目录

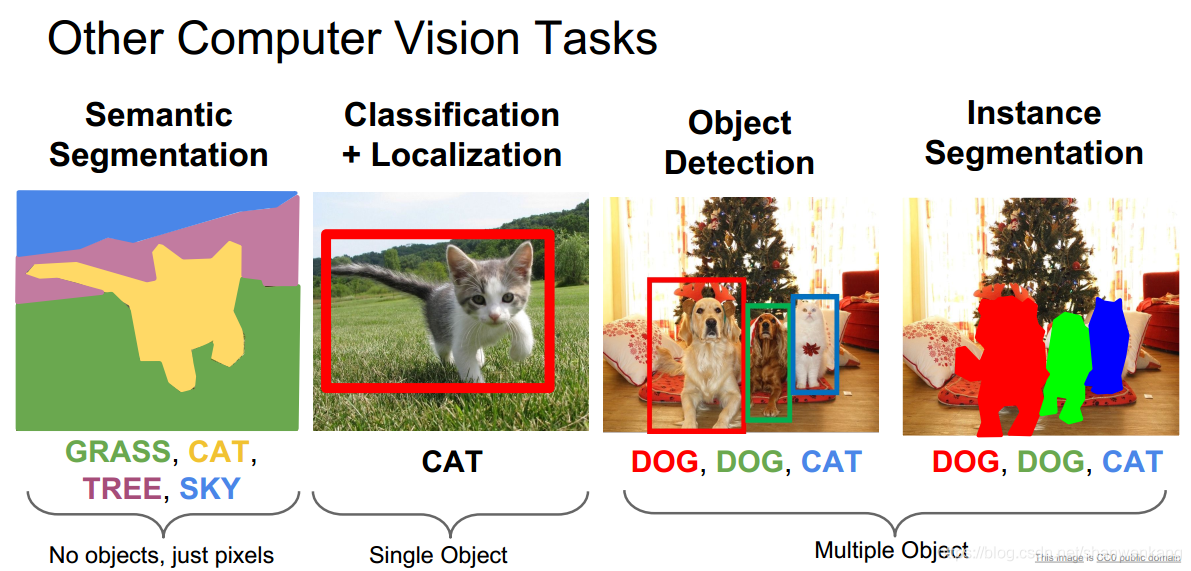
[Semantic Segmentation 3](#_Toc66644238)

[Classification+Localization 8](#_Toc66644239)

[Object Detection 9](#_Toc66644240)

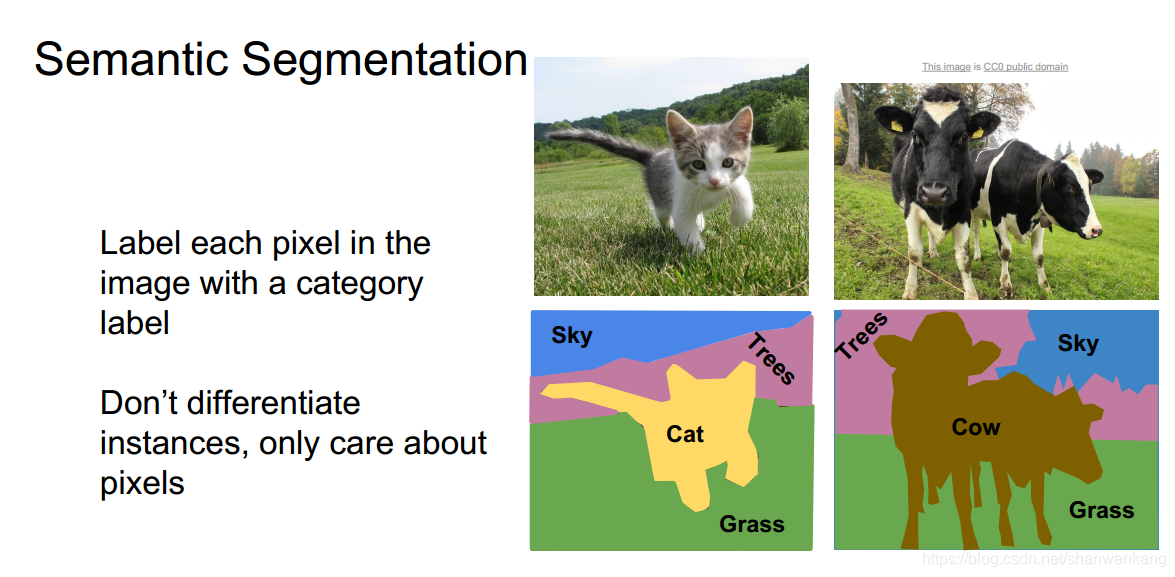
[Instance Segmentation 13](#_Toc66644241)

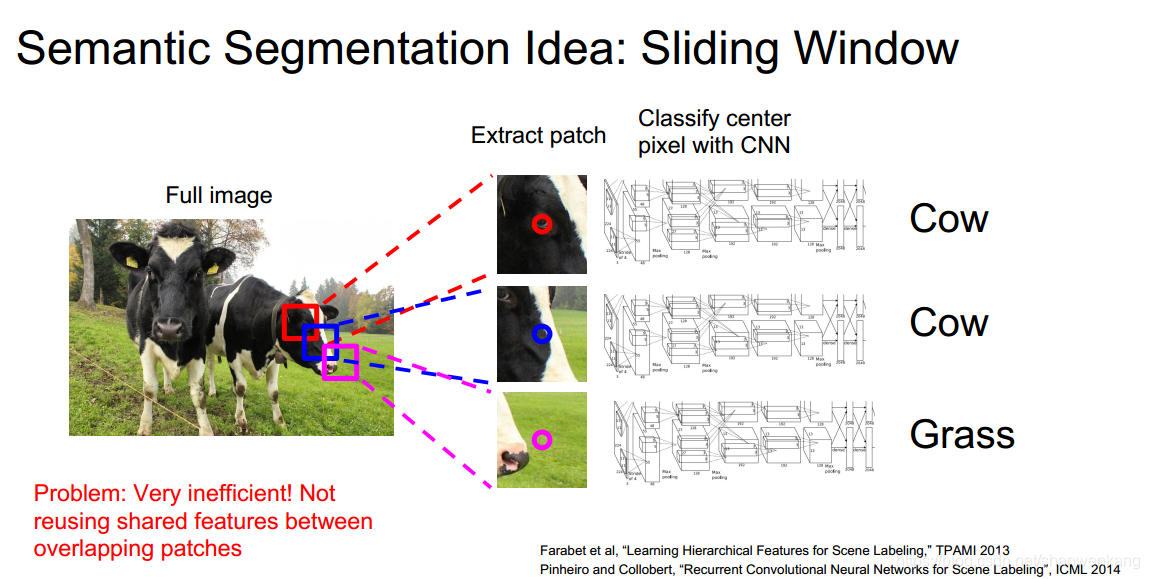
之前讲的都是图像分类的问题，在这一讲我们要介绍一些其他的计算机视觉中的问题，例如图像检测与图像分割。我们主要介绍一下四个部分：图像分割、分类+定位、多目标检测、多目标实例分割



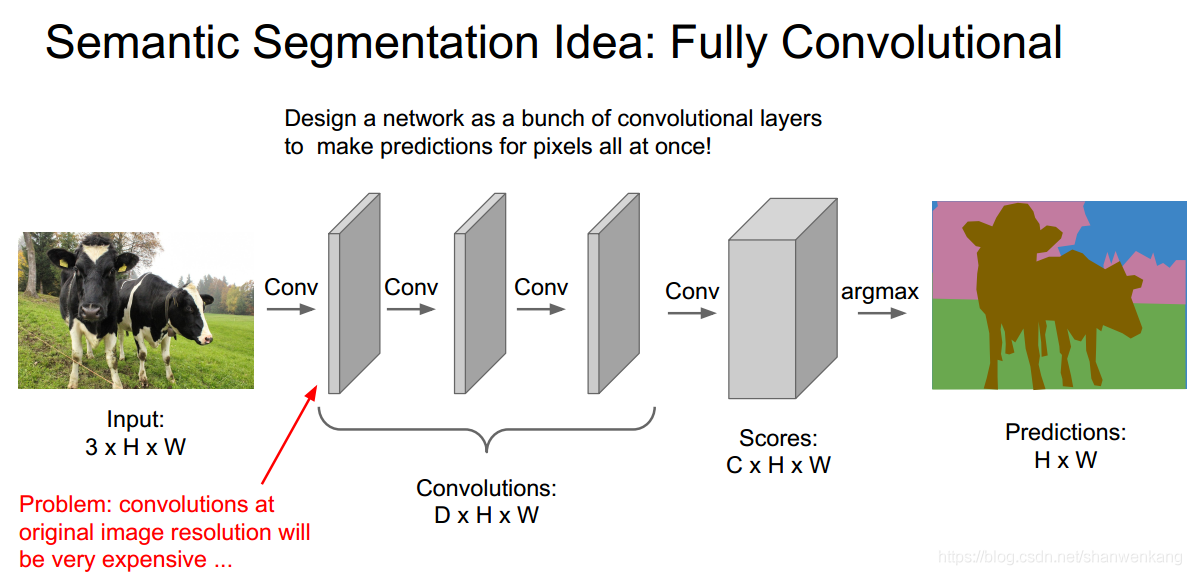
# Semantic Segmentation

第一部分是语义分割。我们需要给每个像素都输出一个值，来代表这个像素属于某一个分类

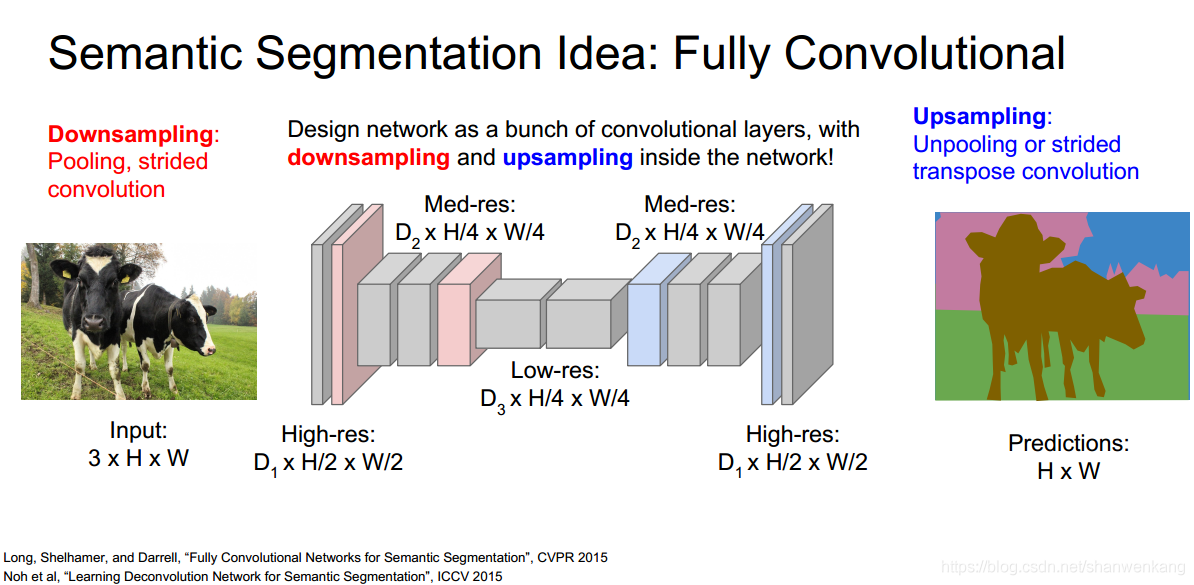


一种传统的方法就是用滑动窗口不断在输入图像上滑动， 然后输出窗口中央的像素属于哪一类。这种方法有很大的冗余性，比如两张非常接近的patch，我们没有利用到他们的相关信息，这样导致的结果就是计算效率十分低

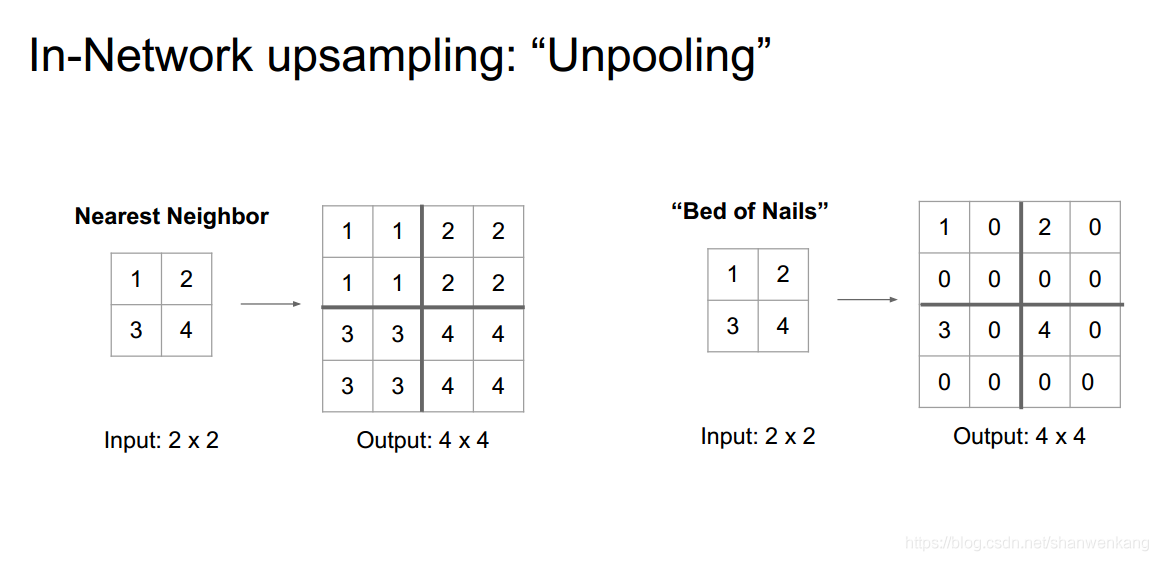
于是我们可以采用CNN的方法，输入一张图片，直接输出这张图片每个像素在每个分类上的打分，然后取分数最大的类作为当前像素的分类，但是这样对于高分辨率的图片计算量也是非常大的



解决方法就是在CNN中先下采样来减小特征维度，再上采样还原图像



我们之前讲过下采样，下面就介绍几种常用的上采用方法，一种就是反池化



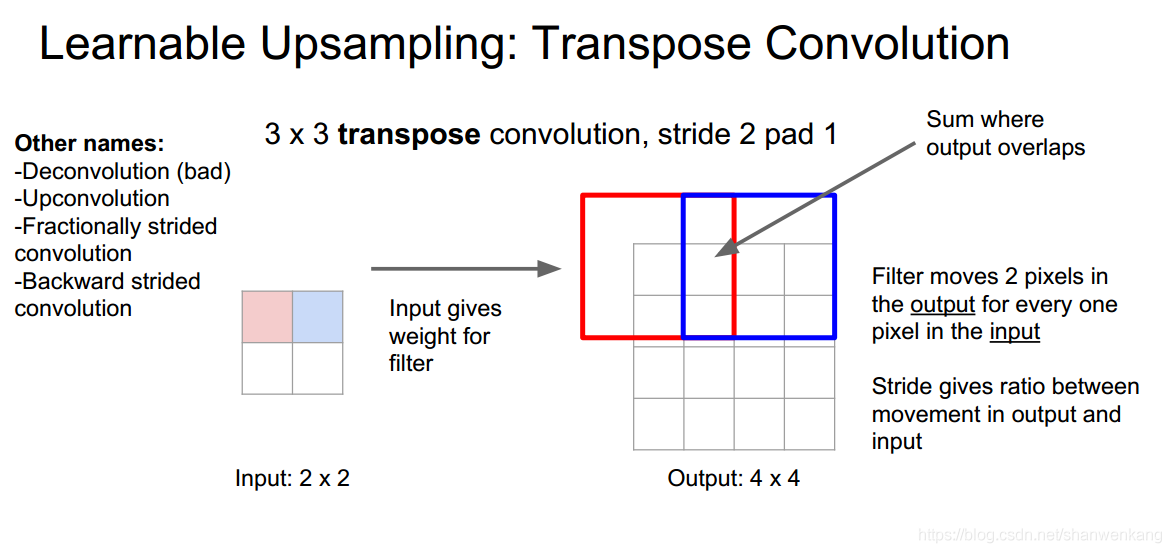
以及保留了池化位置信息的最大反池化



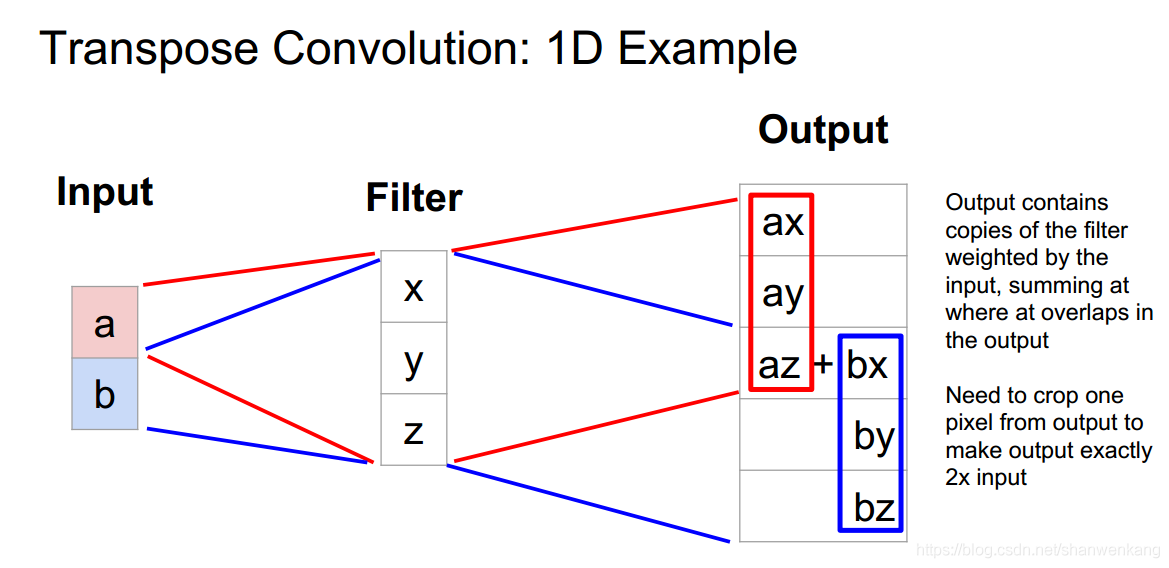
另外一种方法就是转置卷积transpose convolution，我们讲过普通卷积会减小输入的维度



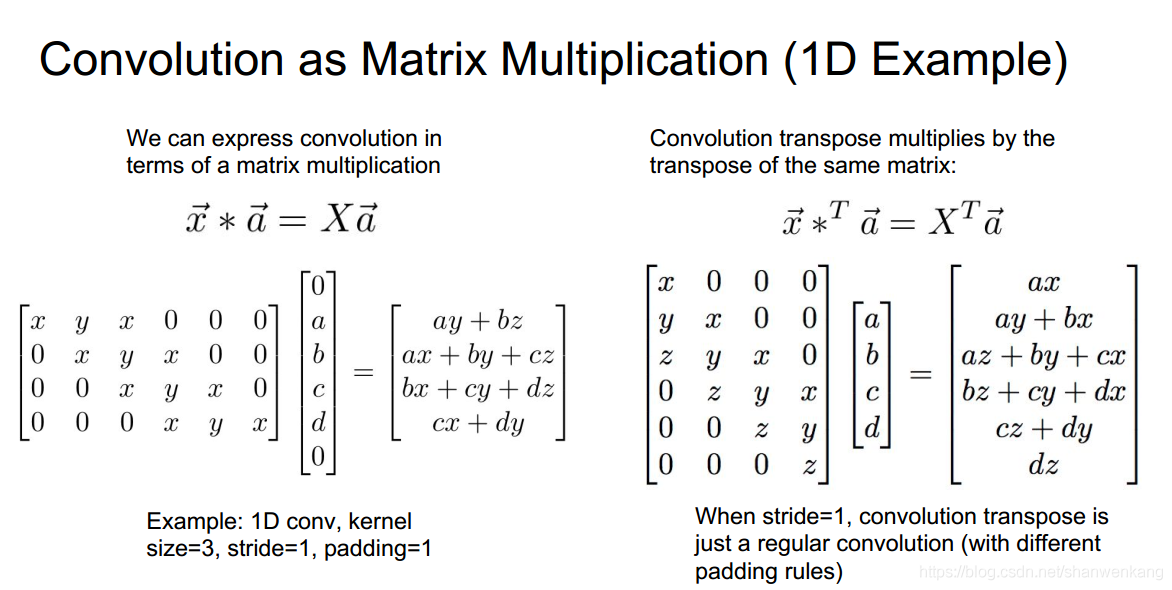
反卷积相反会增加输入的维度，我们在反卷积做的是把输入的值作为权重乘以卷积核，然后在输出窗口上滑动



以下是一个一维的例子以便于我们更好地理解



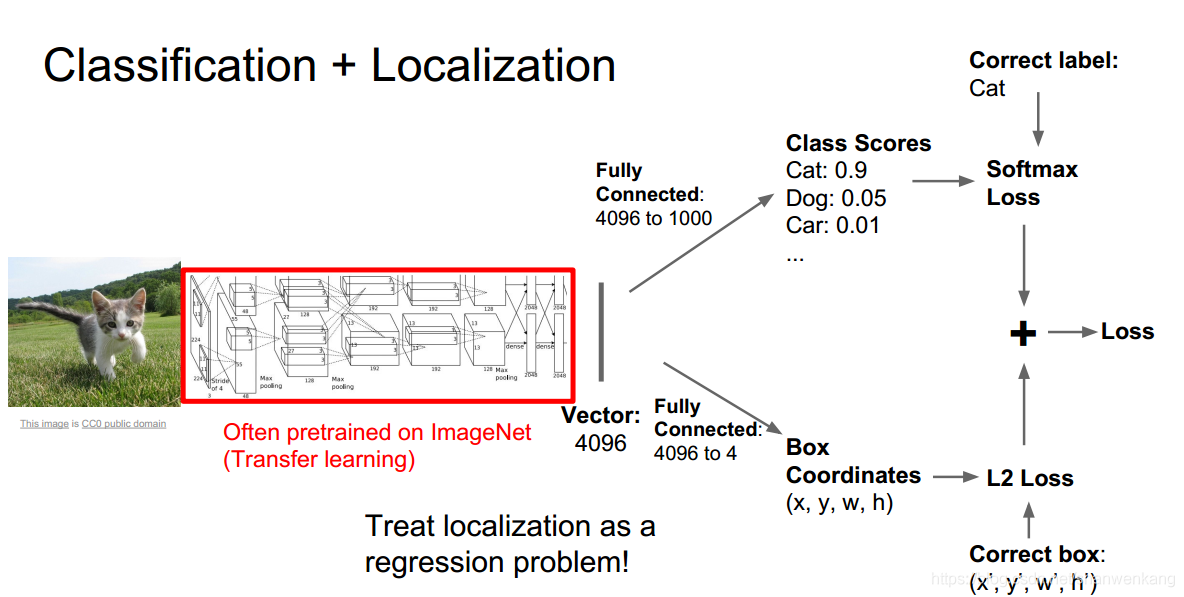
卷积操作可以写成矩阵相乘的形式，而反卷积这个名字的由来就是把其中一个矩阵做转置运算



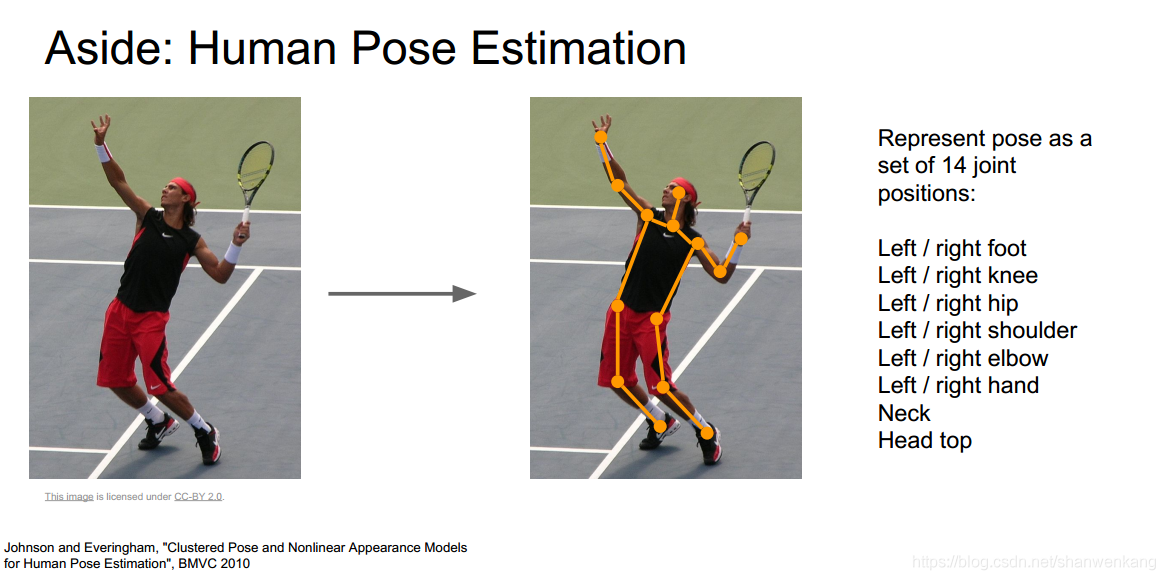


# Classification+Localization

下一个问题是分类加定位，对于这个问题我们可以采样传统的CNN结构，不同的是我们有两个输出，一个是分类的值，一个是bounding box的坐标，然后将两个损失函数加权作为最终的损失函数

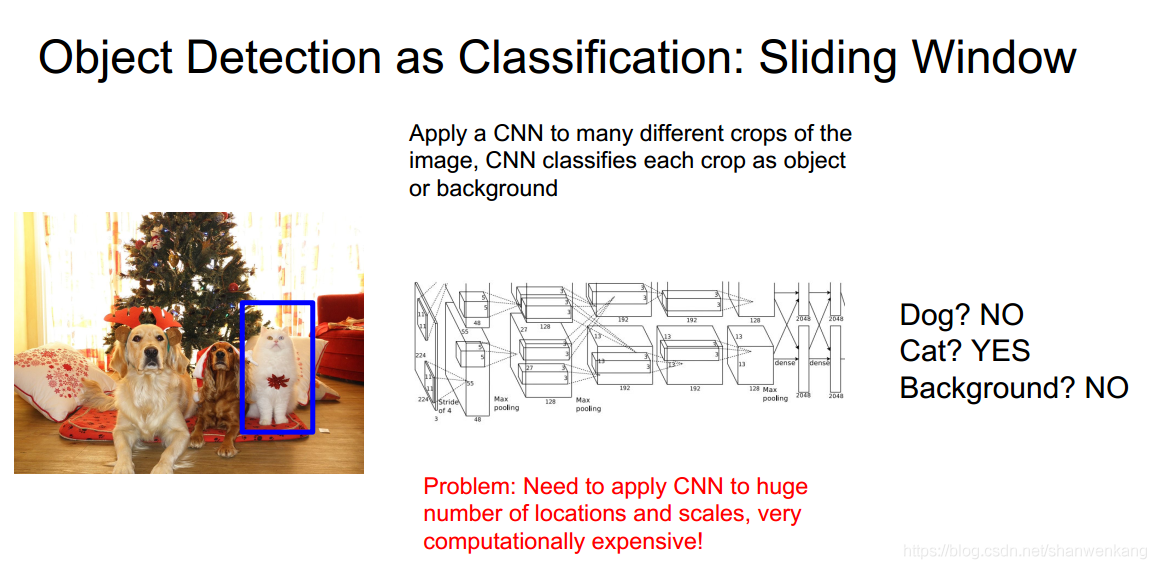


另外这种方法还可以用作人体姿势估计，唯一不同的地方就是最后输出的是每个关节的坐标



# Object Detection

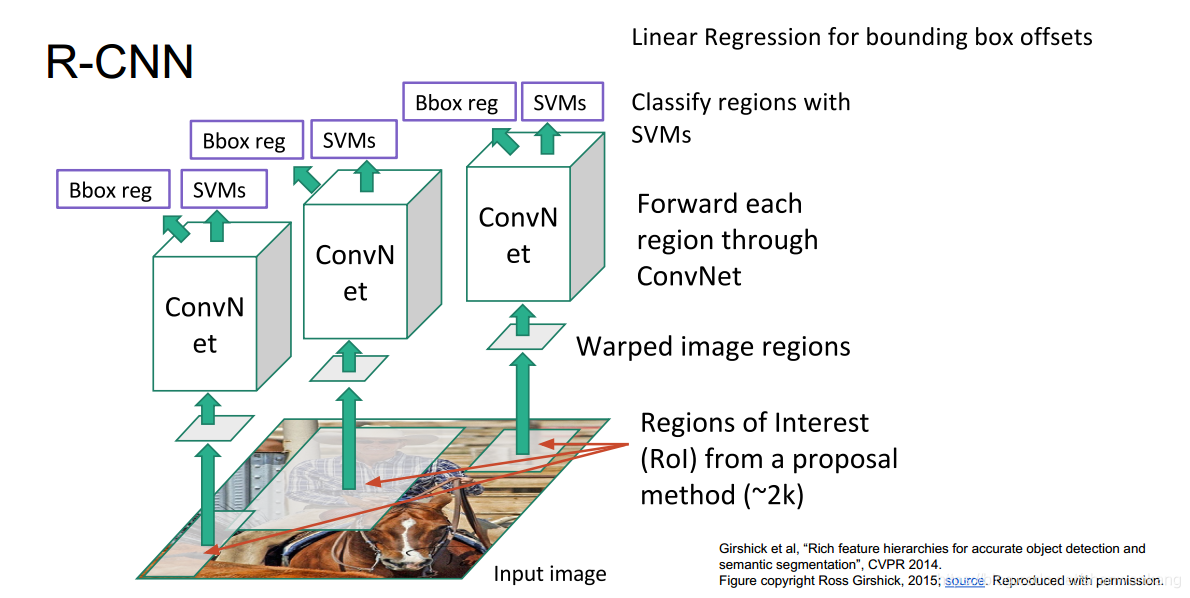
下一个任务是多目标检测，多目标检测中物体的个数不确定，因此我们最容易想到的方法是滑动窗口：窗口在输入图像上滑动，来判断每个窗口中是否存在物体，以及物体的分类。但是由于物体大小不同，我们需要不同位置、不同长宽比、不同大小的窗口，这样的计算量无疑是难以接受的



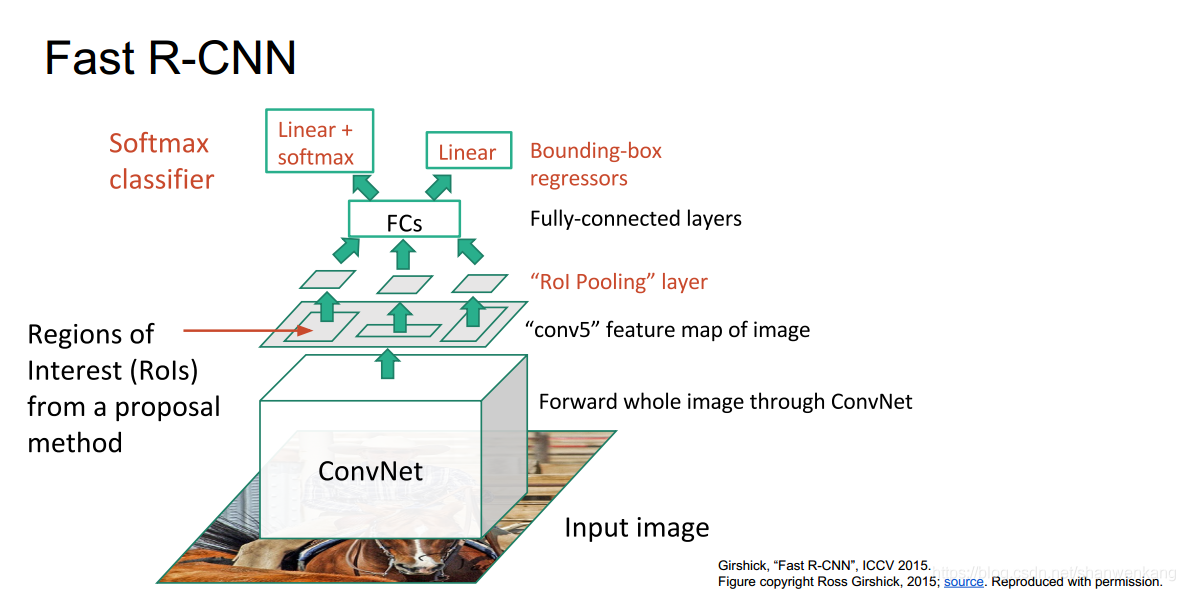
那我们应该用什么方法呢？在计算机视觉中，有一个传统的问题叫候选区域，意思是输入一张图，输出给你1000个区域，尽管这1000个区域中不一定都含有物体，但是图片中的物体大部分都会在这1000个区域中的某一个区域中。这种方法没用到神经网路，但是相对来说耗时非常短

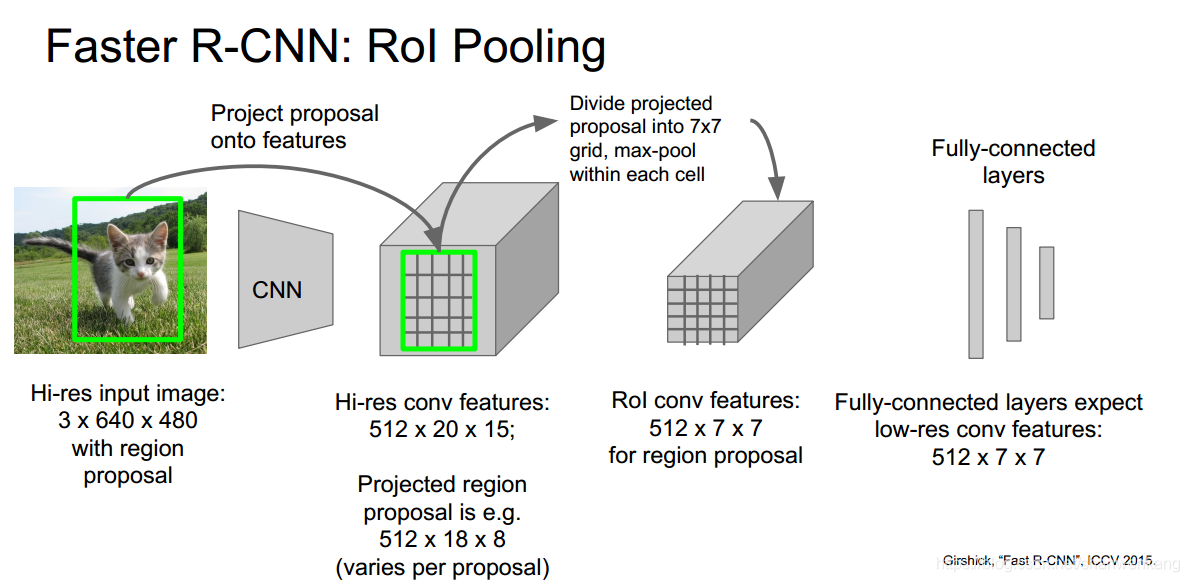


有了这个方法以后，我们就可以提出更快的识别算法了。先把这1000个区域变为同样的大小，然后输入到CNN中，CNN中有两个输出，一个是候选区域的分类（这其中加入了一个背景类，代表什么物体都没有），另外一个个bounding box坐标的矫正参数，来使bounding box更好地选中物体

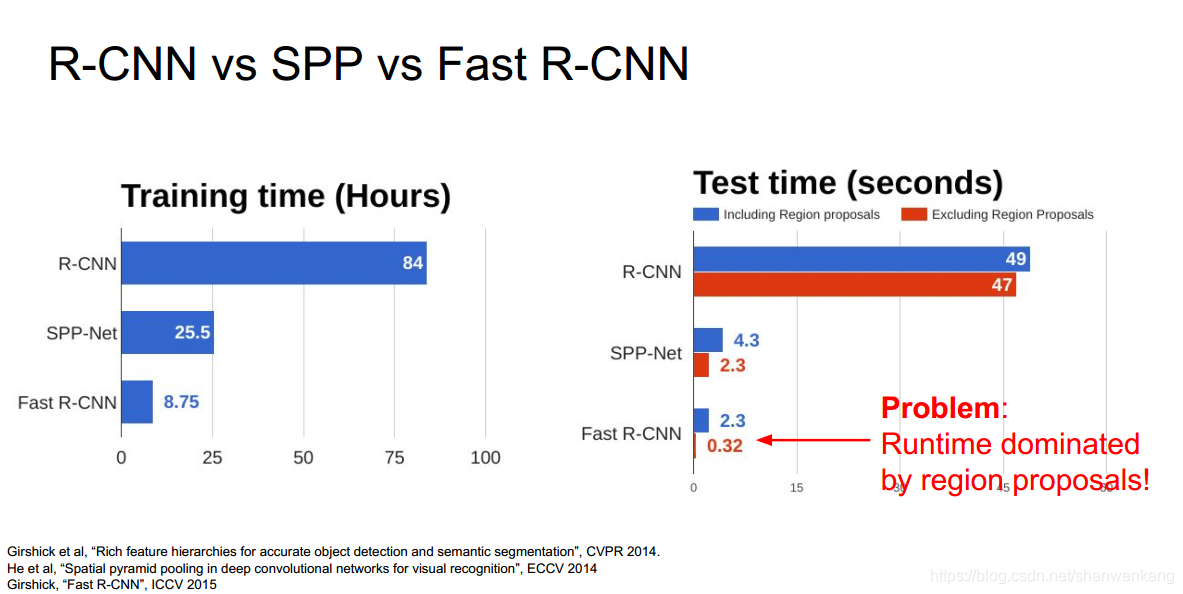


这个网络结构的缺点仍然是计算量大, 进一步改进方法是Fast R-CNN，它不是从原图中提取候选区域，而是先让图片经过卷积神经网络，然后在特征图上选取感兴趣区域（ROI），然后通过全连接层，得到两个输出，其中ROI池化层可以统一感兴趣区域的大小

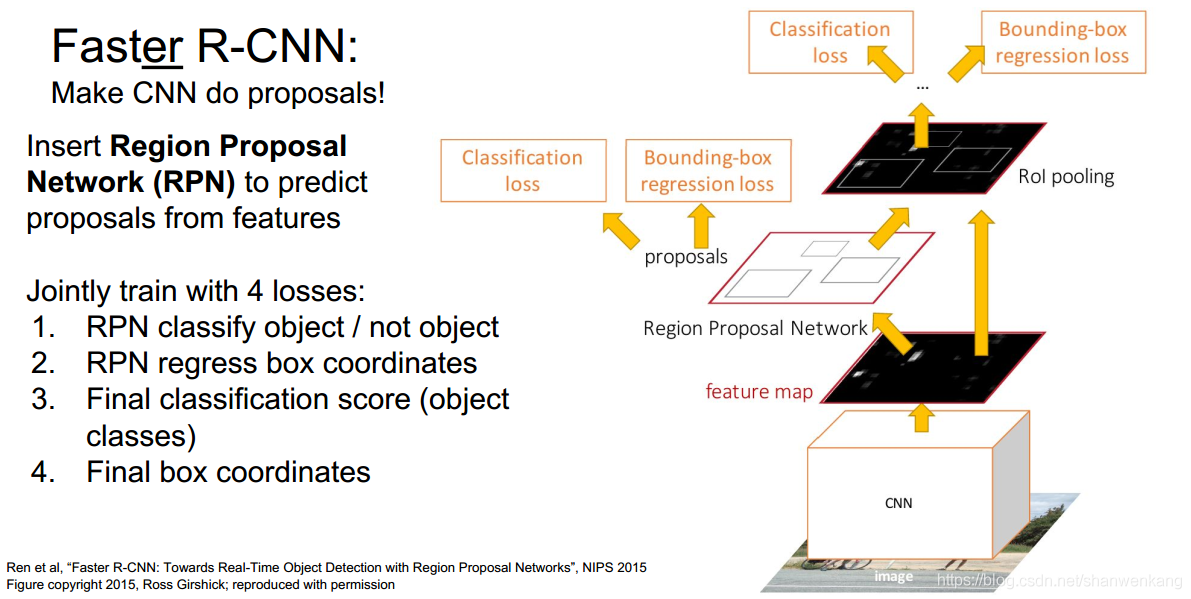




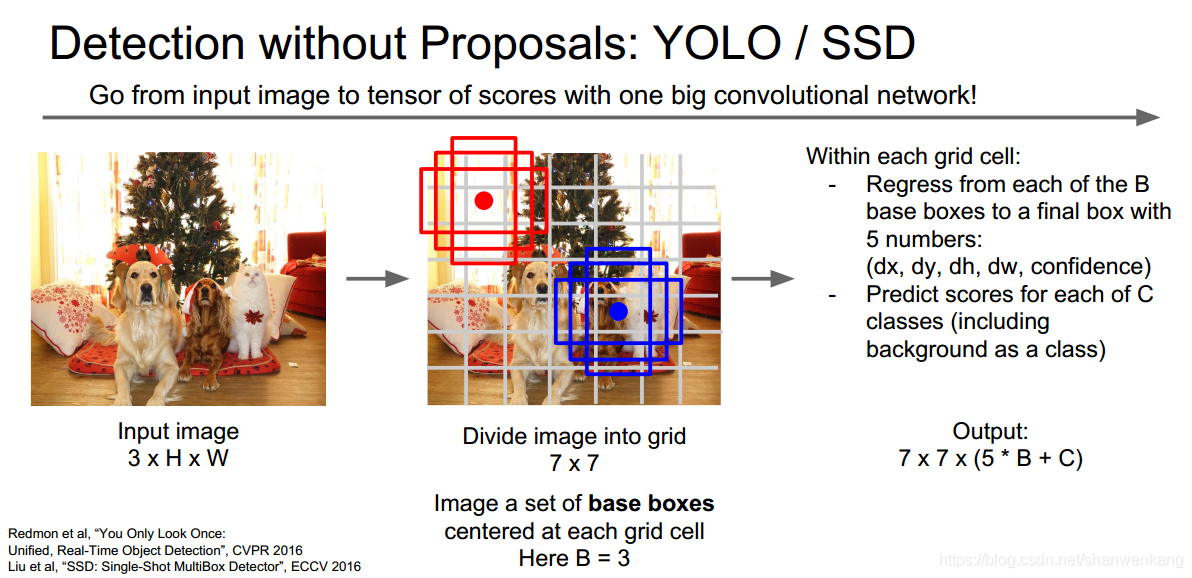
以下是几种方法消耗时间的对比，我们可以看到对于Fast R-CNN，时间大部分都花在了传统的候选区域算法上面



进一步改进方法是我们将候选区域选择也通过神经网络实现。于是我们的损失函数变为了以下四部分：RPN提出的候选区域是否含有物体、RPN选出的bounding box的坐标、最后分类器的分数以及最终经过修正以后的bounding box坐标

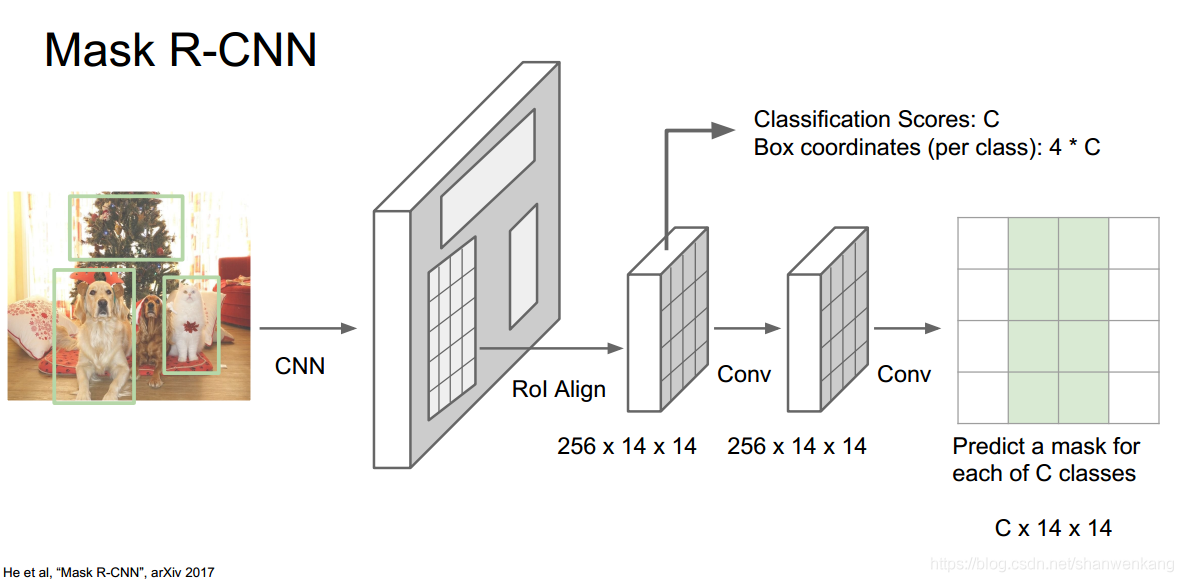


以上提到的方法都是基于候选区域的，另外的不基于候选区域的方法例如YOLO主要的方法就是将图像划分为栅格，对于每个栅格，选取一定数量的base box，然后通过修正base box的位置以及大小来更好地选中无图



# Instance Segmentation

最后一部分是多目标实例分割，这部分与多目标检测十分类似，我们要做的就是计算多目标检测中提出的候选区域中每个像素的分类结果即可



这种方法看起来简单却拥有非常好的效果



Mask R-CNN如果输出关节坐标也能做人体姿势估计

