## RCNN

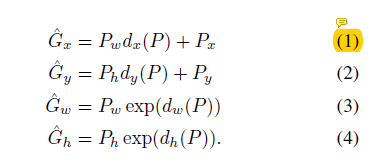
1. Testing：
   1. Region Proposals: use **selective search** extracts 1000-2000 region proposals in one picture, and in order to fit the input size of CNN, the region proposals needs to be warp to 227\*227 pixel;

**Attention**: selective search is **sensitive with image scale**, the number of region proposals may change if the sacle change. So in RCNN, author resize the image into the same pixel 500.

* 1. Feature extraction: use CNN(AlexNet) to extract 4096-d feature vector. The **dimension of feature** vector is related to the layer you choose. Output of P5/ f6/f7 can also be used as feature vector;
  2. Classification: feed the feature vector to **each SVM** trained for classifies, and get the score of each bounding box.

(speed of CNN：13s/image on a GPU or 53s/image on a CPU)

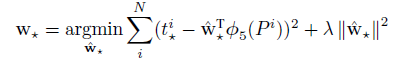
* 1. Location fine-tuning: given all scored regionsin an image, we apply a greedy **non-maximum suppression** (for each class independently) that rejects a region if it has an intersection-over union (IoU) overlap with a higher scoring selected region larger than a learned threshold.
  2. **bounding-box regression can reduce mislocalizations.**

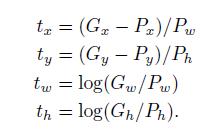


这里使用了一个线性回归模型来微调region propasals的位置，生成最终的bounding box。前两个公式相当于平移变换，后面两个相当于尺寸变换，所以这里一共引入了四个线性函数。



他们都是P的函数，这里使用同一的来表示，并用平方差加岭回归的方式来学习。

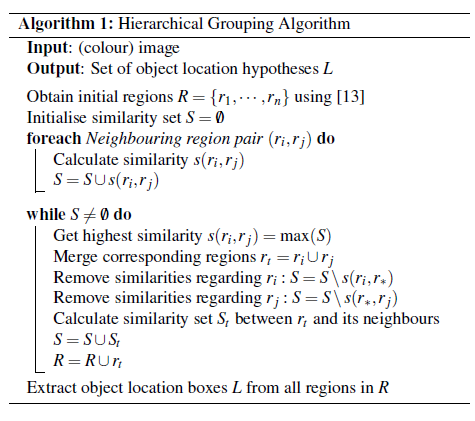
，其中t就是用Ground Truth与预测值的结果求得。



1. Training
   1. Supervised pre-training：先用图像分类的图片集训练AlexNet
   2. Fine-tuning: 修改最后的输出类别维度（20+1background），减慢学习率，然后输入Region Proposals预训练模型，并提取f7（4096d）的特征
   3. Train SVMs：，用b)中得到的特征向量训练svm得到分类结果

### Selective search

RCNN中初步选取Region Proposals选用的是Selective search方法，下面对SS简单地描述一下。SS的主体是一个层级的区域合并过程，但是由于不同物体之间可以通过颜色、纹理、尺寸和填充是个方面来计算相似度，提供合并的依据。



## SPPNet

SPPNet是RCNN之上的又一次进化，针对下列问题进行改进：

1）RCNN中CNN需要单独对每个Region Proposals提取特征带来的冗余与效率低下问题，2）由于FC限制了CNN输入维度的问题，以及为了统一维度进行resize/crop/wrap对图像分割变形带来的扭曲原始图像与部分信息丢失

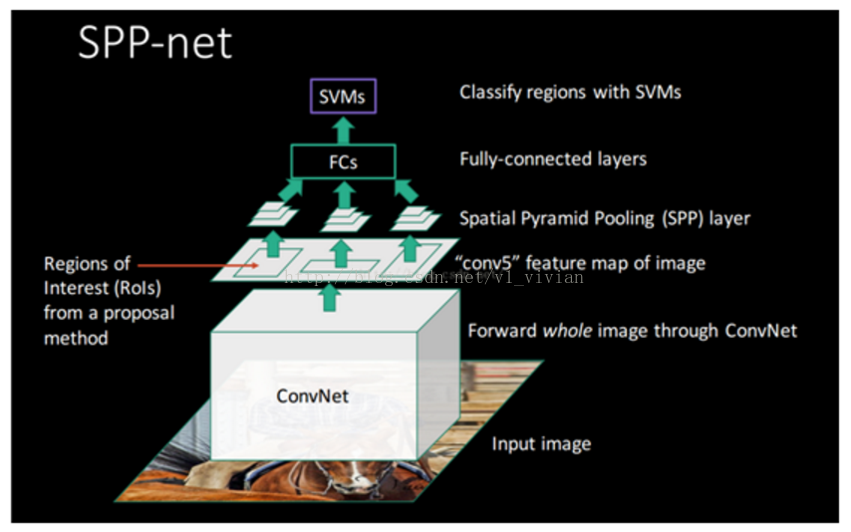
**FC层限制了输入尺寸的原因**：权值W的尺寸与输入输出的大小有关，一个图形，经过卷积层之后输出的维度为30，FC的维度为30，那么此时W的维度为（30×30），如果输入的尺寸变为40，那么理论上完成FC需要的权值维度为（30×40），那么原来的权值尺寸将无法适应变化的输入尺寸，所以在RCNN中，需要限定Region Proposals的大小。

主要思想：空间金字塔池化策略，SPPNet可以生产固定长度的表示，不管图像的大小和规模；对物体变形也很有效。在目标检测中，使用SPPNet的方法只对整个特征图像计算一次。避免了重复计算卷积特征。比R-CNN的方法更快。

SPP layer添加的位置是在最后一个卷积层与第一个FC层之间，因为FC的权重维度是由属于与输出共同决定的，所以卷积之后通过SPP Layer输出一个固定尺寸的向量作为FC的输入，解决了region proposal尺寸受限的问题

而对于SPP-Net，整个过程是：

1. 首先通过选择性搜索，对待检测的图片进行搜索出2000个候选窗口。这一步和R-CNN一样。
2. 特征提取阶段。这一步就是和R-CNN最大的区别了，这一步骤的具体操作如下：把整张待检测的图片，输入CNN中，进行一次性特征提取，得到feature maps，然后在feature maps中找到各个候选框的区域，再对各个候选框采用金字塔空间池化，提取出固定长度的特征向量。而R-CNN输入的是每个候选框，然后在进入CNN，因为SPP-Net只需要一次对整张图片进行特征提取，速度会大大提升。
3. 最后一步也是和R-CNN一样，采用SVM算法进行特征向量分类识别。



SPPnet有一个缺点是只能微调spp层后面的全连接层，所以SPPnet就可以采用随机梯度下降（SGD）来训练。

### Pre-train与fine-tune的联系和差别

Pre-train的model就是指之前被训练好的Model, 比如很大很耗时间的model, 你又不想从头training一遍。这时候可以直接download别人训练好的model，里面保存的都是每一层的parameter配置情况。(Caffe里对于ImageNet的一个model, 我记得是200+M的model大小)。你有了这样的model之后，可以直接拿来做testing, 前提是你的output的类别是一样的。

如果不一样咋办，但是恰巧你又有一小部分的图片可以留着做fine-tuning, 一般的做法是修改最后一层softmax层的output数量，比如从Imagenet的1000类，降到只有20个类，那么自然最后的InnerProducet层，你需要重新训练，然后再经过Softmax层，再训练的时候，可以把除了最后一层之外的所有层的learning rate设置成为0， 这样在traing过程，他们的parameter 就不会变，而把最后一层的learning rate 调的大一点，让他尽快收敛，也就是Training Error尽快等于0.

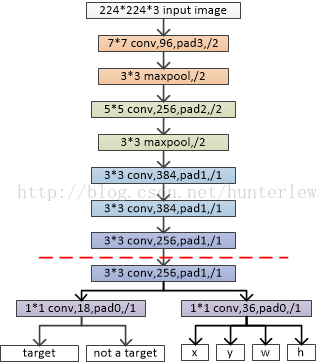
## Fast R-CNN

Fast R-CNN之所以是将原来的先预测region propsal中是否有物体以及具体分类，再修正候选框的串行pipeline修改成并行结构，同时预测出两者。所以在网络结构上做了一些调整，改成了分支结构的输出，一个FC负责分类，一个FC负责region proposal的回归。（分支结构怎么反向传播？？？）

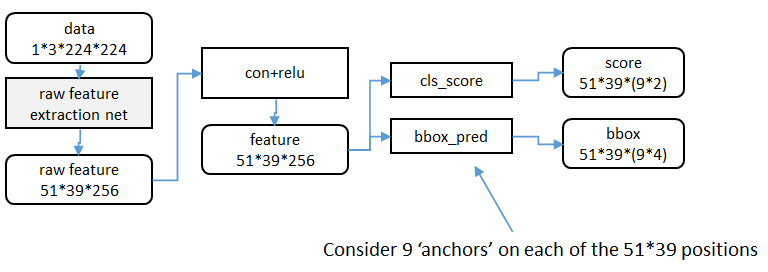
1. R-CNN与SPP Net中使用SVM分类是在CPU之上进行的，速度上远不如GPU，并且CPU与GPU之间的交互也有开销。在Fast R-CNN中在CNN之后添加了Softmax分类取代了原来的SVM分类，使整个网络的进度和速度得到了提升；
2. 在SPP Net中使用的层级pool结构是反向传播机制只能够局限于SPP Layer之上，这里使用了一种特殊情况ROI Pooling即只存在一层Pooling层的SPP Layer，这样使反向传播机制在整个CNN网络上进行，同时又保留了可以不限定输入尺寸的优点。

## Faster R-CNN

输入是共享卷积层的，下图是RPN网络的结构，虚线向上是ZF网络的最后一个卷积成输入，虚线往下就是增加的RPN网络部分，一个包含256个3×3 filter的卷积层以及用于判断有无物体与修正anchor box的两个分支。



### Region Proposal Net



具体的维度转换如上图所示，经过第一列的共享卷积层之后，第二列是RPN独有的卷积层，在这里用256个3×3的filter生成了51×39×256的特征，channel数量就是256。得到第一个卷积层输出之后，将特征分别送入两个不同的卷积层中，其中以一个像素中心的所有特征256d为一个单位，一共可以预测出W\*H\*(2k)用于判断anchor box中是否存在物体（k是anchor box的数量），另一个卷积成可以预测出W\*H\*(4K)用于调整anchor box的大小和比例

1. 网络的输入输出： 入是共享卷积层的feature map（W\*H），出是一系列region proposal以及一个有无物体的评分；（引申阅读：注意力模型）
2. 这里的sliding window就是一个3\*3卷积层，用sliding window进行滑窗时，以中心像素为基点构造9种anchor映射到原始图像中，映射比例为16倍。那么总共可以得到W\*H \*9个anchor (大约2万)；

Attention: 这里解释一下，sliding window在feature map中的运算量其实与anchor box的大小形状没有关系，它是用于判断以中心像素为基点的anchor box有没有物体。通过anchor box映射到原图上的范围与ground true进行对比才是anchor box的调整；

1. 往下网络分为两个分支，一个用于测试以feature map中一个像素点为中心的区域是否存在物体，另一个用于回归anchor box的位置。
2. RPN也有缺点，最大的问题就是对小物体检测效果很差，假设输入为512\*512，经过网络后得到的feature map是32\*32，那么feature map上的一个点就要负责周围至少是16\*16的一个区域的特征表达，那对于在原图上很小的物体它的特征就难以得到充分的表示，因此检测效果比较差。去年年底的工作SSD: Single Shot MultiBox Detector很好的解决了这个问题。
3. 问题：在这里的分支网络中，是如何方向传播调整权重的？？？？？
4. 整个网络是怎么训练的？？

## YOLO

以R-CNN为代表的二刀流region proposals，然后再针对region proposal做分类和位置微调。YOLO系列则是一刀流的代表，直接用一个深度网络，回归出目标的位置和归类，在速度上远胜R-CNN系列。

1. CNN网络输入：网络借鉴的是GoogleNet，整张图片被分成7\*7个grid cell在网络中就是表现为最后一个卷积层输出的特征维度是7\*7\*1024，那么针对每一个grid cell，网络最后的输出是（5+5+20），其中的数值分别代表（x1, y1, w1, h1, confidence1, x2, y2, w2, h2, confidence2, c1, c2, c3, ,,, c20），前10位是两个anchor box的参数，后面是对grid cell对应的分类，维度为7×7×30.
2. Confidence = Pr(object) 0/1 \* IOU，最后的分类概率由Cx \* confidence来决定，最后得到的物体的边界是由分类概率最大的anchor box来决定。
3. （x1, y1, w1, h1）是对anchor box位置的描述，但是x1, y1是受限与grid cell内部的，而w1, h1是对整个图片而言的
4. 训练细节：在计算loss中，坐标与分类概率的重要性不相同，因此加大了坐标的损失权重，减小了分类概率的权重；框比较小和框比较大的物体，对于坐标误差的容忍度也不同，利用开根函数，加大了小物体的坐标损失权重，降低大物体的坐标损失权重；对不含目标的物体不计算loss

## SSD

1. SSD提取了不同尺度的特征图来做检测，大尺度特征图（较靠前的特征图）可以用来检测小物体，而小尺度特征图（较靠后的特征图）用来检测大物体；
2. SSD采用了不同尺度和长宽比的先验框（Prior boxes, Default boxes，在Faster R-CNN中叫做锚，Anchors）。

## FPN

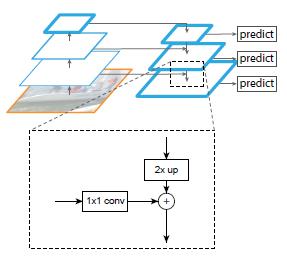
1. 设计理念

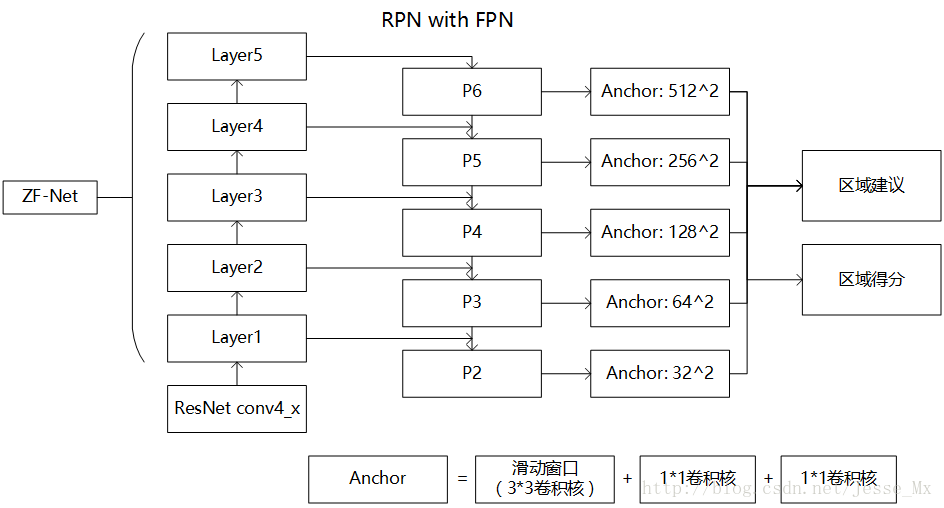
与SSD的构想有相似之处，都想用不同层的特征图来预测结果。与SSD相比进行了一些修改；

* 1. 特征图的生成：bottom-up + lateral connection + top-dowm的形式得到鲁棒性好的高层特征；

不同的stage（把网络中输出规模没有改变的conv layers称为同一个stage）形成了自顶向下的特征融合。假设原来的特征平面是（C2,C3,C4,C5,C6），首先最后一个平面P6就是C6，这是前向传播的最高层特征；接下来对P6进行2倍上采样，并且对C5进行1×1卷积，修改其通道数与P6相同，将两者按照像素累加，最后对累加的结果用3×3卷积（防止混叠效应）得到特征图P5。P5的大小与C5是一样的。同理得到（P2,P3,P4）最终。

* 1. 假设我们现在将FPN用于RPN中，现在分别对（P2,P3,P4,P5,P6）5个不同的特征图选取规模分别为（32^2, 63^2, 128^2, 256^2, 512^2）的anchor box，横纵比为（2：1， 1：1， 1：2）。这五个特征图共享一个卷积层（3×3 sliding window），分别用两个1×1卷积预测得到confidence与localization.





* 1. 将FPN应用与Fast R-CNN的目标检测部分时，ROI Pooling会根据ROI的大小来计算出对应使用的特征层来计算预测结果。

https://img-blog.csdn.net/20170117220352603?watermark/2/text/aHR0cDovL2Jsb2cuY3Nkbi5uZXQvSmVzc2VfTXg=/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70/gravity/SouthEast

## Mask R-CNN

### 主要的技术改进点

1. **网络的改进**
   1. **增加了mask分支**，对每一个ROI Align提取K个M\*M像素的特征（一个物体的分类一共有k类），然后resize成ROI的大小，与ground truth做比较。这个Mask分支实际上是一个FCN来实现的。
   2. **更强大的baseline**：ResNet101+FPN（FPN可以用于RPN，也可以用于主干预测，这里是用于主干中ROI特征层的提取，在FPN中有介绍，根据图片的W×H选择feature map的level来做ROI的feature）
2. **用ROI Align代替ROI Pooling**，两者的不同之处，同样输出2×2的卷积，当pooling框与原图的像素不是一一对应的时候，ROI Align会用**双线性插值**的方法来计算，减少卷积前后的位置误差，使位置的映射更加精确；

**多线性插值**的例子：

一个坐标位置为（0.75, 0.75）像素中心的灰度需要结合（0，0）（0，1）（1，0）（1，1）的灰度值共同计算得到：

f(0.75, 0.75) = (1-x)(1-y)f(0,0) + (1-x)yf(0,1) + x(1-y)f(1,0) + xyf(1,1)，其中点的位置离（1，1）比较近，所以系数为0.75×0.75， 坐标位置离（0，0）比较远，所以系数为0.25×0.25

1. **Multi-Task Loss**：Loss由Classification、Localization与Mask的Loss组成，前两个Loss的设置与Faster RCNN中是一致的，Mask的Loss与只与ground Truth那一类所在的mask有关，其他的mask与Loss无关，用平均二值交叉熵损失来定义Mask Loss（ sigmoid的损失函数）