源地址：http://pjreddie.com/yolo9000/

**摘要：**

我们介绍一个最先进的实时对象检测系统YOLO9000，它可以检测超过9000个对象类别。首先，我们提出了YOLO检测方法的各种改进，既有新颖性，也是从以前的工作中得出的。改进型号YOLOv2是标准检测任务（如PASCAL VOC和COCO）的最先进技术。使用一种新颖的多尺度训练方法，相同的YOLOv2模型可以运行在不同的大小，提供了速度和精度之间的轻松的权衡。在67 FPS，YOLOv2在VOC 2007上获得76.8 mAP。在40 FPS下，YOLOv2获得78.6 mAP，超过最先进的方法，如具有ResNet和SSD的更快的RCNN，而运行速度明显更快。最后，我们提出一种联合训练对象的方法检测和分类。使用这种方法，我们在COCO检测数据集和ImageNet分类数据集上同时训练YOLO9000。我们的联合培训允许YOLO9000预测没有标记检测数据的对象类的检测。我们验证了我们在ImageNet检测任务中的方法。YOLO9000在ImageNet检测验证集上获得19.7 mAP，尽管只有200个类中的44个检测数据。在不在COCO的156个类中，YOLO9000获得了16.0 mAP。但是YOLO可以检测到超过200个类别;它预测超过9000个不同对象类别的检测。它仍然实时运行。

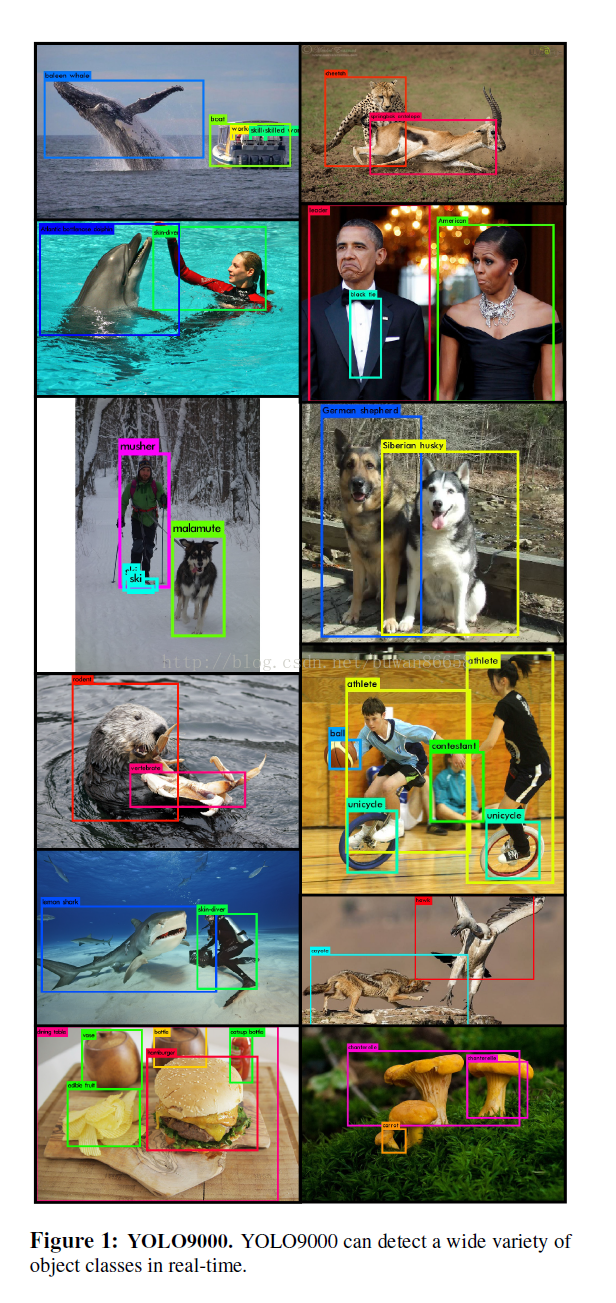
**1.简介：**

通用对象检测应该是快速，准确，能够识别各种各样的对象。自从引入神经网络以来，检测框架越来越快，准确。然而，大多数检测方法仍然限制在一小部分对象上。

当前对象检测数据集与其他任务（如分类和标记）的数据集相比是有限的。最常见的检测数据集包含数以千计的数十万个图像，数十到数百个标签[3] [10] [2]。分类数据集具有数百万个具有数十或数十万个类别的图像[20] [2]。

我们希望检测能够扩展到对象分类的级别。然而，标记图像进行检测比用于分类或标记的标签要贵得多（标签通常由用户自由提供）。因此，我们不太可能在不久的将来看到与分类数据集相同规模的检测数据集。

我们提出了一种新方法来利用我们已经拥有的大量分类数据，并利用它来扩大当前检测系统的范围。我们的方法使用允许我们的对象分类的分层视图  
将不同的数据集合在一起  
  
我们还提出了一种联合训练算法，使我们可以对检测和分类数据进行目标检测。 我们的方法利用标记的检测图像来学习精确定位对象，同时使用分类图像来增加其词汇和鲁棒性。  
  
使用这种方法，我们训练YOLO9000，一个可以检测超过9000个不同对象类别的实时目标检测器。 首先，我们改进了基础的YOLO检测系统来生产YOLOv2，这是现代最先进的实时检测器。然后我们使用我们的数据集合方法和联合训练算法，从ImageNet中对9000多个类别进行了一个模型的训练，以及COCO的检测数据。



**2.更好**

相对于最先进的检测系统，YOLO具有各种不足之处。YOLO与Fast R-CNN相比的错误分析表明，YOLO产生了大量的定位错误。此外，与区域提案方法相比，YOLO的召回率相对较低。因此，我们主要关注改善召回和定位，同时保持分类准确性。

计算机视觉通常趋向于更大，更深的网络[6] [18] [17]。更好的性能往往取决于训练更大的网络或将多个模型组合在一起。然而，使用YOLOv2，我们需要一个更加准确的检测器。我们不需要扩展我们的网络，而是简化网络，并使代码更容易学习。我们从过去的工作中汇集各种想法，以我们的新概念来提高YOLO的表现。结果摘要可以在表2中找到。

**Batch Normalization:**

批量归一化导致收敛的显着改善，而不需要其他形式的正则化[7]。通过在YOLO的所有卷积层上添加批量归一化，我们在mAP中得到超过2％的改进。批量归一化也有助于规范模型。通过批量归一化，我们可以从模型中删除dropuot而不过度拟合。

**High Resolution Classifier:**

所有最先进的检测方法都使用ImageNet上预处理的分类器[16]。从AlexNet开始，大多数分类器对小于256×256的输入图像进行操作[8]。原始的YOLO以224×224的分类器网络进行训练，将分辨率提高到448进行检测。这意味着网络必须同时切换到学习对象检测并适应新的输入分辨率。  
  
对于YOLOv2，我们首先在ImageNet上对10个epchs的448×448分辨率进行微调分类网络。这给网络时间调整滤波器以在更高分辨率的输入上更好地工作。然后，我们会在检测到的时候微调所得网络。这种高分辨率分类网络使我们增加了近4％的mAP。

**Convolutional With Anchor Boxes:**

YOLO直接使用卷积特征提取器顶部的全连接层来预测边界坐标。不需要直接预测坐标更快的R-CNN使用手工选择的先验来预测边界框[15]。仅使用卷积层，更快的R-CNN中的区域提议网络（RPN）预测了锚点框的偏移和置信度。由于预测层是卷积的，所以RPN在特征图中的每个位置预测这些偏移。预测偏移而不是坐标可以简化问题，使网络更容易学习。

我们从YOLO中删除完全连接图层，并使用锚点框来预测边框。首先我们消除一个池化层，使得网络卷积层的输出更高分辨率。我们还缩小网络，以操作416个输入图像，而不是448×448。我们这样做是因为我们想要在我们的功能地图中找到奇数个位置，所以有一个单一的中心单元格。目标，特别是大型的目标，往往占据图像的中心，所以在中心有一个位置能够很好的预测这些目标，而不是四个定位点都在附近。YOLO的卷积层通过一个下采样银子32，所以通过使用416的输入图像，我们得到了13×13的输出特征图。

当我们移动到锚边框时，我们也将类预测机制与空间位置分离，而是预测每个锚边框的类和目标。根据YOLO，在给定类别中，当有目标存在时，预测了ground-truth的IOU和所提出的框以及该类的条件概率。

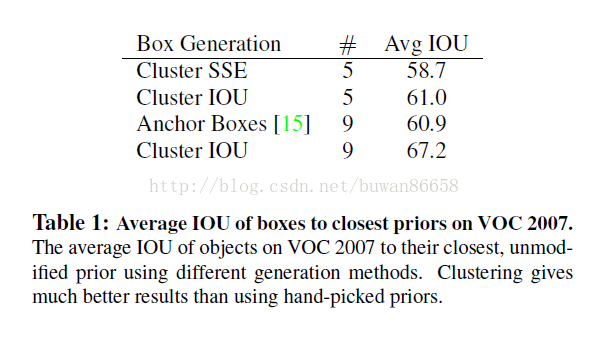
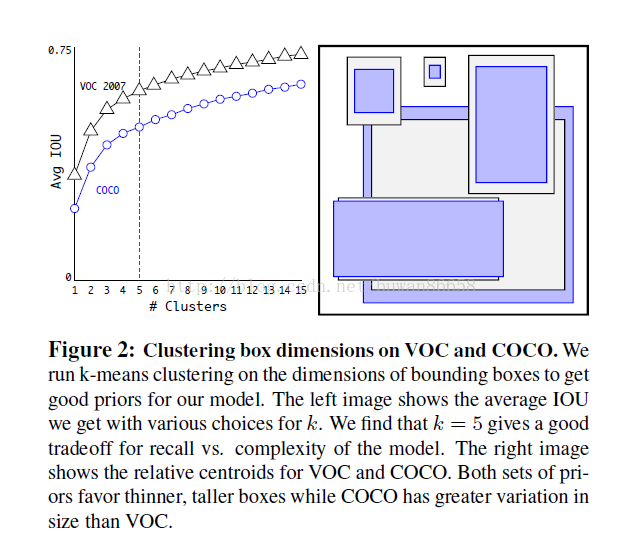
使用anchor boxes，我们精确度降低很少.YOLO仅预测每个图像98个边框，但是使用anchor boxes，我们的模型预测了一千多个。没有anchor boxes，我们的中间模型得到69.5 mAP，召回率为81％。 使用anchor boxes，我们的模型得到69.2 mAP，召回率为88％。尽管mAP减少，但是召回增加意味着我们的模式有更多的改进空间。

**Dimension Clusters：**

当与YOLO一起使用时，我们遇到两个问题。首先是手动选择边框尺寸。网络可以学习适当地调整框，但如果我们从网络中选择更好的先验，我们可以使网络更容易学习预测良好的检测。  
我们不用手动选择先验，而是在训练集边界框上运行k均值聚类，以自动找到优先级。 如果我们使用标准的k-means与欧氏距离较大的盒子比较小的盒子产生更多的错误。然而，我们真正想要的是优先级，导致良好的IOU分数，这与箱子的大小无关。因此，对于我们的距离度量我们使用：d(box,centroid)=1 -IOU(box,centroid)

我们对k的各种值运行k-means，并绘制最接近重心的平均IOU，参见图2.我们选择k = 5作为模型复杂度与高回调之间的一个很好的权衡。集群质心与手动选择的anchor boxes显著不同。短，宽的边框和更高，薄的边框更少。

我们将平均IOU与我们的聚类策略之前的最接近的比较和表1中的手动选择的anchor boxes进行比较。只有5个优先级，质心与9个anchor boxes相似，平均IOU为61.0，而60.9。 如果我们使用9个质心，我们看到一个更高的平均IOU。这表明使用k-means来生成我们的边界框可以用更好的表示来启动模型，使得任务更容易学习。



**Direct location prediction：**

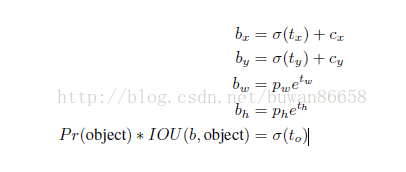
当使用YOLO的锚盒时，我们遇到第二个问题：模型不稳定，特别是在早期迭代期间。大多数不稳定性来自预测边框的（x; y）位置。在区域提案网络中，网络预测值tx和ty，并且（x; y）中心坐标计算为：

                            x = (t\_x\*w\_a) - x\_a

                            y = (t\_y \* h\_a) - y\_a

例如，tx = 1的预测将使边框向右移动anchor box的宽度，tx = -1的预测将向左移动相同的量。  
  
该公式是无约束的，所以任何anchor box可以在图像中的任何点处结束，而不管什么位置预测该盒子。 随机初始化，模型需要很长时间才能稳定以预测合理的偏移量。  
  
而不是预测偏移，我们遵循YOLO的方法，并相对于网格单元的位置预测位置坐标。 这将基本的真相限制在0到1之间。我们使用物流激活来限制网络的预测落在此范围内。

网络预测输出特征图中每个单元格的5个边界框。网络预测每个边界框tx，ty，tw，th和to的5个坐标。如果单元格从图像的左上角偏移（cx; cy），先前的边界框具有宽度和高度pw，ph，则预测对应于：

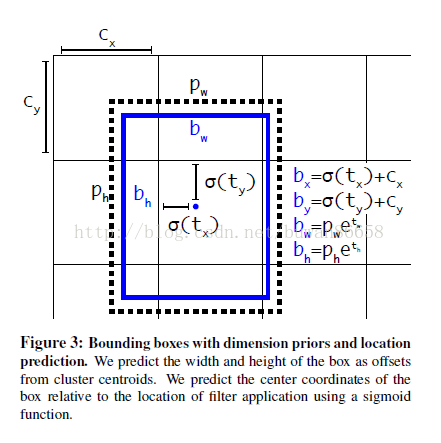


由于我们约束了位置预测，参数化更容易学习，使得网络更加稳定。使用维度集群以及直接预测边界框中心位置将YOLO比使用anchor boxes的版本提高了近5％。

**Fine-Grained Features:**

该修改的YOLO预测13×13特征图上的检测。虽然这对于大目标来说足够，但是它可能会从更细粒度的特征中获益，从而实现小目标定位。faster R-CNN和SSD可以在网络中的各种功能图上运行他们的提案网络，以获得一系列的分辨率。我们采用不同的方法，只需添加一个passthrough层，从26×26分辨率的早期层提取功能。

直通层将较高分辨率的特征与低分辨率特征相连，通过将相邻特征叠加到不同的通道而不是空间位置，类似于ResNet中的身份映射。这将26×26×512的特征图变成13×13×2048的特征图，可以与原来的特征相连。我们的检测器运行在这个扩展的功能图的顶部，以便它可以访问细粒度特征。这提供了一个适度的1％的性能提升。



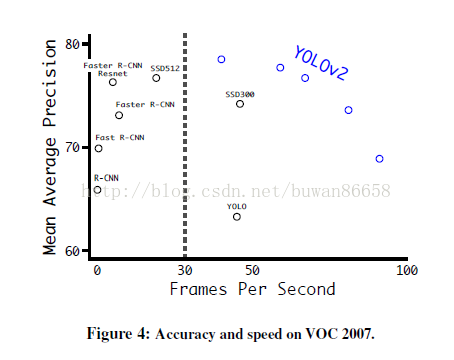
**Multi-Scale Training:**

原来的YOLO使用448×448的输入分辨率。通过添加anchor boxes，我们将分辨率更改为416×416。但是，由于我们的模型仅使用卷积和池化层，因此可以随时调整大小。我们希望YOLOv2能够运行在不同尺寸的图像上，因此我们将其训练到模型中。

而不是修改输入图像大小，我们每隔几个迭代就改变网络。我们的网络每10批次随机选择一个新的图像尺寸大小。由于我们的模型下降了32倍，我们从32的倍数：{320,352，...，608}中抽出。因此，最小的选择是320×320，最大的是608×608。我们将网络调整到该维度并继续训练。

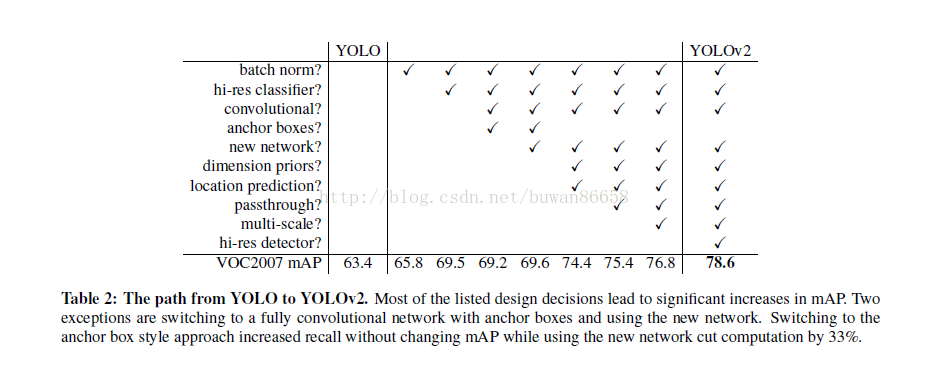
这一制度迫使网络学习如何在各种输入方面进行预测。这意味着相同的网络可以预测不同分辨率的检测。网络运行速度更快，所以YOLOv2在速度和精度之间提供了轻松的折中。  
在低分辨率下，YOLOv2作为廉价，相当准确的检测器运行。 在288×288，它运行在超过90 FPS，mAP几乎和Fast R-CNN一样好。这使它  
适用于较小的GPU，高帧速视频或多个视频流。

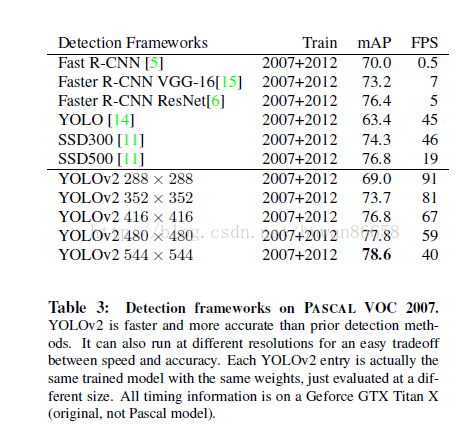
在高分辨率下，YOLOv2是VOC 2007中最先进的检测器，具有78.6 mAP，同时仍然高于实时速度。 有关YOLOv2与VOC 2007中其他框架的比较，请参见表3。图4

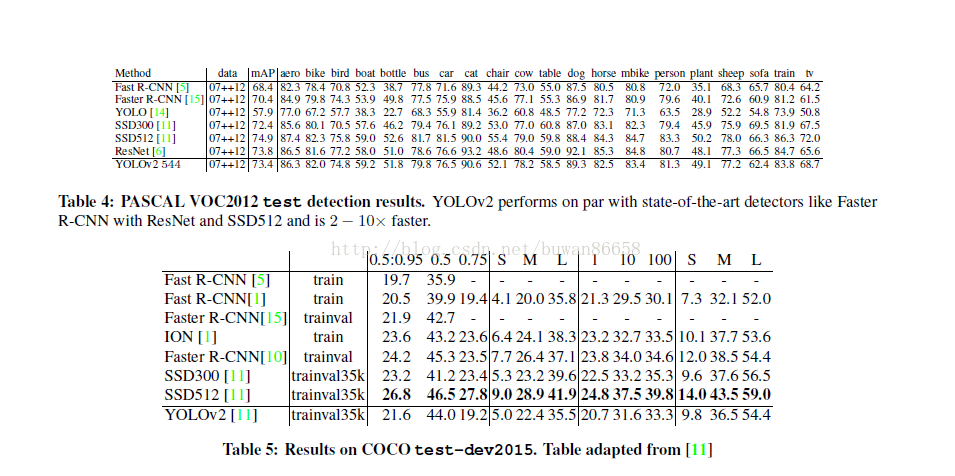


**Furher Experiments:**

我们在VOC 2012上训练YOLOv2进行检测。表4显示了YOLOv2与其他最先进的检测系统的比较性能.YOLOv2获得了73.4mAP的同时运行速度也是远远超过其他比较的方法。我们还对COCO进行训练，并与表5中的其他方法进行比较。在VOC指标（IOU = .5）上，YOLOv2获得44.0 mAP，与SSD和更快的R-CNN相当。







**3.更快**

我们希望检测准确，但我们也希望检测速度更快。用于检测的大多数应用，如机器人或自驾车，都依赖于低延迟预测。为了最大限度地提高性能，我们设计YOLOv2从头开始。  
大多数检测框架依赖于VGG-16作为基本特征提取器[17]。 VGG-16是一个功能强大，准确的分类网络，但不必要的复杂。VGG-16的卷积层需要306.6亿个浮点运算，用于以224×224分辨率对单个图像进行单次传递。  
YOLO框架使用基于Googlenet架构的自定义网络[19]。 这个网络比VGG-16要快，一个前向传播的只需要85.2亿次操作。但是，它的精度略差于VGG-16。对于单作物，在224×224下的top-5，YOLO的定制模型获得88.0％的ImageNet，而VGG-16的90.0％。

**Darknet-19：**

我们提出一种新的分类模型作为YOLOv2的基础。我们的模式建立在网络设计以及现场常识方面的先前工作。与VGG模型类似，我们主要使用3×3滤波器，每次池化长后，通道数量增加一倍[17]。在网络网络（NIN）的工作之后，我们使用全局平均池进行预测以及1×1个滤波器来压缩3×3卷积之间的特征表示[9]。我们使用BN来稳定训练，加快收敛速度，并使模型规范化[7]。

我们的最终模型称为Darknet-19，具有19个卷积层和5个最大填充层。有关完整的说明，请参见表6. Darknet-19仅需要58.8亿次运算来处理图像，但ImageNet上的top-1精度达到了72.9％，top-5精度达到了91.2％。

**Training for classification：**

我们用标准的ImageNet 1000类别分类数据集训练160个时期，使用起始学习率为0.1的随机梯度下降，功率为4的多项式速率衰减，0.0005的重量衰减和0.9的动量，使用Darknet神经网络框架[13]。在训练期间，我们使用标准数据增加技巧，包括随机crop，旋转，色调，饱和度和曝光转换。

如上所述，在我们对224×224的图像进行初始训练后，我们以较大的尺寸对我们的网络进行了微调，448.对于这个微调，我们训练上述参数，但只训练10个epochs，并以0.001的学习速度开始。在这个更高的分辨率下，我们的网络实现了top-1的精度为76.5％，top-5个精度为93.3％。

**Training for detection:**

我们通过去除最后一个卷积层来修改该网络进行检测，而不是在三个3×3的卷积层上加上1024个滤波器，之后是最终的1×1卷积层，其中需要检测的输出数量。对于VOC，我们预测5个框，5个坐标，每个有20个类，所以125个过滤器。我们还从最终的3×3×512层添加了一个passthrough层，到第二个到最后的卷积层，使得我们的模型可以使用细粒度特征。

我们训练160个epochs网络，起始学习率为0.001，在60和90个epochs分成十份。我们使用0.0005的权重衰减和0.9的动量。我们对YOLO和SSD采用与随机crop，色彩转移等相似的数据增加。我们对COCO和VOC采用相同的训练策略。

**4.更强大**

我们提出了分类和检测数据联合训练机制。我们的方法使用标记为检测的图像来学习检测特定信息，如边界框坐标预测和以及如何对常见目标进行分类。它使用只有类标签的图像来扩展它可以检测的类别数量。

在训练期间，我们混合来自检测和分类数据集的图像。当我们的网络看到一个标记为检测的图像时，我们可以基于完整的YOLOv2损失函数进行反向传播。当它看到一个分类图像时，我们只能从架构的分类特定部分反向传播损失。

这种方法提出了一些挑战。检测数据集只有常用的对象和一般标签，如“狗”或“船”。分类数据集具有更广泛和更深层次的标签。 ImageNet拥有一百多种犬，包括“诺福克犬”，“约克郡犬”和“贝灵顿犬”。如果我们要在两个数据集上进行训练，我们需要一个一致的方式来合并这些标签。

大多数分类方法使用所有可能类别的softmax层来计算最终概率分布。使用softmax假定这些类是互斥的。这会给组合数据集带来问题，例如，您不希望使用此模型来组合ImageNet和COCO，因为“诺福克犬”类和“狗”类不是相互排斥的。

**Hierarchical classification：**

ImageNet标签是从WordNet中提取出来的，这是一个构建概念及其相关关系的语言数据库[12]。在WordNet中，“诺福克犬”和“约克夏犬”都是“猎犬”的下标，它是一种“猎犬”，它是一种“狗”，它是“犬”等。大多数分类方法假设标签为扁平结构，但是对于组合数据集，结构正是我们需要的。

WordNet的结构是有向图，而不是树，因为语言是复杂的。例如，“狗”既是“犬”，也是WordNet中同义词的“家畜”类型。而不是使用完整的图形结构，我们通过从ImageNet中的概念构建层次树来简化问题。

为了构建这个树，我们检查ImageNet中的视觉名词，并通过WordNet图查看他们的路径到根节点，在这种情况下是“物理对象”。许多synsets只有一条通过图形的路径，所以首先我们将所有这些路径添加到我们的树中。然后，我们迭代地检查我们剩下的概念，并尽可能少地添加树生长的路径。因此，如果一个概念有两条路径到根，一条路径将添加三个边缘到我们的树，另一个只添加一个边，我们选择较短的路径。

最终的结果是WordTree，视觉概念的层次模型。为了使用WordTree执行分类，我们预测每个节点的条件概率，在给定该synset时，该synset的每个下位词的概率。例如，在“梗”节点，我们预测：

                                                            Pr(Norfolk terrier|terrier)

                                                            Pr(Yorkshire terrier|terrier)

                                                            Pr(Bedlington terrier|terrier)  
                                                                                                ...

如果我们要计算一个特定节点的绝对概率，那么我们只需要将该树的路径跟随到根节点并乘以条件概率。所以如果我们想知道图片是否是诺福克犬，我们计算：

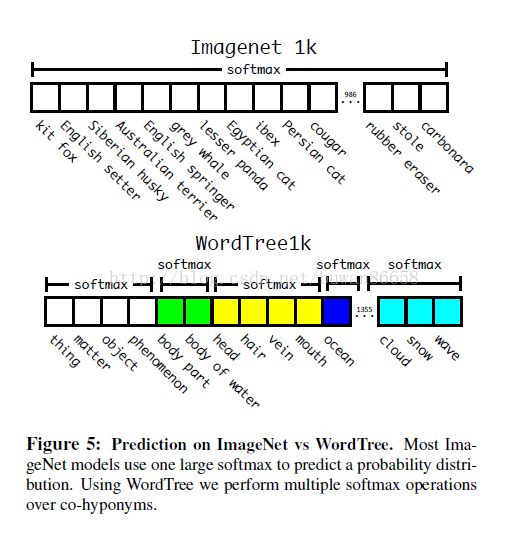
Pr(Norfolk terrier) = Pr(Norfolk terrier|terrier)  
Pr(terrier|hunting dog)  
 \*...\*  
\*Pr(mammal|Pr(animal)  
\*Pr(animal|physical object)

为了分类，我们假设图像包含一个对象：Pr（physical  object）= 1。

为了验证这种方法，我们训练使用1000类ImageNet构建的WordTree上的Darknet-19模型。为了构建WordTree1k，我们添加所有的中间节点，将标签空间从1000扩展到1369.在训练中，我们将树上的真实标签传播到树上，以便如果一个图像被标记为“诺福克犬”，它也被标记为“狗”和“哺乳动物”等。为了计算条件概率，我们的模型预测了1369个值的向量，并且我们计算作为相同概念的下位的所有系统的softmax，参见图5。

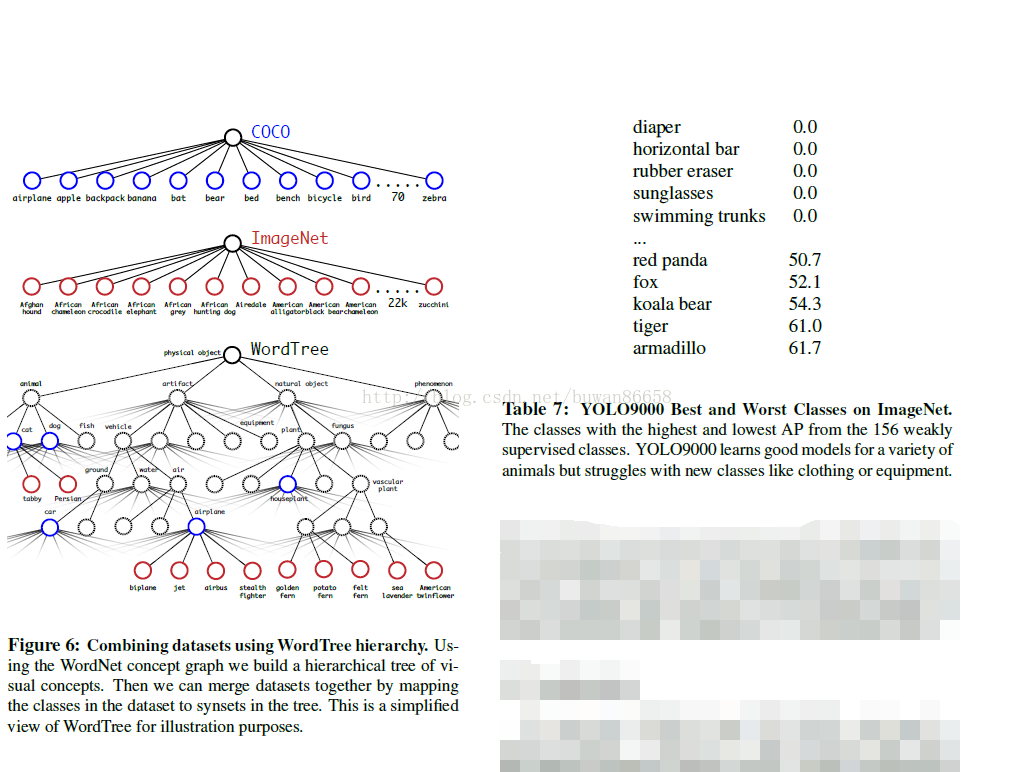
使用与以前相同的训练参数，我们的分级Darknet-19实现了71.9％的top-1精度和90：4％的top-5精度。尽管增加了369个附加概念，并让我们的网络预测了一个树状结构，但是我们的准确性只会略有下降以这种方式进行分类也有一些好处。性能会在新的或未知的对象类别上正常地降级。 例如，如果网络看到一只狗的图片，但是不确定它是什么类型的狗，它仍然会以高度的信心预测“狗”，但是在低位点之间散布较低的信心。

该制剂也适用于检测。现在，我们使用YOLOv2的对象预测器来代替每个图像都有一个对象，而不是给我们Pr（物理对象）的值。检测器预测了边界框和概率树。我们遍历树，在每个分裂处采取最高的置信路径，直到达到一定的阈值，并且我们预测该对象类。



**Dataset combination with wordtree:**

我们可以使用WordTree以合理的方式将多个数据集合在一起。我们简单地将数据集中的类别映射到树中的synsets。图6显示了使用WordTree组合来自ImageNet和COCO的标签的示例。WordNet是非常多样化的，所以我们可以使用这种技术与大多数数据集。



**Joint classification and detection:**

现在我们可以使用WordTree组合数据集，我们可以训练我们的分类和检测联合模型。我们想培训一个超大规模的检测器，因此我们使用COCO检测数据集和完整ImageNet版本的前9000个类创建了我们的组合数据集。我们还需要评估我们的方法，所以我们从ImageNet检测挑战中添加任何尚未包含的类。对于该数据集的WordTree具有9418个类。 ImageNet是一个更大的数据集，所以我们通过对COCO进行过采样来平衡数据集，使得ImageNet只有4：1的倍数。

使用这个数据集我们训练YOLO9000。我们使用基本的YOLOv2架构，但只有3个优先级而不是5个来限制输出大小。当我们的网络看到一个检测图像时，我们正常地传播损失。对于分类损失，我们只能反向传播标签上相应级别以上的损失。 例如，如果标签是“狗”，我们会在树中进一步降低预测值“德国牧羊犬”与“金毛猎犬”的差异，因为我们没有这些信息。

当它看到分类图像时，我们只是反向传播分类损失。为了做到这一点，我们只需找到一个预测该类最高概率的边界框，并计算其预测树上的损失。我们还假设预测的框重叠了至少为.3 IOU的ground truth标签，并且我们基于这个假设反向传播目标损失。

使用这种联合培训，YOLO9000使用COCO中的检测数据学习使用图像中的对象，并学习使用ImageNet中的数据对这些对象进行分类。

我们在ImageNet检测任务上评估YOLO9000。ImageNet在44个对象类别上与COCO共享检测任务，这意味着YOLO9000只能看到大多数测试图像的分类数据，而不是检测数据。YOLO9000在不相交的156个对象类上总共获得19.7 mAP，16.0 mAP，从未见过任何标记的检测数据。该MAP高于DPM实现的结果，但YOLO9000仅在部分监督下对不同数据集进行了训练[4]。它还同时检测9000个其他对象类别，全部是实时的。

当我们分析YOLO9000在ImageNet上的表现时，我们看到它学习动物的新物种能力很好，但是与学习类别（如服装和设备）却不是很好。新的动物更容易学习，因为对象的预测从COCO的动物得到广泛的推广。相反，COCO对于任何类型的服装都没有限制的边框，仅供人使用，所以YOLO9000正在努力模拟“太阳眼镜”或“游泳裤”等类别。

**5.总结**

我们介绍YOLOv2和YOLO9000的实时检测系统。YOLOv2是各种检测数据集中的其他检测系统的最先进和最快速的。此外，它可以以各种图像尺寸运行，以提供速度和精度之间的平滑折衷。

YOLO9000是通过联合优化检测和分类来检测超过9000个对象类别的实时框架。我们使用WordTree将来自各种来源的数据和我们的联合优化技术组合在ImageNet和COCO上同时进行训练。YOLO9000是拉近检测和分类之间的数据集大小差距的强有力的一步。

我们的许多技术在对象检测之外推广。我们的WordTree表示ImageNet为图像分类提供了更丰富，更详细的输出空间。使用层次分类的数据集合在分类和分割域中将是有用的。  
多尺度训练技术可以为各种视觉任务提供益处。

对于未来的工作，我们希望使用类似的技术来进行弱监督的图像分割。我们还计划使用更强大的匹配策略来提升我们的检测结果，将弱标签分配给训练期间的分类数据。计算机视觉拥有大量标签数据。我们将继续寻找方法，将不同的数据源和结构结合在一起，从而形成更强大的视觉世界模型。

**References**

[1] S. Bell, C. L. Zitnick, K. Bala, and R. Girshick. Insideoutsidenet: Detecting objects in context with skip pooling and recurrent neural networks. arXiv preprint  
arXiv:1512.04143, 2015. 6  
[2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database.In Computer Vision and Pattern Recognition, 2009. CVPR2009. IEEE Conference on, pages 248–255. IEEE, 2009. 1  
[3] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge.International journal of computer vision, 88(2):303–338, 2010. 1  
[4] P. F. Felzenszwalb, R. B. Girshick, and D. McAllester.Discriminatively trained deformable part models, release 4.http://people.cs.uchicago.edu/ pff/latent-release4/. 8

[5] R. B. Girshick. Fast R-CNN. CoRR, abs/1504.08083, 2015.5, 6  
[6] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learningfor image recognition. arXiv preprint arXiv:1512.03385,2015. 2, 5, 6  
[7] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift.arXiv preprint arXiv:1502.03167, 2015. 2, 5  
[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems, pages1097–1105, 2012. 2  
[9] M. Lin, Q. Chen, and S. Yan. Network in network. arXiv preprint arXiv:1312.4400, 2013. 5

[10] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan,  
P. Doll´ar, and C. L. Zitnick. Microsoft coco: Common objects in context. In European Conference on Computer Vision, pages 740–755. Springer, 2014. 1, 6  
[11] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, and S. E. Reed.SSD: single shot multibox detector. CoRR, abs/1512.02325,2015. 5, 6  
[12] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J.Miller. Introduction to wordnet: An on-line lexical database.International journal of lexicography, 3(4):235–244, 1990.6  
[13] J. Redmon. Darknet: Open source neural networks in c.http://pjreddie.com/darknet/, 2013–2016. 5

[14] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. arXiv preprint arXiv:1506.02640, 2015. 5, 6  
[15] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks.arXiv preprint arXiv:1506.01497, 2015. 2, 3, 5, 6  
[16] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh,S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein,A. C. Berg, and L. Fei-Fei. ImageNet Large Scale Visual  
Recognition Challenge. International Journal of Computer Vision (IJCV), 2015. 2  
[17] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014. 2, 4

[18] C. Szegedy, S. Ioffe, and V. Vanhoucke. Inception-v4,inception-resnet and the impact of residual connections on learning. CoRR, abs/1602.07261, 2016. 2  
[19] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed,D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich.Going deeper with convolutions. CoRR, abs/1409.4842,2014. 4  
[20] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni,D. Poland, D. Borth, and L.-J. Li. Yfcc100m: The new data in multimedia research. Communications of the ACM,59(2):64–73, 2016. 1