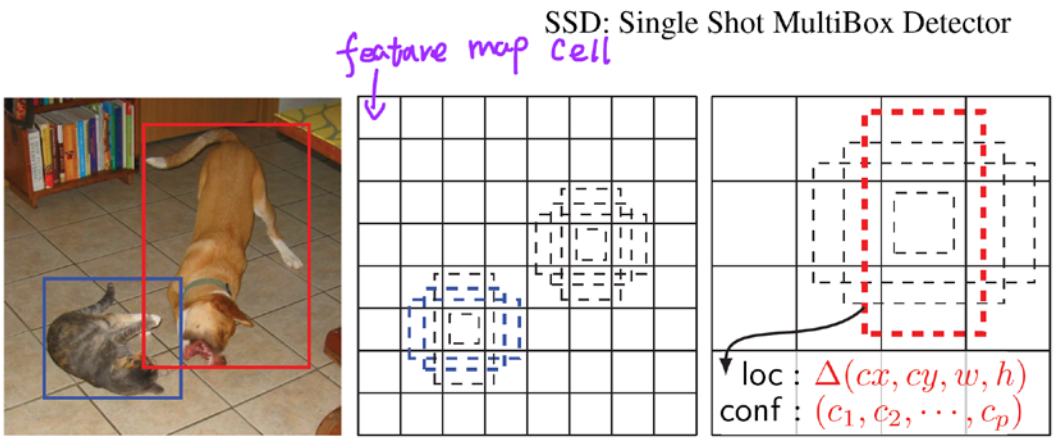
SSD Notes

1. 概述

针对不同大小的目标检测，传统的做法是先将图像转换成不同大小(图像金字塔)，然后分别检测，最后将结果综合起来(NMS)。而 SSD 算法则利用不同卷积层的 feature map 进行综合也能达到同样的效果。算法的基础网络结构是 VGG16，并将最后两个全连接层改成卷积层，并随后增加了 4 个卷积层来构造网络结构。对其中 5 种不同的卷积层的输出（feature map）分别用两个不同的 3×3 卷积核进行卷积，一个输出分类用的 confidence，每个 default box 生成 21 个类别 confidence；一个输出回归用的 localization，每个 default box 生成 4 个坐标值（x, y, w, h）。上述 5 个 feature map 中每一层 default box 的数量是给定的，最后共生成 8732 个 default box。

2. 一些概念



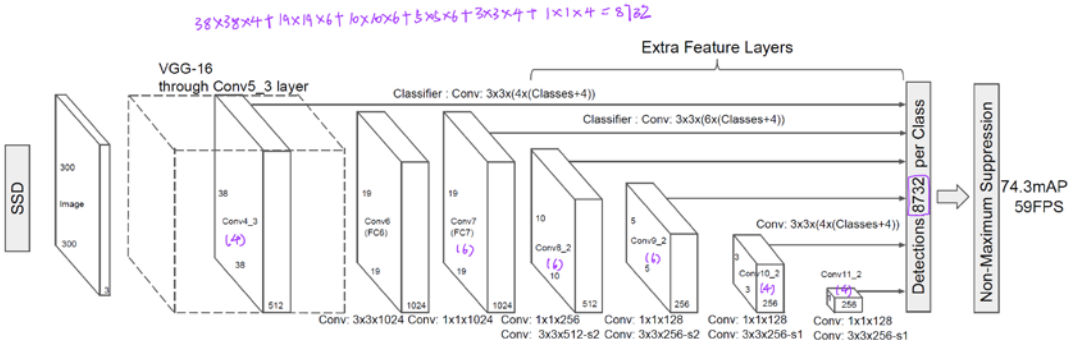
(a) Image with GT boxes (b) 8×8 feature map (c) 4×4 feature map

上图中（a）是画上 GT boxes 的原始图片，（b）是网络结构中的一个 8*8 大小的特征图，（c）是网络结构中的一个 4*4 大小的特征图。

第一个概念是 **feature map cell**，是指 feature map 中每一个小格子，如图中分别有 64 和 16 个 cell。另外有一个概念：**default box**，是指在 feature map 的每个小格(cell)上都有一系列固定大小的 box，如上图中有 4 个（虚线框，仔细看格子的中间有比格子还小的一个 box）。训练中还有一个东西：**prior box**，是指实际中选择的 default box（每一个 feature map cell 不是 k 个 default box 都取）。也就是说 default box 是一种概念，prior box 则是实际的选取。

训练中一张完整的图片送进网络获得各个 feature map，对于正样本训练来说，需要先将 prior box 与 ground truth box 做匹配，匹配成功说明这个 prior box 所包含的是个目标，但离完整目标的 ground truth box 还有段距离，训练的目的是保证 default box 的分类 confidence 的同时将 prior box 尽可能回归到 ground truth box。作者的实验表明 default box 的 shape 数量越多，效果越好。

3. 网络结构



4. Default Boxes 的 scale 和 aspect ratios

假设有 m 个特征图:

$$s_k = s_{\min} + \frac{s_{\max} - s_{\min}}{m-1}(k-1), \quad k \in [1, m]$$

s_{min}=0.2　s_{max}=0.9　s_k 在 0.2~0.9 之间

$a_r \in \{1, 2, 3, 1/2, 1/3\}$

width $(w_k^a = s_k\sqrt{a_r})$　height $(h_k^a = s_k/\sqrt{a_r})$

当 aspect ratio=1 时，增加一个 default box，将 scale 设置成

$$s'_k = \sqrt{s_k s_{k+1}}$$

因此，对于每个 feature map cell 而言，一共有 6 种 default box。可以看出这些 default box 在不同层的 feature map 上有不同的 scale，在同一层的 feature map 上又有不同的 aspect ratio，因此基本上可以覆盖输入图像中的各种形状和大小的 object!

5. 难例挖掘

将分类损失排序，从最高的开始选，使正负样本比例保持在 1:3

6. 训练过程

$$L(x, c, l, g) = \frac{1}{N}(L_{conf}(x, c) + \alpha L_{loc}(x, l, g))$$

N是match成功的default box数

$$L_{loc}(x, l, g) = \sum_{i \in Pos}^{N} \sum_{m \in \{cx, cy, w, h\}} x_{ij}^k \text{smooth}_{\text{L1}}(l_i^m - \hat{g}_j^m)$$

$$\hat{g}_j^{cx} = (g_j^{cx} - d_i^{cx})/d_i^w \qquad \hat{g}_j^{cy} = (g_j^{cy} - d_i^{cy})/d_i^h$$

$$\hat{g}_j^w = \log\left(\frac{g_j^w}{d_i^w}\right) \qquad \hat{g}_j^h = \log\left(\frac{g_j^h}{d_i^h}\right)$$

$$L_{conf}(x,c) = -\sum_{i\in Pos}^{N} x_{ij}^{p} log(\hat{c}_i^p) - \sum_{i\in Neg} log(\hat{c}_i^0) \quad \text{where} \quad \hat{c}_i^p = \frac{\exp(c_i^p)}{\sum_p \exp(c_i^p)}$$

L_{loc}只有在正例上累加损失，L_{conf}在正例和负例上都累加损失