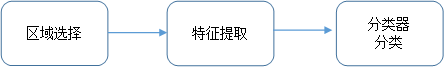
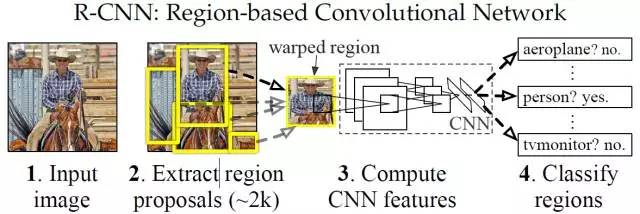
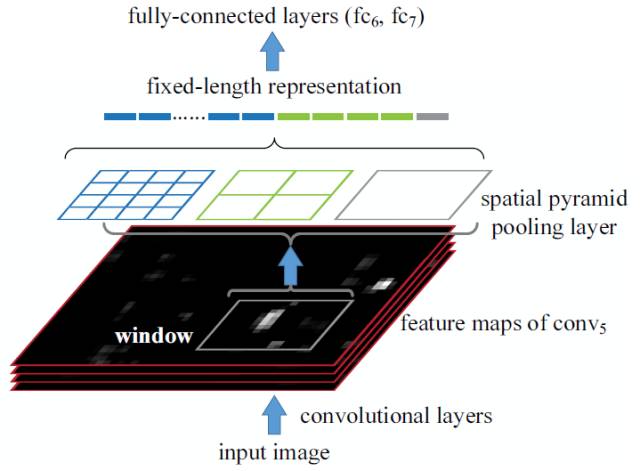
https://mp.weixin.qq.com/s?\_\_biz=MzI1NTE4NTUwOQ==&mid=2650324780&idx=1&sn=1579155aacc83e991a9a39b142a7a0b8&mpshare=1&scene=1&srcid=1027QpLsB0FNKr3gZanOXScB&pass\_ticket=Jmgqm%2BkG8d8DMuS6YDGj06ZSSloWA72GO7jJ4yo7WKhTUP2IRmsxp6GqthL3o0zF#rd

什么物体？他们在什么位置？这还不简单，图中有一个猫和一个人，具体的位置就是上图右侧图像两个边框(bounding-box)所在的位置。其实刚刚的这个过程就是目标检测，目标检测就是“给定一张图像或者视频帧，找出其中所有目标的位置，并给出每个目标的具体类别”。  
 (1) 区域选择   (2) 特征提取    (3) 分类器总结：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。  
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。     (1) 输入测试图像    (2)  利用selective search算法在图像中提取2000个左右的region proposal。    (3) 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。    (4) 将每个region proposal提取到的CNN特征输入到SVM进行分类。    \* 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模型对训练集提特征训练SVM。   \* 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。    \* 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。 小结：R-CNN在PASCAL VOC2007上的检测结果从DPM HSC的34.3%直接提升到了66%(mAP)。如此大的提升使我们看到了region proposal+CNN的巨大优势。    但是R-CNN框架也存在着很多问题:  (1) 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  (2) 训练耗时，占用磁盘空间大：5000张图像产生几百G的特征文件  (3) 速度慢: 使用GPU, VGG16模型处理一张图像需要47s。  针对速度慢的这个问题，SPP-NET给出了很好的解决方案。  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：    (1) 训练分为多个阶段，步骤繁琐: 微调网络+训练SVM+训练训练边框回归器    (2) 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。 (1) 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。 (2) 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)。  (3) 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。  (1) 使用在ImageNet上预训练的模型初始化网络参数，微调RPN网络；  (2) 使用(1)中RPN网络提取region proposal训练Fast R-CNN网络；  (3) 使用(2)的Fast R-CNN网络重新初始化RPN, 固定卷积层进行微调；(4) 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。    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。  
 (1) 给个一个输入图像，首先将图像划分成7\*7的网格 (2) 对于每个网格，我们都预测2个边框（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） (3) 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。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。     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。  
 (1) 难分样本挖掘（hard negative mining） (2) 多层特征融合   (3) 使用上下文信息