中国科学: 信息科学

SCIENTIA SINICA Informationis

论文

# 目标检测中 RCNN 系列算法综述

于泽枭[1], 管昀玫[2], 宋佳蓁[1]

1. 南开大学, 学号 2013814
2. 南开大学, 学号 2013750
3. 南开大学, 学号 2013904
* 通信作者. E-mail:

基金资助

**摘要** 计算机视觉中关于图像识别有四大类任务: 分类、定位、检测和分割, 其中目标检测除了要解决一般的图像分类任务之外, 其难点在于检测目标的不确定性。为了解决这一核心问题, 人们将卷积神经网络应用于目标检测领域。在 2014 至 2015 年间, 三个基于 CNN 的算法陆续被提出, 使目标检测的准确率和效率有了新的提升。这其中包括 2014 年 Ross Girshick 等人提出的引入了 SS 算法的 RCNN 网络, 这是人们第一次将 CNN 引入目标检测。随后, 为了提高 RCNN 算法的效率, 2015 年 Ross Girshick 等人提出了 Fast-RCNN, 它首次使用 CNN 在整张图片上提取特征, 同时引入 RoI pooling。也在 2015 年 Kaiming He 等人提出了 Faster-RCNN 算法, 引入神经网络 RPN, 将多个相对独立的模型完全整合为一个端到端神经网络, 使得检测效率大幅度提升。本文则基于从目标检测的定义、发展, 对三篇论文的内容进行综述, 在文末对三者进行了比较。

**关键词** R-CNN, Fast R-CNN, Faster R-CNN, 目标检测

## 1 背景

### 1.1 目标检测问题

计算机视觉中关于图像识别有四大类任务: 分类、定位、检测和分割, 这四大类任务分别以解决图片或视频中包含什么类别的目标、定位出目标的位置、定位且明确目标是什么、每一个像素属于哪个目标物或场景这四种问题为本质。目标检测要解决的核心问题除了一般的图像分类任务以外, 还需解决的难点是目标的不确定性——目标可能出现在图像的任意位置, 以及目标可能有各种不同的形状和大小等。各类物体不同程度上的干扰, 使得目标检测成为机器视觉领域研究的热门问题之一。

**引用格式:** 于泽枭, 管昀玫, 宋佳蓁. 引用的标题. 中国科学: 信息科学, 在审文章
Xiaoze Yu, Yunmei Guan, Jiazhen Song. Title for citation (in Chinese). Sci Sin Inform, for review

### 1.2 基于传统手工特征的目标检测

早期的目标检测算法多数是基于手工特征所构建的。由于在深度学习诞生之前缺乏有效的图像特征表达方法，研究者们尽其所能去设计更加多元化的检测算法以弥补手工特征表达能力上的缺陷。同时早期计算资源也相对匮乏，研究者们也探寻更加精巧的计算方法来对模型进行加速。

2001 年 VJ 检测器算法的提出在当时有限的资源环境中首次实现了人脸检测。其所采用的目标检测手段即是最保守的滑动窗口检测。VJ 检测器尽管使用了多尺度 Haar 特征的快速计算和级联决策结构，其依然耗费极大的计算开销。

HOG 检测器是基于梯度特征的目标检测器的基础。其沿用了原始的多尺度金字塔和滑动窗口的组合，将图像所在区域划分为不同的细胞单元 (Cell)，并在每个细胞内统计梯度方向直方图信息。但由于其只能生成宽高比固定的矩形框，故 HOG 检测器只在行人检测等问题中能较好地完成任务。连续获得三年检测冠军的 DPM 模型则是 HOG 和 SVM 的扩展，继承了两者优势的 DPM 同样适用于人脸检测任务，也同样具有较强的复杂性和较慢的检测速度。

目标检测的发展在此时经历着“包围框回归”式的技术进步。但这种技术方式也决定了其过度趋于训练机器使用精妙的计算和多元的算法，然而分类的准确率和能力并没有成功取得突破。

### 1.3 从复杂计算到机器学习

1990 年代，CNNS、SVM 相继被广泛使用。2012 年，Krizhevsky 等人在 ImageNet 大规模视觉识别挑战赛上大放异彩，将一个大型的 CNN 应用于 120 万的标签图像上，从此用 CNN 进行目标识别的新篇章正式开启。随后，研究者们在图片分类和目标检测之间建立联系，回答了关于如何将 ImageNet 上的分类结果推广到 PASCAL VOC 数据集的目标检测任务的问题。在图片分类的基础上，选取合适的方式定位目标。将定位问题简单地看成回归问题，或采用滑窗检测定位的方式都不能保持识别的准确度，因此 RCNN 选取的方式为区域推荐。R-CNN 方法在 PASCAL VOC 数据集上 mAP 达到了 53.7%，而在 200 个类别的 ILSVRC2013 检测数据集上，R-CNN 方法的 mAP 是 31.4%，相对于 OverFeat 的 24.3% 来说有很大的提高。

## 2 RCNN

### 2.1 RCNN

区别于多数传统的以图像识别为基础的目标检测方法，RCNN 可以说是第一个成功将深度学习应用到目标检测上的算法。本节中将从 RCNN 的模块设计、模型分析以及 RCNN 在语义分割任务上的挑战几个方面来介绍。

#### 2.1.1 RCNN 模块设计

1. **生成候选框:**生成候选框的其中一种方法是将定位问题看做回归问题,即边界框回归 (Bounding-Box Regression)。边界框回归将候选目标检测框 (region proposal) 作为原像，将真实的目标框 (ground-truth) 作为像，利用平移和尺度放缩组合的映射方式，使得候选框无限接近于真实

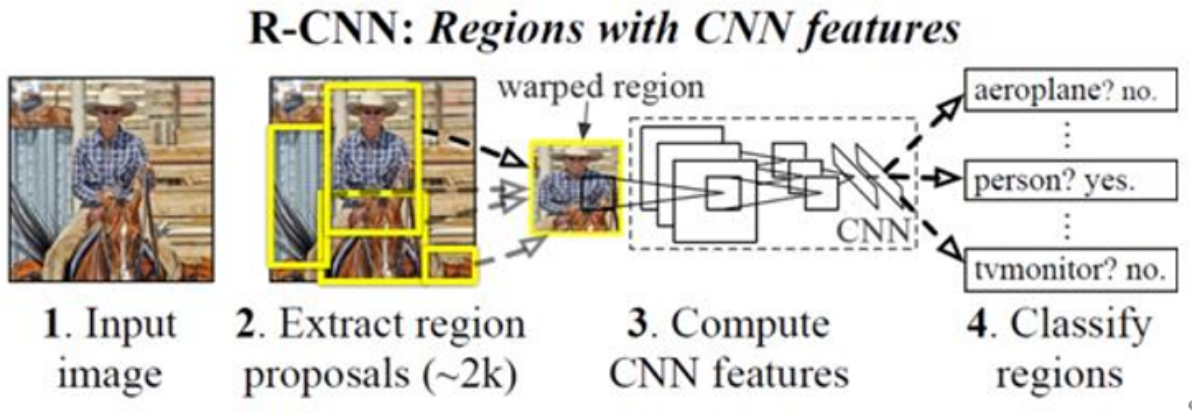


**图 1 RCNN**

**Figure 1** R-CNN: Regions with CNN features

框。每个输入到边界框回归数据集中的数据为目标检测框 Pi 和真实目标框 Gi 的组合，并为 Pi 和 Gi 赋予中心点的 x 坐标、中心点的 y 坐标、目标框的长度、目标框的宽度四维特征参数。衡量回归效果的概念为并交比 (IoU)，即两目标框重合部分面积与两目标框总面积的比值。但 Szegedy 等人使用边界框回归策略进行定位，在 VOC2007 上的 mAP 仅为 30.5%，因此将定位问题单纯作为回归问题来解决并没有达到好的效果。

滑动窗口检测是一种最为保守和传统的方法。将滑动窗口在图像上按照指定的步长进行滑动，对每个滑动得到的区域进行预测分析，判断该区域中存在目标的概率，每个周期接受后相应调整滑动窗口的大小、步长，并继续滑动和预测。这种方式的弊端一是计算成本较高，采用较小粒度时传递给卷积网络的小窗口数目巨大，而较大粒度过于粗糙，可能影响性能；二是滑窗本身大小多数情况下并不完全贴合目标尺寸，对后续的选择影响较大。且对于有 5 个卷积层，195*195 的感受野，32*32 像素的滑动来讲，精确定位的挑战尤其大。

而本文中选取的定位方式是区域推荐 (region proposal)，通过操作”recognition using regions”范式，解决了 CNN 的定位问题。对于每张图片，产生接近 2000 个于类别无关的 region proposal，对每个 CNN 抽取一个固定长度的特征向量，然后借助专门针对特定类别数据的线性 SVM 对每个区域进行分类。这也是 R-CNN 名称的由来——Regions with CNN features。近年来很多研究都提出了与类别无关的区域推荐的方法。由于 RCNN 对于特定区域算法并不关心，所以本文中采用的是选择性搜索 (selective search) 方法，以便于与之前的工作进行可控的比较。

2. **提取特征向量：**对于已经获取的候选区域，需要进一步使用 CNN 提取对应的特征向量。本文中使用模型 AlexNet (2012)。需要注意的是 Alexnet 的输入图像大小是 227*227，而通过选择性搜索产生的候选区域大小不一，为了与 Alexnet 兼容，RCNN 采取的手段是忽视候选区域的大小和形状，统一变换到 227*227 的尺寸。

   该网络在开源 Caffe CNN 库上执行有监督预训练。以 ImageNet 为样本，只训练与分类有关的参数，其中最后一层需要进行 4096 维向量到 1000 维向量的映射。其次进行特定样本下的参数

调优。为了使 CNN 适应检测任务以及变形后的推荐窗口，作者使用变形后的推荐区域对 CNN 参数进行 SGD 训练，将 ImageNet 专用的 1000-way 分类器替换为一个随机初始化的 21-way 分类器（20 代表 VOC 数据集的目标类别数量，1 代表一个背景）。将 IoU>= 0.5 的推荐区域作为正例，反之作为负例，学习率为 0.001。每一轮 SGD 迭代统一使用 32 个正样本和 96 个负样本。

3. **SVM 分类：**通过上述步骤获得候选区域的特征向量，下面进一步使用 SVM 进行物体分类。将 2000*4096 维特征与 20 个 SVM 组成的权值矩阵（4096*20）相乘，获得的 2000*20 维矩阵即为某个物体类别的得分。分别对上述 2000×20 维矩阵中每一列，也即每一类，进行非极大值抑制剔除重叠建议框，得到该列即该类中得分最高的一些候选框。

4. **边界框修正：**使用一个回归器进行边框回归：输入为卷积神经网络 pool5 层的 4096 维特征向量，输出为 x、y 方向的缩放和平移，实现边界框的修正。在进行测试前仍需回归器进行训练。

#### 2.1.2 模型分析

1. **消融研究：**在无微调情况下研究每层的性能时，将所有 CNN 参数仅在 ILSVRC2012 上预训练。逐层分析性能的结果表明，fc7 的特性比 fc6 的特性泛化更差。这意味着 29% 的 CNN 参数可以删除，而不降低 mAP。同时，尽管 pool5 的特征只使用 CNN 的 6% 的参数进行计算，删除 fc7 和 fc6 也会产生相对好的结果。由此可见 CNN 的表示能力大部分来自它的卷积层，而非全连接层。

   有微调情况下从每层看性能的改进，可以发现进步是显著的。微调使 mAP 增加 8.0 个百分点，达到 54.2%，其对 fc6 和 fc7 的促进作用大于 pool5，可见从 Image Net 学习到的 pool5 特征是通用的，并且大多数改进都是通过在它们之上学习针对特定领域的非线性分类器中获得的。

2. **检测误差分析：**作者使用 Hoiem 提出的监测分析工具来理解调参的影响，同时观察相对于 DPM 方法的错误形式。由 FP 类型分布的演变等可以得出结论：微调可以提高检测的鲁棒性，尤其是针对有歧义或遮挡的情况；边界回归能修正候选框的位置，二者对于分类和定位的性能都十分重要。

#### 2.1.3 语义分割

作者尝试使用 RCNN 进行 PASCAL VOC 上的语义分割检测挑战。与目前最好的语义分割算法二阶化池 (OP2) 比较。使用三种策略在 CPMP 区域上计算特征的策略，但效果不甚理想。

## 3 Fast R-CNN

### 3.1 解决的问题

目标检测有两大难点。首先，必须处理大量候选目标位置（“proposals”）。第二，这些候选框仅提供粗略定位，其必须被精细化以实现精确定位。

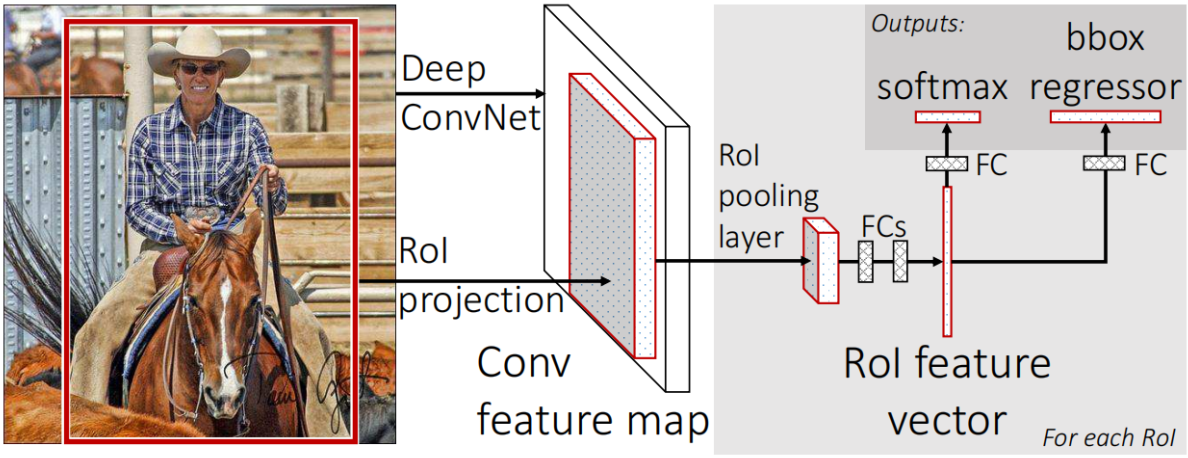


**图 2　模型结构**
**Figure 2**　Model Structue

虽然 R-CNN 已提出了一个可行的方案，但仍不够精确且有效。Fast R-CNN 主要是为了解决 R-CNN 和 SPPnet 的以下几个问题：

1. 训练过程是多级 pipeline。R-CNN 首先使用目标候选框对卷积神经网络使用 log 损失进行 fine-tunes。然后，它将卷积神经网络得到的特征送入 SVM，最后用 bounding-box 进行回归。

2. 训练在时间和空间上开销大。对于 SVM 和 bounding-box 回归训练，从每个图像中的每个目标候选框提取特征，并写入磁盘，这一部分的时间和磁盘空间开销较大。

3. 目标检测速度很慢。在测试时，从每个测试图像中的每个目标候选框提取特征，且每张图片的每个 region proposal 都要做卷积，重复操作多。

4. SPPnet 网络虽提出通过共享计算加速 R-CNN，但训练过程也是一个多级 pipeline，涉及提取特征、使用 log 损失对网络进行 fine-tuning、训练 SVM 分类器以及最后拟合检测框回归，时间和空间消耗仍较大。且在 [1] 中提出的的 fine-tuning 算法不能更新在空间金字塔池之前的卷积层，限制了深层网络的精度。

### 3.2　模型结构

Fast R-CNN 将 R-CNN 中的 SVM 分类器和 bbox 线性回归器放在一体化网络中。网络首先使用几个卷积层和最大池化层来处理整个图像以产生卷积特征图，同时运用了 ROI-pooling 层，将大小不一的 RP 转换成同样的 size，因此可以将整张图像进行过 selective search 后直接输入进 CNN 层，一次性对所有的 RP 完成 softmax 分类和 bbox 回归。

相较于 R-CNN 的改进：

1. 卷积直接对整张图像，而不再是对每个 region proposal 进行，减少很多重复计算。

2. 因为全连接层的输入要求尺寸大小一样，因此用 RoI pooling 进行特征的尺寸变换后进行 SS 就能直接进入 CNN 层了。

3. 将 regressor 放进网络一起训练，每个类别对应一个 regressor，同时用 softmax 代替原来的 SVM 分类器，对于所有三个网络，Softmax 略优于 SVM，mAP 分别提高了 0.1 和 0.8 个点。

4. 使用截短的 SVD 实现更快的检测。较大的全连接层可以轻松通过截短的 SVD 压缩来提升速度。

### 3.3 训练步骤

#### 3.3.1 预训练模型

首先对 VGG-16 进行有监督的分类预训练。实现三个预训练的 ImageNet 网络，每个网络有五个最大池化层和 5 到 13 个卷积层。

1. 最后一层的最大池化层由 RoI 池化层代替。

2. 网络最后一个全连接层和 Softmax 被替换为全连接层和 $K+1$ 个类别的 Softmax 以及特定类别的 bounding-box 回归。

3. 网络修改为两个数据的输入：图像列表和 RoI 的列表。

#### 3.3.2 Fine-tunning

在进行微调之前，还需要以下操作：

- **训练图像的 RP 搜索与 ROI 选择**。在调优训练时，每个 mini-batch 包含 2 张图像，对它们进行 selective search 后，采样出 128 个 region proposal（或者叫 ROI），即每张图像有 64 个 ROI。这些 ROI 中约 25% 的 ROI 作为正样本，正样本和 ground truth 的 IOU 值都大于 0.5。剩下的 ROI 作为负样本，IOU 都小于 0.5。
- **输入网络中进行调优训练**。输入图像是 224*224，不满足条件的先 Resize。经过 5 个卷积层和 2 个降采样层后，进入 RoI 池化层；该层的输入是卷积 5 层的输出和 region proposal，region proposal 的个数大约为 2000。然后再经过两个 output 大小是 4096 的全连接层。最后分别经过 output 个数是 21 和 84 的两个并列的全连接层：前者是分类的输出，代表每个 region proposal 属于每个类别（21 类）的得分；后者是回归的输出，代表每个 region proposal 的四个坐标。最后是两个损失层，分类的是 softmaxWithLoss，输入是 label 和分类层输出的 score；回归的是 SmoothL1Loss，输入是回归层的输出和 target 坐标及 weight。

之后便可进行微调。

1. **多任务损失：** Fast R-CNN 网络具有两个同级输出层。第一个输出在 K+1 个类别上的离散概率分布（每个 RoI），$p = (p_0, ..., p_K)$。第二个输出层输出 bounding-box 回归偏移，即 $t^k = (t_x^k, t_y^k, t_w^k, t_h^k)$，k 表示 K 个类别的索引。对每个标记的 RoI 使用多任务损失 L 以联合训练分类和 bounding-box 回归：

$$L(p,u,t^u,v) = L_{cls}(p,u) + \lambda[u \geqslant]L_{loc}(t^u,v)$$

其中 $L_{cls}(p,u) = -logp_u$, 是类真值 u 的 log 损失

对于检测框回归，使用损失：

$$L_{loc}(t^u,v) = \sum i \in \{x,y,w,h\} smooth_{L1}(t_i^u - v_i)$$

其中：

$$smooth_{L1}(x) = \begin{cases} 0.5x^2, & if|x| < 1 \\ |x| - 0.5, & otherwise \end{cases}$$

第一个公式的超参数 $\lambda$ 控制两个损失之间的平衡。我们将回归真值 $v_i$ 归一化为具有零均值和方差为 1 的分布。所有实验使用 $\lambda = 1$。

2. **RoI 池化层:**RoI 池化层可以说是 SPP（spatial pyramid pooling）的简化版。RoI 池化层去掉了 SPP 的多尺度池化，直接用 MxN 的网格，将每个候选区域均匀分成 M×N 块，对每个块进行 max pooling。从而将特征图上大小不一的候选区域转变为大小统一的特征向量，送入下一层。

3. **采用截短的 SVD 分解改进全连接层:** 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，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，因此在 Fast R-CNN 中可以采用 SVD 分解加速全连接层计算。

4. **SGD 超参数:** 用于 Softmax 分类和检测框回归的全连接层的权重分别使用具有方差 0.01 和 0.001 的零均值高斯分布初始化。偏置初始化为 0。所有层的权重学习率为 1 倍的全局学习率，偏置为 2 倍的全局学习率，全局学习率为 0.001。

### 3.4 尺度不变性

两种实现尺度不变目标检测的方法：（1）通过 “brute force” 学习和（2）通过使用图像金字塔。在 “brute force” 方法中，在训练和测试期间以预定义的像素大小处理每个图像。在多尺度训练期间，我们在每次图像采样时随机采样金字塔尺度。结果发现，单尺度检测几乎与多尺度检测一样好，能够证实：深度卷及网络擅长直接学习到尺度的不变性。另外，具有单尺度的较大网络具有最佳的速度/精度平衡。

### 3.5 Fast R-CNN 的优缺点

优点：

1. 比 R-CNN 和 SPPnet 具有更高的目标检测精度（mAP）。

2. 训练是使用多任务损失的单阶段训练。

3. 训练可以更新所有网络层参数。

4. 不需要磁盘空间缓存特征。

缺点：

1. Fast RCNN 的主要缺点在于 region proposal 的提取使用 selective search，目标检测时间大多消耗在这上面（region proposal 2 3s，而提特征分类只需 0.32s），这也是后续 Faster RCNN 的改进方向之一。

# 4 Fater R-CNN

## 4.1 简介

Faster R-CNN 是一个单一，统一的目标检测网络。基于已知算法对于目标检测的优化，如 Fast R-CNN 实现了近实时检测的速率，目前最先进的目标检测网络主要问题集中于时间开销上面，而时间开销的主要瓶颈是用来推测目标的 region proposal 算法。Faster R-CNN 主要介绍了一种可以与检测网络共享整个图像的卷积特征的网络 Region Proposal Network（RPN），这种网络使得 region proposals 接近于零成本成为可能。它的核心是利用 Anchor 机制将卷积网络与区域生成相联系，从而使计算 proposal 的边际成本达到（10ms/image）接近于零，可以忽略不计，并将检测速度从 fast 的 0.5fps 上升至 7fps。同时，它在 VOC 2012 测试集上实现了 70.4 % mAP 的检测效果，而在 ILSVRC 和 COCO 2015 竞赛中，Faster R-CNN 和 RPN 成为了冠军输入的基础。这些都表现出 Faster R-CNN 的优秀性能。

## 4.2 模型结构

Faster R-CNN 主要可以分为四个主要模块

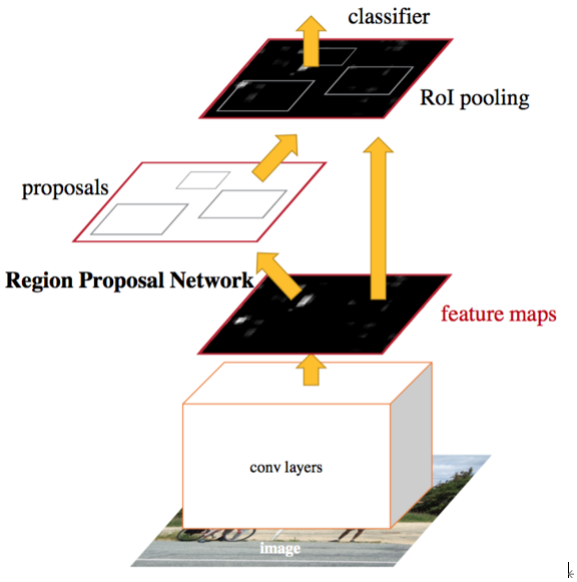


**图 3　Faster R-CNN 模型结构**
**Figure 3**　Faster R-CNN Model Structure

| | |
|---|---|
| **Conv Layers** | 输入图片进行卷积操作，每一个 filter 输出一个 feature map，该 feature maps 被共享用于后续 RPN 层和全连接层。 |
| **Region Proposal Network（RPN）** | 这一网络用于生成 region proposals。它通过 softmax 判断 anchors 究竟是属于 foreground 还是 background，然后利用 bounding box regression 对 anchors 进行修正从而获得精确的 proposals。 |
| **ROI Pooling** | 该层从 feature maps 中取出 proposal 的区域，再把这个区域池化成固定长度的输出，送入后续全连接层判定目标类别。 |
| **Classification** | 利用计算出的 proposal 类别，再通过 bounding box regression 获得检测框最终的精确位置 |

### 4.3　RPN

RPN 是一个全卷积网络，可以同时在每个位置预测目标边界和目标分数。由于 RPN 和最先进目标检测网络共享卷积层 [1]，[2]，因此它计算 proposals 的边际成本很小，经过端到端的训练，可以生成高质量的 region proposals。RPN 模块主要包括 Anchor 生成、正负样本划分，RPN 模块 loss 计算和 Proposal 层这四个部分。

#### 4.3.1 Anchor

1. Anchor 的定义是在特征图的每个像素点上生成的矩形方框。它根据下采样率计算，对应了原图上按比例放大的方框。由于图片的物体位置标签也是矩形方框，因此只需要预测 Anchor 相对于 GroundTruth 的偏移量即可，不需要从零开始拟合。

2. 为了适应不同尺度的物体，作者以特征图的每个像素点为中心，分别设置了 3 种不同尺度和 3 种不同长宽比排列组合的 Anchors，确保了 Anchor 和相关计算函数的平移不变性，在小数据集上有更低的过拟合风险。

3. Anchors size，是根据检测图像设置的，其中有一些人们根据以往的设置经验所总结出的规律，比如在 python demo 中，大概 9 个不同大小的 Anchors 里可以从大到小覆盖 800*600 图像中各种目标。但是这种固定尺寸与比例的 Anchors 在实际中并不能精准的框出目标，所以 9 个也只能是一个大概的值。本文中作者将 Anchors 只作为初始的检测框，后续会用 bounding box regression 修正位置。

#### 4.3.2 正负样本划分

在训练阶段需要人为的划分正负样本，目的是让 RPN 网络去学习 “判断 anchor 是正负样本的能力”。设定正样本为 foreground、负样本为 background。通过将一张图像中产生的所有的 Anchors 与 ground true box 区域计算 IoU，根据以下规则划分正负样本：

1. 对每个标定的 ground true box 区域，将与其 IoU 最大的 Anchor 记为正样本，一个 ground true box 可以对应多个 Anchor。

2. 其余的 Anchor，若其与某个标定区域的 IoU 大于 0.7，记为正样本，若与任意一个标定的 IoU 都小于 0.3，记为负样本。

除此之外剩余的 Anchor 和跨越图像边界的 Anchor 都忽略不用。

#### 4.3.3 Loss 计算

RPN 的 loss 计算，包括 classification loss 和 regression loss 两部分，它们俩按一定比重组成损失函数，定义为：

$$L(pi, ti) = 1/Ncls\ Lcls(pi, pi^*) + k * 1/Nreg\ pi^* Lreg(ti, ti^*)$$

其中，Ncls 为 256，是一个 batch 的大小；Nreg 是 Anchor 的总数，k 是两种 loss 的比例；ti 是卷积网络计算出的 Anchor 的位置预测值，ti* 是对应的 ground truth 的真实值。Lcls 为 foreground 和 background 的对数损失；pi 是 Anchor 预测成为目标的概率，pi* 是 foreground 的标签值，根据 Anchor 的正负决定 1 或 0，它的计算公式是：

$$Lcls(pi, pi^*) = -log[pi^* * pi + (1 - pi^*)(1 - pi)]$$

计算 regression loss 时，函数 Smooth-L1 的公式为, 这一公式与 Fast RCNN 一样:

$$smooth_{L1}(x) = \begin{cases} 0.5x^2, & if|x| < 1 \\ |x| - 0.5, & otherwise \end{cases}$$

#### 4.3.4 Proposal 层

Proposal 层综合所有 foreground Anchors 和其对应的偏移量，它的主要流程如下:

1. 生成 Anchors，利用所有的偏移量对 Anchors 做 bounding box regression 回归
2. 按照输入的 foreground softmax scores 由大到小排序 Anchors，提取修正位置后的 foreground Anchors。
3. 将 foreground Anchors 从映射回原图的尺度，判断它们是否大范围超过边界，剔除严重超出边界的 foreground Anchors。
4. 进行非极大值抑制，再次按照非极大值抑制后的 foreground softmax scores 由大到小排序 foreground Anchors，提取前 post_nms_topN 作为 proposal 输出。

### 4.4 RPN 与 fast R-CNN 共享特征

对于独立训练的两种不同任务的网络模型 RPN 和 Fast R-CNN，即使它们的结构、参数完全一样，但是它们各自卷积层内的卷积核会向着不同方向去改变，导致无法共享权重。作者先讨论提出了三种可能的方式，最后还提出了自己的解决方案—四步交替训练法。

#### 4.4.1 三种可能的方法

1. **交替训练:** 该过程首先训练 RPN，再通过 proposal 来训练 Fast R-CNN。然后将 Fast R-CNN 微调后的再用于初始化 RPN，并且重复这个过程，交替循环训练了两次.。原文中实验也多使用改方法。
2. **近似联合训练:** RPN 和 Fast R-CNN 网络合并到同一个网络中进行训练，这样对于每个 SGD 迭代，前向传递产生区域 proposal 被视为固定，并且再训练 Fast R-CNN 的检测器前先计算 proposal。并对共享层，组合来自 RPN 和 Fast R-CNN 损失的反向传播信号。这个方法可以减少大约 25−50% 的训练时间，但是忽略了关于 proposal 边界框坐标的导数。
3. **非近似联合训练:** 在这一方法中，RPN 预测的边界框也是输入的函数，而相对应的有效的反向传播求解器中应包括关于 proposal 边界框坐标的导数。而上一种方法则忽略了这一导数梯度，即网络响应，因此在这一方案中需要一个关于边界框坐标可微分的 ROI 池化层。

#### 4.4.2 四步教体训练

这种方法是由作者提出的。它的核心是交替优化以学习共享特征。第一步，用 ImageNet 预训练的模型将 RPN 网络初始化，并端到端地调用 Region Proposals；第二步，利用第一步的 Region Proposals，由 Fast R-CNN 训练一个单独的检测网络，这个检测网络也是由 ImageNet 预训练的模型初始化的；第三步，初始化后的 RPN 进行训练，注意一定要固定共享卷积层，只可以微调 RPN 独有的层，因为在这一步两个网络就已经开始共享卷积层；第四步，继续保持共享卷积层固定，这次微调 Fast R-CNN 独有的层，从而使两个网络共享相同的卷积层并且可以构成统一的网络。这个四步交替训练也可以进行更多次的迭代

### 4.5 实验结果

#### 4.5.1 mAP

论文中作者在有超过 20 个对象类别的约 5k 训练图像和 5k 测试图像的 PASCAL VOC 2007（也有部分 PASCAL VOC 2012）、MS COCO 等数据集下，主要评估了检测的 mAP，即目标检测任务的实际标准。作者使用的是 ZF 网络的 fast 版本，该网络有 5 个卷积层和 3 个全连接层，以及公开的 VGG−16 模型，该模型有 13 个卷积层和 3 个全连接层。通过 “fast” 模式生成约 2000 个 proposals，则在 Fast R-CNN 框架下，SS （selective search）的 mAP 为 58.7%，EB （edgeboxes）的 mAP 为 58.6%。当 RPN 网络使用至多 300 个 proposals 时，mAP 达到了 59.9 %。因此，RPN 比 SS 或 EB 基于其更少的共享卷积计算消耗，产生了更快的检测系统，同时它更少的 proposals 也减少了区域方面的全连接层消耗。

#### 4.5.2 RPN 消融实验

这一实验的主要目的是通过控制变量法证明 RPN 作为 proposal 方法的有效性。

1. 只进行四步交替训练的前两步，即不对两个网络进行独立的微调，这样最终的 mAP 结果降低到 58.7 % ，但是幅度很小，不过这也可以表明对于两个网络微调步骤确实可以提高一定的 proposal 的质量。
2. 验证 cls 层会影响排名高的 proposas 的精确性这一猜想。通过关闭 RPN 的 cls 层，从未打分的区域中随机采集 N 个 proposals。发现当 N = 1000 时，mAP 几乎不变（55.8 % 55.8%55.8%），但是当 N=100 时，mAP 显著降低。这表明，cls 的分数能够解释排名最高的 proposals 的精度。
3. 作者评估了更强大的网络对 RPN 的 proposals 的影响。在用 VGG-16 训练时，使用 RPN+VGG，非共享特征的结果是 68.5%，高于 SS。这说明 RPN+VGG 生成的 proposals 比 SS 更准确。

#### 4.5.3 超参数的敏感性

论文中作者默认使用 3 尺度和 3 长宽比，比较每个位置使用不同数量和尺寸的锚时，mAP 的变化。根据论文中的数据发现当每个位置只使用一个锚的时候 mAP 会下降 3-4 %，而使用 1 尺度

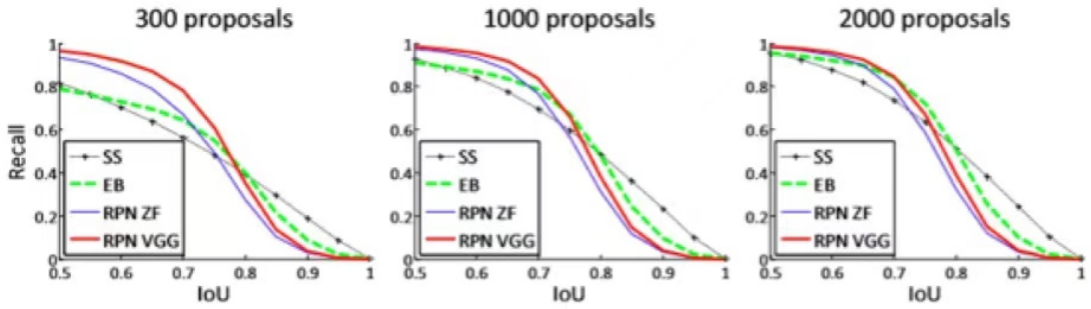


**图 4 IOU 召回率分析**

**Figure 4** Recall vs. IoU overlap ratio on the PASCAL VOC 2007 test set.

和 3 长宽比的时候，mAP 会更高。而使用具有 1 个纵横比 3 个尺度的锚，效果相当于使用具有 3 个纵横比 3 个尺度的锚。这些数据表明使用多个尺寸锚作为参考是有效的，而对于检测的精度而言，尺度和纵横比是其相关维度，具体的相关性还需要进一步分析。

### 4.6 IOU 召回率分析

上图是作者在计算 proposals 为 300,1000,2000 时论文中呈现的对比图，体现了 proposals 与 ground truth 在不同的 IOU 比例下的召回率。由图上结果表明当 proposals 越小，RPN 的表现更好，召回率下降速度也更慢。

### 4.7 一阶段检测 VS 两阶段 proposal + 检测

OverFeat 是一个单级的，类特定的检测流程，论文中使用的是一种新的检测方法。是将一个两级的，与类无关的 proposal 方法和类特定的检测组成的级联方法。Overfeat 和 RPN + Fast R-CNN 的第一级都是使用滑动窗口，前者主要是通过这一方法来确定物体的位置与类别，后者则是为了改进 proposal，希望可以在第二级中将一级的 proposal 自适应池化，提高覆盖区域特征的精确性。论文中的实验结果数据表明，两阶段的方法比一阶段的有更高的 mAP，同时也因为两阶段的方法需要处理的 proposal 更少，它的检测速度更快。

## 5 三篇论文的关系和比较

无论是简单的从命名还是实际算法，R-CNN、Fast R-CNN、Faster R-CNN 都是基于 CNN 一脉相承的目标检测算法。比较三者：

1.R-CNN 的改进是利用了 SS（Selective Search）算法。它通过利用 SS 算法来提取可能的 RoIs 区域，并对每个 RoI 采用 CNN 进行目标分类。

2.Fast R-CNN 的改进是引入了池化操作 RoI pooling。它是通过 CNN 提取图的 feature maps，再从 feature maps 上提取 RoIs，最后进行 classification。这么做的好处是避免同时调用多个 CNN

网络，实现共享。

3.Faster R-CNN 的改进是引入 RPN 替换 SS 算法，减小 region proposal 的时间开销，真正实现了基于深度学习的端到端的目标检测算法。和 Fast R-CNN 相比在达到同样精度的 mAP 时，检测速度从可以从原来的 0.5fps 上升至 7fps，大幅度提高。

## 参考文献


1 K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In ECCV, 2014.

2 Girshick, Ross and Donahue, Jeff and Darrell, Trevor and Malik, Jitendra. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR),2014.

3 Girshick, Ross. Fast R-CNN in Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2015.

4 Ren, Shaoqing and He, Kaiming and Girshick, Ross and Sun, Jian. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks in Advances in Neural Information Processing Systems, 2015, 28.


# Title


Xiaoze Yu[1], Yunmei Guan[2] & Jiazhen Song[3]

1. *Nankai University, SNO* 2013814;
2. *Nankai University, SNO* 2013750;
3. *Nankai University, SNO* 2013904
* Corresponding author. E-mail: