YOLO v2

Joseph Redmon , Ali Farhadi.

YOLO9000: Better, Faster, Stronger. CVPR 2017 (Best Paper Honorable Mention)

https://arxiv.org/abs/1612.08242

YOLO v2

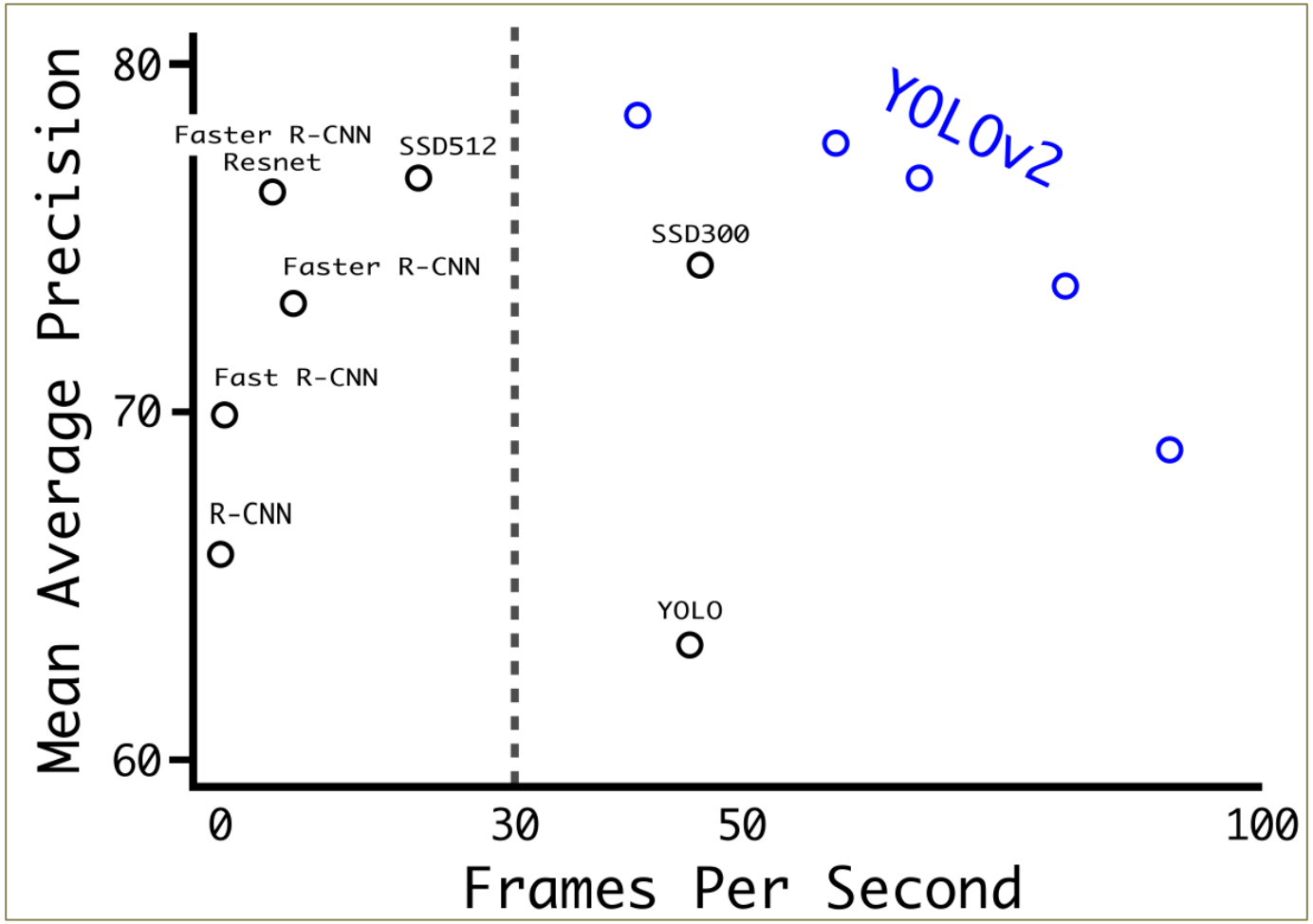
- YOLOv2是YOLO的第二个版本，其目标是在提高速度的同时显著提高准确度。

改进之处

(1) YOLO v2: 使用一系列的方法对YOLO v1进行了改进，在保持原有速度的同时提升准确度

(2) YOLO9000: 提出了一种目标分类与检测的联合训练方法，通过WordTree来混合检测数据集与识别数据集，同时在COCO和ImageNet数据集中进行训练得到YOLO9000，实现9000多种目标的实时检测。

Accuracy comparison for different detectors



Accuracy and speed on VOC 2007

Better

- 使用 批归一化(Batch Normalization) 提高准确度
- 高分辨率分类器(High-resolution classifier)
- 用锚定框的卷积(Convolutional with Anchor Boxes)
- 维度聚类(Dimension Clusters)
- 直接位置预测(Direct Location Prediction)
- 更精细的特征（Fine-Grained Features)
- 多尺度训练(Multi-Scale Training)

Faster

- Darknet-19网络模型

Stronger

- Dataset combination with WordTree
- Joint classification and detection

提高准确度（Accuracy improvements）

- 神经网络学习过程本质就是为了学习数据分布,一旦训练数据与测试数据的分布不同, 那么网络的泛化能力也大大降低; 另外一方面，一旦每批训练数据的分布各不相同, 那么网络的Batch 梯度下降算法就要在每次迭代都去学习适应不同的分布, 这样将会大大降低网络的训练速度。

- 解决办法之一是对数据都要做一个归一化预处理。YOLOv2网络通过在每一个卷积层后添加批归一化(batch normalization)，极大的改善了收敛速度同时减少了对其它正则化方法的依赖（舍弃了Dropout优化后依然没有过拟合），使得mAP获得了2%的提升。

高分辨率分类器(High-resolution classifier)

- 所有State-Of-The-Art的检测方法都在ImageNet上对分类器进行了预训练。

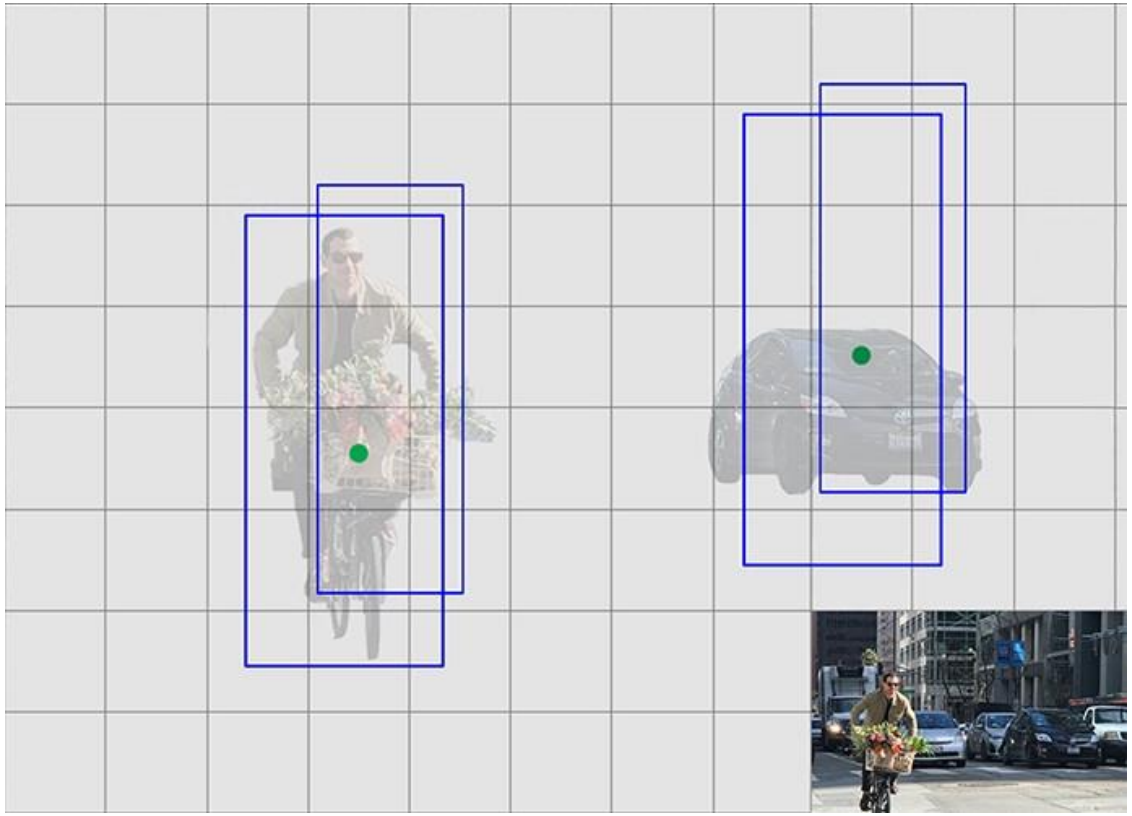
- YOLOv1训练由两个阶段组成。 首先，训练分类器网络； 然后用卷积层替换全连接层，并端到端地重新训练以进行目标检测。 YOLOv1先使用224×224的分辨率来训练分类网络，在训练检测网络的时候再切换到448×448的分辨率，这意味着YOLOv1的卷积层要重新适应新的分辨率，同时YOLOv1的网络还要学习检测网络。

- YOLOv2 以224×224图片开始用于分类器训练，但是然后使用更少的epoch再次用448×448图片重新调整分类器。让网络可以调整滤波器来适应高分辨率, 这使得检测器训练更容易。 使用高分辨率的分类网络提升了将近4%的mAP。

用锚定框的卷积（Convolutional with Anchor Boxes）

YOLO论文指出：早期训练容易受到不稳定梯度的影响。

最初，YOLO对边界框进行任意猜测。 这些猜测可能对某些目标有效，但对其它目标则很糟糕，导致陡峭的梯度变化。 在早期训练中，预测在相互争论选择什么样的特定形状合适。



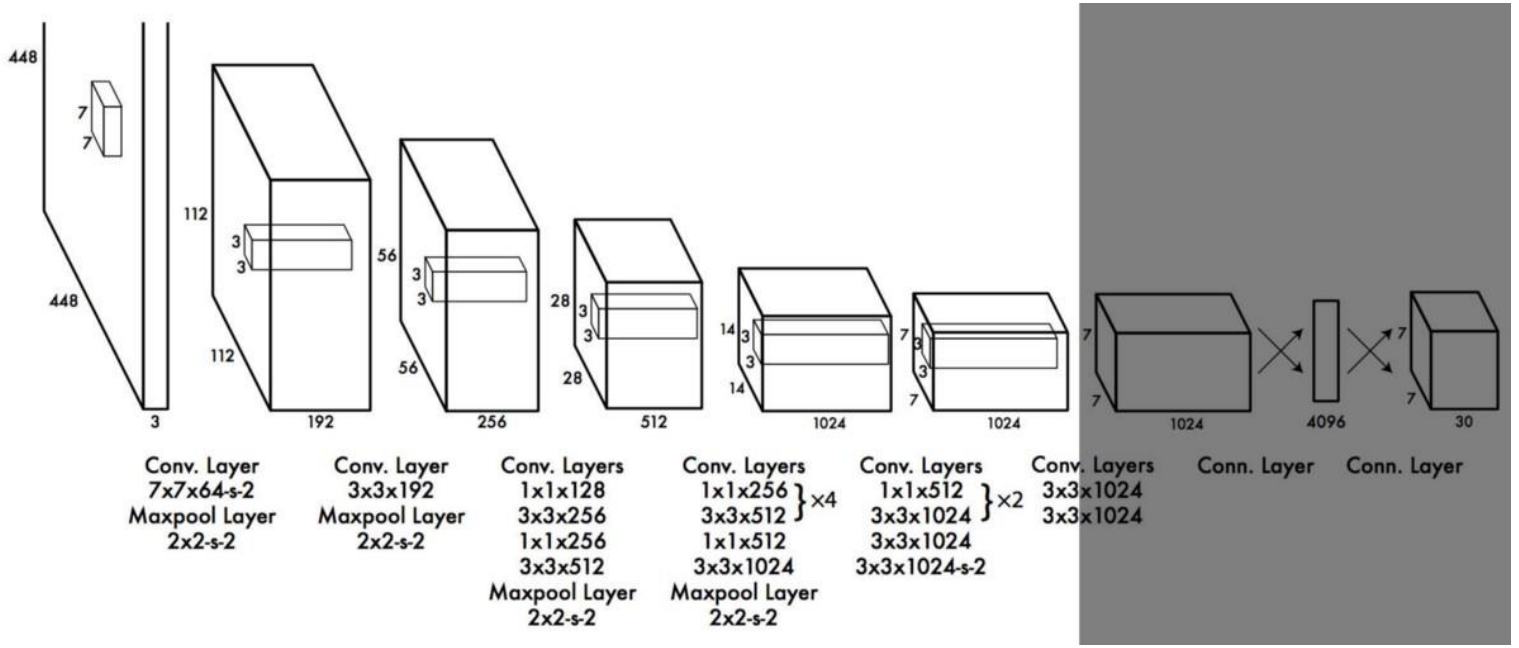
Convolutional with Anchor Boxes

- YOLOv1使用全连接层数据进行边界框预测（要把1470×1的全连接层reshape为7×7×30的最终特征），这会丢失较多的空间信息而定位不准。

- YOLOv2借鉴了Faster R-CNN中的Anchor思想：简单理解为卷积特征图上进行滑动窗采样，每个中心预测5种不同大小和比例的建议框。由于都是卷积不需要reshape，很好的保留的空间信息，最终特征图的每个特征点和原图的每个Cell一 一对应。

总的来说就是移除全连接层（以获得更多空间信息）使用锚定框来预测Bounding boxes。

Convolutional with Anchor Boxes



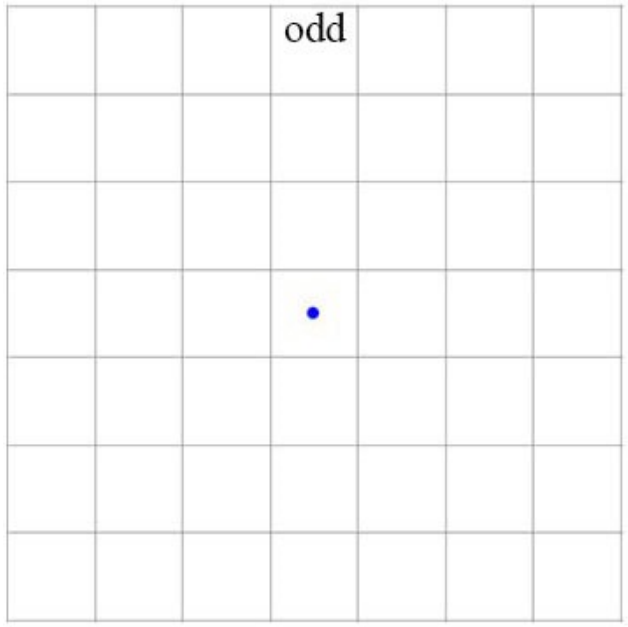
具体做法如下：

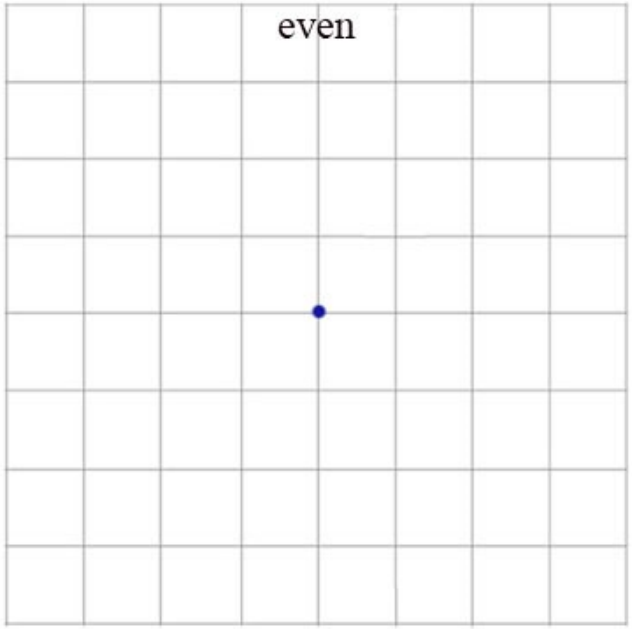
1、去掉最后的池化层确保输出的卷积特征图有更高的分辨率。

2、缩减网络，让图片输入分辨率为416×416，目的是让后面产生的卷积特征图的宽高都为奇数，这样就可以产生一个Center Cell。

3、使用卷积层降采样（factor 为32），使得输入卷积网络的416×416图片最终得到13 ×13的卷积特征图（416/32=13）。

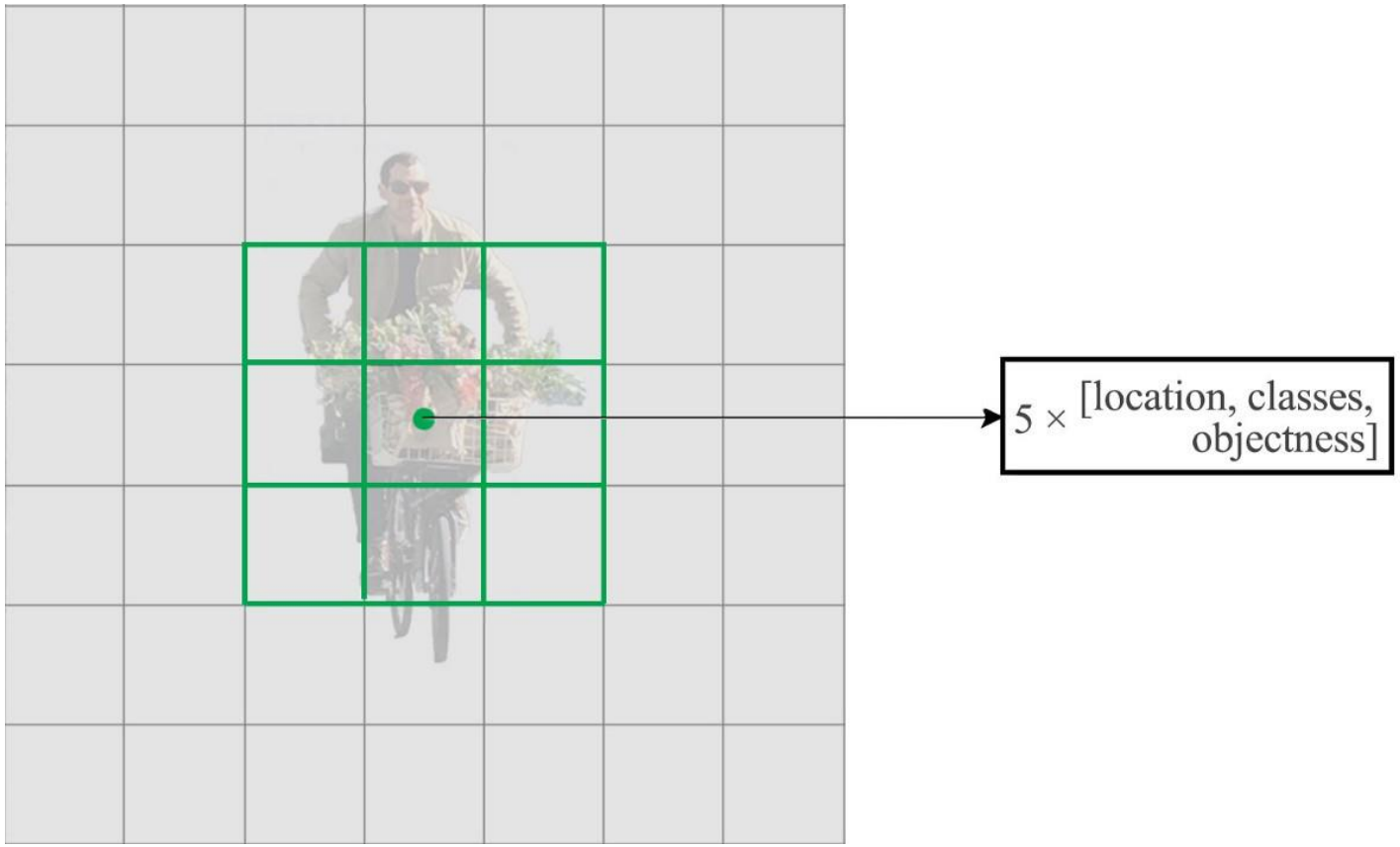
将输入图像大小从448×448更改为416×416。这将创建奇数空间维度（7×7 v.s. 8×8 grid cell）。图片的中心通常被大目标占据。 对于奇数网格单元，可以更确定目标所属的位置。

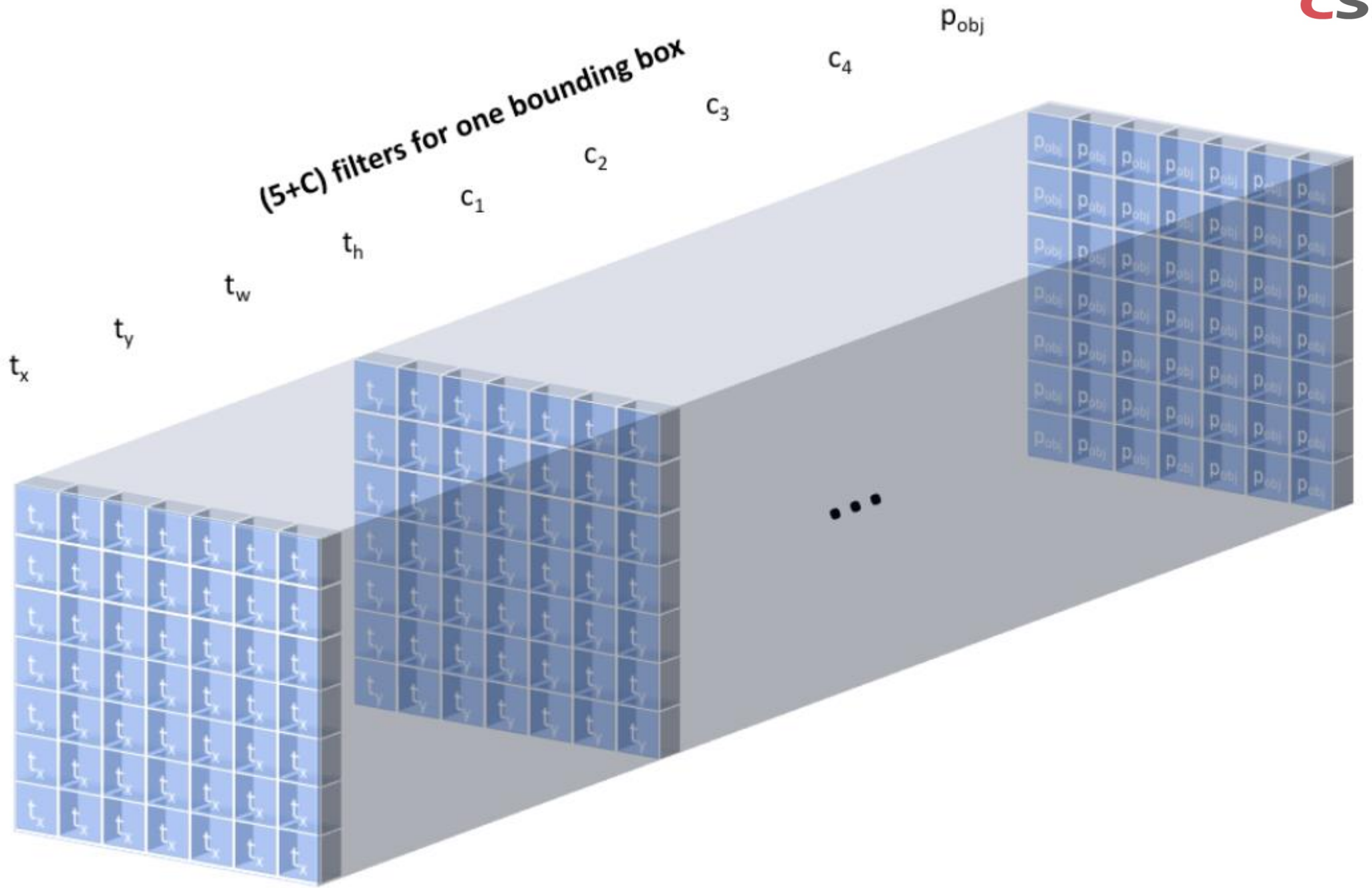


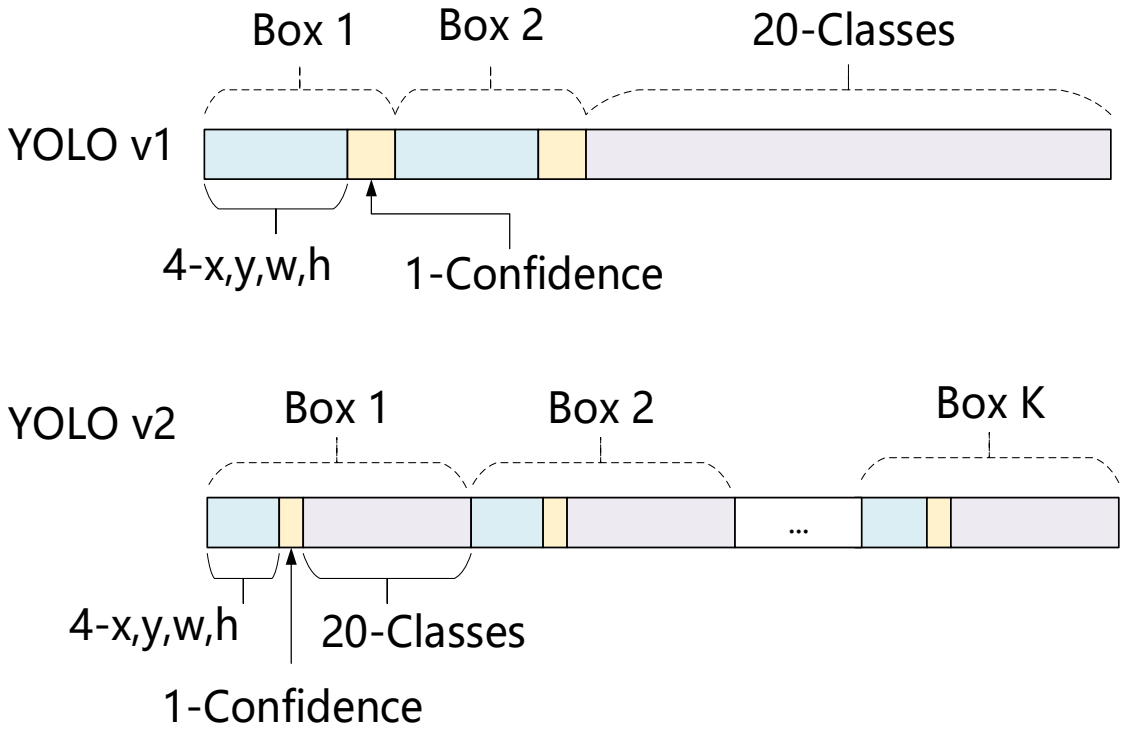


对于一些大目标，它们中心点往往落入图片中心位置，此时使用特征图的一个中心点的cell去预测这些目标的边界框相对容易些，否则就要用中间的4个Cells来进行预测。

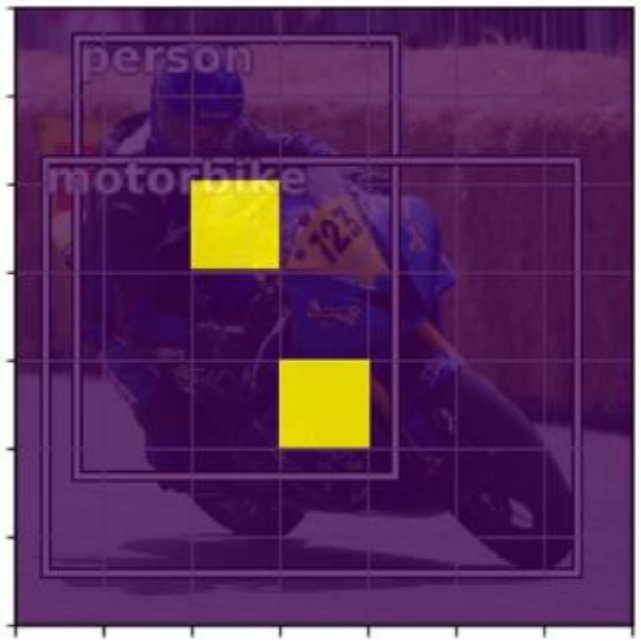
为了生成形状为7×7×125的预测，用三个3×3卷积层替换最后一个卷积层，每个有1024个输出通道。 然后用最终的1×1卷积层将7×7×1024输出转换为7×7×125 (Pascal VOC 数据集)。

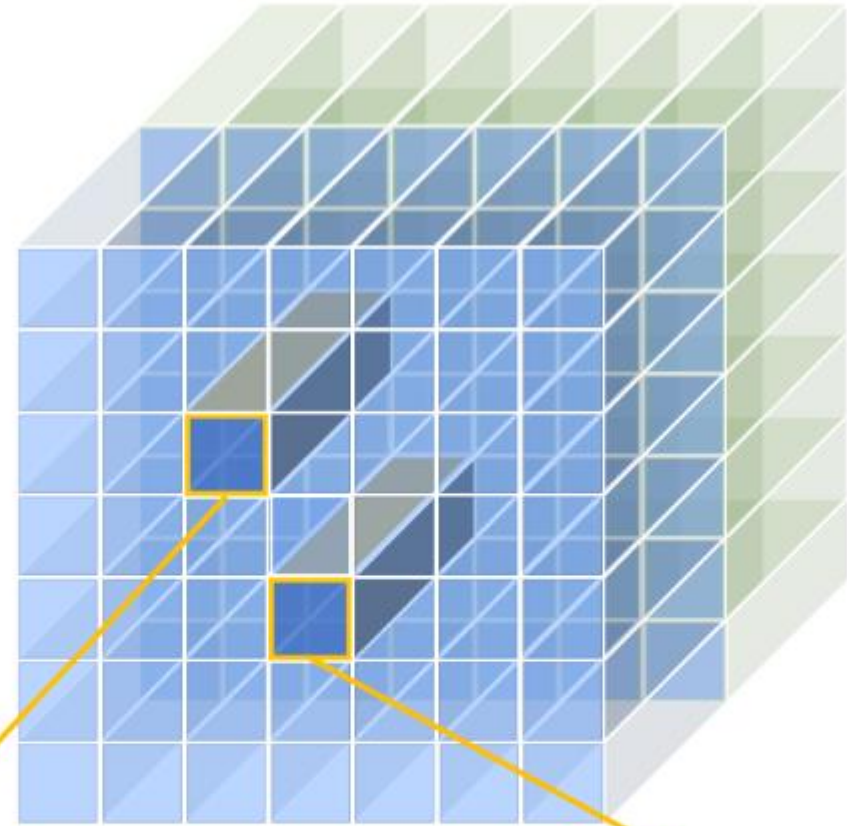



(5+C) filters for one bounding box
t_x
t_y
t_w
t_h
c_1
c_2
c_3
c_4
p_{obj}



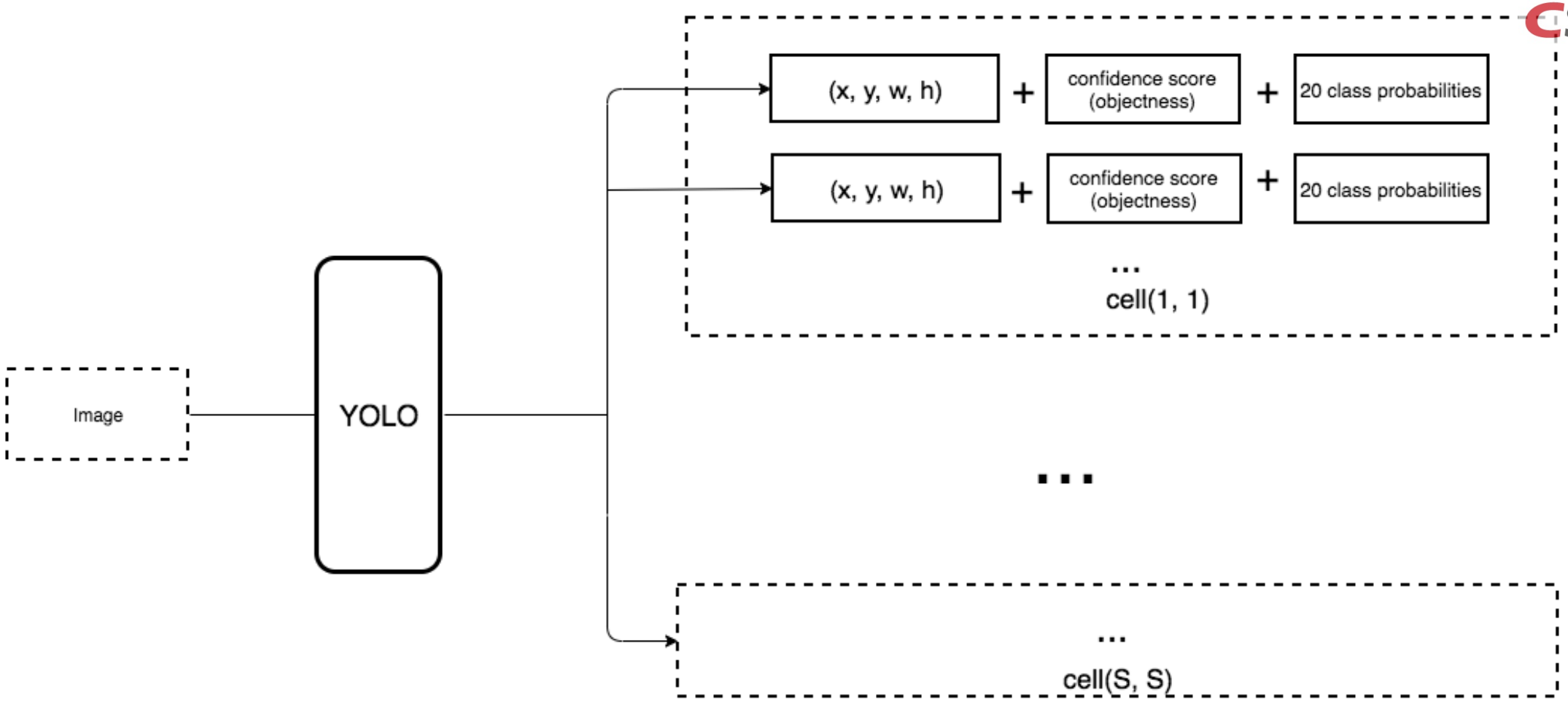
- 把预测类别的机制从空间位置（Cell）中解耦，由Anchor Box同时预测类别和坐标。
- YOLO v1是由每个Cell来负责预测类别，每个Cell对应的2个Bounding Box 负责预测坐标
- YOLOv2中，不再让类别的预测与每个Cell（空间位置）绑定一起，而是全部放到Anchor Box中。





Person bounding box descriptor

Motorbike bounding box descriptor



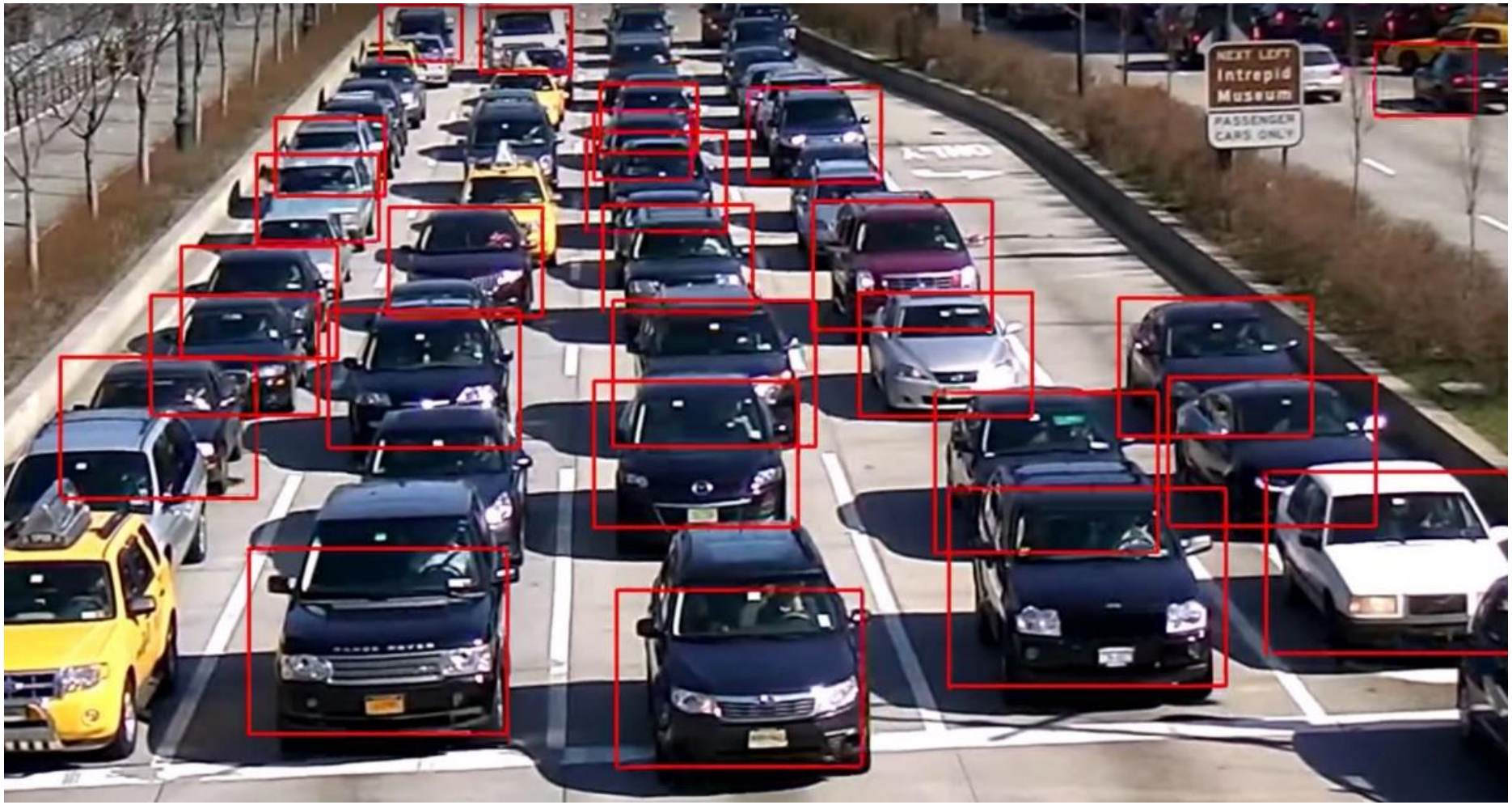
- 将类别预测从cell级别转移到边界框级别。 现在，每个预测包括边界框的4个参数，1个框置信度得分（objectness）和20个类别概率。 即具有25个参数的5个边界框：每个网格单元(grid cell) 有125个参数。

- YOLOv1只能预测98个边界框（7×7×2），而YOLOv2使用anchor boxes之后可以预测更多的边界框（13×13×5=845）。所以使用anchor boxes之后，YOLOv2的召回率大大提升，由原来的81%升至88%。

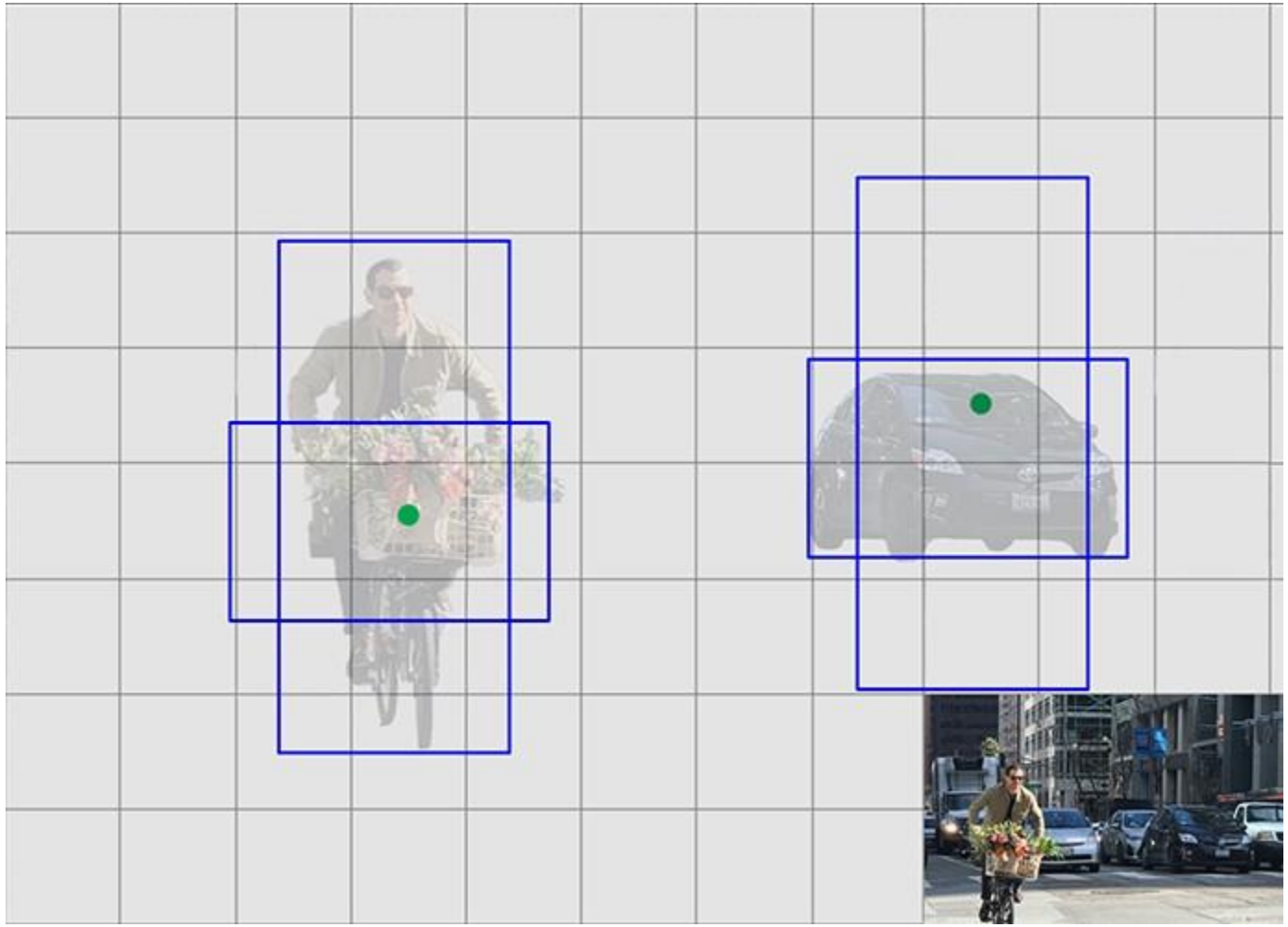
- YOLOv2的锚定框将mAP从69.5略微降低至69.2，但召回率从81%提高到88%。 准确度稍微降低，但它增加了检测所有GT 目标的机会。

现实中，边界框不是任意的

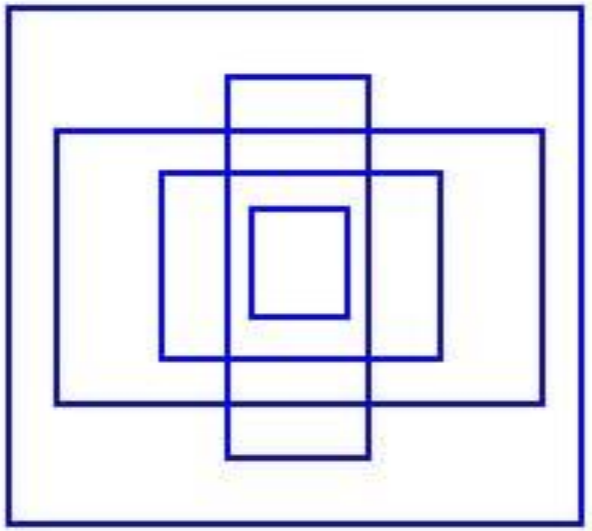
例如汽车的形状非常相似，行人的宽高比约为0.41

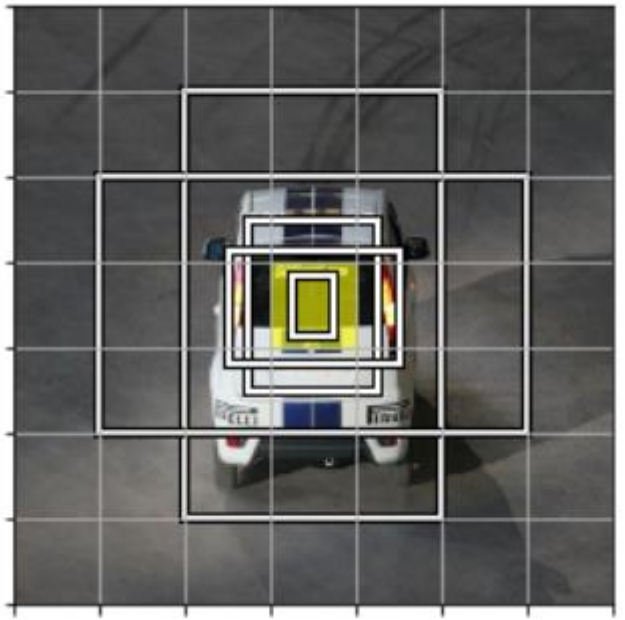


由于只需要一个猜测是正确的，如果从对现实生活对象常见的猜测开始，初始训练将更加稳定。



例如，可以创建5个具有以下形状的锚定框





- 预测上面每个锚定框的偏移量，而不是预测5个任意的边界框。
- 如果约束偏移值，可以保持预测的多样性，并使每个预测集中在特定的形状上。 初始训练将更加稳定。

最终，网络在特征图（13×13 =169）的每个Cell上预测5个边界框，每一个边界框预测5个值：t_x,t_y,t_w,t_h，t_o，其中前四个是坐标，t_o是置信度。

维度聚类(Dimension Clusters)

- 使用Anchor时，作者发现Faster-RCNN中锚定框的个数和宽高维度往往是手动挑选的先验框（Hand-Picked Priors）。设想能否一开始就选择了更好的、更有代表性的先验框维度，那么网络就应该更容易学到准确的预测位置。

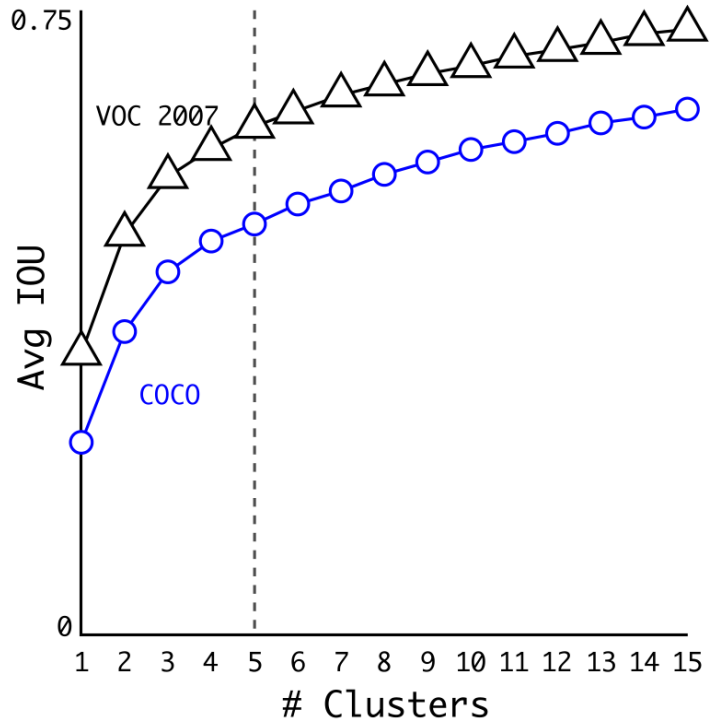
- 解决办法就是统计学习中的K-means聚类方法，通过对数据集的GT框做聚类，找到GT框的统计规律。以聚类个数k为锚定框个数，以k个聚类中心Box的宽高维度为宽高的维度。

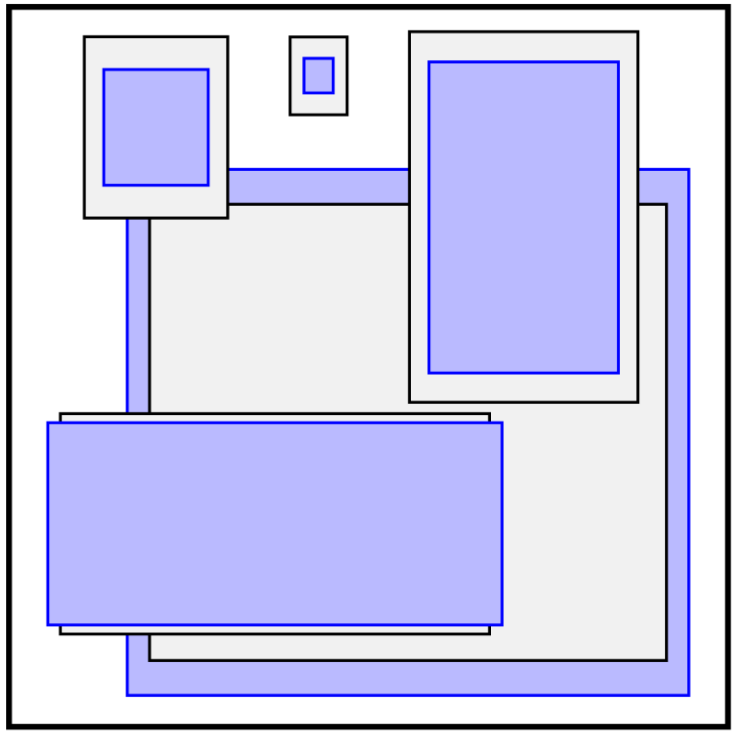
  如果按照标准K-means使用欧式距离函数，大框比小框产生更多误差。但是，我们真正想要的是使得预测框与GT框的有高的IOU得分，而与框的大小无关。因此采用了如下距离度量

$$d(\text{box}, \text{centroid}) = 1 - \text{IOU}(\text{box}, \text{centroid})$$

即聚类分析时选用box与聚类中心box之间的IOU值作为距离指标

聚类结果如下图：





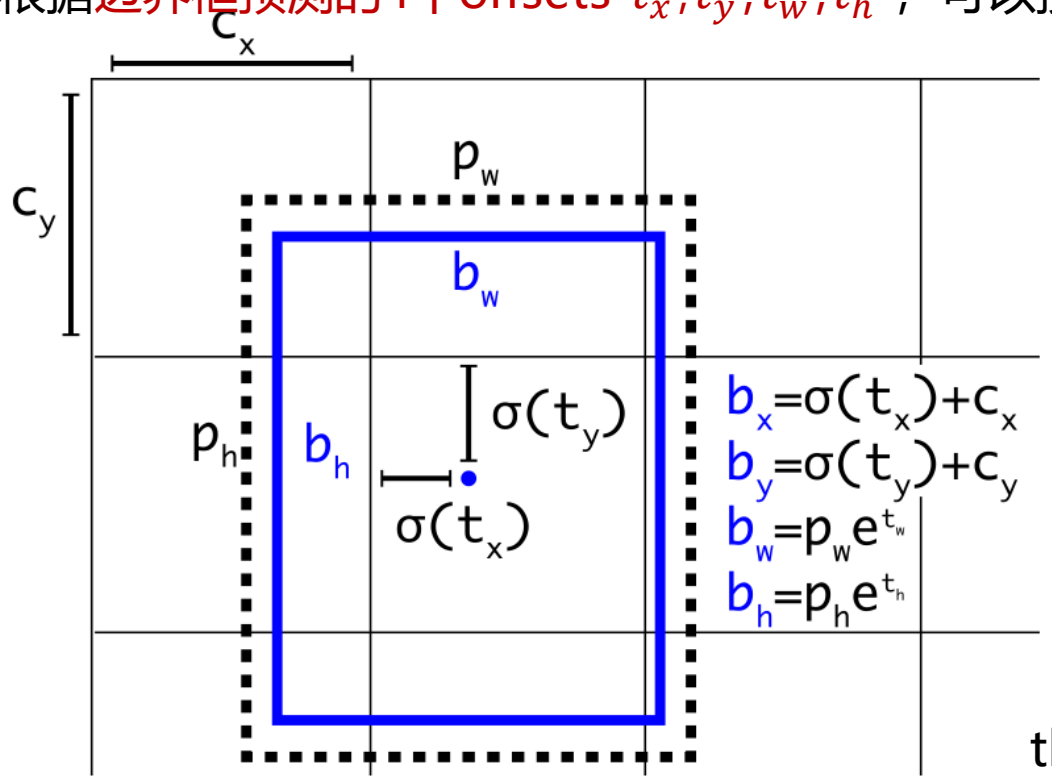
- 上面左图：随着k的增大，IOU也在增大（高召回率），但是复杂度也在增加。所以平衡复杂度和IOU之后，最终得到k值为5。
- 上面右图：5个聚类的中心与手动挑选的框是不同的，扁长的框较少，瘦高的框较多。
- K-means方法的生成的框更具有代表性，使得检测任务更容易学习。

直接位置预测(Direct Location Prediction)

- 引入Sigmoid函数预测边界框的offset，解决了anchor boxes的预测位置问题，采用了新的损失函数
- 由于预测层是卷积，所以RPN预测offset是全局性的。预测offset值而不是坐标值简化了实际问题，并且更便于网络学习。
- 作者借鉴了RPN网络使用的锚定框去预测边界框相对于图片分辨率的offset，通过(x,y,w,h)四个参数去确定锚定框的位置，但是这样在早期迭代中x,y会非常不稳定，因为RPN是一个区域预测一次，但是YOLO中是169个gird cells一起预测，处于一个gird cell 的x,y可能会跑到别的gird cell中，导致不稳定。
- 作者巧妙的应用了sigmoid函数来规约x,y的值在（0,1）轻松解决了这个offset的问题。关于w,h的也改进了YOLOv1中误差平方的计算方法，用了RPN中的log函数。

- YOLOv2沿用YOLOv1的方法，就是预测边界框中心点相对于对应cell左上角位置的相对偏移值。
- 为了将边界框中心点约束在当前cell中，使用sigmoid函数处理偏移值，这样预测的偏移值在(0,1)范围内。

总结来看，根据边界框预测的4个offsets t_x, t_y, t_w, t_h ，可以按如下公式计算出边界框实际位置（坐标值）和大小：



$$b_x = \sigma(t_x) + c_x$$

$$b_y = \sigma(t_y) + c_y$$

$$b_w = p_w e^{t_w}$$

$$b_h = p_h e^{t_h}$$

the bounding box prior has width and height p_w, p_h

使用Dimension Clusters和Direct Location Prediction这两项Anchor Boxes改进方法，mAP获得了5%的提升。

```c
/**
** 存储目标矩形框的结构体
*/
typedef struct box {
    float x, y, w, h;
} box;
```

$$b_x = \sigma(t_x) + c_x$$

$$b_y = \sigma(t_y) + c_y$$

$$b_w = p_w e^{t_w}$$

$$b_h = p_h e^{t_h}$$

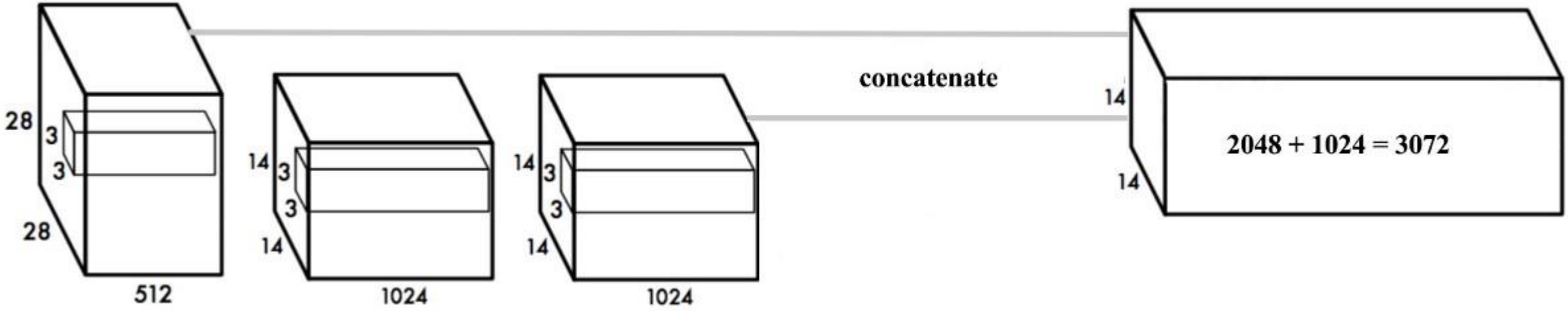
```c
box encode_box(box b, box anchor)
{
    box encode;
    encode.x = (b.x - anchor.x) / anchor.w;
    encode.y = (b.y - anchor.y) / anchor.h;
    encode.w = log2(b.w / anchor.w);
    encode.h = log2(b.h / anchor.h);
    return encode;
}

box decode_box(box b, box anchor)
{
    box decode;
    decode.x = b.x * anchor.w + anchor.x;
    decode.y = b.y * anchor.h + anchor.y;
    decode.w = pow(2., b.w) * anchor.w;
    decode.h = pow(2., b.h) * anchor.h;
    return decode;
}
```

更细粒度的特征（Fine-Grained Features）

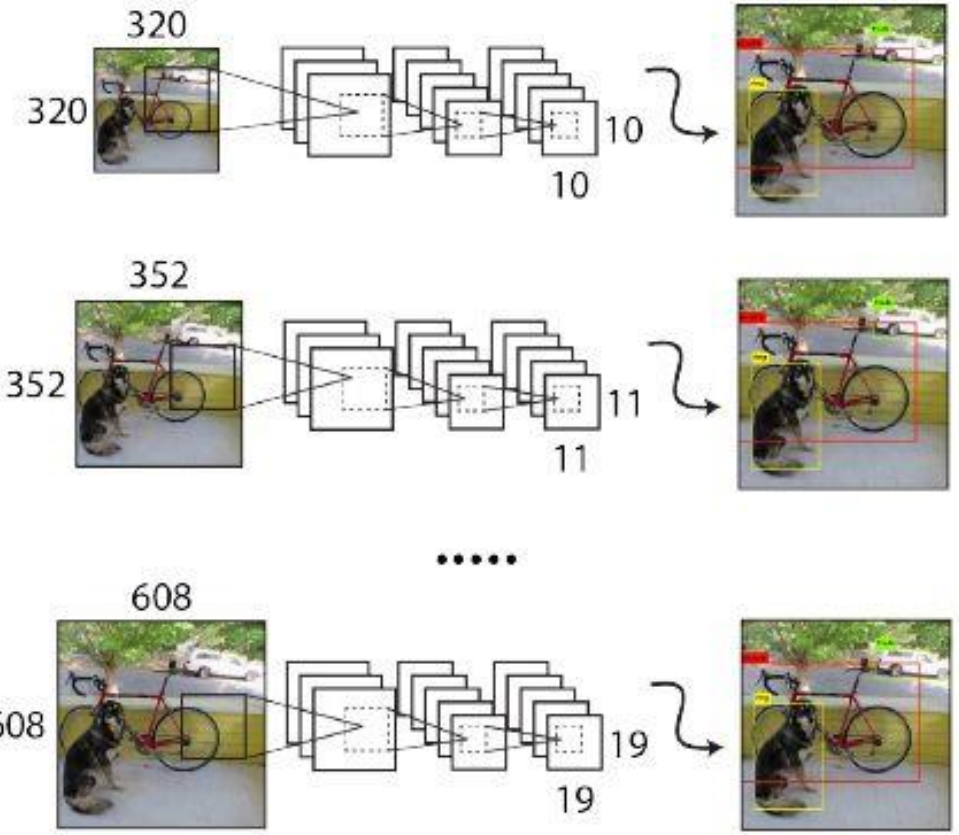
- 卷积层逐渐减小空间维度。 随着相应的分辨率降低，检测小目标变得更加困难。 可以从不同的特征图层找到目标。 每一层专注于不同的尺度。
- YOLO采用了一种称为passthrough的不同方法。 它将28×28×512层重整形为14×14×2048。然后将其与原始的14×14×1024输出层连接。 在新的14×14×3072层上应用卷积滤波器来进行预测。

类似于Resnet中的Identity Mappings



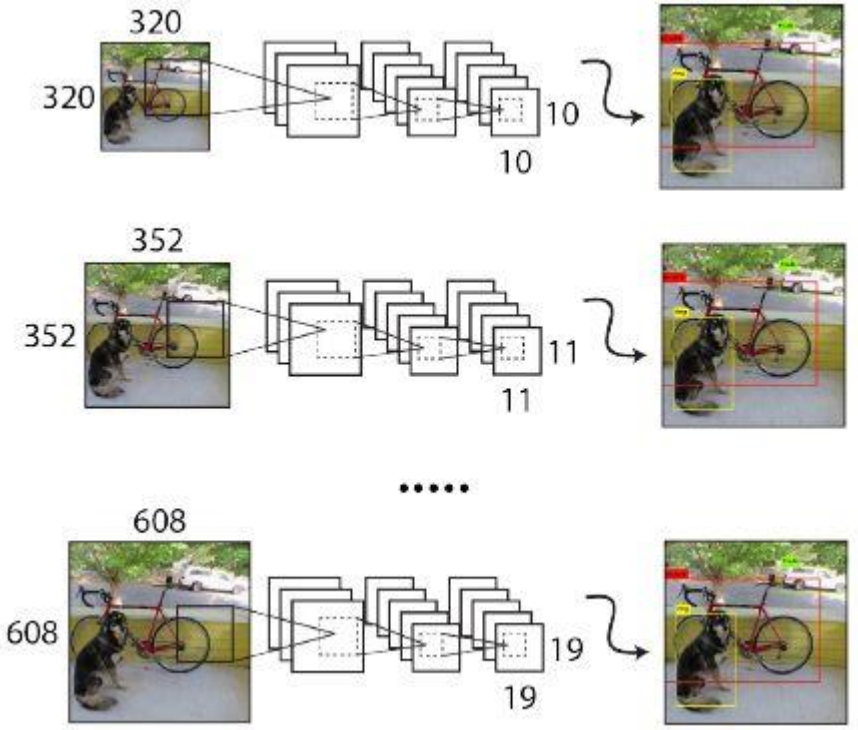
YOLOv2的检测器使用的就是经过扩展后的的特征图，它可以使用细粒度特征，使得模型的性能获得了1%的提升

多尺度训练(Multi-Scale Training)



- 由于YOLOv2模型移除全连接层后只有卷积层和池化层，所以YOLOv2的输入可以不限于 416×416 大小的图片。
- 为了增强模型的鲁棒性，YOLOv2采用了多尺度输入训练策略，具体来说就是在训练过程中每间隔一定的迭代（iterations）之后改变模型的输入图片大小。由于YOLOv2的为32倍下采样，输入图片大小选择一系列为32倍数的值：{320, 352,..., 608}，输入图片最小为 320×320，此时对应的特征图大小为 10 × 10；而输入图片最大为 608 × 608，对应的特征图大小为 19 × 19 。
- 在训练过程，每隔10个迭代（iterations）随机选择一种输入图片大小，然后只需要修改对最后检测层的处理就可以重新训练。

Multi-Scale Training



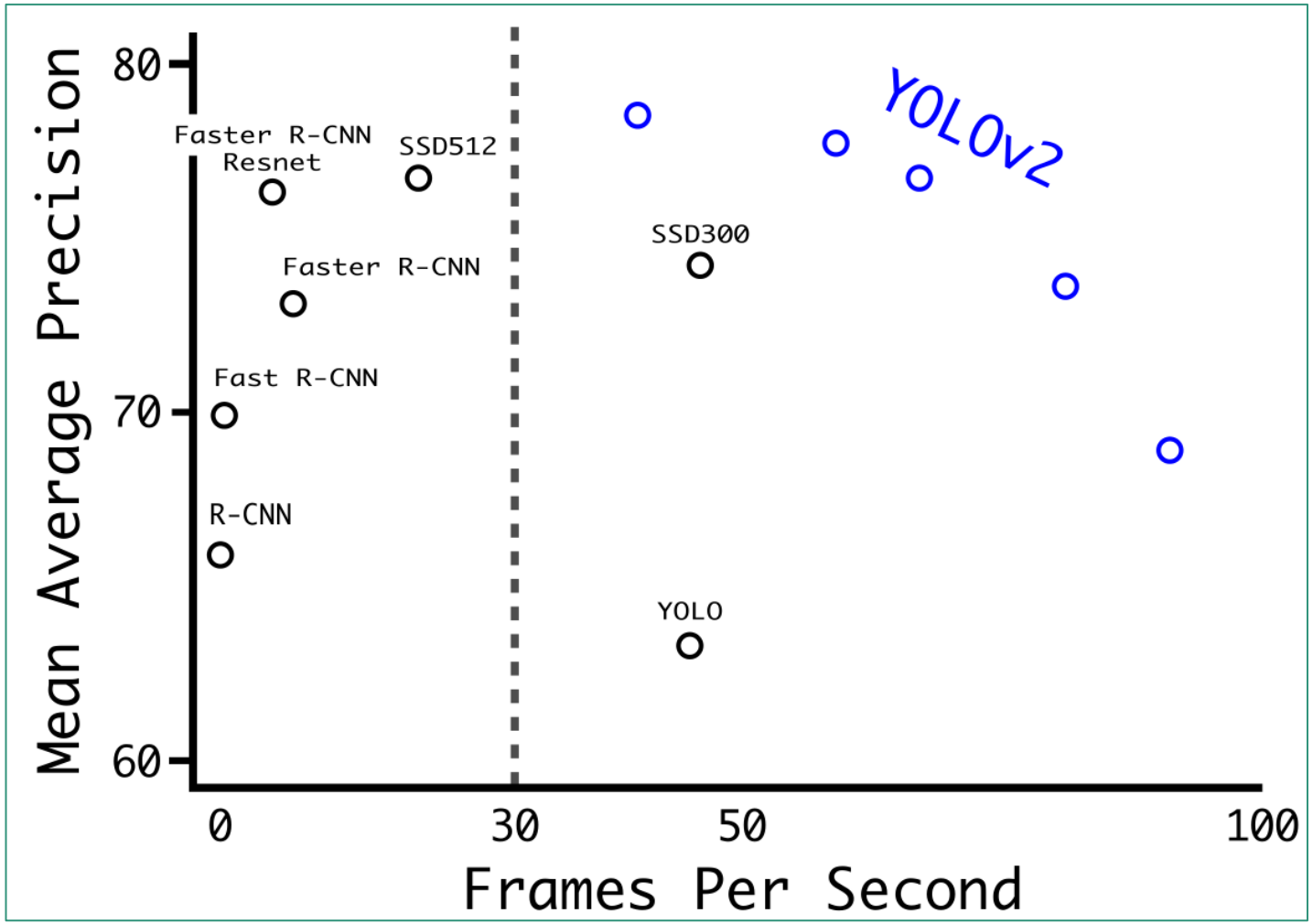
- 另外，可以使用较低分辨率的图像进行目标检测，但代价是准确度。 这对于低GPU设备的速度来说是一个很好的权衡。
- 在288×288时，YOLO的运行速度超过90 FPS，mAP几乎与Fast R-CNN一样好。 在高分辨率下，YOLO在VOC 2007上实现了78.6 mAP。

Accuracy

Here is the accuracy improvements after applying the techniques discussed so far:

	YOLO							YOLOv2
batch norm?		✓	✓	✓	✓	✓	✓	✓
hi-res classifier?			✓	✓	✓	✓	✓	✓
convolutional?				✓	✓	✓	✓	✓
anchor boxes?				✓	✓			
new network?					✓	✓	✓	✓
dimension priors?						✓	✓	✓
location prediction?						✓	✓	✓
passthrough?							✓	✓
multi-scale?								✓
hi-res detector?								✓
VOC2007 mAP	63.4	65.8	69.5	69.2	69.6	74.4	75.4	76.8	**78.6**

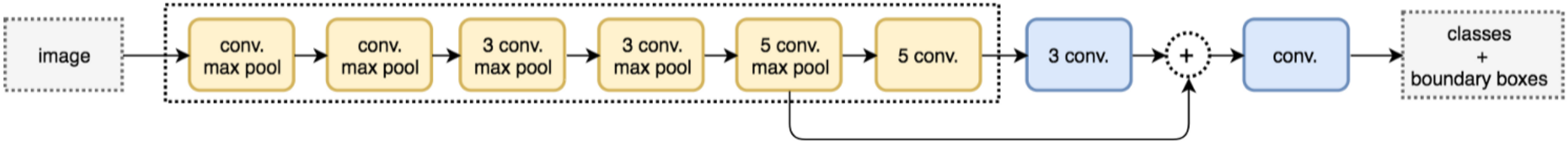
Accuracy comparison for different detectors:



Accuracy and speed on VOC 2007

更快速

YOLOv2采用了一个新的基础模型（特征提取器），称为Darknet-19，包括19个卷积层和5个max pooling层。



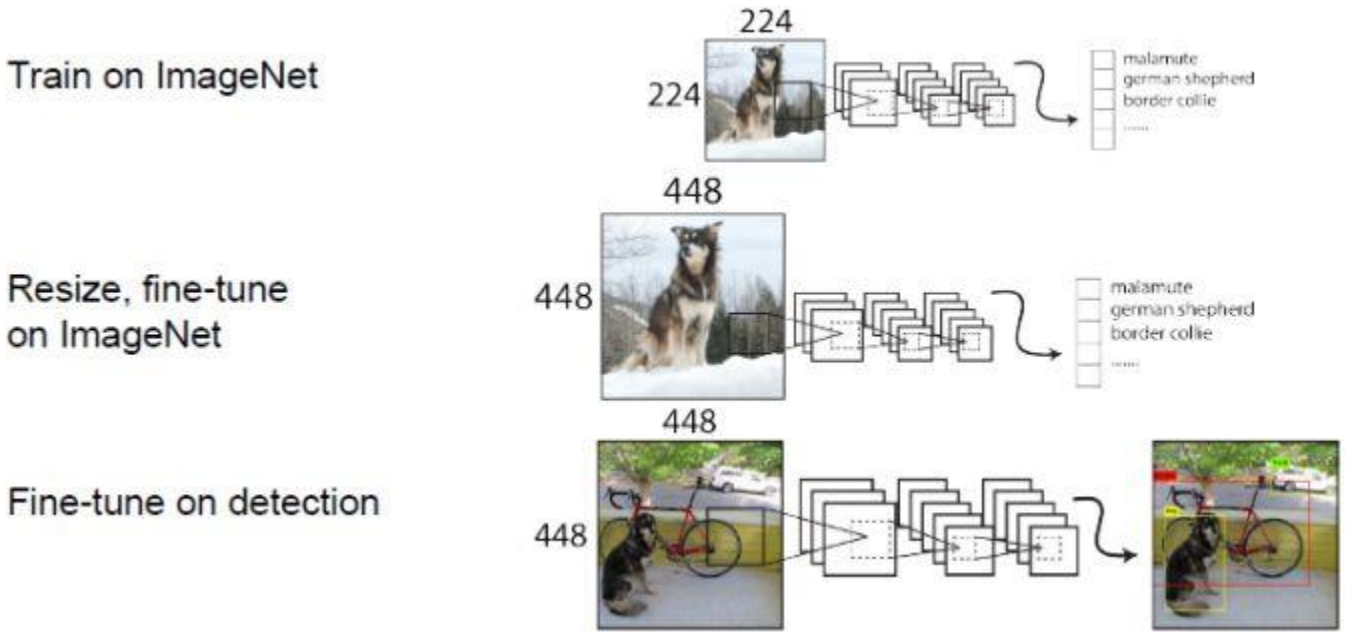
- Darknet-19与VGG16模型设计原则是一致的，主要采用 3×3 卷积，采用 2×2 的最大池化层之后，特征图维度降低2倍，而同时将特征图的通道增加两倍。
- 与NIN(Network in Network)类似，Darknet-19最终采用global avg pooling做预测，并且在 3×3 卷积之间使用 1×1 卷积来压缩特征图通道以降低模型计算量和参数。
- Darknet-19每个卷积层后面同样使用了batch norm层以加快收敛速度，降低模型过拟合。
- 在ImageNet分类数据集上，Darknet-19的top-1准确度为72.9%，top-5准确度为91.2%，但是模型参数相对小一些。
- 使用Darknet-19之后，YOLOv2的mAP值没有显著提升，但是计算量却可以减少约33%。

- Darknet-19，包括19个卷积层和5个max pooling层。
- 采用global avg pooling+Softmax做预测

Type	Filters	Size/Stride	Output
Convolutional	32	3×3	224×224
Maxpool		$2 \times 2/2$	112×112
Convolutional	64	3×3	112×112
Maxpool		$2 \times 2/2$	56×56
Convolutional	128	3×3	56×56
Convolutional	64	1×1	56×56
Convolutional	128	3×3	56×56
Maxpool		$2 \times 2/2$	28×28
Convolutional	256	3×3	28×28
Convolutional	128	1×1	28×28
Convolutional	256	3×3	28×28
Maxpool		$2 \times 2/2$	14×14
Convolutional	512	3×3	14×14
Convolutional	256	1×1	14×14
Convolutional	512	3×3	14×14
Convolutional	256	1×1	14×14
Convolutional	512	3×3	14×14
Maxpool		$2 \times 2/2$	7×7
Convolutional	1024	3×3	7×7
Convolutional	512	1×1	7×7
Convolutional	1024	3×3	7×7
Convolutional	512	1×1	7×7
Convolutional	1024	3×3	7×7
Convolutional	1000	1×1	7×7
Avgpool		Global	1000
Softmax			

YOLOv2的训练

YOLOv2的训练主要包括三个阶段：



- 第一阶段就是先在ImageNet分类数据集上预训练Darknet-19，此时模型输入为 224×224 ，共训练160个epochs。
- 然后第二阶段将网络的输入调整为 448 × 448 ，继续在ImageNet数据集上微调分类模型，训练10个epochs，此时分类模型的top-1准确度为76.5%，而top-5准确度为93.3%。
- 第三个阶段就是修改Darknet-19分类模型为检测模型，并在检测数据集上继续微调网络。

YOLO v2总结

- 总结来看，虽然YOLOv2做了很多改进，但是大部分都是借鉴其它论文的一些技巧，如Faster R-CNN的锚定框，YOLOv2采用锚定框和卷积做预测，与SSD模型非常类似，而且SSD也是借鉴了Faster R-CNN的RPN网络。
- 从某种意义上来说，YOLOv2和SSD这两个单阶段模型与RPN网络本质上无异，只不过RPN不做类别的预测，只是简单地区分目标与背景。
  - 在两阶段方法中，RPN起到的作用是给出region proposals，其实就是作出粗糙的检测，所以另外增加了一个阶段，即采用R-CNN网络来进一步提升检测的准确度（包括给出类别预测）。
  - 而单阶段方法想要一步到位，直接采用“RPN”网络作出精确的预测，要因此要在网络设计上做很多的技巧（tricks）。
- YOLOv2的一大创新是采用Multi-Scale Training策略，这样同一个模型就可以适应多种尺寸的图片。

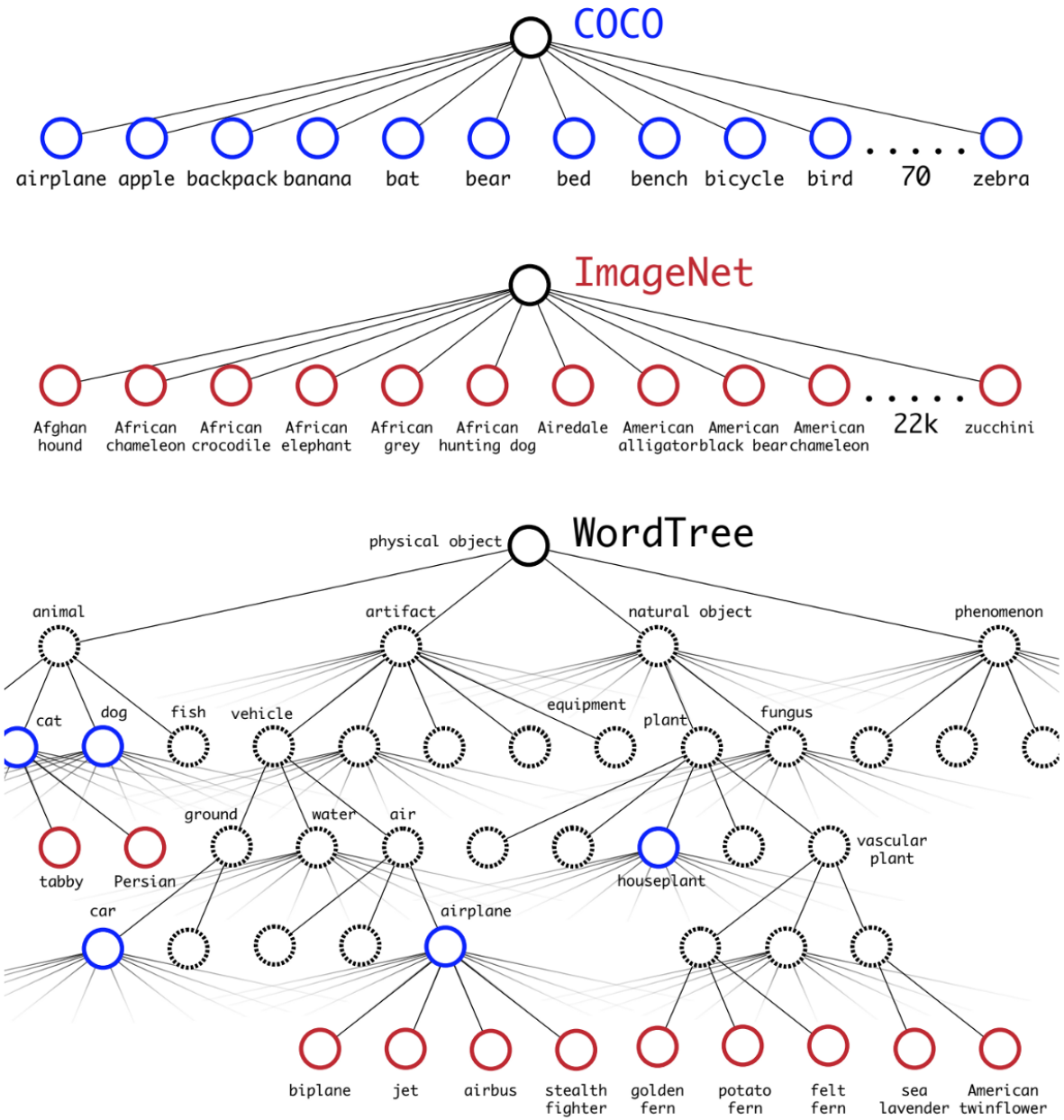
更强大 → YOLO9000

- YOLO9000是在YOLOv2的基础上提出的一种可以检测超过9000个类别的模型，其主要贡献点在于提出了一种分类和检测的联合训练策略。

- 众多周知，检测数据集的标注要比分类数据集打标签繁琐的多，所以ImageNet分类数据集比VOC等检测数据集高出几个数量级。在YOLO中，边界框的预测其实并不依赖于目标的标签，所以YOLO可以实现在分类和检测数据集上的联合训练。对于检测数据集，可以用来学习预测目标物体的边界框、置信度以及为目标分类，而对于分类数据集可以仅用来学习分类，但是其可以大大扩充模型所能检测的目标类别。

- 作者选择在COCO和ImageNet数据集上进行联合训练，但是遇到的第一问题是两者的类别并不是完全互斥的，比如“Norfolk terrier”明显属于“dog”，所以作者提出了一种层级分类方法（Hierarchical classification），主要思路是根据各个类别之间的从属关系（根据WordNet）建立一种树结构WordTree。

Combining COCO and ImageNet labels to a hierarchical WordTree

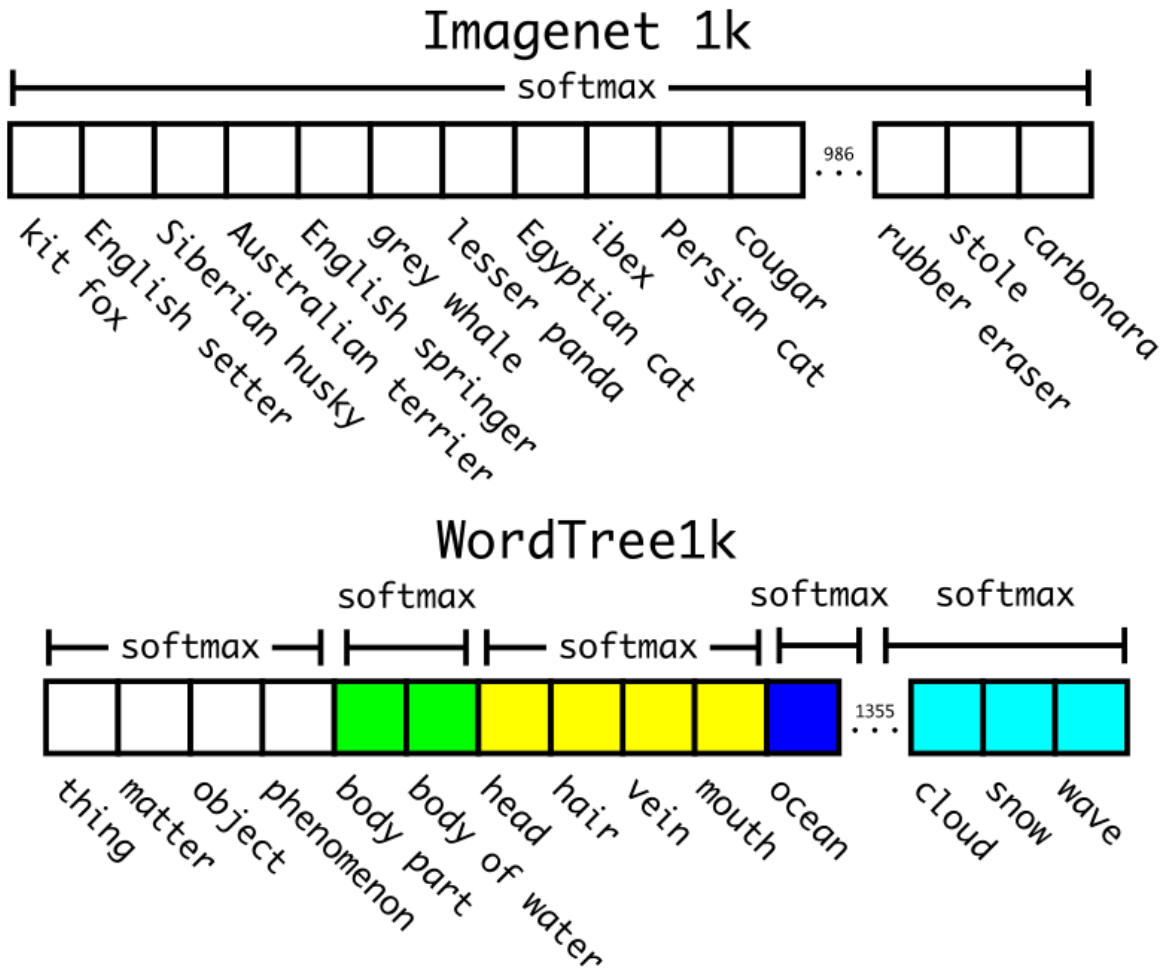


- 可以使用WordTree把多个数据集整合在一起。只需要把数据集中的类别映射到树结构中的同义词集合（Synsets）。

- 使用WordTree整合ImageNet和COCO的标签如图所示。

WordTree中的根节点为"physical object"，每个节点的子节点都属于同一子类，可以对它们进行softmax处理。在给出某个类别的预测概率时，需要找到其所在的位置，遍历这个path，然后计算path上各个节点的概率之积。

Prediction on ImageNet vs WordTree



- 大多数ImageNet模型使用一个大的softmax来预测概率分布
- 使用WordTree，在co-hyponyms上执行多个softmax操作

“玫瑰”是“花”的一个下义词（hyponym）；被包括在同一个上义词项内的词项称作同下义词（co-hyponyms），如被包括在上义词“花”之内的“玫瑰、牡丹、水仙、腊梅”等是同下义词。

联合分类和检测 (Joint Classification and Detection)

- YOLO9000扩展YOLO以检测超过9000个类别的目标，使用了9418节点WordTree的分层分类方法。它结合了COCO的样本和ImageNet的前9000个类别。
- YOLO为每个COCO数据采样四个ImageNet数据。 它学习使用COCO中的检测数据查找目标，并使用ImageNet样本对这些目标进行分类。
- 采用这种联合训练，YOLO9000从COCO检测数据集中学习如何在图片中寻找目标，从ImageNet数据集中学习更广泛的目标分类。
- 通过联合训练策略，YOLO9000可以快速检测出超过9000个类别的目标，总体mAP值为19.7%。


YOLO v2
特点
YOLOv2是YOLO的第二个版本，其目标是在提高速度的同时显著提高准确度。
Better
使用 Batch Normalization提高准确度
高分辨率分类器((High-resolution classifier)
用锚定框的卷积(Convolutional with Anchor Boxes)
维度聚类(Dimension Clusters)
直接位置预测(Direct Location Prediction)
更精细的特征（Fine-Grained Features)
多尺度训练(Multi-Scale Training)
Faster
Darknet-19网络模型
实验结果
YOLOv2: mAP=78.6%; YOLOv1=63.4%
Stronger(YOLO9000)
提出了一种层级分类方法（Hierarchical classification)
提出了一种分类和检测的联合训练策略