[基本概念 3](#_Toc151461334)

[一 IoU（Intersection over Union） 5](#_Toc1260933413)

[二 基于深度学习的目标检测 7](#_Toc992642446)

[1、传统目标检测 7](#_Toc1394885192)

[1.1 区域选择 7](#_Toc1729373955)

[1.2 特征提取 8](#_Toc135728087)

[1.3 分类器 8](#_Toc1135031389)

[1.4 总结 9](#_Toc1544108298)

[2、基于深度学习的目标检测 9](#_Toc794498848)

[2.1相关研究 12](#_Toc690193300)

[2.2、基于区域提名的方法 16](#_Toc1885469050)

[（1）R-CNN 17](#_Toc1366233230)

[（2）SPP-net 19](#_Toc1307937151)

[（3）Fast R-CNN（ICCV2015） 21](#_Toc1684201300)

[（4）Faster R-CNN（NIPS2015） 23](#_Toc1409832549)

[3、基于回归方法的深度学习目标检测 26](#_Toc1253848778)

[3.1 YOLO（CVPR2016） 26](#_Toc1531139529)

[3.2 SSD 29](#_Toc148404238)

[3.3 基于残差（Residual）方法的深度学习目标检测 30](#_Toc858059263)

[（1）相关概念 30](#_Toc1682320843)

[（2）深度残差学习（Deep Residual Learning）的思想 32](#_Toc1887856328)

[三、提高目标检测方法 36](#_Toc690355154)

[1、难分样本挖掘（hard negative mining） 36](#_Toc849886691)

[2、多层特征融合 36](#_Toc1309916214)

[3、使用上下文信息 37](#_Toc75145655)

目标识别有关

# 基本概念

1）CNN：Convolutional Neural Networks

2）FC：Fully Connected 全连接

3）IoU：Intersection over Union （IoU的值定义：Region Proposal与Ground Truth的窗口的交集比并集的比值，如果IoU低于0.5，那么相当于目标还 是没有检测到）

4）ICCV：International Conference on Computer Vision

5）R-CNN：Region-based Convolutional Neural Networks

6）AR：Average Recall

7）mAP：mean Average Precision 平均精度

8）RPN：Region Proposal Networks

基于区域提名的神经网络 or 基于候选区域的神经网络

9）FAIR：Facebook AI Research

10）w.r.t.：with respect to

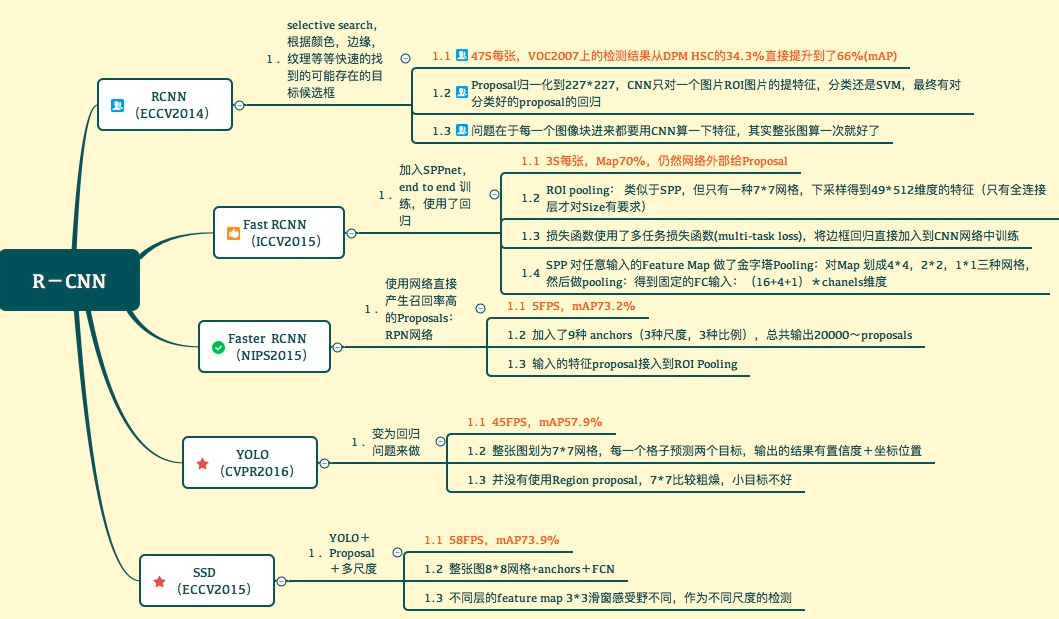
11）crop：截取原图片的一个固定大小的patch

warp：将原图片的ROI缩放到一个固定大小的patch

无论是crop还是warp，都无法保证在不失真的情况下将图片传入到CNN当中：

- crop：物体可能会产生截断，尤其是长宽比大的图片。

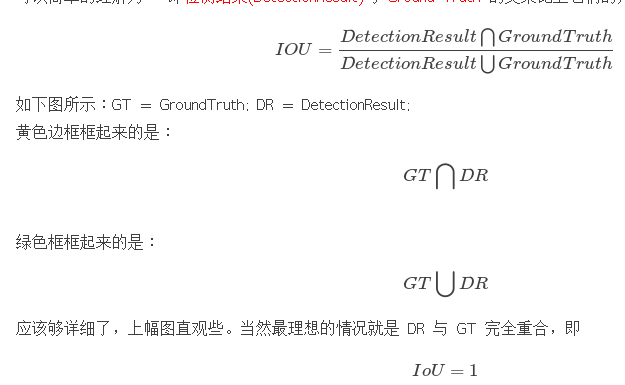
- warp：物体被拉伸，失去“原形”，尤其是长宽比大的图片

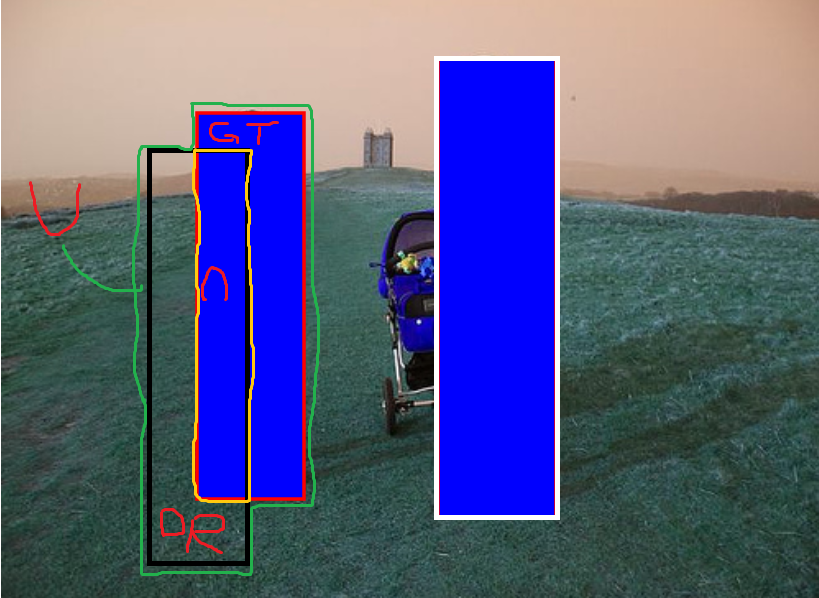


# 一 IoU（Intersection over Union）

检测评价函数IOU。

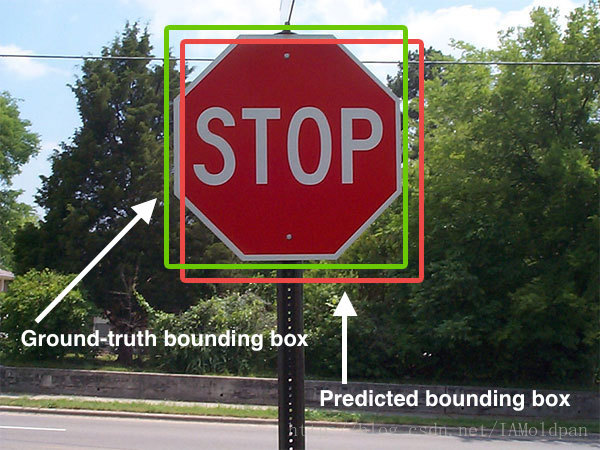
在目标检测的评价体系中，有一个参数叫做 IoU ，简单来讲就是模型产生的目标窗口和原来标记窗口的交叠率。具体我们可以简单的理解为： 即检测结果(DetectionResult)与 Ground Truth 的交集比上它们的并集，即为检测的准确率 IoU 。





通常我们在 HOG + Linear SVM object detectors 和 Convolutional Neural Network detectors (R-CNN, Faster R-CNN, YOLO)中使用该方法检测其性能。注意，这个测量方法和你在任务中使用的物体检测算法没有关系。

IoU是一个简单的测量标准，只要是在输出中得出一个预测范围(bounding boxex)的任务都可以用IoU来进行测量。为了可以使IoU用于测量任意大小形状的物体检测，我们需要：   
1、 ground-truth bounding boxes（人为在训练集图像中标出要检测物体的大概范围）；   
2、我们的算法得出的结果范围。



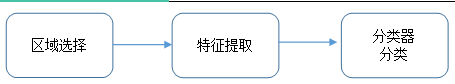
# 二 基于深度学习的目标检测

普通的深度学习监督算法主要是用来做分类，而在ILSVRC（ImageNet Large Scale Visual Recognition Challenge)竞赛以及实际的应用中，还包括目标定位和目标检测等任务。其中目标定位是不仅仅要识别出来是什么物体（即分类），而且还要预测物体的位置，位置一般用边框（bounding box）标记。而目标检测实质是多目标的定位，即要在图片中定位多个目标物体，包括分类和定位。

目标检测对于人类来说并不困难，通过对图片中不同颜色模块的感知很容易定位并分类出其中目标物体，但对于计算机来说，面对的是RGB像素矩阵，很难从图像中直接得到狗和猫这样的抽象概念并定位其位置，再加上有时候多个物体和杂乱的背景混杂在一起，目标检测更加困难。

## 1、传统目标检测

传统的目标检测一般使用滑动窗口的框架，主要包括三个步骤：



### 1.1 区域选择

利用不同尺寸的滑动窗口框住图中的某一部分作为候选区域。

这一步是为了对目标的位置进行定位。由于目标可能出现在图像的任何位置，而且目标的大小、长宽比例也不确定，所以最初采用滑动窗口的策略对整幅图像进行遍历，而且需要设置不同的尺度，不同的长宽比。这种穷举的策略虽然包含了目标所有可能出现的位置，但是缺点也是显而易见的：时间复杂度太高，产生冗余窗口太多，这也严重影响后续特征提取和分类的速度和性能。（实际上由于受到时间复杂度的问题，滑动窗口的长宽比一般都是固定的设置几个，所以对于长宽比浮动较大的多类别目标检测，即便是滑动窗口遍历也不能得到很好的区域）

### 1.2 特征提取

提取候选区域相关的视觉特征，比如人脸检测常用的Harr特征；行人检测和普通目标检测常用的HOG特征等；

由于目标的形态多样性，光照变化多样性，背景多样性等因素使得设计一个鲁棒的特征并不是那么容易。然而提取特征的好坏直接影响到分类的准确性。（这个阶段常用的特征有SIFT、HOG等）

### 1.3 分类器

利用分类器进行识别，比如常用的SVM模型。

传统的目标检测中，多尺度形变部件模型DPM（Deformable Part Model）[13]是出类拔萃的，连续获得VOC（Visual Object Class）2007到2009的检测冠军，2010年其作者Felzenszwalb Pedro被VOC授予”终身成就奖”。DPM把物体看成了多个组成的部件（比如人脸的鼻子、嘴巴等），用部件间的关系来描述物体，这个特性非常符合自然界很多物体的非刚体特征。DPM可以看做是HOG+SVM的扩展，很好的继承了两者的优点，在人脸检测、行人检测等任务上取得了不错的效果，但是DPM相对复杂，检测速度也较慢，从而也出现了很多改进的方法。正当大家热火朝天改进DPM性能的时候，基于深度学习的目标检测横空出世，迅速盖过了DPM的风头，很多之前研究传统目标检测算法的研究者也开始转向深度学习。

### 1.4 总结

传统目标检测存在的两个主要问题：一个是基于滑动窗口的区域选择策略没有针对性，时间复杂度高，窗口冗余；二是手工设计的特征对于多样性的变化并没有很好的鲁棒性。

## 2、基于深度学习的目标检测

基于深度学习的目标检测发展起来后，其实效果也一直难以突破。比如文献[6]中的算法在VOC 2007测试集合上的mAP只能30%多一点，文献[7]中的OverFeat在ILSVRC 2013测试集上的mAP只能达到24.3%。2013年R-CNN诞生了，VOC 2007测试集的mAP被提升至48%，2014年时通过修改网络结构又飙升到了66%，同时ILSVRC 2013测试集的mAP也被提升至31.4%。

那么，对于传统目标检测任务存在的两个主要问题，我们该如何解决呢？

对于滑动窗口存在的问题，region proposal提供了很好的解决方案。region proposal（候选区域）是预先找出图中目标可能出现的位置。但由于region proposal利用了图像中的纹理、边缘、颜色等信息，可以保证在选取较少窗口（几千个甚至几百个）的情况下保持较高的召回率。这大大降低了后续操作的时间复杂度，并且获取的候选窗口要比滑动窗口的质量更高（滑动窗口固定长宽比）。比较常用的region proposal算法有selective Search和edge Boxes，如果想具体了解region proposal可以看一下PAMI2015的[“What makes for effective detection proposals？”](https://www.mpi-inf.mpg.de/departments/computer-vision-and-multimodal-computing/research/object-recognition-and-scene-understanding/how-good-are-detection-proposals-really/)

有了候选区域，剩下的工作实际就是对候选区域进行图像分类的工作（特征提取+分类）。对于图像分类，不得不提的是2012年ImageNet大规模视觉识别挑战赛（ILSVRC）上，机器学习泰斗Geoffrey Hinton教授带领学生Krizhevsky使用卷积神经网络将ILSVRC分类任务的Top-5 error降低到了15.3%，而使用传统方法的第二名top-5 error高达 26.2%。此后，卷积神经网络占据了图像分类任务的绝对统治地位，微软最新的ResNet和谷歌的Inception V4模型的top-5 error降到了4%以内多，这已经超越人在这个特定任务上的能力。所以目标检测得到候选区域后使用CNN对其进行图像分类是一个不错的选择。

2014年，RBG（Ross B. Girshick）大神使用region proposal+CNN代替传统目标检测使用的滑动窗口+手工设计特征，设计了R-CNN框架，使得目标检测取得巨大突破，并开启了基于深度学习目标检测的热潮。

R-CNN是Region-based Convolutional Neural Networks的缩写，中文翻译是基于区域的卷积神经网络，是一种结合区域提名（Region Proposal）和卷积神经网络（CNN）的目标检测方法。Ross Girshick在2013年的开山之作《Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation》[1]奠定了这个子领域的基础，这篇论文后续版本发表在CVPR 2014[2]，期刊版本发表在PAMI 2015[3]。

其实在R-CNN之前已经有很多研究者尝试用Deep Learning的方法来做目标检测了，包括OverFeat[7]，但R-CNN是第一个真正可以工业级应用的解决方案，这也和深度学习本身的发展类似，神经网络、卷积网络都不是什么新概念，但在本世纪突然真正变得可行，而一旦可行之后再迅猛发展也不足为奇了。

R-CNN这个领域目前研究非常活跃，先后出现了R-CNN[1,2,3,18]、SPP-net[4,19]、Fast R-CNN[14, 20] 、Faster R-CNN[5,21]、R-FCN[16,24]、YOLO[15,22]、SSD[17,23]等研究。Ross Girshick作为这个领域的开山鼻祖总是神一样的存在，R-CNN、Fast R-CNN、Faster R-CNN、YOLO都和他有关。这些创新的工作其实很多时候是把一些传统视觉领域的方法和深度学习结合起来了，比如选择性搜索（Selective Search)和图像金字塔（Pyramid）等。

深度学习相关的目标检测方法也可以大致分为两派：

（1）基于区域提名的，如R-CNN、SPP-net、Fast R-CNN、Faster R-CNN、 R-FCN；

（2）端到端（End-to-End），无需区域提名的，如YOLO、SSD。

目前来说，基于区域提名的方法依然占据上风，但端到端的方法速度上优势明显，后续的发展拭目以待。

### 2.1相关研究

（1）region proposal

目标检测的第一步是要做区域提名（Region Proposal），也就是找出可能的感兴趣区域（Region Of Interest, ROI）。区域提名类似于-光学字符识别（OCR）领域的切分，OCR切分常用过切分方法，简单说就是尽量切碎到小的连通域（比如小的笔画之类），然后再根据相邻块的一些形态学特征进行合并。但目标检测的对象相比OCR领域千差万别，而且图形不规则，大小不一，所以一定程度上可以说区域提名是比OCR切分更难的一个问题

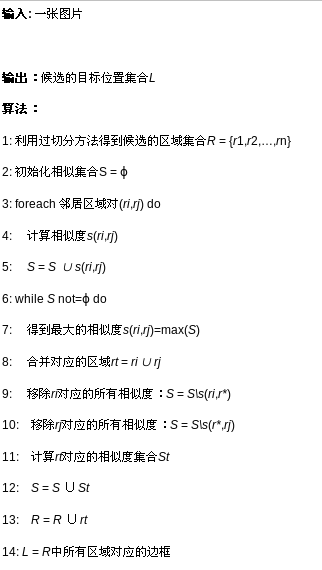
区域提名（候选区域）可能的方法有：

A）滑动窗口。滑动窗口本质上就是穷举法，利用不同的尺度和长宽比把所有可 能的大大小小的块都穷举出来，然后送去识别，识别出来概率大的就留下来。 很明显，这样的方法复杂度太高，产生了很多的冗余候选区域，在现实当中 不可行。

B）规则块。在穷举法的基础上进行了一些剪枝，只选用固定的大小和长宽比。 这在一些特定的应用场景是很有效的，比如拍照搜题APP小猿搜题中的汉 字检测，因为汉字方方正正，长宽比大多比较一致，因此用规则块做区域提 名是一种比较合适的选择。但是对于普通的目标检测来说，规则块依然需要 访问很多的位置，复杂度高。

C）选择性搜索。从机器学习的角度来说，前面的方法召回是不错了，但是精度 差强人意，所以问题的核心在于如何有效地去除冗余候选区域。其实冗余候 选区域大多是发生了重叠，选择性搜索利用这一点，自底向上合并相邻的重 叠区域，从而减少冗余。

选择性搜索的具体算法细节[8]如算法1所示。总体上选择性搜索是自底向上不断合并候选区域的迭代过程。



从算法不难看出，R中的区域都是合并后的，因此减少了不少冗余，相当于准确率提升了，但是别忘了我们还需要继续保证召回率，因此算法1中的相似度计算策略就显得非常关键了。如果简单采用一种策略很容易错误合并不相似的区域，比如只考虑轮廓时，不同颜色的区域很容易被误合并。选择性搜索采用多样性策略来增加候选区域以保证召回，比如颜色空间考虑RGB、灰度、HSV及其变种等，相似度计算时既考虑颜色相似度，又考虑纹理、大小、重叠情况等。

总体上，选择性搜索是一种比较朴素的区域提名方法，被早期的基于深度学习的目标检测方法（包括Overfeat和R-CNN等）广泛利用，但被当前的新方法弃用了。

区域提名并不只有以上所说的三种方法，实际上这块是非常灵活的，因此变种也很多，有兴趣的读者不妨参考一下文献[12]。

（2）OverFeat

OverFeat[7][9]是用CNN统一来做分类、定位和检测的经典之作，作者是深度学习大神之一————Yann Lecun在纽约大学的团队。OverFeat也是ILSVRC 2013任务3（分类+定位）的冠军得主[10]。

OverFeat的核心思想有三点：

A）区域提名：结合滑动窗口和规则块，即多尺度（multi-scale)的滑动窗口；

B）分类和定位：统一用CNN来做分类和预测边框位置，模型与AlexNet[12] 类似，其中1-5层为特征抽取层，即将图片转换为固定维度的特征向量，6-9 层为分类层(分类任务专用)，不同的任务（分类、定位、检测）公用特征抽 取层（1-5层），只替换6-9层；

C）累积：因为用了滑动窗口，同一个目标对象会有多个位置，也就是多个视角； 因为用了多尺度，同一个目标对象又会有多个大小不一的块。这些不同位置 和不同大小块上的分类置信度会进行累加，从而使得判定更为准确。

OverFeat的关键步骤有四步：

A）利用滑动窗口进行不同尺度的区域提名，然后使用CNN模型对每个区域进 行分类，得到类别和置信度。从图2中可以看出，不同缩放比例时，检测出 来的目标对象数量和种类存在较大差异；

B）利用多尺度滑动窗口来增加检测数量，提升分类效果，如图3所示；

C）用回归模型预测每个对象的位置，从图4中来看，放大比例较大的图片，边 框数量也较多；

D）边框合并。

Overfeat是CNN用来做目标检测的早期工作，主要思想是采用了多尺度滑动窗口来做分类、定位和检测，虽然是多个任务但重用了模型前面几层，这种模型重用的思路也是后来R-CNN系列不断沿用和改进的经典做法。

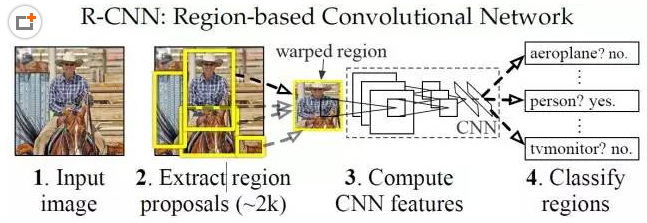
当然Overfeat也是有不少缺点的，至少速度和效果都有很大改进空间，后面的R-CNN系列在这两方面做了很多提升。

### 2.2、基于区域提名的方法

主要介绍基于区域提名的方法，包括R-CNN、SPP-net、Fast R-CNN、Faster R-CNN、R-FCN。

#### （1）R-CNN

如前面所述，早期的目标检测，大都使用滑动窗口的方式进行窗口提名，这种方式本质是穷举法，R-CNN[1,2,3]采用的是Selective Search。



上面的框架图清晰的给出了R-CNN的目标检测流程：

a) 输入测试图像

b) 利用selective search算法在图像中提取2000个左右的region proposal。

c) 将每个region proposal缩放（warp）成227x227的大小并输入到CNN， 将CNN的fc7层的输出作为特征。

d) 将每个region proposal提取到的CNN特征输入到SVM进行分类。

针对上面的框架给出几点解释：

A）上面的框架图是测试的流程图，要进行测试我们首先要训练好提取特征的 CNN模型，以及用于分类的SVM：使用在ImageNet上预训练的模型 （AlexNet/VGG16）进行微调得到用于特征提取的CNN模型，然后利用 CNN模型对训练集提取特征训练SVM。

B）对每个region proposal缩放到同一尺度是因为CNN全连接层输入需要保 证维度固定。

C）上图少画了一个过程——对于SVM分好类的region proposal做边框回归 （bounding-box regression)，边框回归是对region proposal进行纠正 的线性回归算法，为了让region proposal提取到的窗口跟目标真实窗口更 吻合。因为region proposal提取到的窗口不可能跟人手工标记那么准，如 果region proposal跟目标位置偏移较大，即便是分类正确了，但是由于 IoU(region proposal与Ground Truth的窗口的交集比并集的比值)低于 0.5，那么相当于目标还是没有检测到。

Overfeat可以看做是R-CNN的一个特殊情况，只需要把Selective Search换成多尺度的滑动窗口，每个类别的边框回归器换成统一的边框回归器，SVM换为多层网络即可。但是Overfeat实际比R-CNN快9倍，这主要得益于卷积相关的共享计算。

事实上，R-CNN有很多缺点：

A）重复计算：R-CNN虽然不再是穷举，但依然有两千个左右的候选框，这些 候选框都需要进行CNN操作，计算量依然很大，其中有不少其实是重复计 算；

B）SVM模型：而且还是线性模型，在标注数据不缺的时候显然不是最好的选 择；

C）训练测试分为多步：区域提名、特征提取、分类、回归都是断开的训练的过 程，中间数据还需要单独保存；

D）训练的空间和时间代价很高：卷积出来的特征需要先存在硬盘上，这些特征 需要几百G的存储空间；

E）慢：前面的缺点最终导致R-CNN出奇的慢，GPU上处理一张图片需要13 秒，CPU上则需要53秒[2]。

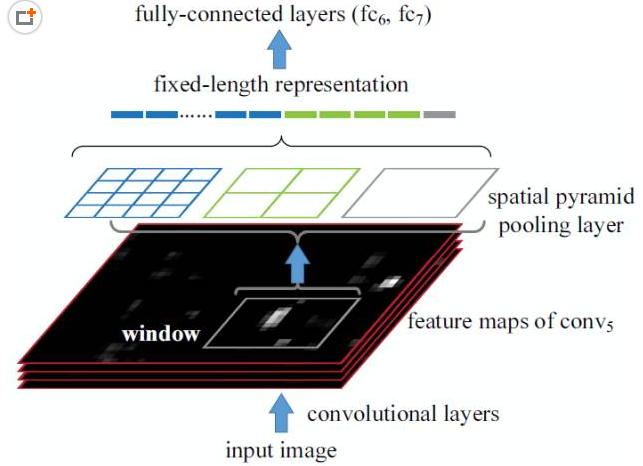
当然，R-CNN这次是冲着效果来的，其中ILSVRC 2013数据集上的mAP由Overfeat的24.3%提升到了31.4%，第一次有了质的改变。

#### （2）SPP-net

(Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition)

先看一下R-CNN为什么检测速度这么慢，一张图都需要47s！仔细看下R-CNN框架发现，对图像提完region proposal（2000个左右）之后将每个proposal当成一张图像进行后续处理(CNN提特征+SVM分类)，实际上对一张图像进行了2000次提特征和分类的过程！

有没有方法提速呢？好像是有的，这2000个region proposal不都是图像的一部分吗，那么我们完全可以对图像提一次卷积层特征，然后只需要将region proposal在原图的位置映射到卷积层特征图上，这样对于一张图像我们只需要提一次卷积层特征，然后将每个region proposal的卷积层特征输入到全连接层做后续操作。（对于CNN来说，大部分运算都耗在卷积操作上，这样做可以节省大量时间）。现在的问题是每个region proposal的尺度不一样，直接这样输入全连接层肯定是不行的，因为全连接层输入必须是固定的长度。SPP-NET恰好可以解决这个问题：



上图对应的就是SPP-NET的网络结构图，任意给一张图像输入到CNN，经过卷积操作我们可以得到卷积特征（比如VGG16最后的卷积层为conv5\_3，共产生512张特征图）。图中的window是就是原图一个region proposal对应到特征图的区域，只需要将这些不同大小window的特征映射到同样的维度，将其作为全连接的输入，就能保证只对图像提取一次卷积层特征。SPP-NET使用了空间金字塔采样（spatial pyramid pooling）：将每个window划分为4\*4, 2\*2, 1\*1的块，然后每个块使用max-pooling下采样，这样对于每个window经过SPP层之后都得到了一个长度为(4\*4+2\*2+1)\*512维度的特征向量，将这个作为全连接层的输入进行后续操作。

小结：使用SPP-NET相比于R-CNN可以大大加快目标检测的速度，但是依然存在着很多问题：

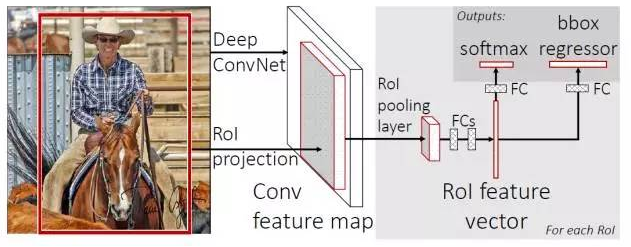
a) 训练分为多个阶段，步骤繁琐: 微调网络+训练SVM+训练边框回归器

b) SPP-NET在微调网络的时候固定了卷积层，只对全连接层进行微调，而对于一个新的任务，有必要对卷积层也进行微调。（分类的模型提取的特征更注重高层语义，而目标检测任务除了语义信息还需要目标的位置信息）

针对这两个问题，RBG又提出Fast R-CNN, 一个精简而快速的目标检测框架。

#### （3）Fast R-CNN（ICCV2015）

 有了前边R-CNN和SPP-NET的介绍，我们直接看Fast R-CNN的框架图：



与R-CNN框架图对比，可以发现主要有两处不同：一是最后一个卷积层后加了一个ROI pooling layer，二是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。

A）ROI pooling layer实际上是SPP-NET的一个精简版，SPP-NET对每个proposal使用了不同大小的金字塔映射，而ROI pooling layer只需要下采样到一个7x7的特征图。对于VGG16网络conv5\_3有512个特征图，这样所有region proposal对应了一个7\*7\*512维度的特征向量作为全连接层的输入。

B）R-CNN训练过程分为了三个阶段，而Fast R-CNN直接使用softmax替代SVM分类，同时利用多任务损失函数边框回归也加入到了网络中，这样整个的训练过程是端到端的(除去region proposal提取阶段)。

C）Fast R-CNN在网络微调的过程中，将部分卷积层也进行了微调，取得了更好的检测效果。

小结：Fast R-CNN融合了R-CNN和SPP-NET的精髓，并且引入多任务损失函数，使整个网络的训练和测试变得十分方便。在Pascal VOC2007训练集上训练，在VOC2007测试的结果为66.9%(mAP)，如果使用VOC2007+2012训练集训练，在VOC2007上测试结果为70%（数据集的扩充能大幅提高目标检测性能）。使用VGG16每张图像总共需要3s左右。

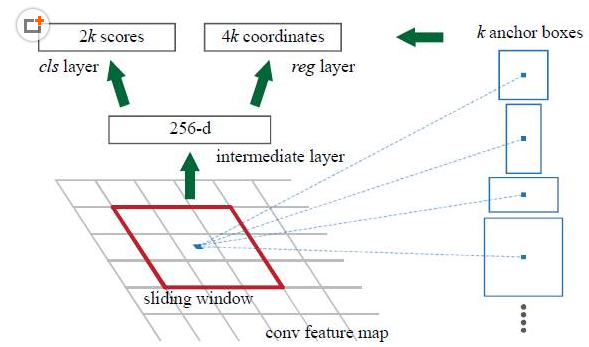
缺点：region proposal的提取使用selective search，目标检测时间大多消耗在这上面（提region proposal 2~3s，而提特征分类只需0.32s），无法满足实时应用，而且并没有实现真正意义上的端到端训练测试（region proposal使用selective search先提取处来）。那么有没有可能直接使用CNN直接产生region proposal并对其分类？Faster R-CNN框架就是符合这样需要的目标检测框架。

#### （4）Faster R-CNN（NIPS2015）

(Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks)

在region proposal + CNN分类的这种目标检测框架中，region proposal质量好坏直接影响到目标检测任务的精度。如果找到一种方法只提取几百个或者更少的高质量的预选窗口，而且召回率很高，这不但能加快目标检测速度，还能提高目标检测的性能（假阳例少）。RPN(Region Proposal Networks)网络应运而生。

RPN的核心思想是使用卷积神经网络直接产生region proposal，使用的方法本质上就是滑动窗口。RPN的设计比较巧妙，RPN只需在最后的卷积层上滑动一遍，因为anchor机制和边框回归可以得到多尺度多长宽比的region proposal。



我们直接看上边的RPN网络结构图（使用了ZF模型???），给定输入图像（假设分辨率为600\*1000），经过卷积操作得到最后一层的卷积特征图（大小约为40\*60）。在这个特征图上使用3\*3的卷积核（滑动窗口）与特征图进行卷积，最后一层卷积层共有256个feature map，那么这个3\*3的区域卷积后可以获得一个256维的特征向量，后边接cls layer和reg layer分别用于分类和边框回归（跟Fast R-CNN类似，只不过这里的类别只有目标和背景两个类别）。3\*3滑窗对应的每个特征区域同时预测输入图像3种尺度（128,256,512），3种长宽比（1:1,1:2,2:1）的region proposal，这种映射的机制称为anchor。所以对于这个40\*60的feature map，总共有约20000(40\*60\*9)个anchor，也就是预测20000个region proposal。

这样设计的好处是什么呢？虽然现在也是用的滑动窗口策略，但是：滑动窗口操作是在卷积层特征图上进行的，维度较原始图像降低了16\*16倍（中间经过了4次2\*2的pooling操作）；多尺度采用了9种anchor，对应了三种尺度和三种长宽比，加上后边接了边框回归，所以即便是这9种anchor外的窗口也能得到一个跟目标比较接近的region proposal。

NIPS2015版本的Faster R-CNN使用的检测框架是RPN网络+Fast R-CNN网络分离进行的目标检测，整体流程跟Fast R-CNN一样，只是region proposal现在是用RPN网络提取的（代替原来的selective search）。同时作者为了让RPN的网络和Fast R-CNN网络实现卷积层的权值共享，训练RPN和Fast R-CNN的时候用了4阶段的训练方法:

A）使用在ImageNet上预训练的模型初始化网络参数，微调RPN网络；

B）使用(1)中RPN网络提取region proposal训练Fast R-CNN网络；

C）使用(2)的Fast R-CNN网络重新初始化RPN, 固定卷积层进行微调；

D）固定(2)中Fast R-CNN的卷积层，使用(3)中RPN提取的region proposal微调网络。

权值共享后的RPN和Fast R-CNN用于目标检测精度会提高一些。

使用训练好的RPN网络，给定测试图像，可以直接得到边缘回归后的region proposal，根据region proposal的类别得分对RPN网络进行排序，并选取前300个窗口作为Fast R-CNN的输入进行目标检测，使用VOC07+12训练集训练，VOC2007测试集测试mAP达到73.2%（selective search + Fast R-CNN是70%）， 目标检测的速度可以达到每秒5帧（selective search+Fast R-CNN是2~3s一张）。

需要注意的是，最新的版本已经将RPN网络和Fast R-CNN网络结合到了一起——将RPN获取到的proposal直接连到ROI pooling层，这才是一个真正意义上的使用一个CNN网络实现端到端目标检测的框架。

小结：Faster R-CNN将一直以来分离的region proposal和CNN分类融合到了一起，使用端到端的网络进行目标检测，无论在速度上还是精度上都得到了不错的提高。然而Faster R-CNN还是达不到实时的目标检测，预先获取region proposal，然后在对每个proposal分类计算量还是比较大。比较幸运的是YOLO这类目标检测方法的出现让实时性也变的成为可能。

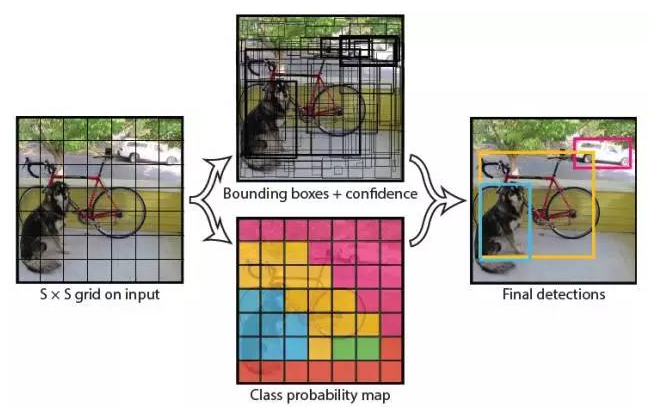
总的来说，从R-CNN, SPP-NET, Fast R-CNN, Faster R-CNN一路走来，基于深度学习目标检测的流程变得越来越精简，精度越来越高，速度也越来越快。可以说基于region proposal的R-CNN系列目标检测方法是当前目标最主要的一个分支。

## 3、基于回归方法的深度学习目标检测

Faster R-CNN的方法目前是主流的目标检测方法，但是速度上并不能满足实时的要求。YOLO一类的方法慢慢显现出其重要性，这类方法使用了回归的思想，即给定输入图像，直接在图像的多个位置上回归出这个位置的目标边框以及目标类别。

### 3.1 YOLO（CVPR2016）

(You Only Look Once: Unified, Real-Time Object Detection)



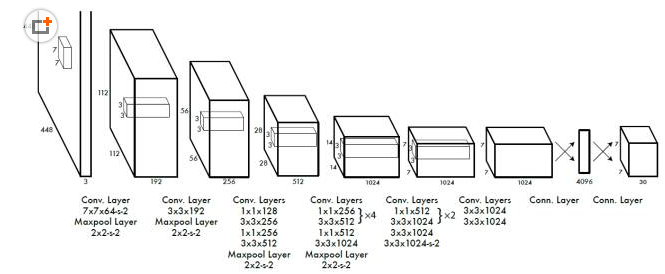
我们直接看上面YOLO的目标检测的流程图：

A）给个一个输入图像，首先将图像划分成7\*7的网格

B）对于每个网格，我们都预测2个值（包括每个边框是目标的置信度以及每个边框区域在多个类别上的概率）

C）根据上一步可以预测出7\*7\*2个目标窗口，然后根据阈值去除可能性比较低的目标窗口，最后NMS去除冗余窗口即可。

可以看到整个过程非常简单，不需要中间的region proposal在找目标，直接回归便完成了位置和类别的判定。



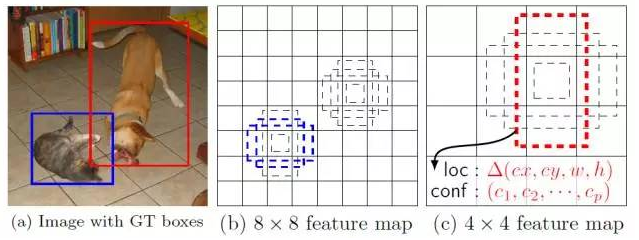
那么如何才能做到直接在不同位置的网格上回归出目标的位置和类别信息呢？上面是YOLO的网络结构图，前边的网络结构跟GoogLeNet的模型比较类似，主要的是最后两层的结构，卷积层之后接了一个4096维的全连接层，然后后边又全连接到一个7\*7\*30维的张量上。实际上这7\*7就是划分的网格数，现在要在每个网格上预测两个目标可能的位置以及这个位置的目标置信度和类别，也就是每个网格预测两个目标，每个目标的信息有4维坐标信息(中心点坐标+长宽)，1个是目标的置信度，还有类别数20(VOC上20个类别)，总共就是(4+1)\*2+20 = 30维的向量。这样可以利用前边4096维的全图特征直接在每个网格上回归出目标检测需要的信息（边框信息加类别）。

小结：YOLO将目标检测任务转换成一个回归问题，大大加快了检测的速度，使得YOLO可以每秒处理45张图像。而且由于每个网络预测目标窗口时使用的是全图信息，使得false positive比例大幅降低（充分的上下文信息）。但是YOLO也存在问题：没有了region proposal机制，只使用7\*7的网格回归会使得目标不能非常精准的定位，这也导致了YOLO的检测精度并不是很高。

### 3.2 SSD

  (SSD: Single Shot MultiBox Detector)

上面分析了YOLO存在的问题，使用整图特征在7\*7的粗糙网格内回归对目标的定位并不是很精准。那是不是可以结合region proposal的思想实现精准一些的定位？SSD结合YOLO的回归思想以及Faster R-CNN的anchor机制做到了这点。



上图是SSD的一个框架图，首先SSD获取目标位置和类别的方法跟YOLO一样，都是使用回归，但是YOLO预测某个位置使用的是全图的特征，SSD预测某个位置使用的是这个位置周围的特征（感觉更合理一些）。那么如何建立某个位置和其特征的对应关系呢？可能你已经想到了，使用Faster R-CNN的anchor机制。如SSD的框架图所示，假如某一层特征图(图b)大小是8\*8，那么就使用3\*3的滑窗提取每个位置的特征，然后这个特征回归得到目标的坐标信息和类别信息(图c)。

不同于Faster R-CNN，这个anchor是在多个feature map上，这样可以利用多层的特征并且自然的达到多尺度（不同层的feature map 3\*3滑窗感受也不同）。

小结：SSD结合了YOLO中的回归思想和Faster R-CNN中的anchor机制，使用全图各个位置的多尺度区域特征进行回归，既保持了YOLO速度快的特性，也保证了窗口预测的跟Faster R-CNN一样比较精准。SSD在VOC2007上mAP可以达到72.1%，速度在GPU上达到58帧每秒。

总结：YOLO的提出给目标检测一个新的思路，SSD的性能则让我们看到了目标检测在实际应用中真正的可能性。

### 3.3 基于残差（Residual）方法的深度学习目标检测

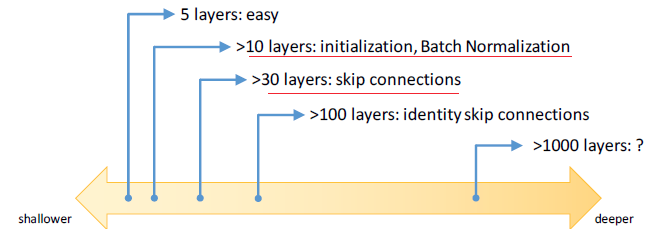
#### （1）相关概念

它使用**残差学习**的这一思想使得学习更深的网络成为可能，从而学习到更好的表达。

层数越多的神经网络越难以训练。当层数超过一定数量后，传统的深度网络就会因优化问题而出现欠拟合（underfitting）的情况。残差学习框架大幅降低训练更深层网络的难度，也使准确率得到显著提升。在 ImageNet 和 COCO 2015 竞赛中，共有 152 层的深度残差网络 ResNet 在图像分类、目标检测和语义分割各个分项都取得最好成绩，相关论文更是连续两次获得 CVPR 最佳论文。

     最新研究发现，当残差网络将身份映射作为 skip connection（跳跃连接） 并实现 inter-block activation（块间激活），正向和反向信号能够直接从一个区块传播到另一个区块，这样就达到了 1001 层的残差网络。由此可见，神经网络的深度这一非常重要的因素，还有很大的提升空间。

A）深度谱

B）为使用网络层数更多，通常采用的方法有：初始化算法，BN方法

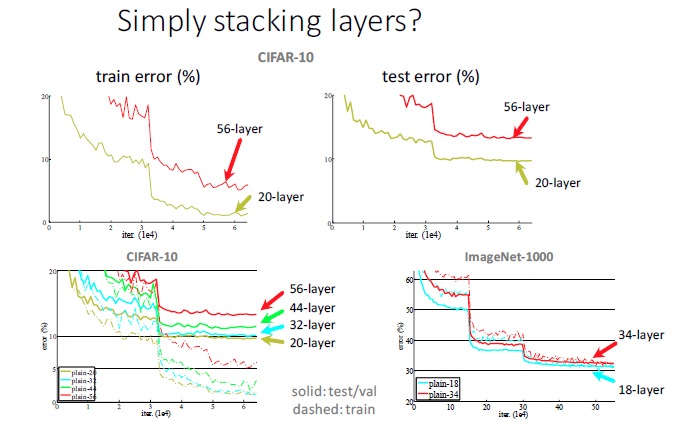
C）是否简单堆叠的网络层数越多，训练误差和测试误差就越小?答案正好相反

D）目前流行的深度神经网络结构大致可以分为三类：

- 直线型（如AlexNet, VGGNet）

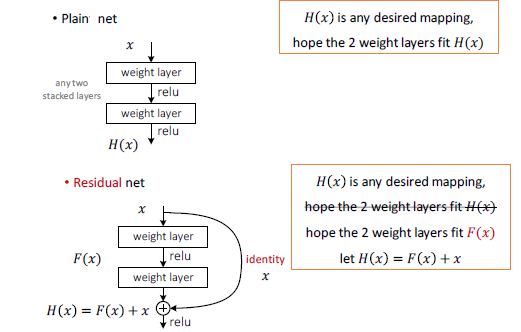
- 局部双分支型（ResNet）

- 局部多分支型（GoogleNet）



#### （2）深度残差学习（Deep Residual Learning）的思想

假如目前有一个可以工作的很好的网络A，这时来了一个比它更深的网络B，只需要让B的前一部分与A完全相同，后一部分只实现一个恒等映射（identity mapping），这样B最起码能获得与A相同的性能，而不至于更差。深度残差学习的思想也由此而产生，既然B后面的部分完成的是恒等映射，何不在训练网络的时候加上这一先验（在网络训练过程中，加入先验信息指导非常重要，合理的先验往往会取得非常好的效果），于是构造网络的时候加入了捷径（shortcut）连接，即每层的输出不是传统神经网络中的输入的映射，而是输入的映射和输入的叠加，如下图中的"Residual net"所示。



在Residual net中：

A）identity：为恒等映射，此条路径一直存在

B）F(x)：为需要学习的残差函数(residual function)：H(x)-x = F(x)

问题的重新表示或预处理会简化问题的优化!

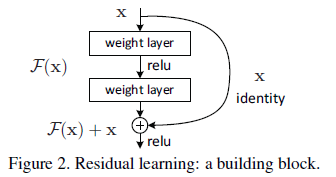
假设我们期望的网络层关系映射为 H(x), 我们让 the stacked nonlinear layers 拟合另一个映射， F(x):= H(x)-x , 那么原先的映射就是 F(x)+x。 这里我们假设优化残差映射F(x) 比优化原来的映射 H(x)容易。

这里我们首先求取残差映射 F(x):= H(x)-x，那么原先的映射就是 F(x)+x。尽管这两个映射应该都可以近似理论真值映射 the desired functions (as hypothesized)，但是它俩的学习难度是不一样的。

这种改写启发于"网络层数越多，训练和测试误差越大"性能退化问题违反直觉的现象。如果增加的层数可以构建为一个恒等映射(identity mappings)，那么增加层数后的网络训练误差应该不会增加，与没增加之前相比较。性能退化问题暗示多个非线性网络层用于近似identity mappings 可能有困难。使用残差学习改写问题之后，如果identity mappings 是最优的，那么优化问题变得很简单，直接将多层非线性网络参数趋0。

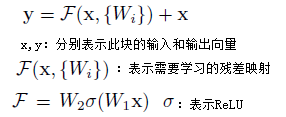
实际中，identity mappings 不太可能是最优的，但是上述改写问题可能帮助预处理问题。如果最优函数接近identity mappings，那么优化将会变得容易些。 实验证明该思路是对的。

F(x)+x 可以通过shortcut connections 来实现，如下图所示：



上图中的shortcut connections执行一个简单的恒等映射；既没有参数，也没有计算复杂度。公式分析如下：

A）需要学习的残差映射



B） x和F的维数必须相同。如果x和F的维数不相同，则对x进行线性投影（linear projection）使用其与F的维数一致，公式如下：、

2018-01-26 17-25-05 的屏幕截图

关于残差网络的详细分析会重新分析

# 三、提高目标检测方法

R-CNN系列目标检测框架和YOLO目标检测框架给了我们进行目标检测的两个基本框架。除此之外，研究人员基于这些框架从其他方面入手提出了一系列提高目标检测性能的方法。

## 1、难分样本挖掘（hard negative mining）

R-CNN在训练SVM分类器时使用了难分样本挖掘的思想，但Fast R-CNN和Faster R-CNN由于使用端到端的训练策略并没有使用难分样本挖掘（只是设置了正负样本的比例并随机抽取）。CVPR2016的Training Region-based Object Detectors with Online Hard Example Mining(oral)将难分样本挖掘(hard example mining)机制嵌入到SGD算法中，使得Fast R-CNN在训练的过程中根据region proposal的损失自动选取合适的region proposal作为正负例训练。实验结果表明使用OHEM（Online Hard Example Mining）机制可以使得Fast R-CNN算法在VOC2007和VOC2012上mAP提高 4%左右。

## 2、多层特征融合

Fast R-CNN和Faster R-CNN都是利用了最后卷积层的特征进行目标检测，而由于高层的卷积层特征已经损失了很多细节信息（pooling操作），所以在定位时不是很精准。HyperNet等一些方法则利用了CNN的多层特征融合进行目标检测，这不仅利用了高层特征的语义信息，还考虑了低层特征的细节纹理信息，使得目标检测定位更精准。

## 3、使用上下文信息

在提取region proposal特征进行目标检测时，结合region proposal上下文信息，检测效果往往会更好一些。（Object detection via a multi-region & semantic segmentation-aware CNN model以及Inside-Outside Net等论文中都使用了上下文信息）

参考文献：

[1] R. Girshick, J. Donahue, T. Darrell, J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. ImageNet Large-Scale Visual Recognition Challenge workshop, ICCV, 2013. R-CNN

[2] R. Girshick, J. Donahue, T. Darrell, J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014. R-CNN

[3] R. Girshick, J. Donahue, T. Darrell, J. Malik. Region-Based Convolutional Networks for Accurate Object Detection and Segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, May. 2015. R-CNN

[18] R-CNN: Region-based Convolutional Neural Networks: <https://github.com/rbgirshick/rcnn> R-CNN

[4] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In ECCV. 2014. SPP-net

[19] SPP-net: <https://github.com/ShaoqingRen/SPP_net> SPP-net

[14] Girshick, R. Fast R-CNN. ICCV 2015. Fast R-CNN

[20] Fast R-CNN: <https://github.com/rbgirshick/fast-rcnn> Fast R-CNN

[5] S. Ren, K. He, R. Girshick, J. Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. Advances in Neural Information Processing Systems 28 (NIPS), 2015. Faster R-CNN

[21] Faster R-CNN: <https://github.com/rbgirshick/py-faster-rcnn> Faster R-CNN

[16] R-FCN: Object Detection via Region-based Fully Convolutional Networks. Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. Conference on Neural Information Processing Systems (NIPS), 2016. R-FCN

[24] R-FCN: <https://github.com/daijifeng001/r-fcn> R-FCN

[15] Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: CVPR. (2016) YOLO

[22] YOLO: <http://pjreddie.com/darknet/yolo/> YOLO

[17] Liu W, Anguelov D, Erhan D, et al. SSD: Single Shot MultiBox Detector[J]. arXiv preprint arXiv:1512.02325, 2015. SSD

[23] SSD: <https://github.com/weiliu89/caffe/tree/ssd> SSD

[6] C. Szegedy, A. Toshev, D. Erhan. Deep Neural Networks for Object Detection. Advances in Neural Information Processing Systems 26 (NIPS), 2013.

[7] P. Sermanet, D. Eigen, X.Zhang, M. Mathieu, R. Fergus, and Y. LeCun. OverFeat: Integrated recognition, localization and detection using convolutional networks. In ICLR, 2014. OverFeat

[9] OverFeat source code:

<http://cilvr.nyu.edu/doku.php?id=software:OverFeat:start> OverFeat

[8] J.R. Uijlings, K.E. vandeSande, T. Gevers, and A.W. Smeulders. Selective search for object recognition. IJCV, 2013. 选择性搜索

[12] J. Hosang, R. Benenson, P. Dolla ́r, and B. Schiele. What makes for effective detection proposals? TPAMI, 2015. 区域提名变种

[13] Felzenszwalb P F, Girshick R B, McAllester D, et al. Object detection with discriminatively trained part-based models[J]. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 2010, 32(9): 1627-1645.

DPM 多尺度形变部件模型

[10] ILSVRC 2013 results:

http://www.image-net.org/challenges/LSVRC/2013/results.php

[11] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural net- works. In NIPS, 2012.

[25] Detection Results: VOC2012 Competition – “comp4” (train on own data): http://host.robots.ox.ac.uk:8080/leaderboard/displaylb.php?challengeid=11&compid=4