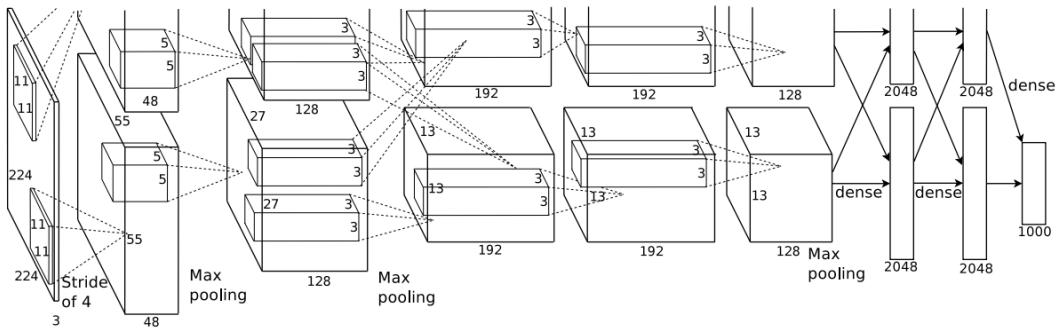


AlexNet



1. ReLU
2. Training on Multiple GPUs
3. Local Response Normalization (LRN层, 现已不使用)
4. Overlapping Pooling (重叠的池化, 现已不使用)
5. Data Augmentation (image translations and horizontal reflections, 颜色变化)
6. Dropout

训练阶段 每一个 batch
随机掐死一半的神经元
(将神经元输出设为0)
阻断该神经元的前向-反向传播
预测阶段 保留所有神经元
预测结果乘 0.5

Dropout 为什么能减少过拟合?

- A. 模型集成 $p=0.5$ 意味着 2^n 个共享权重的潜在网络
- B. 记忆随机抹去 不再死记硬背
- C. Dropout: 减少神经元之间的联合依赖性
每个神经元都被逼着独当一面
- D. 有性繁殖 每个基因片段都要与来自另一个随机个体的
基因片段协同工作

E. 数据增强 总可以找到一个图片 使神经网络中间层结果
与 Dropout 后相同 相当于增加了这张图片到数据集中
F. 稀疏性

YOLO家族

YOLOV1

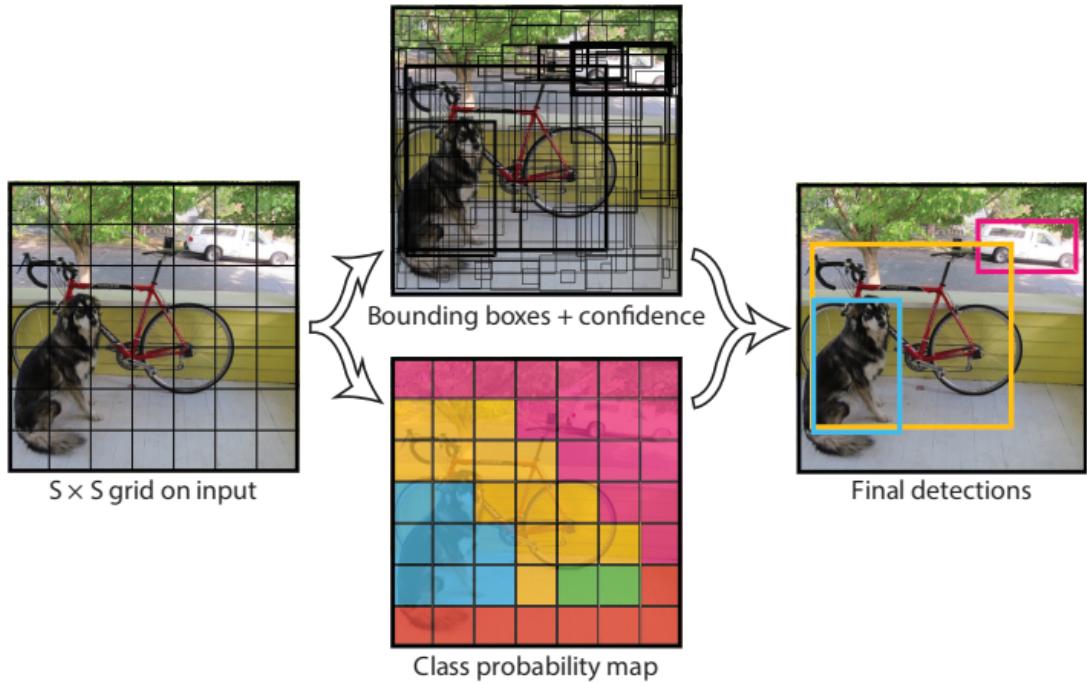
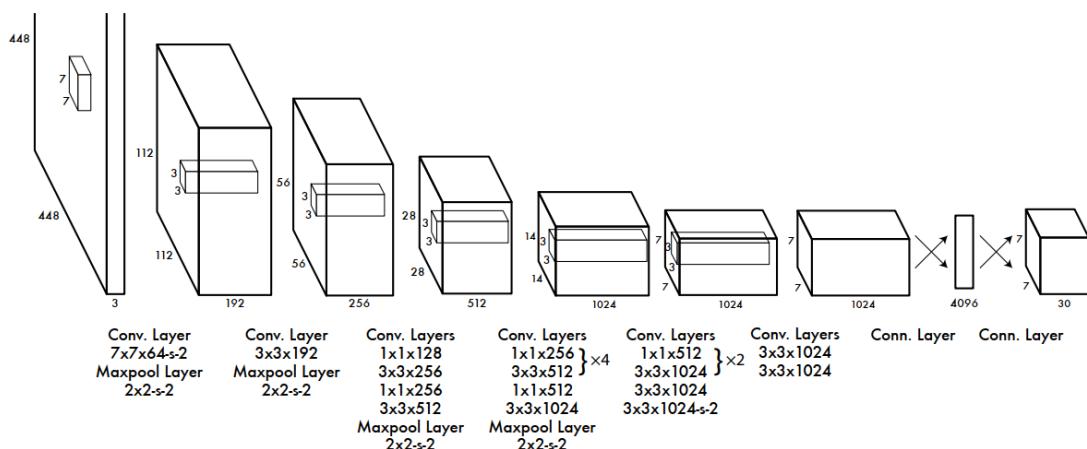


Figure 2: The Model. Our system models detection as a regression problem. It divides the image into an $S \times S$ grid and for each grid cell predicts B bounding boxes, confidence for those boxes, and C class probabilities. These predictions are encoded as an $S \times S \times (B * 5 + C)$ tensor.



预测阶段

Input: $448 \times 448 \times 3$

将图像大小分为 $7 \times 7 = 49$ 个部分 (grid cell)

每个 grid cell 包含两个 bounding box

所以最后的 output 为 $7 \times 7 \times 30$ 的张量

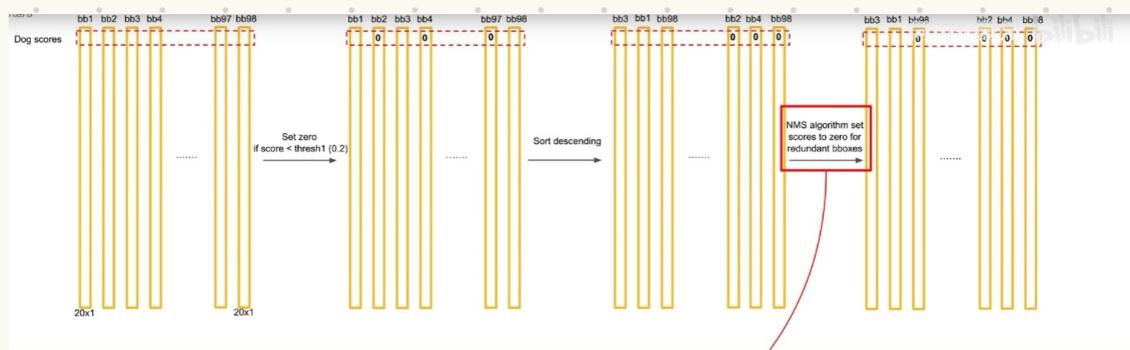
其中 30: 1-5 为第一个 bbox 的 P(置信度), X, Y, Width, Height

6-10 为第二个 bbox 的 P(置信度), X, Y, Width, Height

11-30 为某一个类别别的条件概率 (Pascal VOC 有 20 个类别, 所以此处为 20 维)

所以每一个 bbox 对应一个 20×1 的概率向量 (20×1 的条件概率向量 $\times P = 全概率$)

共有 $7 \times 7 \times 2 = 98$ 个 20×1 的概率向量



选择 20 维中的第一维数 (即该类 dog 的概率)

1. 设定一个概率阈值, 低于的都置 0

2. 不重叠放在前面, 为重叠放在后面 (从高到低排序)

3. NMS

NMS: 非极大值抑制 (可检测多目标)

1. 将所有框与最大的框计算 Iou, 大于阈值的, 从重叠放到一个列表, 直接剔除

2. 将所有框与第 2 大的

3. 依此类推, 最后剩下的框为检测框

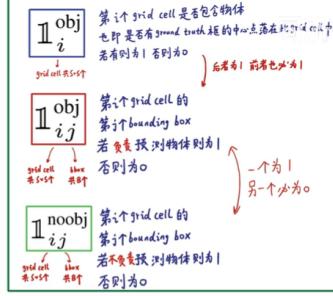
YOLO V1 损失函数

每一项都是平方和误差 将目标检测问题当作回归问题

批注: 同济子豪兄

loss function:

$$\begin{aligned}
 & \text{负责检测物体的bbox} \quad \text{中心点定位误差} \\
 & \lambda_{\text{coord}} \sum_{i=0}^S \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} [(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2] \\
 & \text{负责检测物体的bbox} \quad \text{宽高定位误差} \\
 & \text{宽高定位误差} \\
 & \text{若相等则对误差更敏感} \\
 & + \lambda_{\text{coord}} \sum_{i=0}^S \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} \left[(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2 \right] \\
 & \text{负责检测物体的bbox} \quad \text{Confidence 误差} \\
 & \text{Confidence 误差} \\
 & \text{不负责检测物体的bbox} \quad \text{Confidence 误差} \\
 & + \sum_{i=0}^S \sum_{j=0}^B \mathbb{1}_{ij}^{\text{obj}} (C_i - \hat{C}_i)^2 \\
 & \text{负责检测物体的grid cell} \quad \text{分类误差} \\
 & \text{分类误差} \\
 & + \lambda_{\text{noobj}} \sum_{i=0}^S \sum_{j=0}^B \mathbb{1}_{ij}^{\text{noobj}} (C_i - \hat{C}_i)^2 \\
 & \text{类别预测误差} + \sum_{i=0}^S \mathbb{1}_i^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2 \quad (3)
 \end{aligned}$$



1. Leaky ReLU

2. Grid Cell

模型泛化迁移能力较强，小物体检测性能较差（由于grid cell的限制导致）

定位误差较大，但区分背景能力强（对比RCNN，可以看到整图信息，而不是region proposal）

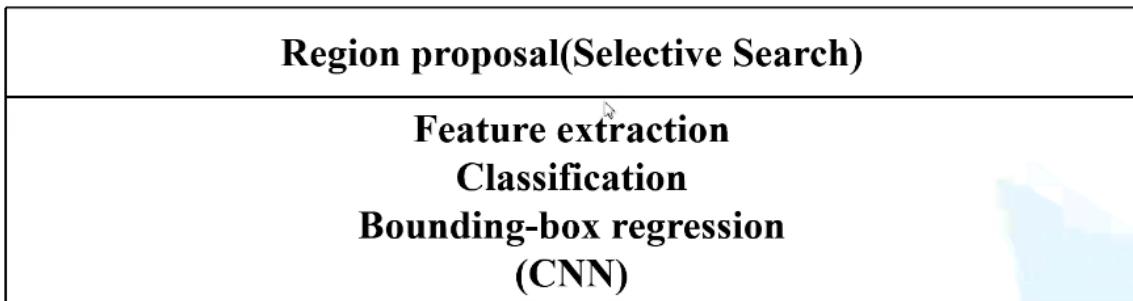
R-CNN家族

框架对比

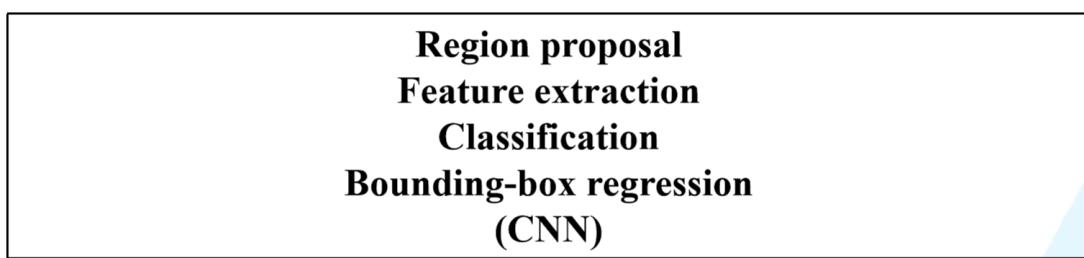
R-CNN框架

Region proposal(Selective Search)	
Feature extraction(CNN)	
Classification (SVM)	Bounding-box regression (regression)

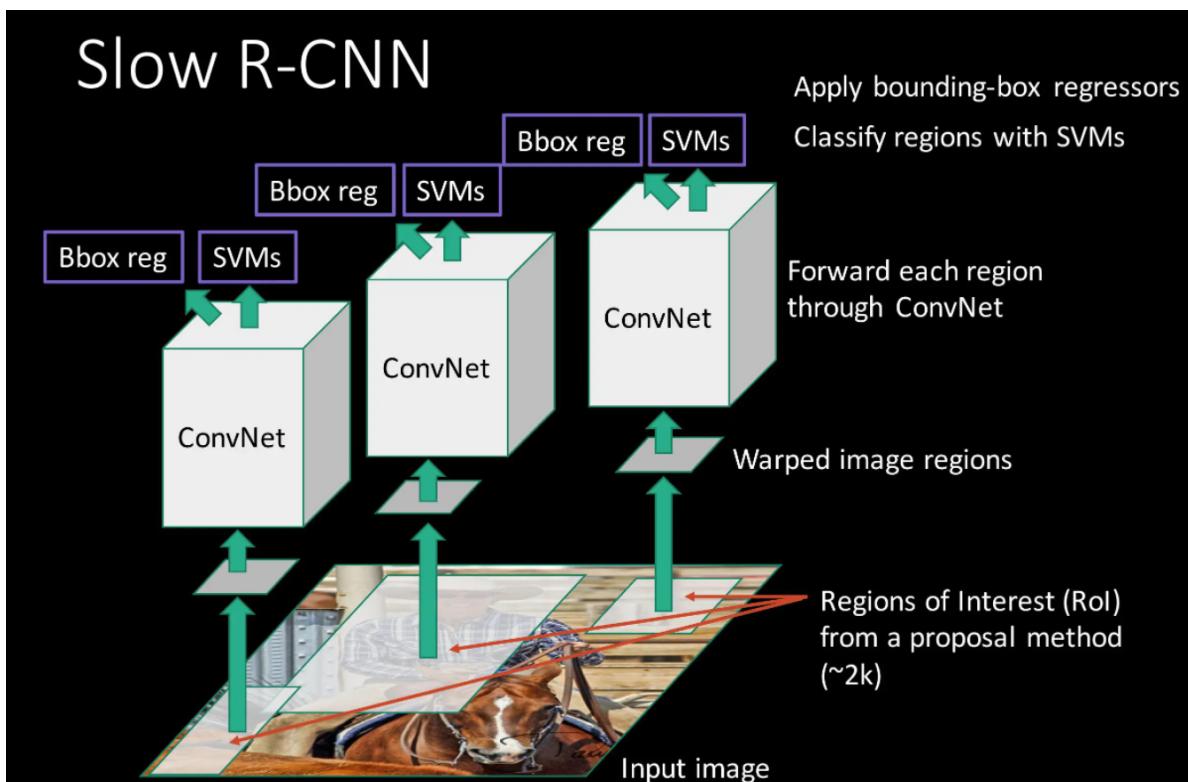
Fast R-CNN框架



Faster R-CNN框架



R-CNN



1. pre-trained + fine-tuning (应对小数据集时，用在大数据集上训练的预训练模型，再在小数据集上微调)
2. pre-trained模型的主要信息来自于conv层而不是fc层
3. VGG16作为backbone
4. region proposal

5. selective search
6. bounding box regression
7. SVM

SPP-Net

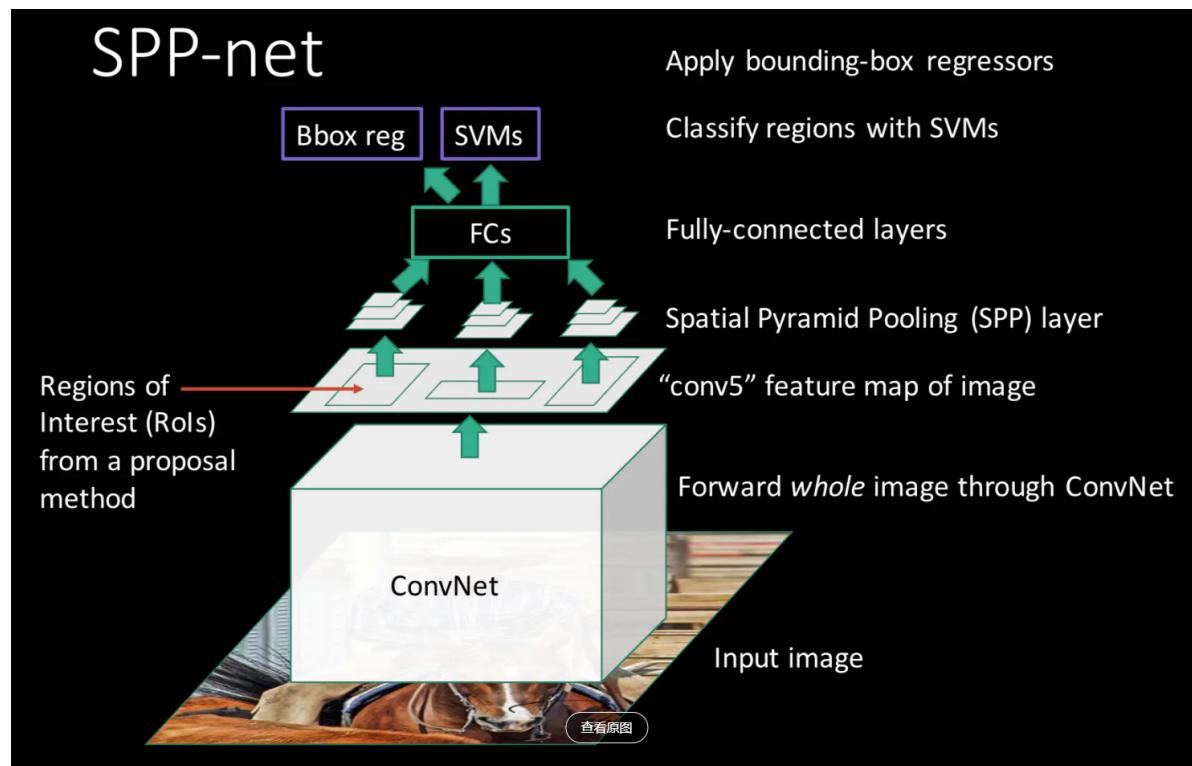
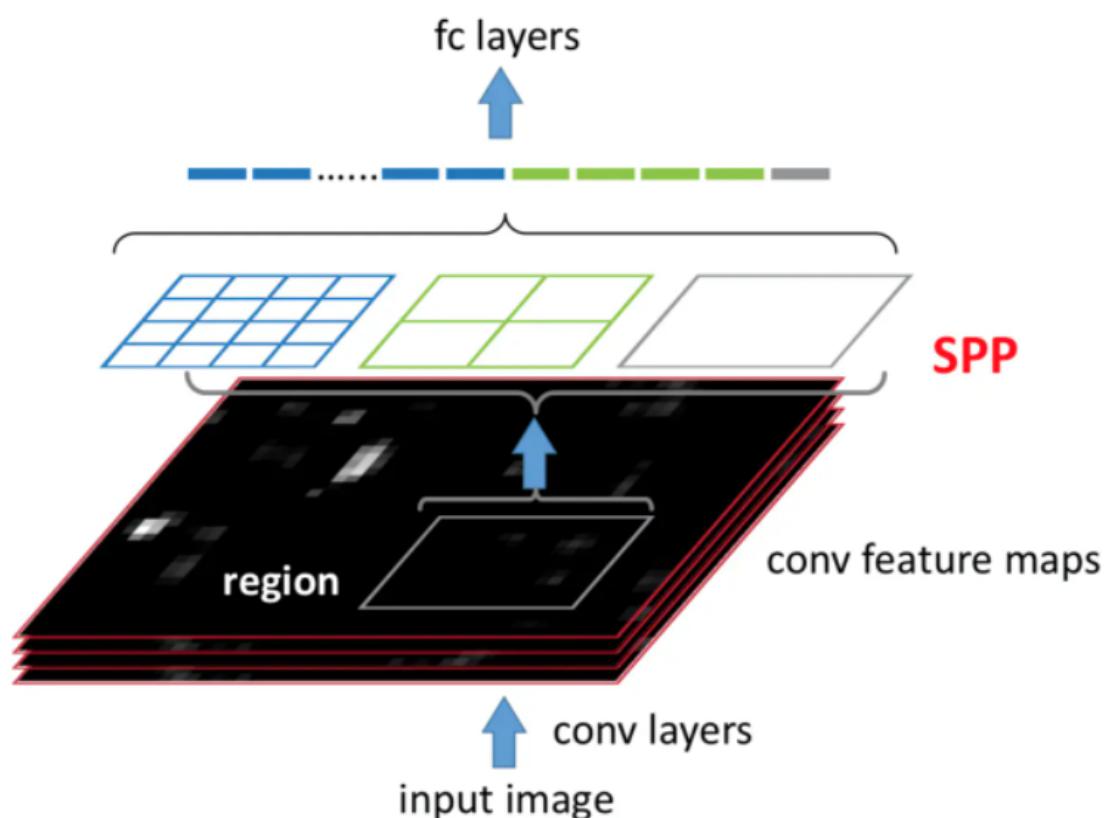


image → conv layers → spatial pyramid pooling → fc layers → output

1. spp层



输入：最后一层卷积输出的特征(我们称为feature map), feature map为上图的黑色部分表示, SPP层的输入为与候选区域对应的在feature map上的一块区域

输出：一共有256个feature map, 每个feature map通过 $4 * 4, 2 * 2, 1 * 1$ 的max pooling的之后得到一个 $(16+4+1) * 256$ 的固定输出的特征向量

2. 候选区域在原图与feature map之间的映射关系

类型	大小
第 l 层的输入尺寸	$W^{[l-1]} * H^{[l-1]}$
第 l 层的输出尺寸	$W^{[l]} * H^{[l]}$
第 l 层的卷积核大小	$f^{[l]} * f^{[l]}$
第 l 层的卷积步长	$S^{[l]}$
第 l 层的填充大小	$p^{[l]}$

输入的尺寸大小与输出的尺寸大小有如下关系：

$$W^{[l]} = (W^{[l-1]} + 2p^{[l]} - f^{[l]}) / S^{[l]} + 1$$

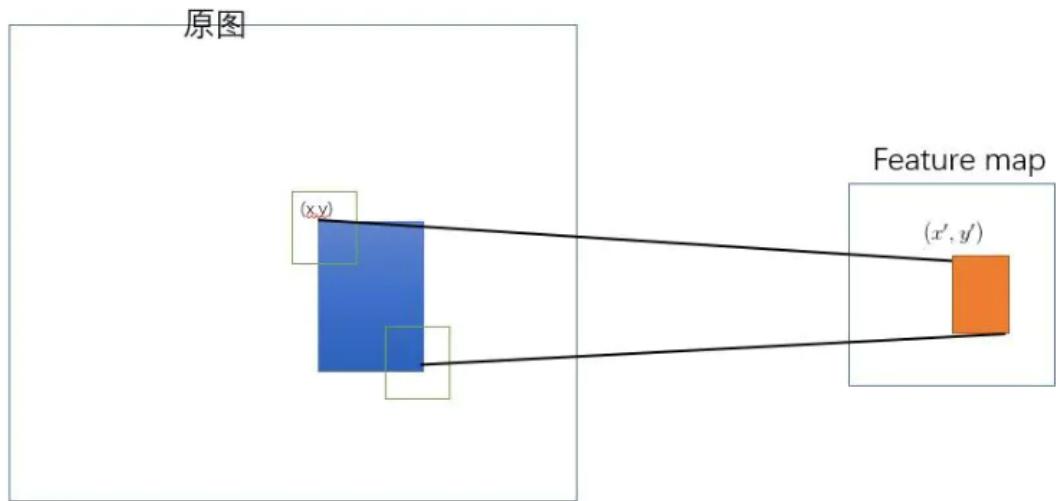
$$H^{[l]} = (H^{[l-1]} + 2p^{[l]} - f^{[l]}) / S^{[l]} + 1$$

上面是区域尺寸大小的对应关系，下面看一下坐标点之间的对应关系

$$p_i = s_i \cdot p_{i+1} + ((k_i - 1)/2 - padding)$$

含义	符号
在 i 层的坐标值	p_i
i 层的步长	s_i
i 层的卷积核大小	k_i
i 层填充的大小	padding

spp-net设置每层的padding = $k_i/2$,这样 $p_i = s_i * p_{i+1}$

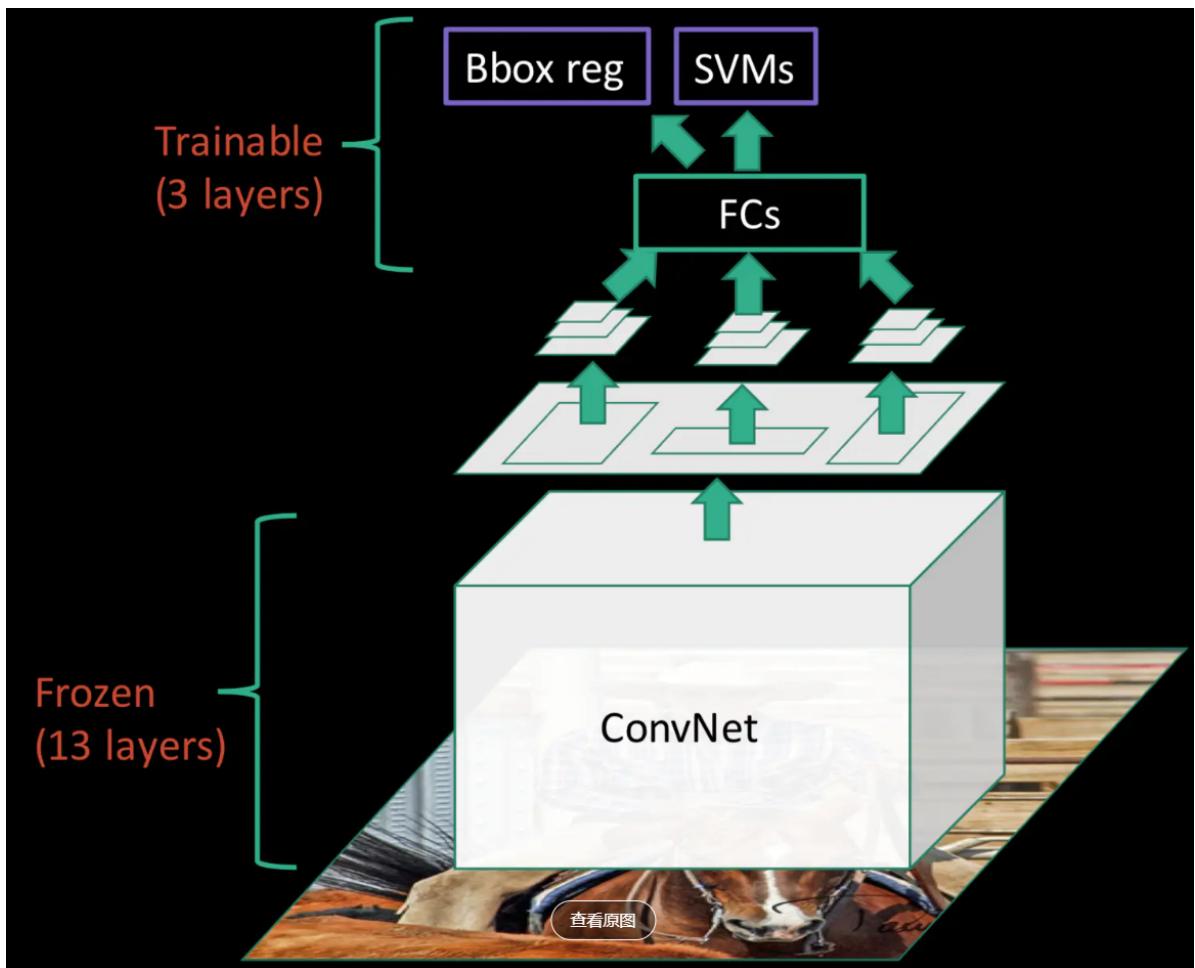


从原图坐标 (x, y) 到特征图中坐标 (x', y') 的映射关系为

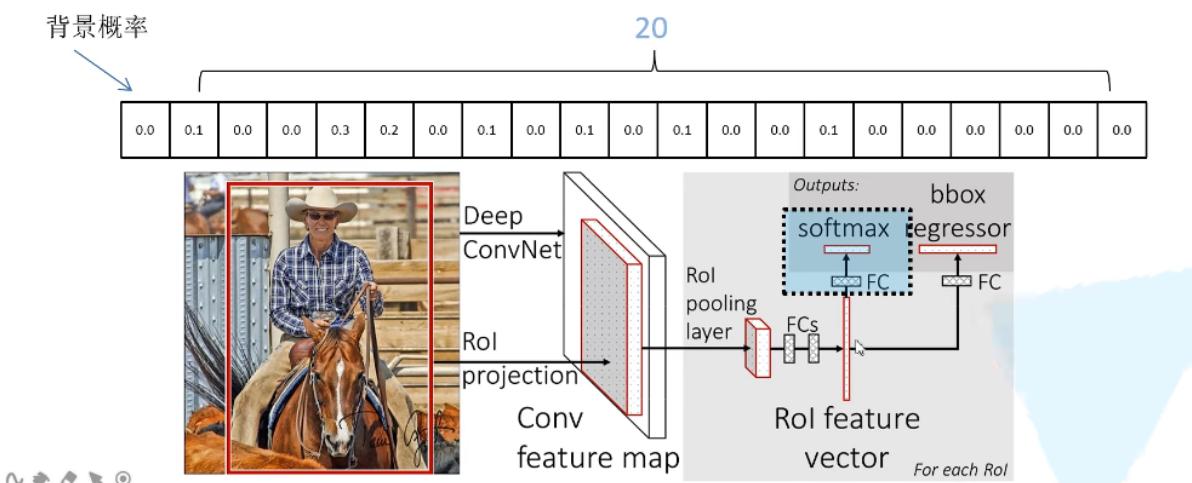
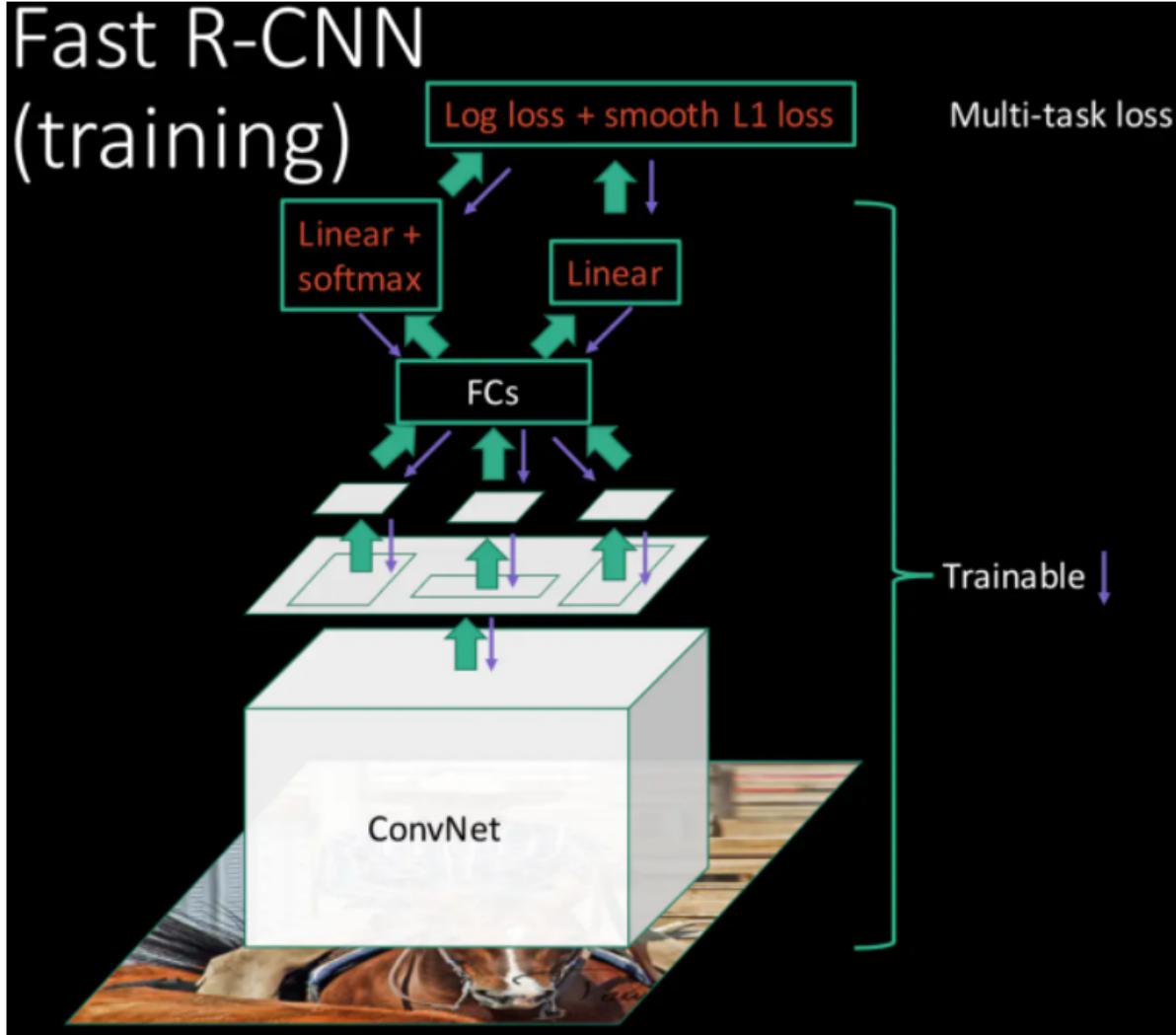
- 前面每层都填充padding/2 得到的简化公式 : $p_i = s_i \cdot p_{i+1}$
- 需要把上面公式进行级联得到 $p_0 = S \cdot p_{i+1}$ 其中 $(S = \prod_0^i s_i)$
- 对于feature map 上的 (x', y') 它在原始图的对应点为 $(x, y) = (Sx', Sy')$
- 论文中的最后做法: 把原始图片中的ROI映射为 feature map中的映射区域(上图橙色区域)其中左上角取: $x' = \lfloor x/S \rfloor + 1, y' = \lfloor y/S \rfloor + 1$; 右下角的点取: y' 的 x 值:
 $x' = \lceil x/S \rceil - 1, y' = \lceil y/S \rceil - 1$ 。

Fast R-CNN

SPP-Net的缺点: 在fune-tuning阶段不能对SPP层下面所有的卷积层进行后向传播



Fast R-CNN (training)



Fast R-CNN结合了R-CNN与SPP，并作出改进

1. VGG16作为backbone
2. selective search生成约2000候选框，在候选框中随机取N个正样本和N个负样本训练，其中正样本为与GT IOU>0.5,负样本为与GT 0.1<IOU<0.5
3. SPP层，仅用一个尺度的pooling (SPP为多尺度)
4. 引入多任务损失函数，用于同时计算bbox回归和分类的损失

Multi-task loss

$$L(p, u, t^u, v) = L_{cls}(p, u) + \lambda[u \geq 1]L_{loc}(t^u, v)$$

分类损失 边界框回归损失

p 是分类器预测的softmax概率分布 $p = (p_0, \dots, p_k)$

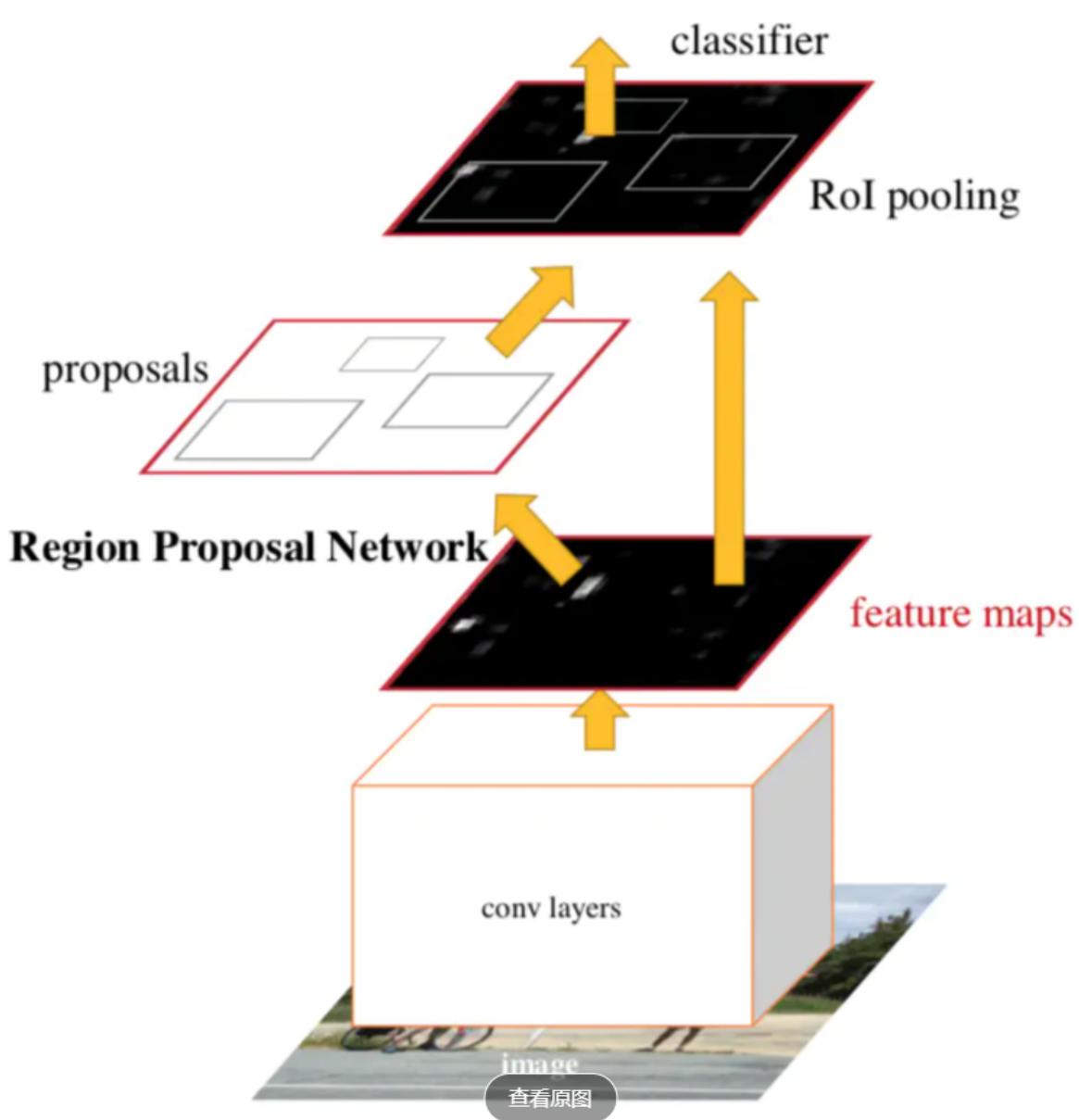
u 对应目标真实类别标签

t^u 对应边界框回归器预测的对应类别 u 的回归参数 $(t_x^u, t_y^u, t_w^u, t_h^u)$

v 对应真实目标的边界框回归参数 (v_x, v_y, v_w, v_h)



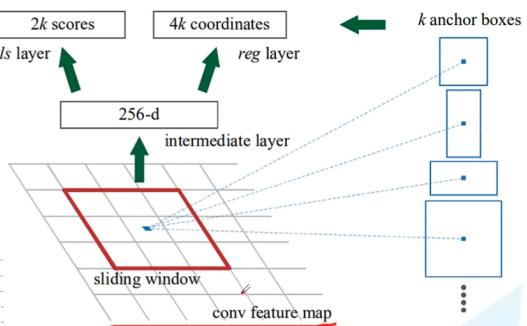
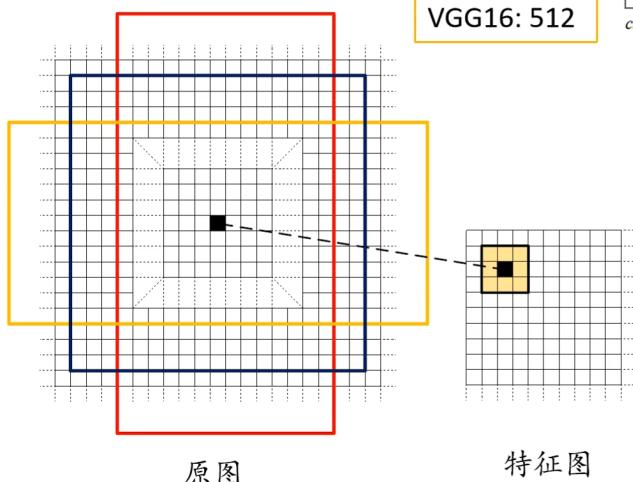
Faster R-CNN



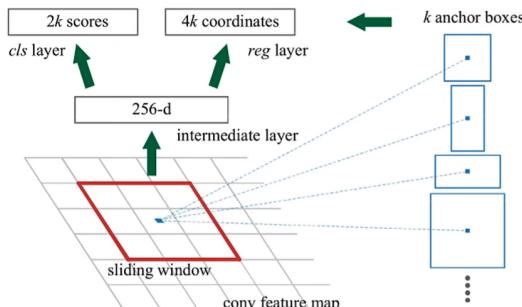
可以看作RPN (region proposal network) +Fast R-CNN

Faster R-CNN

RPN的原论文名字是啥呀



对于特征图上的每个 3×3 的滑动窗口，计算出滑动窗口中心点对应原始图像上的中心点，并计算出 k 个anchor boxes(注意和proposal的差异)。



三种尺度(面积){ $128^2, 256^2, 512^2$ }

三种比例{ 1:1, 1:2, 2:1 }

每个位置在原图上都对应有 $3 \times 3 = 9$ anchor

对于一张 $1000 \times 600 \times 3$ 的图像，大约有 $60 \times 40 \times 9 (20k)$ 个anchor，忽略跨越边界的anchor以后，剩下约 $6k$ 个anchor。对于RPN生成的候选框之间存在大量重叠，基于候选框的cls得分，采用非极大值抑制，IoU设为0.7，这样每张图片只剩 $2k$ 个候选框。



RPN Multi-task loss

$$L(\{p_i\}, \{t_i\}) = \frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*) + \lambda \frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*)$$

分类损失 边界框回归损失

p_i 表示第*i*个anchor预测为真实标签的概率

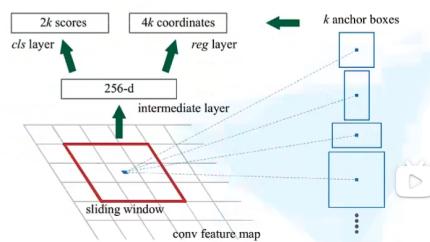
p_i^* 当为正样本时为1,当为负样本时为0

t_i 表示预测第*i*个anchor的边界框回归参数

t_i^* 表示第*i*个anchor对应的GT Box

N_{cls} 表示一个mini-batch中的所有样本数量256

N_{reg} 表示anchor位置的个数(不是anchor个数)约2400



Faster R-CNN

Faster R-CNN训练

直接采用RPN Loss+ Fast R-CNN Loss的联合训练方法

原论文中采用分别训练RPN以及Fast R-CNN的方法

(1)利用ImageNet预训练分类模型初始化前置卷积网络层参数，并开始单独训练RPN网络参数；

(2)固定RPN网络独有的卷积层以及全连接层参数，再利用ImageNet预训练分类模型初始化前置卷积网络参数，并利用RPN网络生成的目标建议框去训练Fast RCNN网络参数。

(3)固定利用Fast RCNN训练好的前置卷积网络层参数，去微调RPN网络独有的卷积层以及全连接层参数。

(4)同样保持固定前置卷积网络层参数，去微调Fast RCNN网络的全连接层参数。最后RPN网络与Fast RCNN网络共享前置卷积网络层参数，构成一个统一网络。

现在直接联合训练，原论文中采用分步训练的方法

