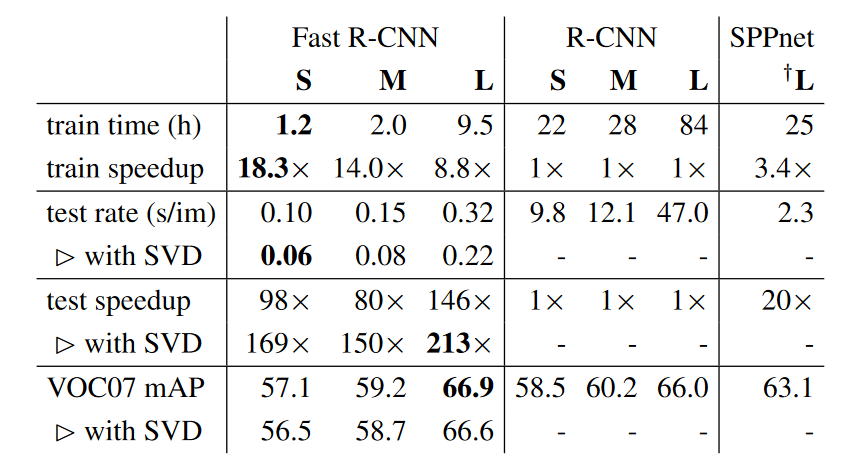
本文使用Fast RCNN架构在VOC2007，VOC2010以及VOC2012数据集上分别进行性能测试，并将结果与当时性能最好的几种方法，即SPPnet，BabyLearning，SegDeepM以及NUS\_NIN\_c2000。从下表可以看到，在额外数据集的辅助下，Fast RCNN的性能最强。







此外，作者统计了测试时间，结果如上表所示。可以看到Fast RCN相比于RCNN和SPPnet在训练速度上有极大深度地提高。综合以上两点，Fast RCNN架构是非常优秀的目标检测架构。



三个主要结果证实了本文提到的共享：

1、在VOC2010和VOC2012数据集上的最新mAP

2、相比于R-CNN和SPPnet更快的训练速度

3、VGG16的微调卷积层对mAP的提高

此外，SPPnet论文中考虑到的较浅网络中，仅微调全连结层就足以获得良好的准确性。作者假设这一结果对较深的网络不适用。为了验证对VGG16进行卷积层微调是否重要，作者使用Fast RCNN进行微调，但保持卷积层不变，只让全连结层学习。结果如表5所示，这也证实了作者的猜想：通过ROI池化层进行训练对于非常深的网络是重要的。同时，这并不意味着所有的卷积层都应该微调，一般来说从conv3开始向上更新是必要的。

最后，作者论证了一系列细节问题，这里就不再叙述了。

**4 讨论**

贡献：

1、基本实现了端到端训练

2、特征共享

3、提高训练速度

4、提高目标检测性能

缺陷：

1、仍然使用selective search提取候选框，慢

2、任然未实现完全的端到端，候选框提取仍然是独立的

3、无法完全做到实时检测