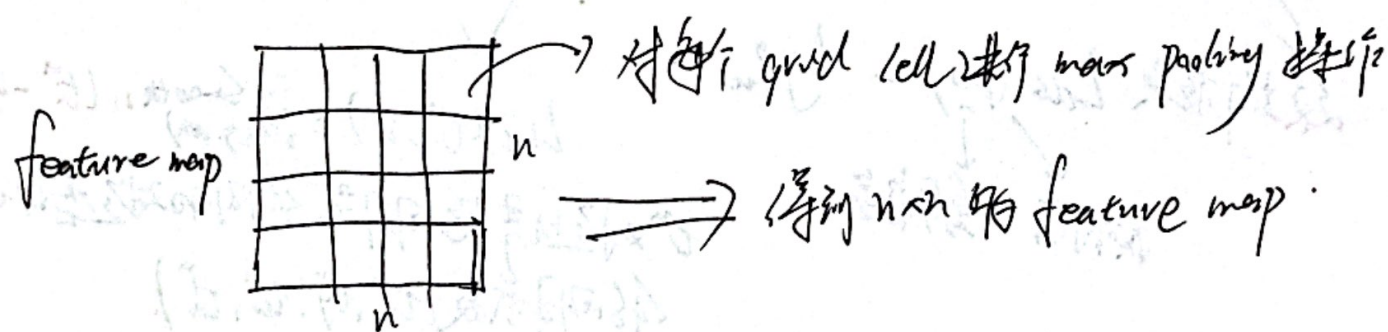


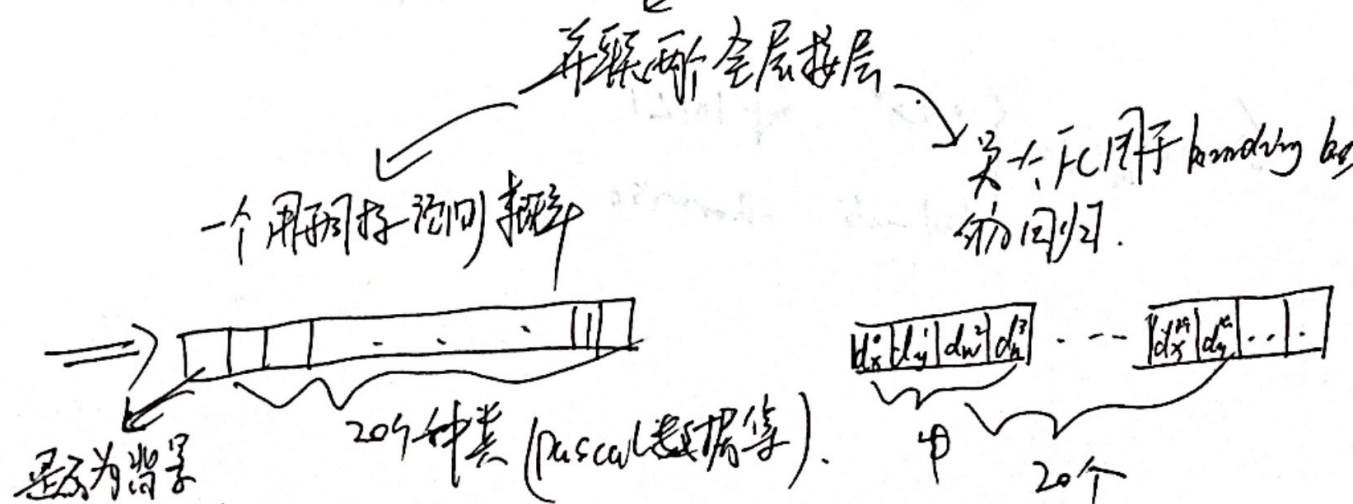
Fast R-CNN

1. 将整张图用SS算法生成1k-2k个候选框。
 2. 将整张图输入卷积网络得到相应的feature map。同时将候选框投影到特征图上得到相应的前景矩阵 (将候选框投影到feature map的相应位置)。
 3. 将每个特征图矩阵通过ROI pooling层缩放到7x7大小的特征图。
- ROI: Region of Interest 感兴趣区域。

ROI pooling: 对feature map的 $n \times n$ 的网格 (grid cell)



4. 将ROI操作后的特征图扁平化通过一到FC层得到预测结果。



候选框的同参数 dx, dy, dw, dh

$$\hat{L}_x = p_w dx(p) + p_x$$

p_x, p_y, p_w, p_h 分别为候选框中的 x, y 坐标, 以及宽高

$$\hat{L}_y = p_h dy(p) + p_y$$

$$\hat{L}_w = p_w \cdot \exp(dw(p))$$

$\hat{L}_x, \hat{L}_y, \hat{L}_w, \hat{L}_h$ 分别为最终预测的边界框中的 x, y 坐标

$$\hat{L}_h = p_h \cdot \exp(dh(p))$$

$$\begin{cases} u = -n & n \geq 1 \\ u = 0 & u < 1 \end{cases}$$

$$Loss = \text{分类损失} + \text{边界框损失} = L_{cls}(p, u) + \lambda |u| L_{loc}(t^n, v)$$

艾弗森符号.

边界框损失 $L_{cls}(p, u) = -\log p_u$

\downarrow 预测值 \downarrow 真实标签

$$L_{loc}(t^n, v) = \sum_{i \in \{x, y, w, h\}} \text{Smooth}_L(t_i^n - v_i)$$

t^n 对应边界框同意的预测值的同参数 $(t_x^n, t_y^n, t_w^n, t_h^n)$

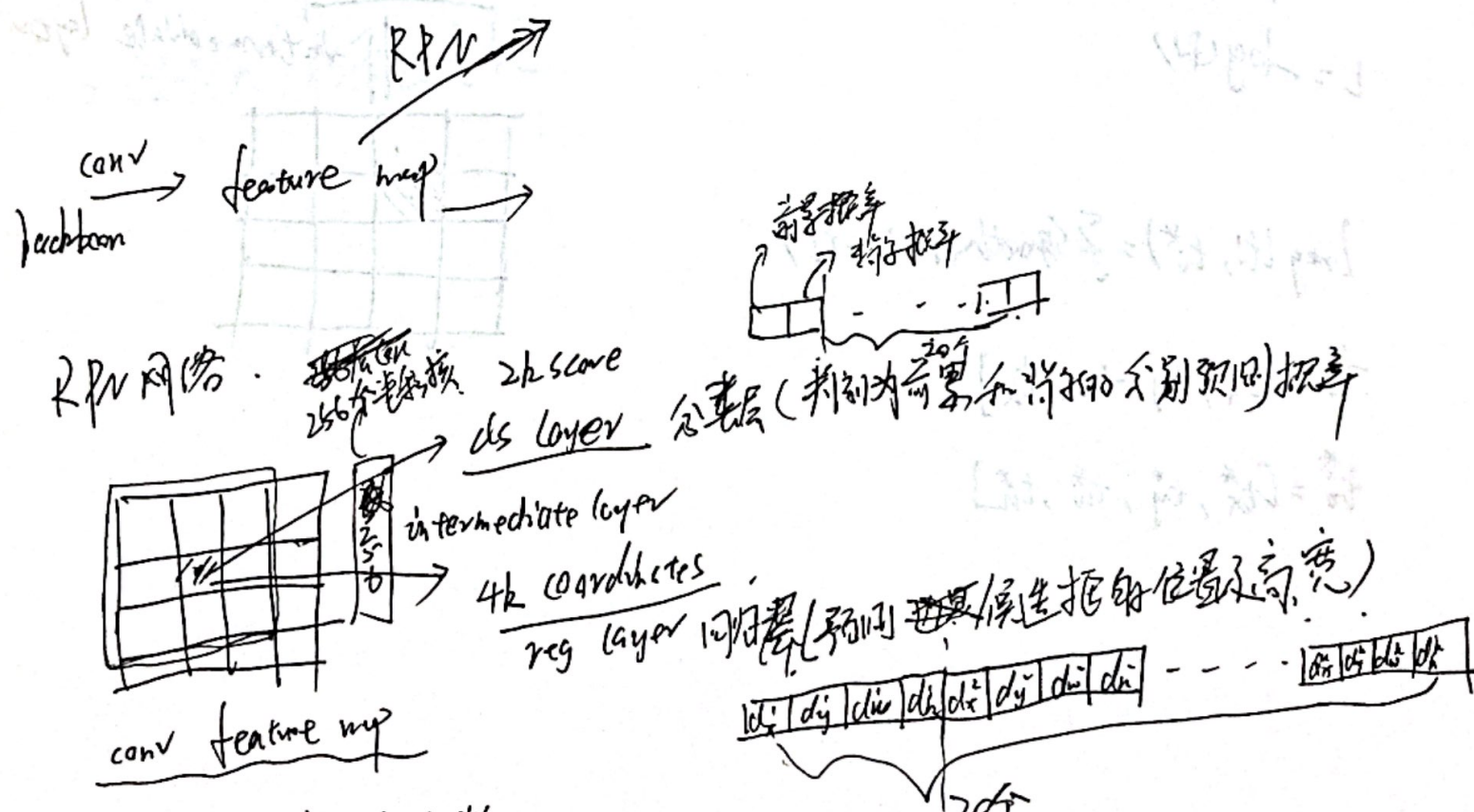
v 对真实同意的边界框同参数 (v_x, v_y, v_w, v_h)

$$\text{Smooth}_L(x) = \begin{cases} 1.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases}$$

Faster R-CNN (RPN + Fast R-CNN).

同样使用 VGG16 作为 backbone → 候选框生成网络.

1. 将整张图片输入到 backbone 得到特征图.
2. 使用 RPN 结构生成候选框 将 RPN 生成的候选框投影到特征图获得 ~~候选框~~ 对应的特征矩阵 (替代原 RPN 的 SS 算法)
3. 将特征矩阵通过 ROI pooling 层得到 7x7 大小的 feature map
4. 将 ROI pooling 操作后的特征图扁平化通过一系列全连接得到预测结果



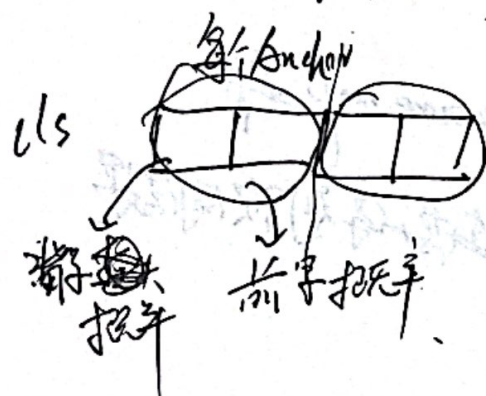
k : anchor box 的个数
 Anchor 的 66 例 $\{1:1, 1:2, 2:1\}$
 尺度 $\{128^2, 256^2, 512^2\}$.
 总共生成 $3 \times 3 = 9$ 个 Anchor

Anchor RPN 的 21 候选框

RPN Multi-task loss

$$L(\{p_i\}, \{t_i\}) = \underbrace{\frac{1}{N_{cls}} \sum_i L_{cls}(p_i, p_i^*)}_{\text{分类损失}} + \lambda \underbrace{\frac{1}{N_{reg}} \sum_i p_i^* L_{reg}(t_i, t_i^*)}_{\text{回归损失}}$$

分类损失: 使用到 softmax cross Entropy.

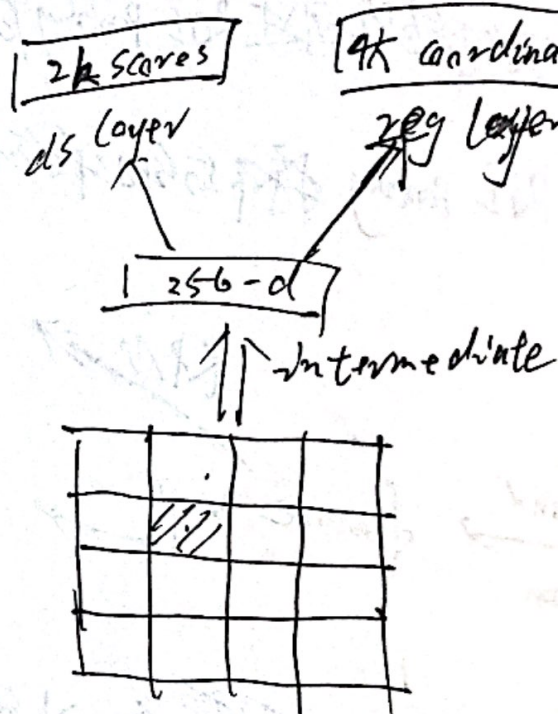


$$L = -\log(p)$$

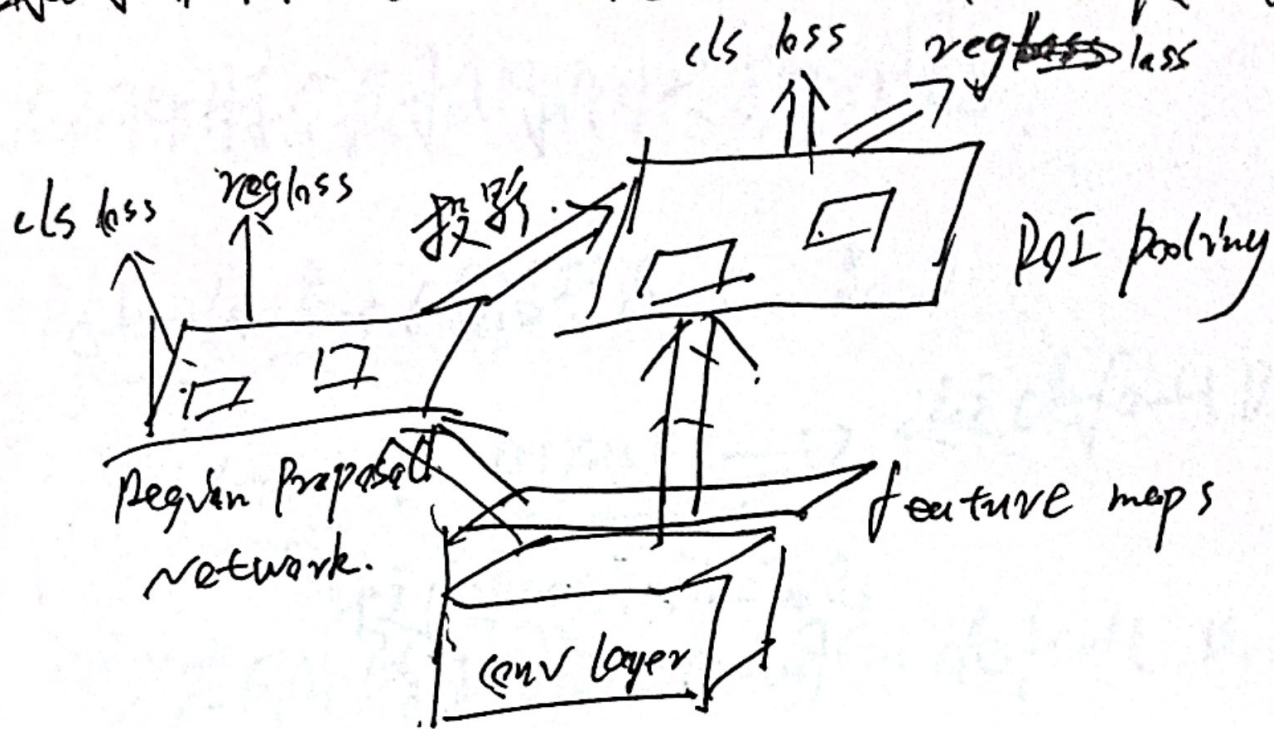
$$L_{reg}(t_i, t_i^*) = \sum_i \text{Smooth}_L(t_i - t_i^*)$$

$$t_i = [t_x, t_y, t_w, t_h]$$

$$t_i^* = [t_x^*, t_y^*, t_w^*, t_h^*]$$



直接采用 RPN loss + Fast RCNN loss 联合训练



Fast RCNN 存在的问题:

1. 对小目标检测效果差
2. 模型大, 训练速度慢.