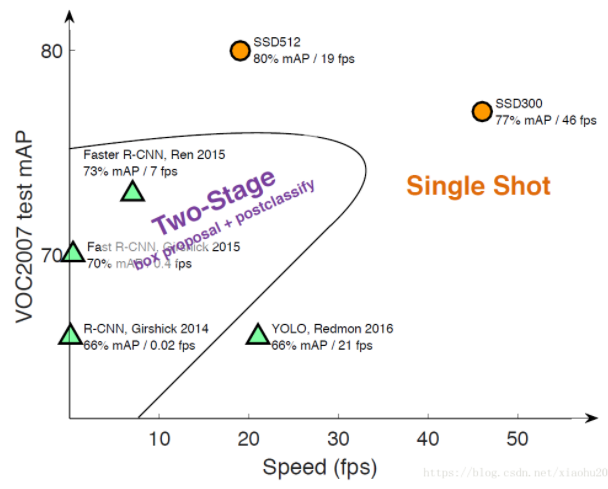
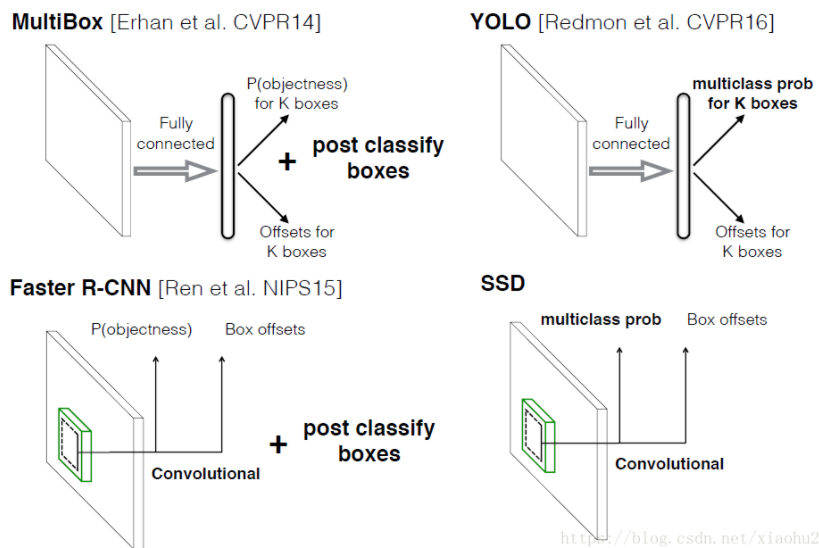


从上面可以大致的看出，Faster-RCNN的准确度更加精确，而RFCN和SSD更快。

（1）two-stage方法，如R-CNN系算法，其主要思路是先通过启发式方法（selective search）或者CNN网络（RPN)产生一系列稀疏的候选框，然后对这些候选框进行分类与回归，two-stage方法的优势是准确度高；（2）one-stage方法，如Yolo和SSD，其主要思路是均匀地在图片的不同位置进行密集抽样，抽样时可以采用不同尺度和长宽比，然后利用CNN提取特征后直接进行分类与回归，整个过程只需要一步，所以其优势是速度快，但是均匀的密集采样的一个重要缺点是训练比较困难，这主要是因为正样本与负样本（背景）极其不均衡（参见Focal Loss），导致模型准确度稍低。不同算法的性能如图1所示，可以看到两类方法在准确度和速度上的差异。

RCNN->SppNET->Fast-RCNN->Faster-RCNN

《Rich feature hierarchies for accurate oject detection and semantic segmentation》

针对高准确度的目标检测与语义分割的多特征层级 2014

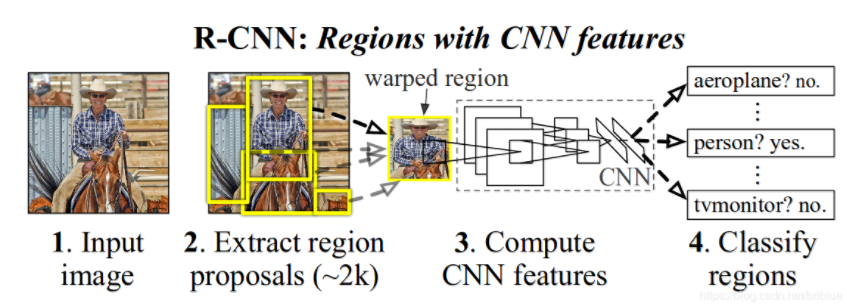
1. 在 Pascal VOC 2012 的数据集上，能够将目标检测的验证指标 mAP 提升到 53.3%,这相对于之前最好的结果提升了整整 30%.

2. 这篇论文证明了可以讲神经网络应用在自底向上的候选区域，这样就可以进行目标分类和目标定位。

3. 这篇论文也带来了一个观点，那就是当你缺乏大量的标注数据时，比较好的可行的手段是，进行神经网络的迁移学习，采用在其他大型数据集训练过后的神经网络，然后在小规模特定的数据集中进行 fine-tune 微调。

在过去的十多年时间里，传统的机器视觉领域，通常采用特征描述子来应对目标识别任务，这些特征描述子最常见的就是 SIFT 和 HOG.而 OpenCV 有现成的 API 可供大家实现相关的操作。SIFT 和 HOG 的王者地位最近被卷积神经网络撼动。

1. 给定一张输入图片，从图片中提取 2000 个类别独立的候选区域。
2. 对于每个区域利用 CNN 抽取一个固定长度的特征向量。
3. 再对每个区域利用 SVM 进行目标分类。



迁移学习

针对每个类，通过计算 IoU 指标，采取**非极大性抑制**

**bbox 回归**

bbox 的值其实就是物体方框的位置，预测它就是回归问题，而不是分类问题。

受 DPM 的启发，作者训练了一个线性的回归模型，这个模型能够针对候选区域的 pool5 数据预测一个新的 box 位置。具体细节，作者放在补充材料当中

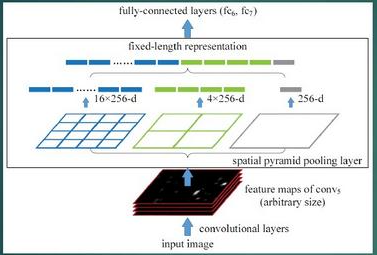
利用 CPMC 生成候选区域，然后将这些区域调整大小为 227x227,送到神经网络当中，这是 full 阶段，区域中有背景也有前景

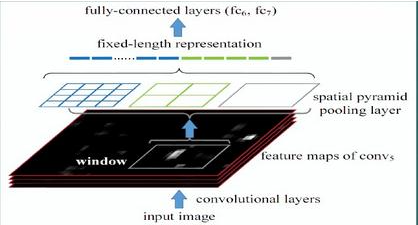
**SPP Net**

Spatial Pyramid Pooling（空间金字塔池化）

一般CNN后接全连接层或者分类器，他们都需要固定的输入尺寸，因此不得不对输入数据进行crop或者warp，这些预处理会造成数据的丢失或几何的失真。SPP Net的第一个贡献就是将金字塔思想加入到CNN，实现了数据的多尺度输入。

如下图所示，在卷积层和全连接层之间加入了SPP layer。此时网络的输入可以是任意尺度的，在SPP layer中每一个pooling的filter会根据输入调整大小，而SPP的输出尺度始终是固定的。



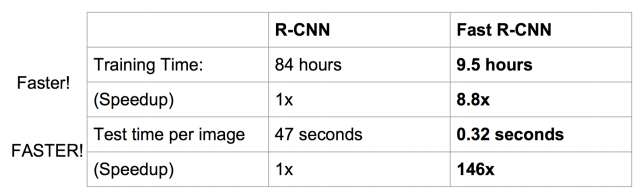
2.只对原图提取一次卷积特征  
在R-CNN中，每个候选框先resize到统一大小，然后分别作为CNN的输入，这样是很低效的。  
所以SPP Net根据这个缺点做了优化：只对原图进行一次卷积得到整张图的feature map，然后找到每个候选框zaifeature map上的映射patch，将此patch作为每个候选框的卷积特征输入到SPP layer和之后的层。节省了大量的计算时间，比R-CNN有一百倍左右的提速。

# Fast-RCNN

先说RCNN的缺点：即使使用了selective search等预处理步骤来提取潜在的bounding box作为输入，但是RCNN仍会有严重的速度瓶颈，原因也很明显，就是计算机对所有region进行特征提取时会有重复计算，Fast-RCNN正是为了解决这个问题诞生的。

大牛提出了一个可以看做单层sppnet的网络层，叫做ROI Pooling，这个网络层可以把不同大小的输入映射到一个固定尺度的特征向量，而我们知道，conv、pooling、relu等操作都不需要固定size的输入，因此，在原始图片上执行这些操作后，虽然输入图片size不同导致得到的feature map尺寸也不同，不能直接接到一个全连接层进行分类，但是可以加入这个神奇的ROI Pooling层，对每个region都提取一个固定维度的特征表示，再通过正常的softmax进行类型识别。

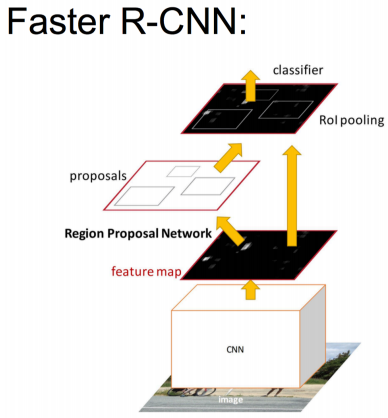
，之前RCNN的处理流程是先提proposal，然后CNN提取特征，之后用SVM分类器，最后再做bbox regression，而在Fast-RCNN中，作者巧妙的把bbox regression放进了神经网络内部，与region分类和并成为了一个multi-task模型，实际实验也证明，这两个任务能够共享卷积特征，并相互促进。Fast-RCNN很重要的一个贡献是成功的让人们看到了Region Proposal+CNN这一框架实时检测的希望，原来多类检测真的可以在保证准确率的同时提升处理速度，也为后来的Faster-RCNN做下了铺垫。



**Faster R-CNN**

Fast R-CNN存在的问题：存在瓶颈：选择性搜索，找出所有的候选框，这个也非常耗时。那我们能不能找出一个更加高效的方法来求出这些候选框呢？  
解决：加入一个提取边缘的神经网络，也就说找到候选框的工作也交给神经网络来做了。  
做这样的任务的神经网络叫做Region Proposal Network(RPN)。

具体做法：  
　　• 将RPN放在最后一个卷积层的后面  
　　• RPN直接训练得到候选区域



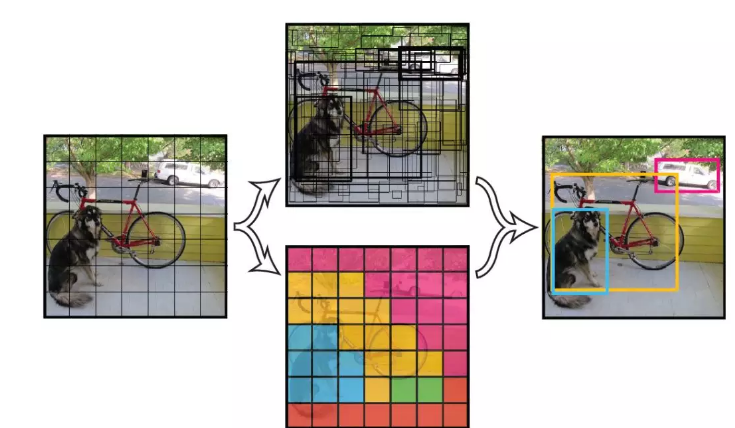
一种网络，四个损失函数;  
　　• RPN calssification(anchor good.bad)  
　　• RPN regression(anchor->propoasal)  
　　• Fast R-CNN classification(over classes)  
　　• Fast R-CNN regression(proposal ->box)

Faster R-CNN的主要贡献是设计了提取候选区域的网络RPN，代替了费时的选择性搜索，使得检测速度大幅提高。

YOLO

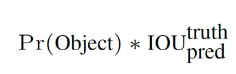
YOLO 的核心思想就是利用整张图作为网络的输入，直接在输出层回归 bounding box（边界框） 的位置及其所属的类别。

将一幅图像分成 SxS 个网格（grid cell），如果某个 object 的中心落在这个网格中，则这个网格就负责预测这个 object。



每个网格要预测 B 个 bounding box，每个 bounding box 除了要回归自身的位置之外，还要附带预测一个 confidence 值。

这个 confidence 代表了所预测的 box 中含有 object 的置信度和这个 box 预测的有多准这两重信息，其值是这样计算的：



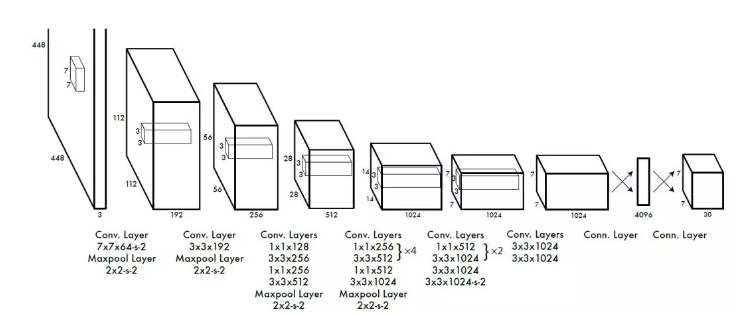
其中如果有 object 落在一个 grid cell 里，第一项取 1，否则取 0。 第二项是预测的 bounding box 和实际的 groundtruth 之间的 IoU 值。

每个 bounding box 要预测 (x, y, w, h) 和 confidence 共5个值，每个网格还要预测一个类别信息，记为 C 类。则 SxS个 网格，每个网格要预测 B 个 bounding box 还要预测 C 个 categories。输出就是 S x S x (5\*B+C) 的一个 tensor。

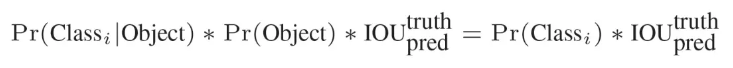
注意：class 信息是针对每个网格的，confidence 信息是针对每个 bounding box 的。

举例说明: 在 PASCAL VOC 中，图像输入为 448x448，取 S=7，B=2，一共有20 个类别（C=20），则输出就是 7x7x30 的一个 tensor。

整个网络结构如下图所示



在 test 的时候，每个网格预测的 class 信息和 bounding box 预测的 confidence信息相乘，就得到每个 bounding box 的 class-specific confidence score:



等式左边第一项就是每个网格预测的类别信息，第二、三项就是每个 bounding box 预测的 confidence。这个乘积即 encode 了预测的 box 属于某一类的概率，也有该 box 准确度的信息。

得到每个 box 的 class-specific confidence score 以后，设置阈值，滤掉得分低的 boxes，对保留的 boxes 进行 NMS 处理，就得到最终的检测结果。

**YOLO 的缺点**

YOLO 对相互靠的很近的物体，还有很小的群体检测效果不好，这是因为一个网格中只预测了两个框，并且只属于一类。

同一类物体出现的新的不常见的长宽比和其他情况时，泛化能力偏弱。

由于损失函数的问题，定位误差是影响检测效果的主要原因。尤其是大小物体的处理上，还有待加强。

YOLOv2：代表着目前业界最先进物体检测的水平，它的速度要快过其他检测系统（FasterR-CNN，ResNet，SSD），使用者可以在它的速度与精确度之间进行权衡。

YOLO9000：这一网络结构可以实时地检测超过 9000 种物体分类，这归功于它使用了 WordTree，通过 WordTree 来混合检测数据集与识别数据集之中的数据。

联合训练算法的基本思路就是：同时在检测数据集和分类数据集上训练物体检测器（Object Detectors ），用监测数据集的数据学习物体的准确位置，用分类数据集的数据来增加分类的类别量、提升鲁棒性。

**Batch Normalization**

使用 Batch Normalization 对网络进行优化，让网络提高了收敛性，同时还消除了对其他形式的正则化（regularization）的依赖。通过对 YOLO 的每一个卷积层增加 Batch Normalization，最终使得 mAP 提高了 2%，同时还使模型正则化。使用 Batch Normalization 可以从模型中去掉 Dropout，而不会产生过拟合。

**High resolution classifier**

目前业界标准的检测方法，都要先把分类器（classiﬁer）放在ImageNet上进行预训练。从 Alexnet 开始，大多数的分类器都运行在小于 256\*256 的图片上。而现在 YOLO 从 224\*224 增加到了 448\*448，这就意味着网络需要适应新的输入分辨率。

为了适应新的分辨率，YOLO v2 的分类网络以 448\*448 的分辨率先在 ImageNet上进行微调，微调 10 个 epochs，让网络有时间调整滤波器（filters），好让其能更好的运行在新分辨率上，还需要调优用于检测的 Resulting Network。最终通过使用高分辨率，mAP 提升了 4%。

**Convolution with anchor boxes**

YOLO 一代包含有全连接层，从而能直接预测 Bounding Boxes 的坐标值。  Faster R-CNN 的方法只用卷积层与 Region Proposal Network 来预测 Anchor Box 偏移值与置信度，而不是直接预测坐标值。作者发现通过预测偏移量而不是坐标值能够简化问题，让神经网络学习起来更容易。

所以最终 YOLO 去掉了全连接层，使用 Anchor Boxes 来预测 Bounding Boxes。作者去掉了网络中一个池化层，这让卷积层的输出能有更高的分辨率。收缩网络让其运行在 416\*416 而不是 448\*448。由于图片中的物体都倾向于出现在图片的中心位置，特别是那种比较大的物体，所以有一个单独位于物体中心的位置用于预测这些物体。YOLO 的卷积层采用 32 这个值来下采样图片，所以通过选择 416\*416 用作输入尺寸最终能输出一个 13\*13 的特征图。 使用 Anchor Box 会让精确度稍微下降，但用了它能让 YOLO 能预测出大于一千个框，同时 recall 达到88%，mAP 达到 69.2%。

**Dimension clusters**

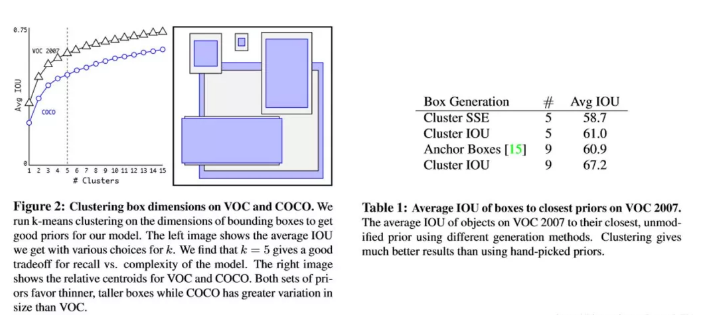
之前 Anchor Box 的尺寸是手动选择的，所以尺寸还有优化的余地。 为了优化，在训练集的 Bounding Boxes 上跑一下 k-means聚类，来找到一个比较好的值。

如果我们用标准的欧式距离的 k-means，尺寸大的框比小框产生更多的错误。因为我们的目的是提高 IOU 分数，这依赖于 Box 的大小，所以距离度量的使用：



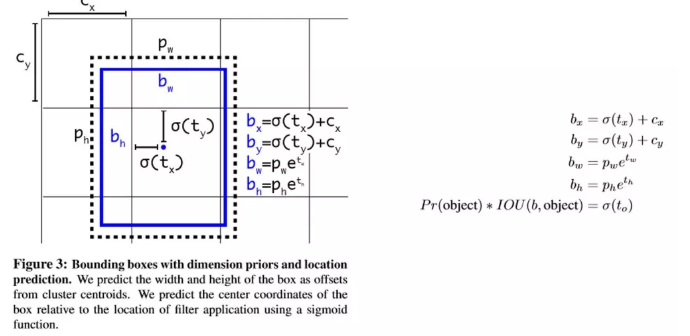
通过分析实验结果（Figure 2），左图：在模型复杂性与 high recall 之间权衡之后，选择聚类分类数 K=5。右图：是聚类的中心，大多数是高瘦的 Box。

Table1 是说明用 K-means 选择 Anchor Boxes 时，当 Cluster IOU 选择值为 5 时，AVG IOU 的值是 61，这个值要比不用聚类的方法的 60.9 要高。选择值为 9 的时候，AVG IOU 更有显著提高。总之就是说明用聚类的方法是有效果的。



**Direct location prediction**

用 Anchor Box 的方法，会让 model 变得不稳定，尤其是在最开始的几次迭代的时候。大多数不稳定因素产生自预测 Box 的（x,y）位置的时候。按照之前 YOLO的方法，网络不会预测偏移量，而是根据 YOLO 中的网格单元的位置来预测坐标，这就让 Ground Truth 的值介于 0 到 1 之间。而为了让网络的结果能落在这一范围内，网络使用一个 Logistic Activation 来对于网络预测结果进行限制，让结果介于 0 到 1 之间。 网络在每一个网格单元中预测出 5 个 Bounding Boxes，每个 Bounding Boxes 有五个坐标值 tx，ty，tw，th，t0，他们的关系见下图（Figure3）。假设一个网格单元对于图片左上角的偏移量是 cx、cy，Bounding Boxes Prior 的宽度和高度是 pw、ph，那么预测的结果见下图右面的公式：



因为使用了限制让数值变得参数化，也让网络更容易学习、更稳定。Dimension clusters和Direct location prediction，使 YOLO 比其他使用 Anchor Box 的版本提高了近5％。

**Fine-Grained Features**

YOLO 修改后的特征图大小为 13\*13，这个尺寸对检测图片中尺寸大物体来说足够了，同时使用这种细粒度的特征对定位小物体的位置可能也有好处。Faster-RCNN、SSD 都使用不同尺寸的特征图来取得不同范围的分辨率，而 YOLO 采取了不同的方法，YOLO 加上了一个 Passthrough Layer 来取得之前的某个 26\*26 分辨率的层的特征。这个 Passthrough layer 能够把高分辨率特征与低分辨率特征联系在一起，联系起来的方法是把相邻的特征堆积在不同的 Channel 之中，这一方法类似与 Resnet 的 Identity Mapping，从而把 26\*26\*512 变成 13\*13\*2048。YOLO 中的检测器位于扩展后（expanded ）的特征图的上方，所以他能取得细粒度的特征信息，这提升了 YOLO 1% 的性能。

**Multi-Scale Training**

作者希望 YOLOv2 能健壮地运行于不同尺寸的图片之上，所以把这一想法用于训练模型中。

区别于之前的补全图片的尺寸的方法，YOLOv2 每迭代几次都会改变网络参数。每 10 个 Batch，网络会随机地选择一个新的图片尺寸，由于使用了下采样参数是  32，所以不同的尺寸大小也选择为 32 的倍数 {320，352…..608}，最小 320\*320，最大 608\*608，网络会自动改变尺寸，并继续训练的过程。

这一政策让网络在不同的输入尺寸上都能达到一个很好的预测效果，同一网络能在不同分辨率上进行检测。当输入图片尺寸比较小的时候跑的比较快，输入图片尺寸比较大的时候精度高，所以你可以在 YOLOv2 的速度和精度上进行权衡。

## YOLOv3

1.多尺度预测 （类FPN）

2.更好的基础分类网络（类ResNet）和分类器 darknet-53，见下图

3.分类器-类别预测：

YOLOv3 不使用 Softmax 对每个框进行分类，主要考虑因素有：

1. Softmax 使得每个框分配一个类别（得分最高的一个），而对于 Open Images这种数据集，目标可能有重叠的类别标签，因此 Softmax不适用于多标签分类。
2. Softmax 可被独立的多个 logistic 分类器替代，且准确率不会下降。
3. 分类损失采用 binary cross-entropy loss.

**多尺度预测**

每种尺度预测 3 个 box, anchor 的设计方式仍然使用聚类，得到9个聚类中心，将其按照大小均分给 3 个尺度。

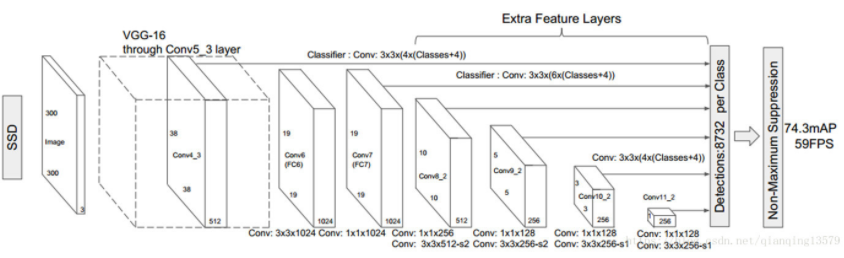
* 尺度1: 在基础网络之后添加一些卷积层再输出box信息。
* 尺度2: 从尺度1中的倒数第二层的卷积层上采样(x2)再与最后一个 16x16 大小的特征图相加，再次通过多个卷积后输出 box 信息，相比尺度1变大两倍.
* 尺度3: 与尺度2类似，使用了 32x32 大小的特征图

总之，yolo就是天生“灵活”，所以特别适合作为**工程**算法。

# SSD

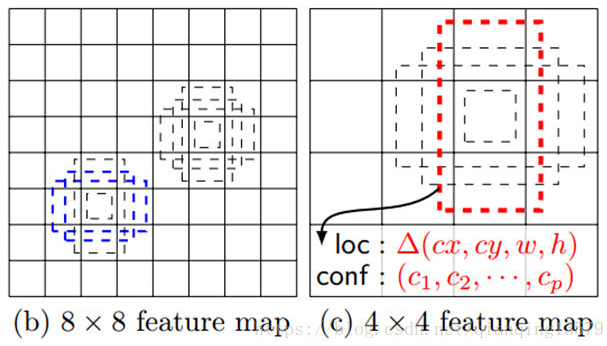
1. 多尺度   
2. 设置了多种宽高比的default box(anchor box)   
3. 数据增强

# 原因1：多尺度



由SSD的网络结构可以看出，SSD使用6个不同特征图检测不同尺度的目标。低层预测小目标，高层预测大目标。

# 原因2：设置了多种宽高比的default box

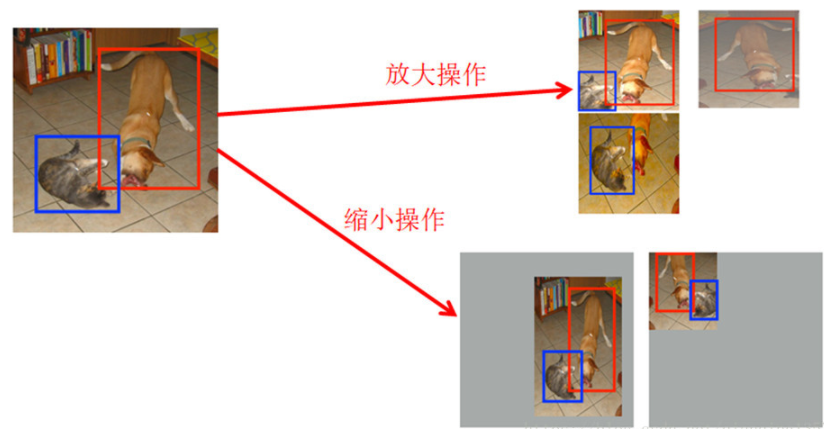


# 原因3：数据增强

SSD中使用了两种数据增强的方式

放大操作: 随机crop，patch与任意一个目标的IOU为0.1,0.3,0.5,0.7,0.9，每个patch的大小为原图大小的[0.1,1],宽高比在1/2到2之间。能够生成更多的尺度较大的目标

缩小操作: 首先创建16倍原图大小的画布，然后将原图放置其中，然后随机crop，能够生成更多尺度较小的目标



SSD主要缺点：SSD对小目标的检测效果一般，作者认为小目标在高层没有足够的信息。

**基于区域的全卷积神经网络（R-FCN）**

**1.2.5 R-FCN**

前面的目标检测方法都可以细分为两个子网络：

1. 共享的全卷积网络；
2. 不共享计算的ROI相关的子网络（比如全连接网络）。

R-FCN则将最后的全连接层之类换为了一个位置敏感的的卷积网络，从而让所有计算都可以共享。具体来说，先把每个提名区域划分为*k*×*k*个网格，比如R-FCN原文中*k*的取值为3，则对应的九个网格分别表示：左上top-left，上中top-center，……，右下bottom-right，对应图12中的九宫格及图13中的不同颜色的块，每个Grid都有对应的编码，但预测时候会有*C*+1个输出，*C*表示类别数目，+1是因为有背景类别，全部的输出通道数量为*k2*×(*C*+1)。