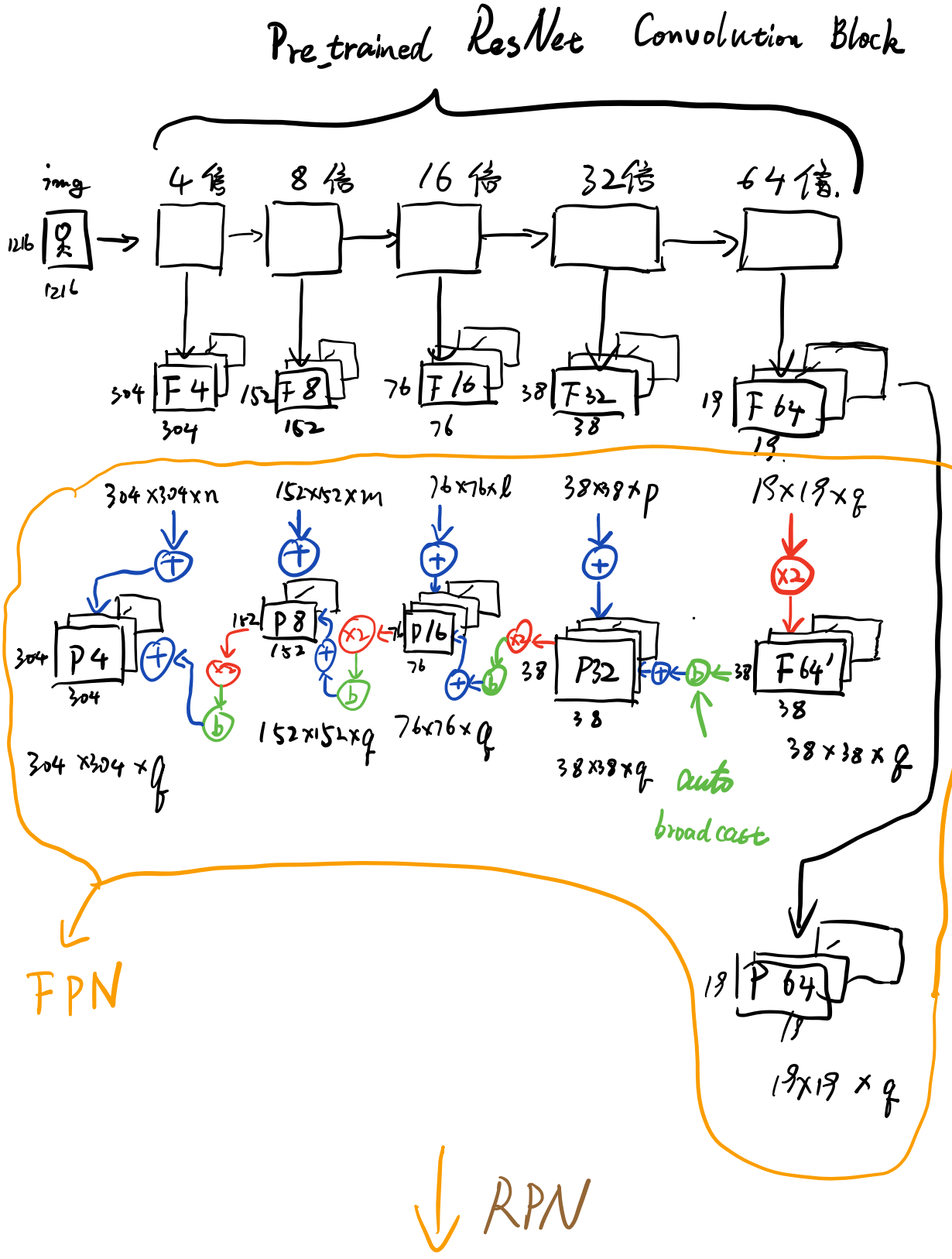
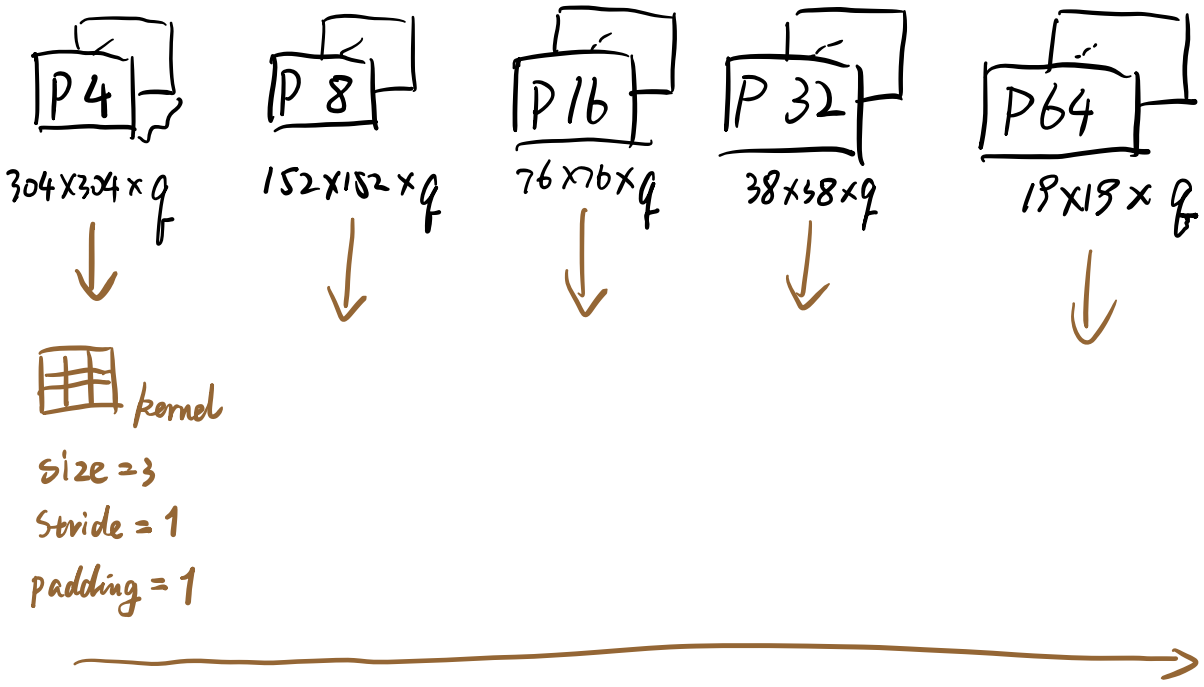

Pre_trained ResNet Convolution Block
img
4倍
8倍
16倍
32倍
64倍
1216
1216
304
304
F4
152
152
F8
76
76
F16
38
38
F32
19
19
F64
304x304xn
152x152xm
76x76xl
38x38xp
19x19xq
P4
P8
P16
P32
F64'
304x304xq
152x152xq
76x76xq
38x38xq
38x38xq
19x19xq
x2
auto
broad cast
FPN
RPN



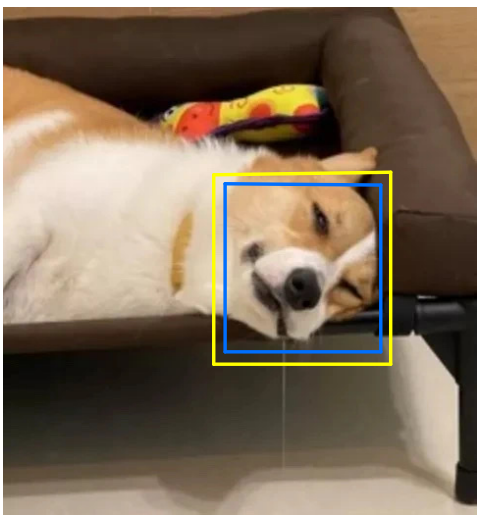
感受野从小到大，

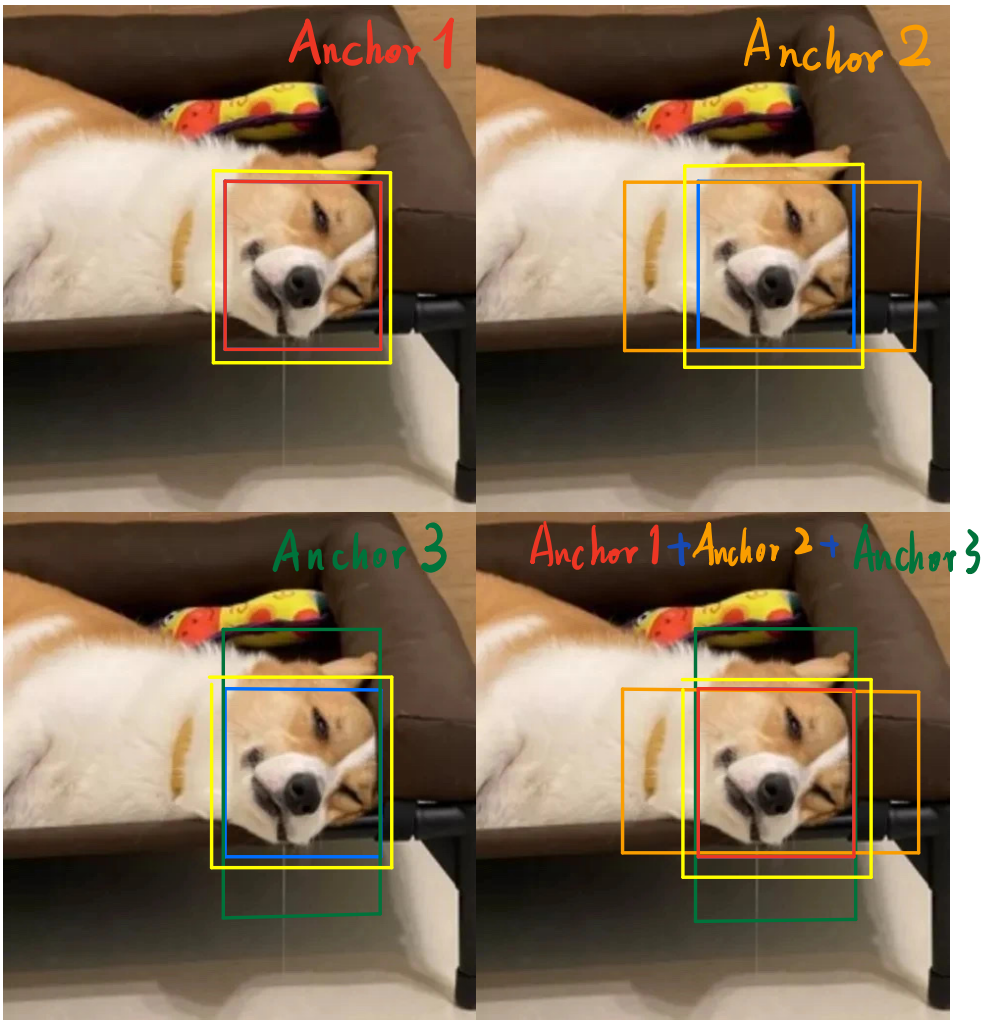
在特征图中每个像素对应的原图中的感受野上

生成 3 个 anchor 比例为 1:1，1:2，2:1

设蓝框为感受野

黄框为 Ground Truth



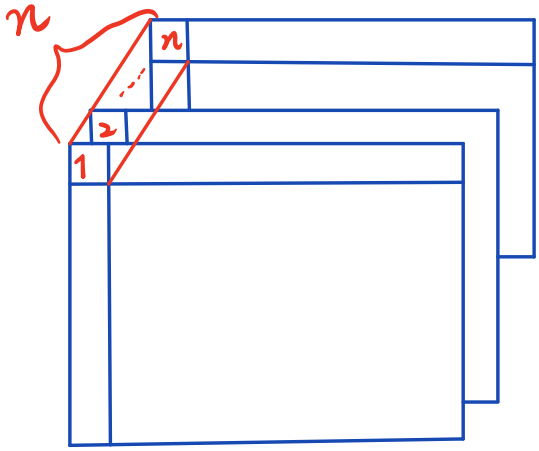


对于P4 304×304×9.

304×304个像素 × 3个anchor.

304×304×3 = 277248个anchor.

使用P中每个像素对应的长度为n的向量V

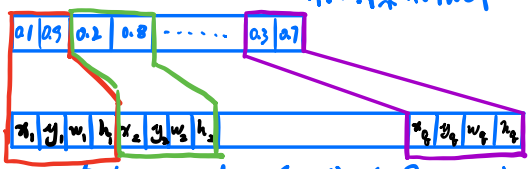


对应向量V

V = | 1 | 2 | - - - - - - - - - - - - | g |

使用V计算 {
① 分类 → 输出2k个值
每个anchor中的内容是前景还是背景的概率
② 回归 → 输出4k个值.
每个anchor的中心点偏移量和长宽缩放量.

分类输出：anchor中内容属于前、背景的概率



回归输出：anchor的偏移量和缩放量.

Proposal层

①利用回归结果对所有anchor进行修正，得到修正后检测框.

②根据分类结果，按前景概率降序排序，取前6000个框.

③限定超出图像边界的框，防止后续ROI pooling超出边界

④对剩余框进行非极大值抑制

⑤输出对应输入图像尺度归一化后的坐标值 $[x_1, y_1, x_2, y_2]$

ROI pooling.

由于有 FPN 输出了多个 Feature maps

在ROI pooling 时要计算一个 k

来确定应该使用哪个 Feature map.

$$k = [k_0 + \log_2(\sqrt{wh}/224)]$$

224：图像输入尺寸

w、h：ROI区域的长宽 ≠7

k_0：基准值 =4

例

ROI =112×112.

$$k = [4 + \log_2(\frac{112}{224})]$$

$= [4 - 1]$

$= 3$

取 P_3 层.

k要取整，且在2-5间，超出就截断