此篇文档参考：

<https://github.com/kratzert/finetune_alexnet_with_tensorflow>

<https://kratzert.github.io/2017/02/24/finetuning-alexnet-with-tensorflow.html>

<https://blog.csdn.net/shenxiaolu1984/article/details/51066975>

<https://blog.csdn.net/hjimce/article/details/50187029>

<https://www.jianshu.com/p/11df675c9699>

**分类**(classify)：

把一张图像正确分类到很多种类去。

**定位**(classify)：

找到一张图片中的一个单一物体的位置。

**目标检测**(object detection)：

目标检测就是定位和分类一张图像中的多个物体就是把定位和分类任务结合起来，我们就需要同时进行一张图片中的多个目标的检测和分类。

目标检测和图片分类的区别：图片分类不需要定位，而物体检测需要定位出物体的位置，也就是相当于把物体的bounding box检测出来，还有一点物体检测是要把所有图片中的物体都识别定位出来。

**传统方法**

第一个是2001年Paul Viola和Micahel Jones在论文“鲁棒实时目标检测”提出的Viola-Jones 框架。这个方法速度快，相对简单，因此傻瓜相机的实时脸部检测就是使用的这个算法，它的运算量很小。

另外一个与之相似的传统方法是梯度方向直方图（HOG）特征和支持向量机（SVM）来分类。它仍然依赖多尺度滑动窗口，虽然它优于Viola-Jones方法，但它仍旧比较慢。

**深度学习方法**

**Overfeat**

第一个使用深度学习做目标检测并取得很大进展的方法是纽约大学在2013年提出的OverFeat。他们提出了一个使用卷积神经网络(CNN)的多尺度滑动窗口算法。

**R-CNN**

在OverFeat提出不久，伯克利大学的Ross Girshick等人于2014年CVPR上的经典论文《Rich feature hierarchies for Accurate Object Detection and Segmentation》中提出了RCNN(Regions with Convolutional Neural Network Features，基于区域的卷积特征)，它在目标检测比赛上相比其他方法取得了50%的性能提升。他们提出了一个三阶的方法：

* 使用区域提议方法（最流行的方法是选择性搜索Selective Search）提取可能的目标；
* 使用CNN在区域上提取特征；
* 使用支持向量机（SVM）对区域进行分类

之后的各种关于深度学习的物体检测的算法，基本上都是继承了这个思想。在三年时间中，目标识别的方法经历了RCNN->Fast-RCNN->Faster-RCNN->Mask-RCNN->SSD->YOLO的演化。(如果想要详细了解这些方法，可以研读以下论文)

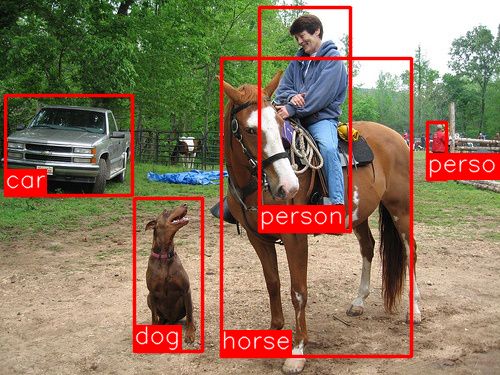
* R-CNN：<https://arxiv.org/abs/1311.2524>
* Fast R-CNN：<https://arxiv.org/abs/1504.08083>
* Faster R-CNN：<https://arxiv.org/abs/1506.01497>
* Mask R-CNN：<https://arxiv.org/abs/1703.06870>
* SSD： <https://arxiv.org/abs/1512.02325>
* YOLO： <https://arxiv.org/abs/1506.02640>
* YOLOv2： <https://arxiv.org/abs/1612.08242>

在这篇文档中，重点介绍的是这一系列方法的起始RCNN，因为这个思想直接基于CNN，相对而言，较容易理解。虽然RCNN取得了了不起的成绩，但是训练过程中有很多的问题。训练它，你首先需要生成训练集的提议区域，然后在每个区域使用CNN特征提取器来提取特征(在VOC Pascal2012的训练集上，需要生成大于200GB的中间文件)，然后训练SVM分类器。R-CNN计算量大，效率低的缺点，从而催生出了其他的改进算法。当然，RCNN也是理解其他算法的基础。

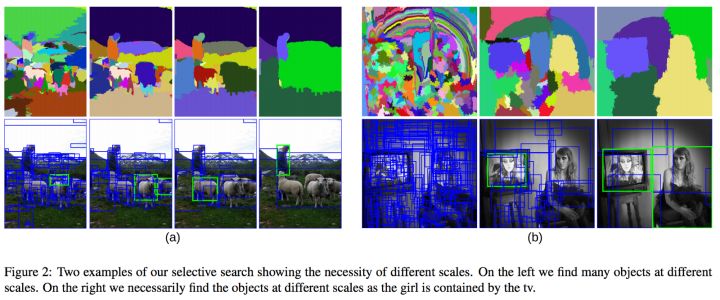
**理解R-CNN**

R-CNN的目标是分析图像，并正确识别图像中主要对象，通过边界框标出对象的具体位置。

* **输入：**图像
* **输出：**图像中每个对象的边界框和标签

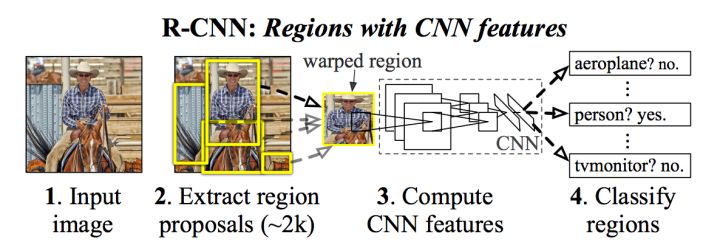


但是我们如何确定这些边界框的大小和位置呢？R-CNN网络通过以下操作完成了一个我们直观上可以完成的任务：在图像中提出了多个边框，并判断这其中的任何一个是否对应着一个具体对象。



以上例子展示了使用不同大小的选择边框的必要性。R-CNN网络使用了选择性搜索的方法，来创建这些边界框或区域建议(region proposal)。在图6中，选择性搜索通过不同大小的边框分析图像，并且对于每个图像块，尝试通过纹理、颜色或强度将相邻像素组合在一起，以识别对象。

下图描述了**RCNN的工作流程**：



* 输入图像
* 在图像中提取约2000个region proposals(可以理解为提取2000个可能包含需要检测物体的区域)
* 利用已经训练好的CNN网络对于每一个提取出的region得到feature。
* 根据提取出来的feature对图中的2000个regions运用svm进行分类。

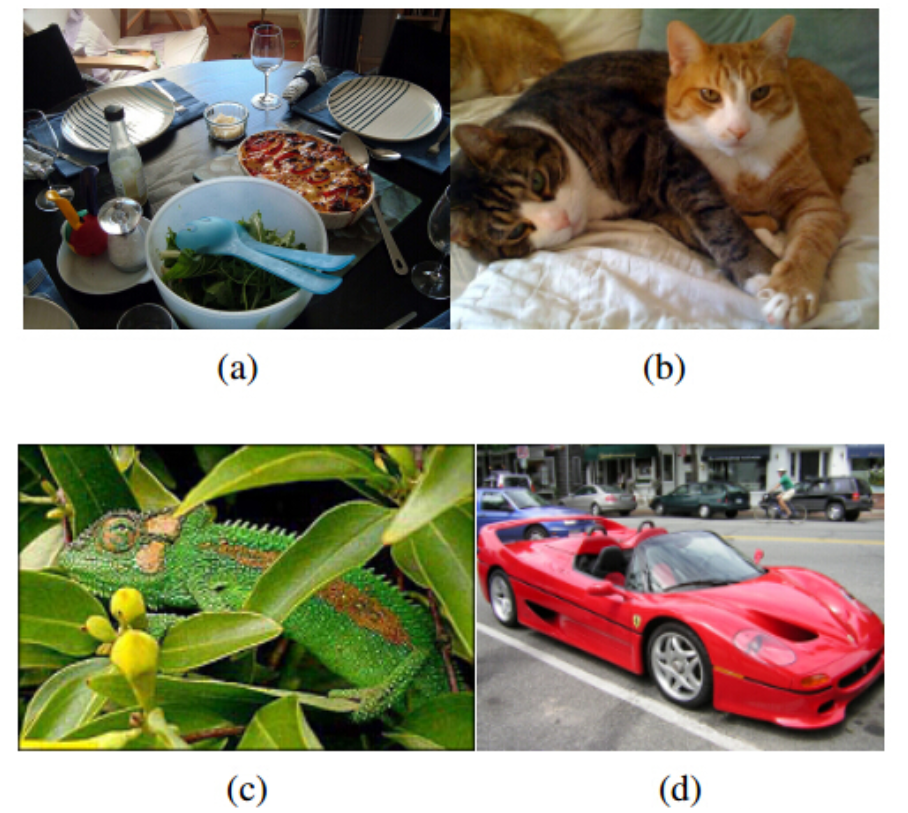
**深入理解R-CNN**

以上部分是对RCNN的简单描述。现在，我们要更深入地理解其中的一些细节，这样，才能清楚应该如何实现它。

1. **Region proposal 提取区域方案**

region proposal的目标就是**从输入图像中提取出可能包含检测物体的region**，而图像（Image）包含的信息非常的丰富，其中的物体（Object）有不同的形状（shape）、尺寸（scale）、颜色（color）、纹理（texture），要想从图像中识别出一个物体非常的难，还要找到物体在图像中的位置，这样就更难了。

下图给出了四个例子，来说明目标检测（Object Recognition）的复杂性以及难度。



由于我们事先不知道需要检测哪个类别，因此图(a)的桌子、瓶子、餐具都是一个个候选目标，而餐具包含在桌子这个目标内，勺子又包含在碗内。这张图展示了目标检测的层级关系以及尺度关系，那我们如何去获得这些可能目标的位置呢。常规方法是通过穷举法，就是在原始图片上进行不同尺度不同大小的滑窗，获取每个可能的位置。而这样做的缺点也显而易见，就是计算量实在是太大了，而且由于不可能每个尺度都兼顾到，因此得到的目标位置也不可能那么准。那么我们能不能通过视觉特征去减少这种分类的可能性并提高精确度呢。这就是本文想做的事情。

可用的特征有很多，到底什么特征是有用的呢？我们看图片(b)的两只猫咪，他们的纹理是一样的，因此纹理特征肯定不行了。而如果通过颜色则能很好区分。但是图(c)变色龙可就不行了，这时候边缘特征、纹理特征又显得比较有用。而在图(d)中，我们很容易把车和轮胎看作是一个整体，但是其实这两者的特征差距真的很明显啊，无论是颜色还是纹理或是边缘都差的太远了。

所以，我们**通过什么特征去区分？应该区分到什么尺度？**

《Rich feature hierarchies for accurate object detection and semantic segmentation》中在Region proposal方案中选用的是selective search，selective search的策略是，既然是不知道尺度是怎样的，那我们就尽可能遍历所有的尺度好了，但是不同于暴力穷举，我们可以先得到小尺度的区域，然后一次次合并得到大的尺寸就好了，这样也符合人类的视觉认知。既然特征很多，那就把我们知道的特征都用上。

具体的流程为：

* 利用Efficient Graph-Based Image Segmentation图像分割方法将原始输入图像分割成很多的初始regions的集合
* 相邻的region计算相似度，保存在一个相似度集合中。
* 在相似度集合中找到最大相似度的两个region，合并成一个region，将这个region存入一个区域集合R中，并更新其与相邻region的相似度产生更新的相似度集合。
* 重复上面的步骤知道整个输入图像成为一个region
* 区域集合R中的所有区域就是利用分层算法得到的待检测物体的可能位置。

简单讲就是通过基于图的图像分割方法初始化原始区域，就是将图像分割成很多很多的小块。然后我们使用贪心策略，计算每两个相邻的区域的相似度，然后每次合并最相似的两块，直到最终只剩下一块完整的图片。然后这其中每次产生的图像块包括合并的图像块我们都保存下来，这样就得到图像的分层表示了。而对于步骤二中，相邻region的相似度该如何计算呢？论文考虑了颜色、纹理、尺寸和空间交叠这4个参数。

1. 颜色相似度
2. 纹理相似度
3. 尺寸相似度，如果仅仅是通过颜色和纹理特征合并的话，很容易使得合并后的区域不断吞并周围的区域，后果就是多尺度只应用在了那个局部，而不是全局的多尺度。即尽量让小的区域先合并。
4. 吻合相似度，不仅要考虑每个区域特征的吻合程度，区域的吻合度也是重要的，吻合度的意思是合并后的区域要尽量规范，不能合并后出现断崖的区域，这样明显不符合常识，体现出来就是区域的外接矩形的重合面积要大。

综合各种相似度。现在各种距离都计算出来，我们要做的就是整合这些距离，最简单的方法当然是加权：

通过这个算法我们搜索出约2000个候选框，然后从上面的总流程中可以看到，搜出的候选框是矩形的，而且是大小各不相同。然而CNN对输入图片的大小是有固定的(如在ALexNet网络中的图像输入大小为227px\*227px)，如果把搜索到的矩形选框不做处理，就扔进CNN中，肯定不行。因此对于每个输入的候选框都需要缩放到固定的大小。

1. **CNN特征提取**

创建了区域边框后，R-CNN网络将该区域块变形标准的方形，并将其输入到改进版的AlexNet网络(AlexNet是在ImageNet 2012比赛中的最优网络，R-CNN灵感就来源于它，我们的代码也使用了AlexNet。)

1. 网络结构设计阶段

我们选择经典的Alexnet，Alexnet特征提取部分包含了5个卷积层、2个全连接层，在Alexnet中p5层神经元个数为9216、 f6、f7的神经元个数都是4096，通过这个网络训练完毕后，最后提取特征每个输入候选框图片都能得到一个4096维的特征向量。

1. 网络有监督预训练阶段

所谓的有监督预训练，我们也可以把它称之为迁移学习。比如你已经有一大堆标注好的人脸年龄分类的图片数据，训练了一个CNN，用于人脸的年龄识别。然后当你遇到新的项目任务是：人脸性别识别，那么这个时候你可以利用已经训练好的年龄识别CNN模型，去掉最后一层，然后其它的网络层参数就直接复制过来，继续进行训练。这就是所谓的迁移学习，说的简单一点就是把一个任务训练好的参数，拿到另外一个任务，作为神经网络的初始参数值,这样相比于你直接采用随机初始化的方法，精度可以有很大的提高。同时，物体标签训练数据少，如果要直接采用随机初始化CNN参数的方法，那么目前的训练数据量是远远不够的。所以paper在设计网络结构的时候，是直接用Alexnet的网络，然后连参数也是直接采用它的参数，作为初始的参数值，然后再fine-tuning训练。

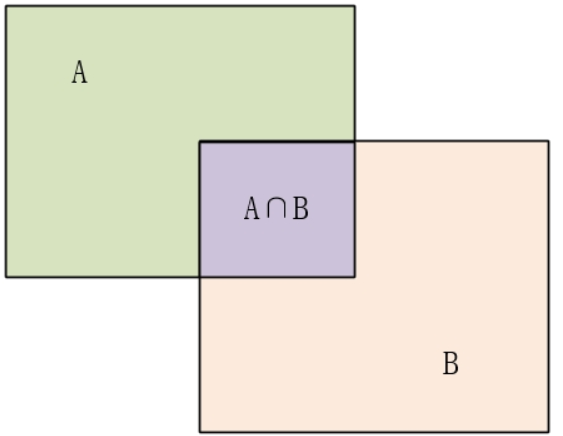
1. fine-tuning阶段

我们接着采用 selective search 搜索出来的候选框 （PASCAL VOC 数据库中的图片）继续对上面预训练的CNN模型进行fine-tuning训练。假设要检测的物体类别有N类，那么我们就需要把上面预训练阶段的CNN模型的最后一层给替换掉，替换成N+1个输出的神经元(加1，表示还有一个背景) (20 + 1bg = 21)。在每次训练的时候，我们batch size大小选择128，其中32个是正样本、96个是负样本。

**关于正负样本问题：**

一张照片我们得到了2000个候选框。然而人工标注的数据一张图片中就只标注了正确的bounding box，我们搜索出来的2000个矩形框也不可能会出现一个与人工标注完全匹配的候选框。因此在CNN阶段我们需要用IOU为2000个bounding box打标签。如果用selective search挑选出来的候选框与物体的人工标注矩形框（PASCAL VOC的图片都有人工标注）的重叠区域IOU大于0.5，那么我们就把这个候选框标注成物体类别（正样本），否则我们就把它当作背景类别（负样本）。

IOU的定义如下：IOU表示了两个region A和B的bounding box的重叠度。



IOU的计算公式：

IOU=(A∩B)/(A∪B)

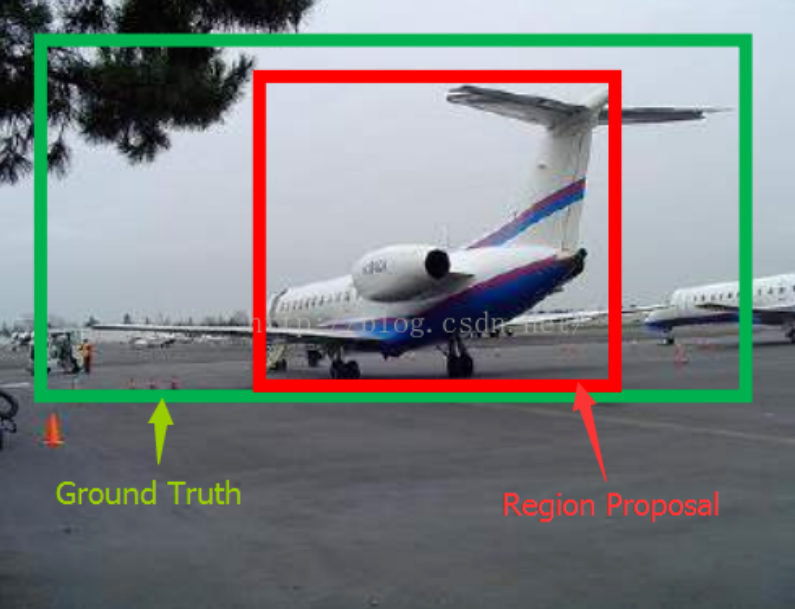
就是矩形框A、B的重叠面积占A、B并集的面积比例:

IOU=SI/(SA+SB-SI)

1. **SVM训练**

这是一个二分类问题，我么假设我们要检测车辆。我们知道只有当bounding box把整量车都包含在内，那才叫正样本；如果bounding box 没有包含到车辆，那么我们就可以把它当作负样本。但问题是当我们的检测窗口只有部分包好物体，那该怎么定义正负样本呢？作者测试了IOU阈值各种方案数值0,0.1,0.2,0.3,0.4,0.5。通过训练发现，如果选择IOU阈值为0.3效果最好（选择为0精度下降了4个百分点，选择0.5精度下降了5个百分点）,即当重叠度小于0.3的时候，就把它标注为负样本。一旦CNN f7层特征被提取出来，那么将为每个类别训练一个SVM分类器。当用CNN提取2000个候选框，可以得到2000\*4096这样的特征向量矩阵，然后只需要把这样的一个矩阵与SVM权值矩阵4096\*N点乘(N为分类类别数目，因为我们训练的N个SVM，每个svm包好了4096个W)，就可以得到结果了。

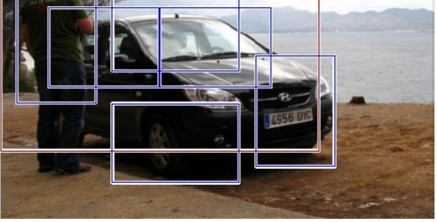
1. **边框回归(Bounding-box regression)**



如上图所示，绿色的框为飞机的Ground Truth，红色的框是提取的Region Proposal。那么即使红色的框被分类器识别为飞机，但是由于红色的框定位不准(IOU<0.5)，那么这张图相当于没有正确的检测出飞机。如果能对红色的框进行微调，使得经过微调后的窗口跟Ground Truth更接近，定位会更准确。Bounding-box regression 就是用来微调这个窗口。

1. **非极大值抑制**

RCNN算法，会从一张图片中找出n多个可能是物体的矩形框，然后为每个矩形框为做类别分类概率：



就像上面的图片一样，定位一个车辆，最后算法就找出了一堆的方框，我们需要判别哪些矩形框是没用的。非极大值抑制：先假设有6个矩形框，根据分类器类别分类概率做排序，从小到大分别属于车辆的概率分别为A、B、C、D、E、F。

* 从最大概率矩形框F开始，分别判断A~E与F的重叠度IOU是否大于某个设定的阈值;
* 假设B、D与F的重叠度超过阈值，那么就扔掉B、D；并标记第一个矩形框F，是我们保留下来的。
* 从剩下的矩形框A、C、E中，选择概率最大的E，然后判断E与A、C的重叠度，重叠度大于一定的阈值，那么就扔掉；并标记E是我们保留下来的第二个矩形框。

就这样一直重复，找到所有被保留下来的矩形框。

**总结**

RCNN的过程要点可以简单的表示为：

* 利用selective search方法对输入图像进行region of interest ROI提取
* 对每个候选区域，使用深度网络提取特征
* 特征送入每一类的SVM 分类器，判别是否属于该类
* 使用回归器精细修正候选框位置

RCNN的主要缺点有：

* 检测速度慢：每一个ROI都需要在CNN网络中进行一次前向传播。
* 训练速度慢：RCNN需要fine-tuning一个SVM分类器，在这之后还要训练一个bounding-box的归回模型。
* 训练过程耗时耗内存：每一个ROI和Feature都需要在训练的过程中存在电脑的内存中，很容易就产生数百G的存储消耗。