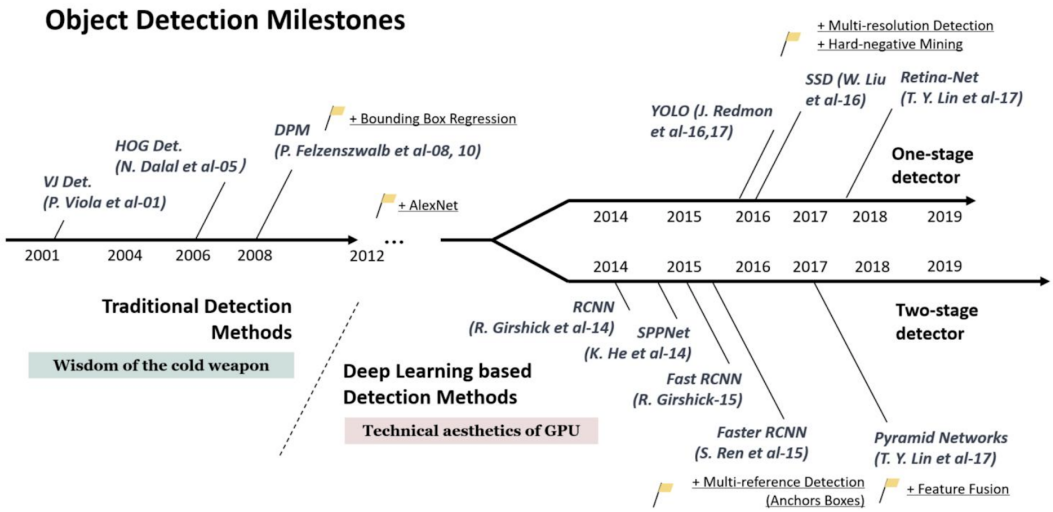
目标检测综述

目标检测(Object Detection)任务是计算机视觉中非常重要的基础问题，也是解决图像分割、目标跟踪、图像描述等问题的基础。目标检测是检测输入图像是否存在给定类别的物体，如果存在，输出物体在图像中的位置信息（矩形框的坐标值表示，Xmin、Ymin、Xmax、Ymax）。
早期，传统目标检测算法还没有使用深度学习，一般分为三个阶段：区域选取、特征提取、特征分类。

- 区域选取：采用滑动窗口(Sliding Windows)算法，选取图像中可能出现物体的位置，这种算法会存在大量冗余框，并且计算复杂度高。
- 特征提取：通过手工设计的特征提取器（如SIFT和HOG等）进行特征提取。
- 特征分类：使用分类器(如SVM)对上一步提取的特征进行分类。

2014年的R-CNN（Regions with CNN features）使用深度学习实现目标检测，从此拉开了深度学习做目标检测的序幕。目标检测大致可以分为一阶段(One Stage)模型和二阶段(Two Stage)模型。目标检测的一阶段模型是指没有独立地提取候选区域(Region Proposal)，直接输入图像得到图中存在的物体类别和相应的位置信息。典型的一阶段模型有SSD(Single Shot multibox-Detector)、YOLO(You Only Look Once)系列模型等。二阶段模型是有独立地候选区域选取，要先对输入图像筛选出可能存在物体的候选区域，然后判断候选区域中是否存在目标，如果存在输出目标类别和位置信息。经典的二阶段模型有R-CNN、SPPNet、Fast R-CNN、Faster R-CNN
下图总结了目标检测中一些经典模型的发展历程：



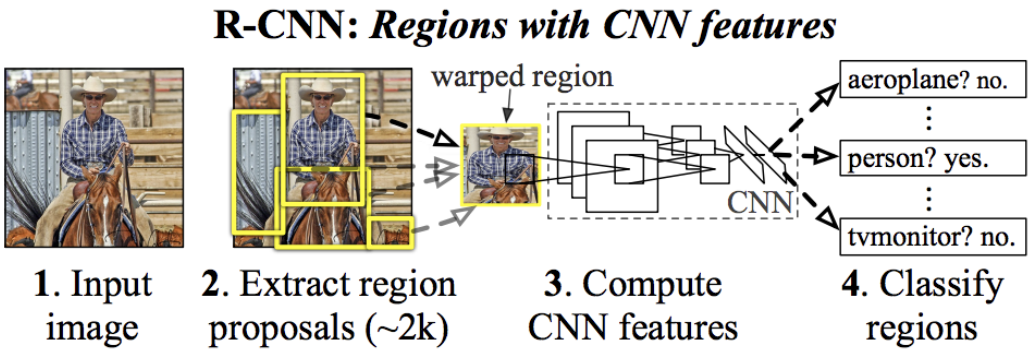
一般来说，一阶段模型在计算效率上有优势，两阶段在检测精度上有优势。对于一阶段和二阶段模型在速度上和精度上的差异，一般有以下原因：

1. 多数一阶段模型是利用预设的锚框（Anchor Box）来捕捉图像可能存在物体的区域，图像中包含物体的框远少于总共的锚框，因而在训练分类器时正负样本数目极不平衡，这会导致分类器训练的效果不好。
2. 二阶段模型在会修正候选框的位置，带来更高的定位精度，同时也增加了模型复杂度。

接下来，简单介绍二阶段模型的发展过程。

R-CNN

首先使用无监督的选择性搜索(Selective Search, SS)方法将输入图像中颜色、纹理相近的区域合并，产生2000个候选区域；
然后截取这些候选区域相应的图像，裁剪缩放至固定的尺寸，依次送入CNN特征提取网络提取特征；
特征送入每一类的SVM分类器，判断是否属于此类；
使用线性分类器修正框位置和大小，最后对检测结果进行非极大值抑制（Non-Maximum Suppression,NMS）。



SPPNet

在RCNN中，要对候选区域裁剪缩放至固定的尺寸，会破坏截取图像的长宽比，损失一些信息。针对以上问题，SPPNet提出了空间金字塔池化(Spatial Pyramid Pooling)层，该层置于CNN的末端，输入不需要缩放至指定的大小。下图第一行是R-CNN，第二行是SPPNet，对比可以发现它们的区别。

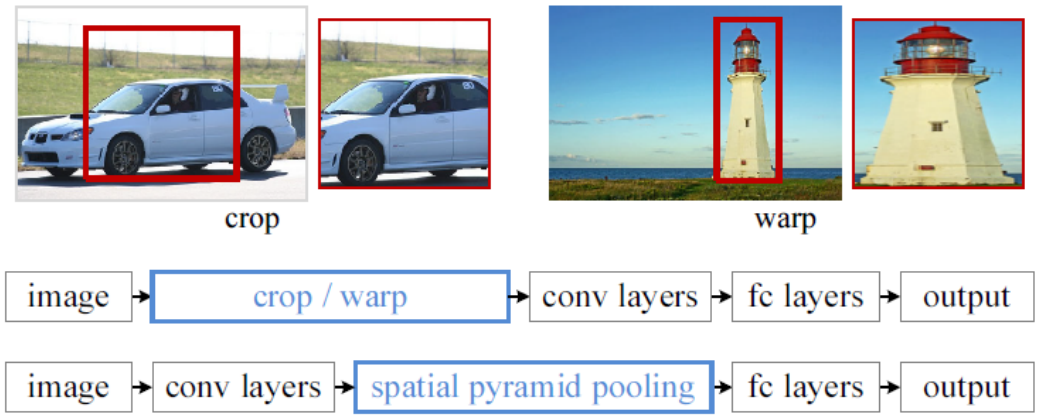


Figure 1: Top: cropping or warping to fit a fixed size. Middle: a conventional CNN. Bottom: our spatial pyramid pooling network structure.

SPPNet的思路是对于任意大小的feature map首先分成16、4、1个块，然后在每个块上最大池化，池化后的特征拼接得到一个固定维度的输出。

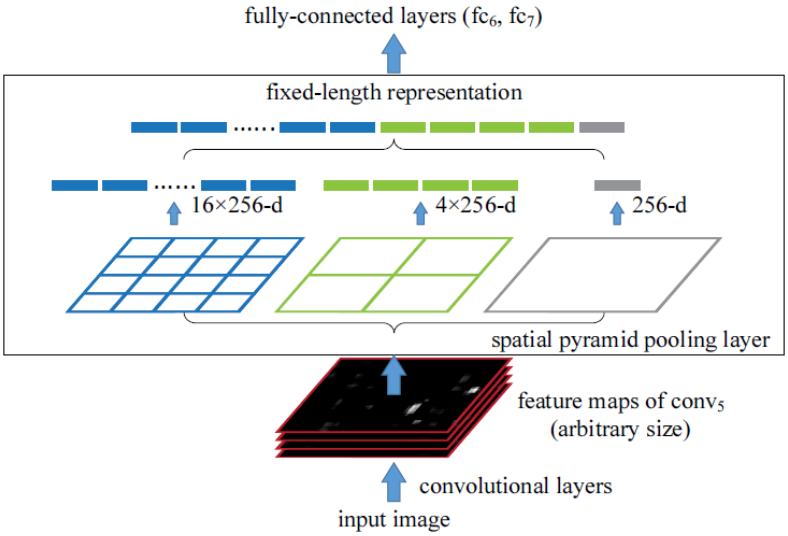
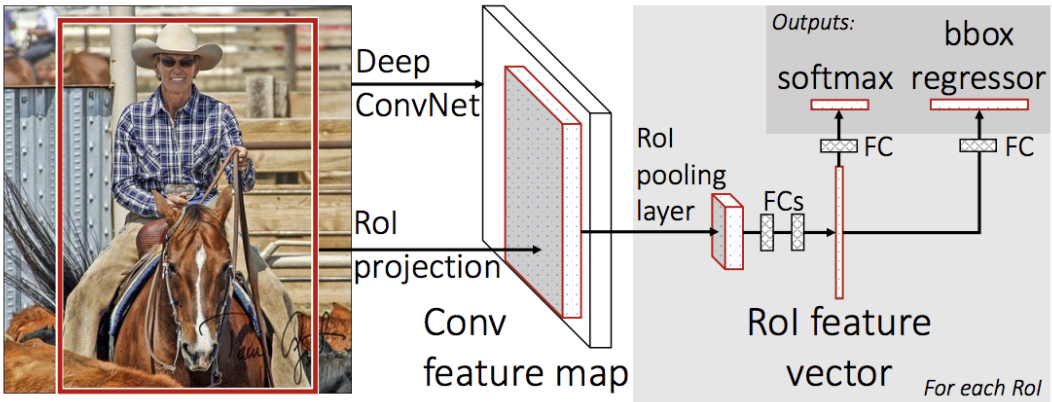


Figure 3: A network structure with a **spatial pyramid pooling layer**. Here 256 is the filter number of the $conv_5$ layer, and $conv_5$ is the last convolutional layer.

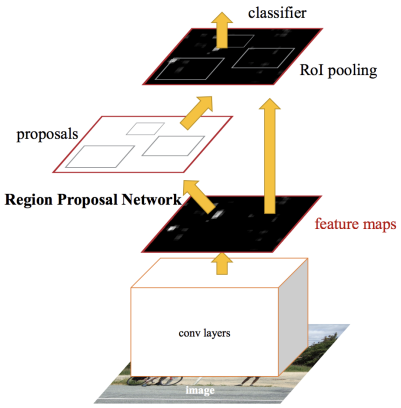
Fast R-CNN

Fast R-CNN的思路与SPPNet一致，区别在于Fast R-CNN使用感兴趣区域池化(Region-of-Interest Pooling)而非空间金字塔池化。Fast R-CNN相比R-CNN使用全连接网络代替之前的SVM分类器和线性回归器进行物体分类和检测框的修正。Fast R-CNN有两个输出，一个是通过softmax层进行类别预测，另一个输出物体的检测框。



Faster R-CNN

Faster R-CNN 在 Fast R-CNN的基础上，将其最耗时的候选区域提取用一个区域候选网络（Region Proposal Network，RPN）进行替代。在faster R-CNN中，一幅输入图像先由RPN提取候选区域，再取出各个候选区域对应的特征图，送入Fast R-CNN(独立于RPN的后半部分)进行物体分类和位置回归。



接下来，简单介绍一阶段模型的发展过程。

R-CNN系列将目标检测问题归结为分类问题，即先寻找目标可能存在的区域（Bounding box），然后对这些Box分类，从而确定目标。Yolo则将目标检测问题转换为一个回归问题（Regreesion problem），直接预测出boudning box和相关的类别信息。Yolo是一个可以端到端训练的单个网络（single network），它不需要单独的搜索Region Proposals，也不需要单独的Classifier，因此其检测速度特别快，Yolo可以达到45FPS，而Fast Yolo可以达到155FPS。Yolo对背景的识别效果较好，且有一定的迁移性，但是Yolo最大的问题是对小目标的检测不准确。

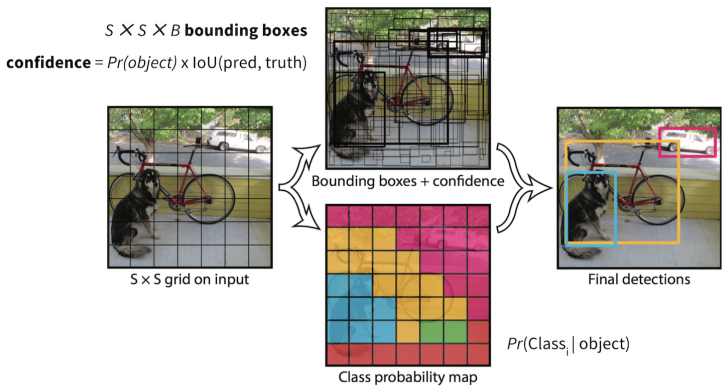
YOLO v1

1、将输入图像($m \times n$)划分为 $S \times S$ 个网格，如果一个物体的中心落在某个网格，则这个网格负责这个物体的检测。

2、每个网格预测 B 个Bounding Box的位置，这个Box的置信度得分，以及Box中是否存物体的概率。

- Bounding Box包含五个参数（中心x坐标，中心y坐标，宽度，高度，置信度）
- **置信度得分**表示该网格包含一个对象的可能性：`Pr(containing an object) x IoU(pred, truth)`；其中`Pr`=概率。
- 如果网格包含一个对象，则它将预测该对象属于每个类别的**概率**

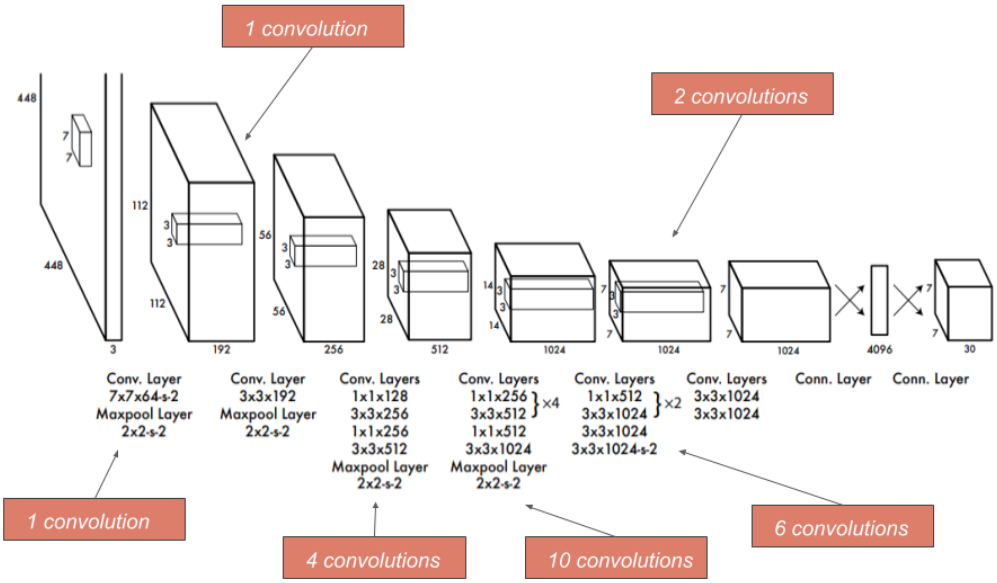
3、将输入图像划分为 $S \times S$ 个网格，每个网格预测B 个Bounding Box以及置信度，则最终的预测编码为 $S \times S \times （B \times 5 + C）$ 。



YOLO v2

YOLO v2在YOLO v1的基础上做出了改进，大体可以分为网络结构的改善、先验框的设计及训练技巧。

1、网络结构的改善，提出了一个全新的网络结构，称之为DarkNet。



- BN层：在卷积层后面添加了批归一化(BN)层。
- 用连续3×3卷积替代了v1版本中的7×7卷积，这样既减少了计算量，又增加了网络深度。此外，DarkNet去掉了全连接层与Dropout层。
- Passthrough层：DarkNet还进行了深浅层特征的融合。

2、**先验框的设计**，YOLO v2首先使用了聚类的算法来确定先验框的尺度。

3、**训练技巧**，YOLO v2采取了多种尺度的图片作为训练的输入。模型在训练过程中，每隔10个批次就改变输入图片的大小。

YOLO v3

YOLO v3在YOLO v2的基础上做出了一些改动。

1、YOLO v3 是使用了Logistic函数代 Softmax函数。原因在于，Softmax函数输出的多个类别预测之间会相互抑制，只能预测出一个类别，而Logistic分类器相互独立，可以实现多类别的预测。

2、YOLO v3 采用了更深的网络作为特征提取器（DarkNet-53），包含53个卷积层。为了避免深层网络带来的梯度消失问题，DarkNet-53借鉴了ResNet的残差思想，在基础网络中大量使用了残差连接。

本篇涉及的论文

Selective Search for Object Recognition

R-CNN

Fast R-CNN

Faster R-CNN

YOLO v1-You Only Look Once: Unified, Real-Time Object Detection

YOLO v2-YOLO9000: Better, Faster, Stronger

YOLO v3: An Incremental Improvement

参考文献：

1、深度学习之PyTorch物体检测实战，董洪义

2、百面深度学习，葫芦娃

3、目标检测（3）-SPPNet

4、YOLO系列模型总结

5、Object Detection Part 4: Fast Detection Models

6、RCNN、Fast RCNN、Faster RCNN对比