* 1. 摘要

本文提出了YOLO9000，一个最新的实时目标检测系统，能够检测超过9000个目标类别。首先，对于YOLO目标检测方法，我们在以前工作的基础上进行了一些创新性的改进。改进后的YOLOv2模型是在PASCAL VOC以及COCO标准目标检测任务上的最新工作。创新性的使用多放缩尺度的训练方法，同一个YOLOv2模型可以在多种尺寸的图片上运行，能够轻易地提供检测速度与准确度的权衡。在VOC2007数据集上，YOLOv2能够达到67帧每秒的速度，mAP达到76.8%。在40帧每秒的速度之下，YOLOv2的准确度达到78.6%，相比于以ResNet和SSD为骨干网络的Faster R-CNN，同时运行速度也有显著的提高。最后，提出了一种目标检测与分类联合训练的方法。利用这种方法，我们在COCO检测数据集和ImageNet分类数据集上同时训练YOLO9000。我们的联合训练使YOLO9000能够预测没有标记检测数据。我们在ImageNet检测任务上验证我们的方法。YOLO9000 在ImageNet的验证集上的mAP只达到19.7，尽管数据集中只包含200个类表中的44个类别的数据。在不存在COCO中，包含156个类别的数据上，YOLO9000的mAP达到16.0。但是，YOLO能够200个类，预测超过9000个不同的目标类别。并且，能够达到实时运行。

* 1. 引言

通用的目标检测应该是快速准确，而且能够识别各种各样的目标类别。因此，引入神经网络，目标检测变得越来越快速且准确。然而，大部分的目标检测方法仍然只能检测少数的目标。

与分类等其他的数据集相比，目标检测数据集是十分有限的。目前，常见的目标检测数据集包含的图片数量在几千到数十万之间，有几十到几百个标签类别。分类数据集的图片达到了百万级别，类别数达到了几万到几十万。

我们希望检测更多的目标类别，但是，为目标检测任务标记图片比起为分类任务标记图片更加昂贵。因此，即使在不久的将来，我们也不太可能看到与分类数据集同等规模的目标检测数据集。

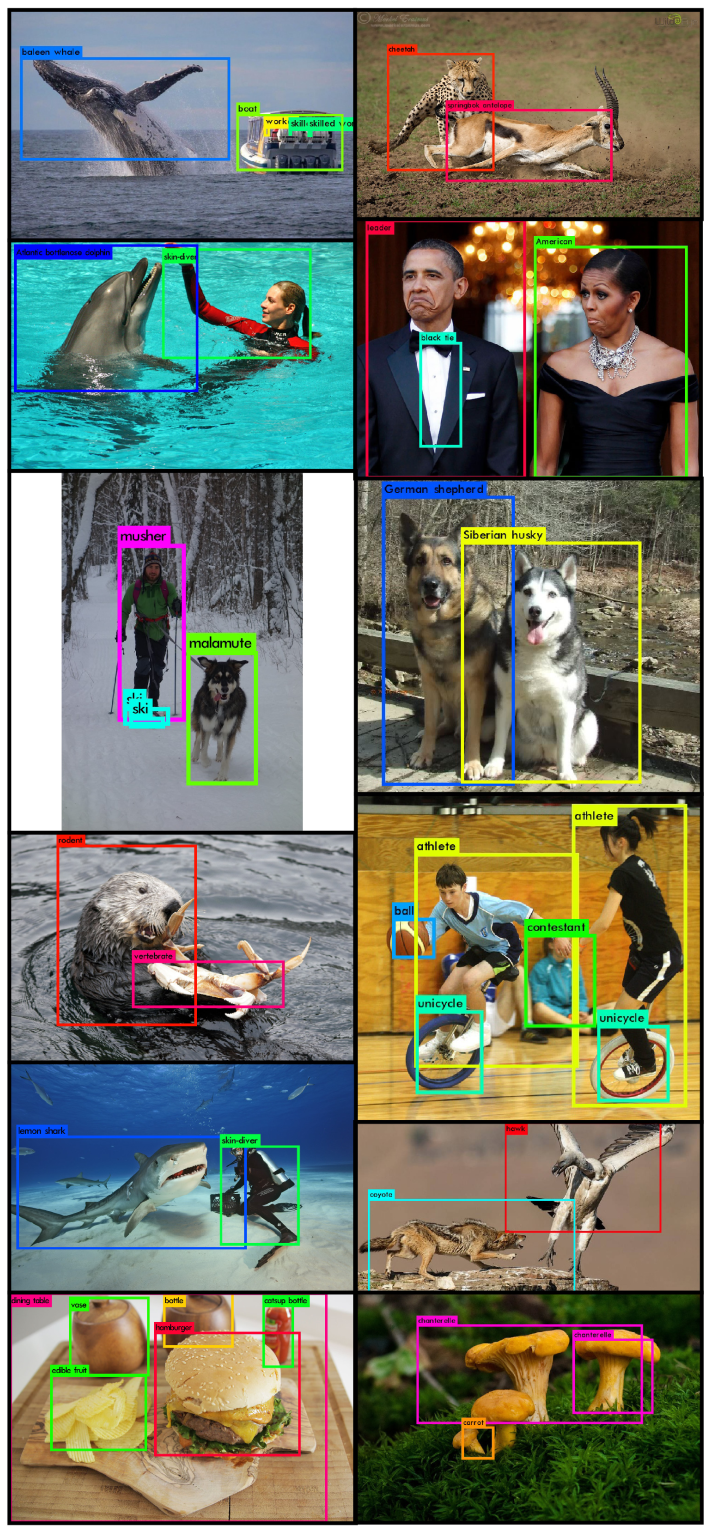


图1 YOLO9000 YOLO9000能够实时检测广泛的目标类别

我们提出了一个新方法利用已有的大量分类数据，来扩展当前检测系统的检测范围。我们的方法是使用分类对象的分层视图，可以将不同的数据集组合在一起。

我们同时提出了一个联合训练算法，允许我们在检测和分类数据上训练目标检测器。我们的方法通过标记的目标检测图像来学习精确定位目标，通过分类图像来增加系统的识别范围和鲁棒性。

使用这个方法来训练YOLO9000，能够实时检测超过9000个不同的目标类别。首先，我们对基本的YOLO检测系统进行了改进，提出了最新的实时检测器YOLOv2（AKA YOLO9000）。 然后，我们使用我们的数据集组合方法和联合训练算法对ImageNet上的9000多个类别以及COCO的检测数据进行训练。

我们的所有代码以及预训练模型可以在 <http://pjreddie.com/yolo9000/> 得到。

* 1. 更好

与最先进的检测系统相比，YOLO存在许多缺点。 与Fast R-CNN相比，通过误差分析表明，YOLO会产生大量的定位错误。 此外，与基于检测区域推荐的方法相比，YOLO的召回率相对较低。 因此，我们主要致力于改善召回率和定位误差，同时保持分类准确性。

计算机视觉通常趋向于使用更大、更深的网络。更好的性能通常依赖于训练更大的网络，或者将更多的模型组合在一起。然而，对于YOLOv2，我们想要一个准确率更高。同时检测快速的模型。我们简化了网络，使特征表示更加易于学习，而不是扩展网络的规模。我们将过去的工作中的各种想法与我们自己的新颖想法融合在一起，以提高YOLO的性能。 结果摘要可在表2中找到。

**批归一化。**批归一化能够显著提升模型的收敛性能，从而减少其他形式的正则化需求，通过在YOLO模型中所有的卷积层前增加批归一化，我们的模型mAP获得了

超过2%的提升。批归一化能够帮助模型实现正则化的功能。通过批归一化，我们可以移除dropout，而不必担心过拟合的问题**。**

**高分辨率分类器。**目前所有的目标检测模型使用的分类器在ImageNet进行了预训练。从AlexNet开始，大多数分类器将输入图片缩小到小于256X256。原始的YOLO模型使用224X224的图片来训练分类器，将分辨率增加到448进行检测。这意味着网络必须能够同时切换到学习目标检测并且调整到新的输入分辨率。

对于YOLOv2，我们首先在ImageNet上以448X448分辨率微调分类网络10个epoch。使网络能够调整他的卷积核(filter)在高分辨率的输入上更好的工作。然后，我们在对检测网络进行微调。高分辨率的分类网络能够给我们的模型带来大约4%的mAP提升。

**锚框(Anchor Boxes)卷积。**YOLO通过使用在卷积特征提取器之上的全连接层来直接预测边界框的坐标。Faster R-CNN使用手选的先验框来预测边界框，而不是直接预

预测其坐标。Faster R-CNN使用只有卷积层的区域建议网络（RPN）预测锚框的偏移和置信度。由于预测层是卷积的，因此RPN会在特征图中的每个位置预测这些偏移。 预测偏移量而不是坐标可以简化问题，并使网络更容易学习。

我们去掉了原始YOLO模型中的全连接层，使用锚框来预测边界框。首先，我们去掉池化层，使卷积层的输出具有更高的分辨率。我们缩小网络来处理416X416分辨率的输入图片而不是448X448分辨率的图片。之所以这样做，是因为我们希望特征图中的位置数量是奇数，以至于只有一个中心网格。目标，尤其是对大目标而言，趋向于占据图片的中心。因此，最好使用图片中心的一个位置去预测这些对象，而不是使用周围的四个位置。YOLO的卷积层使用因子32来对图片进行下采样。因此，使用416X416的输入图片，最后得到13X13的输出特征图。

当我们转而使用锚框时，我们同时将类别预测与空间位置预测解耦，而是为每一个锚框预测类别以及目标。与YOLO相同，对象预测仍然是预测GT与锚框的交并比，类别预测预测的是在锚框存在对象的条件下，目标的类别的条件概率。

使用锚框，模型在准确率上有微小的下降。YOLOv1每张图片仅仅能预测98个边界框，但是，通过锚框我们的模型可以预测超过一千个边界框。不使用锚框的情况下，模型的达到69.5%的mAP，81%的召回率，使用锚框的情况下，模型的mAP为69.2%，召回率为88%。虽然，模型的mAP的有微小的下降，但是召回率有较大空间的提升。

**锚框尺寸聚类。**当我们在YOLO模型中使用锚框时，遇到了两个问题。第一个是，锚框的尺寸是手选的。网络可以适当的学习调整锚框，但是，如果我们为网络选择更好尺寸的锚框去学习，我们能够使网络更容易学习，并作出好的预测。

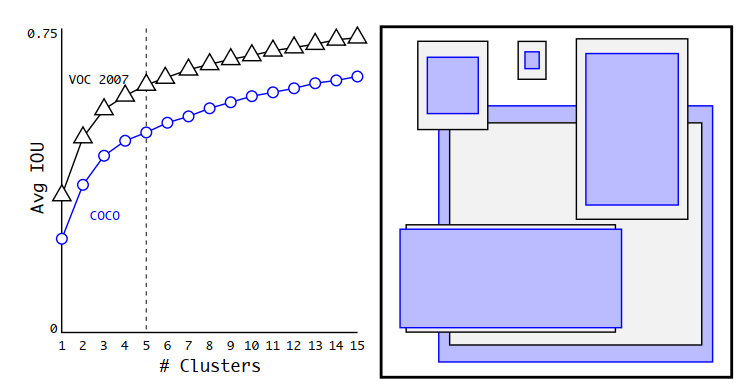


图2. VOC以及COCO数据集上进行锚框尺寸聚类。在边界框尺寸上进行K-Means聚类，为我们的模型得到更好的先验框。左图展示了平均IOU随着聚类数量k的变化趋势，当k=5时，模型的复杂度与召回率能够取得较好的平衡。右图展示了VOC与COCO中的聚类的相对质心，两个数据集都倾向于选择更高，更瘦的边界框尺寸。而COCO数据集的尺寸差异要大于VOC数据集。

我们通过在训练集数据的边界框上进行K-Means聚类，来发现好的先验框，而不是通过手选的方式。如果我们使用欧氏距离的标准K-Means算法，大尺寸的框会比小尺寸的框产生更多的误差。但是，我们真正想要的是能够获得良好IOU的先验值，而这与盒子的大小无关。因此，我们使用的距离度量:

d(box, centroid) = 1 − IOU(box, centroid)

我们选择多个K值进行K-Means聚类，用最接近质心的聚类框的尺寸计算平均的IOU，如图2所示，我们选择k值为5，能够在模型复杂度和召回率之间取得良好的平衡。聚类框的质心与手选的锚框十分不同，矮宽的框少，瘦高的框多。

我们将聚类得到的先验框与手选锚框的平均IOU进行了对比，如表1所示。在仅有5个聚类框时，平均IOU达到61.0，与9个手选锚框达到的平均IOU60.9相似。如果9个聚类框，则平均IOU更高。这表明，使用k-means生成边界框可以更好地表示模型，从而使任务更易于学习。

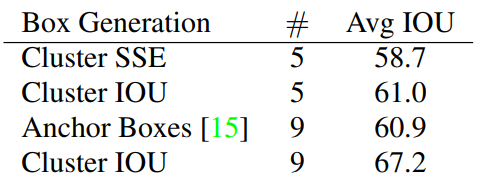


表1. VOC2007上的最接近的聚类框的平均IOU。使用不同方式产生的边界框的平均交并比比较说明，聚类比手选具有更好的结果。

**直接位置预测。**当在YOLO模型中使用锚框时，我们遇到了第二个问题：模型不稳定，特别是在早期的迭代中。主要的不稳定因素来源于边界框的位置(x, y)预测。在区域建议网络（RPN）中，网络的预测值以及，中心坐标(x,y)计算如下：

x = ( \* ) \*

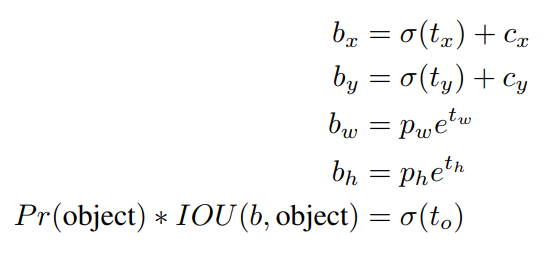
y = ( \* ) \*

例如，预测值 = 1 表示将锚框右移1个锚框宽度的距离，反之亦然。

这个公式没有添加任何约束，所以任何一个锚框可能在图片的任何一个像素结束。使用随机初始化，模型要经过很长的时间才能稳定下来，预测到合理的输出。

相比于预测偏移量，我们使用YOLO中的方式，预测相对于网格的边界框位置坐标。这将真实标签的值限制在0到1之间。我们使用逻辑斯蒂激活函数（sigmoid）函数将网络的输出限制在这个范围之内。

网络预测在输出的特征图中每个网格上预测5个边界框。每一个边界框预测5个坐标， （相对于grid cell的偏移量）以及 。如果网格相对于图片左上角的偏移量为(, ),并且先验框（anchor）的宽和高分别为 和 ，与之对应的预测为：



由于我们限制了位置预测的大小，因此参数更易于学习，从而使网络更稳定。 与使用锚框的版本相比，使用尺寸聚类以及直接预测边界框的中心位置可使YOLO提高近5％。

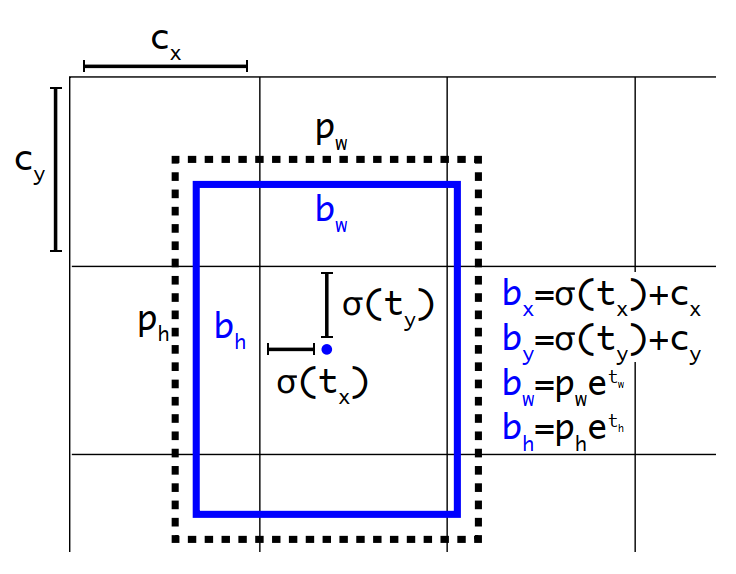
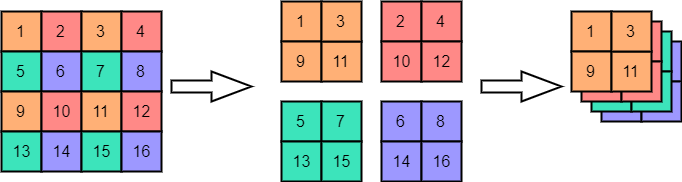


图3. 通过先验框预测边界框位置. 我们将边界框的框的宽度和高度预测为与簇质心的偏移量。 我们使用sigmoid函数预测边界框相对于网格左上角位置的中心坐标。

**细粒度特征.** 这个改进使YOLO能够在13X13的特征图上预测。尽管这可能对于大目标已经足够了，但是更加细粒度的特征对于定位更小的目标可能会更有益处。Faster R-CNN以及SSD在多个特征图上运行区域建议网络来获得多种尺寸的锚框。我们采用了不同的方式，简单的增加passthrough层将早期网络层的的特征以26X26的分辨率尺寸传递到后面。

passthrough层通过将相邻特征图按照通道堆叠到一起（而不是按空间位置堆叠），从而将高分辨率特征与低分辨率特征连接起来，类似于ResNet中的恒等映射。可以将26 X 26 X 512的特征图转变成13 X 13 X 2048的特征图，然后与原始的特征图在channel维度连接在一起。我们的探测器在这个扩展的特征图上进行检测。这个改进给模型带来了1%的性能提升。

Passthrough层，宽高变为原来的一半，深度变为原来的4倍。



**多尺度训练.** 原始的YOLO模型使用的输入图片的分辨率为448X448。随着锚框的使用，我们将输入图片的分辨率调整到416X416。但是，由于我们的模型仅使用卷积和池化层，因此可以随时调整输入图片的分辨率大小。 我们希望YOLOv2能够在不同尺寸的图像上运行，因此我们使用多种图片尺寸训练模型中。

我们每隔若干个迭代就要改变我们的网络结构，而不是改变输入图片的尺寸。每隔10个批次，我们的网络就会随机选择一个新的图片尺寸。我们的模型以32为因子进行下采样，因此，我们从以下32的倍数中提取：{320，352，…，608}。因此，最小的图片尺寸选择是320 X 320，最大的选择是 608 X 608。我们将网络调整到该图片的尺寸并继续培训。

这种方法能够迫使网络对与多种输入尺寸的图片进行学习，并进行良好的预测。 这意味着同一网络可以对不同分辨率的图片做出预测。 网络在较小的尺寸下运行速度更快，因此YOLOv2可以在速度和准确性之间轻松权衡。

在低分辨率下，YOLOv2检测器计算快速，准确度颇高。 在288×288的分辨率下，它的检测速度超过90FPS，mAP指标几乎与Fast R-CNN相当。 这使其成为小型GPU，高帧率视频或多个视频流检测的理想选择。

在高分辨率下，YOLOv2是在VOC 2007上的mAP达到78.6%，是目前最先进的目标检测器，同时仍可以保持实时速度运行。 有关VOC 2007上YOLOv2其他框架的比较，请参见表3。图4。

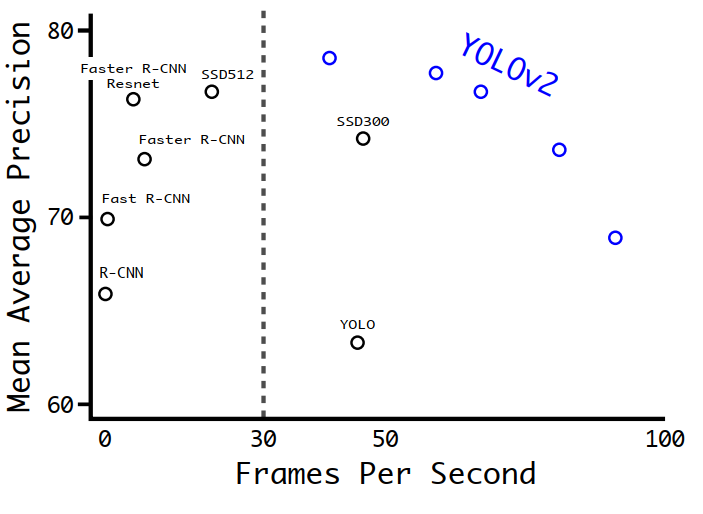


图4. VOC2007上，各个框架的比较

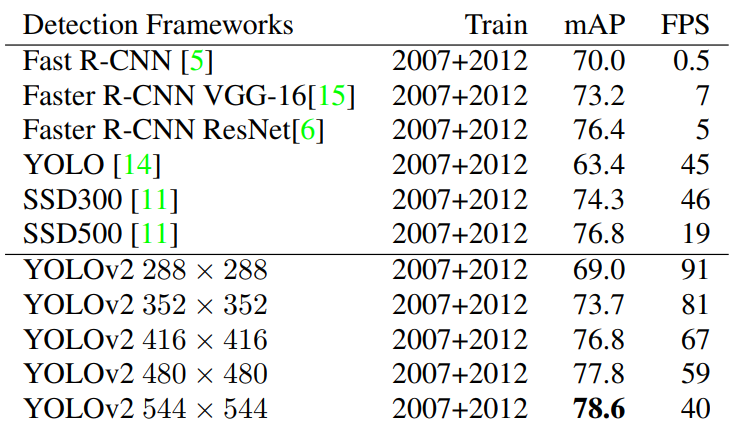


表3 检测框架在PASCAL VOC 2007 上的比较。YOLOv2比以前的检测方法更快，更准确。 它还可以在不同的分辨率下运行，以便在速度和精度之间轻松权衡。 每个YOLOv2模型实际上都是具有相同权重的相同训练模型，只是以不同的大小进行了评估。 所有时序信息均位于Geforce GTX Titan X（原始型号，而非Pascal型号）上。

**进一步实验.** 我们在VOC 2012对YOLOv2进行训练。表4显示了YOLOv2与其他最新检测系统的性能比较。 YOLOv2的mAP达到了73.4，而运行速度却比其他竞争方法快得多。 我们将模型在COCO数据集上进行了训练，并与表5中的其他方法进行了比较。在VOC度量标准（IOU = 0.5）上，YOLOv2获得44.0的mAP，与SSD和Faster R-CNN相当。

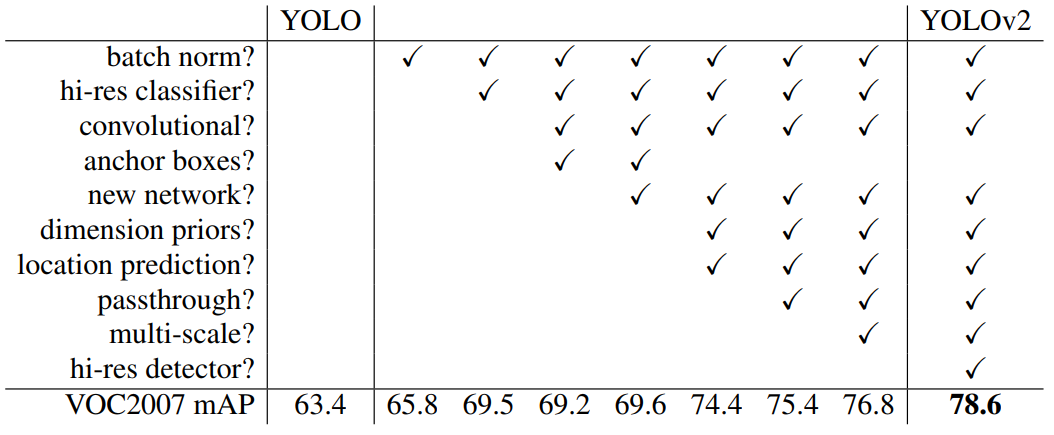


表2. 从YOLO到YOLOv2的变化，以及带来的mAP提升。列出的大多数设计都会使mAP显着增加。 两个例外是切换到具有锚点盒的完全卷积网络并使用新网络。 切换到使用锚框的方式，可以在不更改mAP的情况下增加召回率，同时使用新的网络可以减少33%计算量。

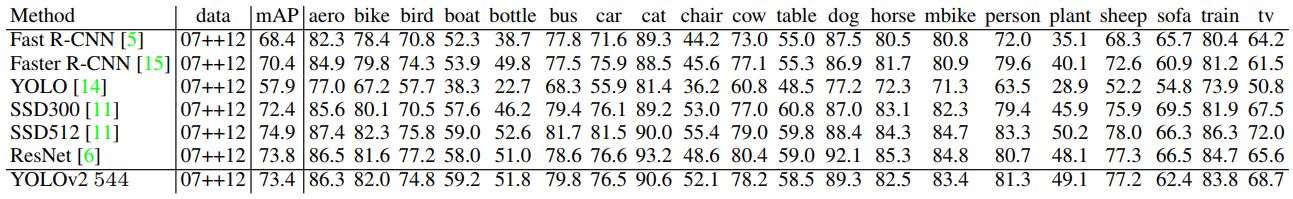


表4. PASCAL VOC2012 检测测试结果。YOLOv2与目前最先进的目标检测器（Faster R-CNN, SSD）性能相当，但是检测速度要快2-10倍。

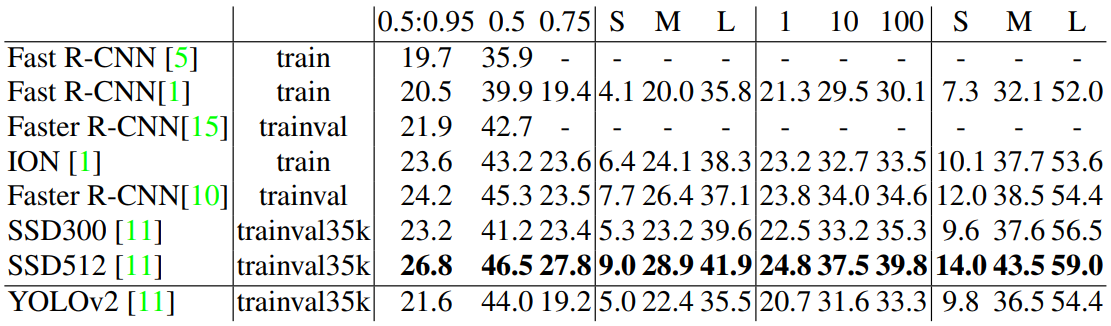


表5 COCO test-dev2015数据集的测试结果，表格改编自文献[11]

* 1. 更快

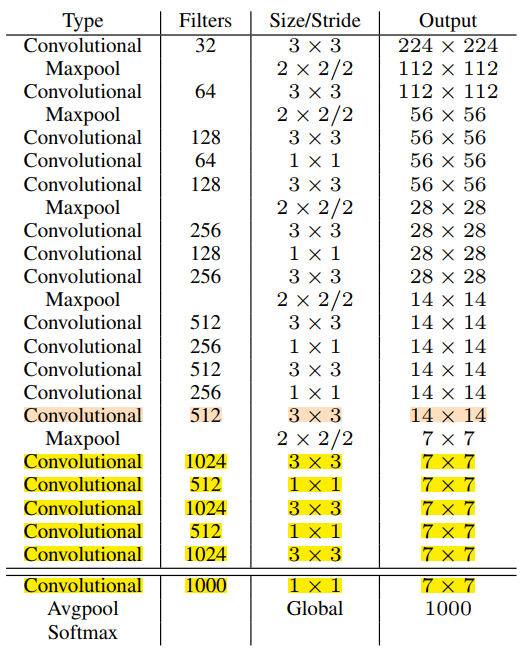
我们想让模型检测即快且准。大多数检测应用，如机器人或自动驾驶汽车，依赖于低延迟预测。为了最大限度地提高性能，我们将YOLOv2设计成尽可能快速检测的模型。

大多数检测框架依赖VGG-16作为基本特征提取器[17]。VGG-16是一个强大、准确的分类网络，但它过于复杂。VGG-16的卷积层在224×224分辨率下对单个图像进行一次处理需要306.9亿次浮点运算。

YOLO框架使用基于Googlenet架构的自定义网络[19]。 该网络比VGG-16更快，一次前向传播仅需要85.2亿次浮点运算。 但是，其准确性比VGG-16差一点。 对于单一裁剪，图片尺寸为为224×224的图片，YOLO的自定义模型在ImageNet上的获得88.0％的Top 5准确率，而VGG-16的Top5 准确率为90.0％。

**Darknet-19.** 我们提出了一个新的分类模型作为YOLOv2的骨干网络。我们的模型建立在以往网络设计的经验和该领域的通用理论的基础上。与VGG模型类似，我们主要使用3×3滤波器，并且在每个池化步骤后将通道数加倍[17]。根据网络中微型网络(Network in Network, NIN)的工作经验，我们使用全局平均池化做预测，以及在3X3的特征之间使用1X1的卷积核对特征表示进行压缩。我们使用批归一化来稳定训练，加速网络的收敛，并且正则化模型。

我们的最终模型称为Darknet-19，具有19个卷积层和5个maxpooling层。 有关完整说明，请参阅表6。Darknet-19仅需要55.8亿次操作即可处理图像，但在ImageNet上达到了72.9％的top-1精度和91.2％的top-5精度。



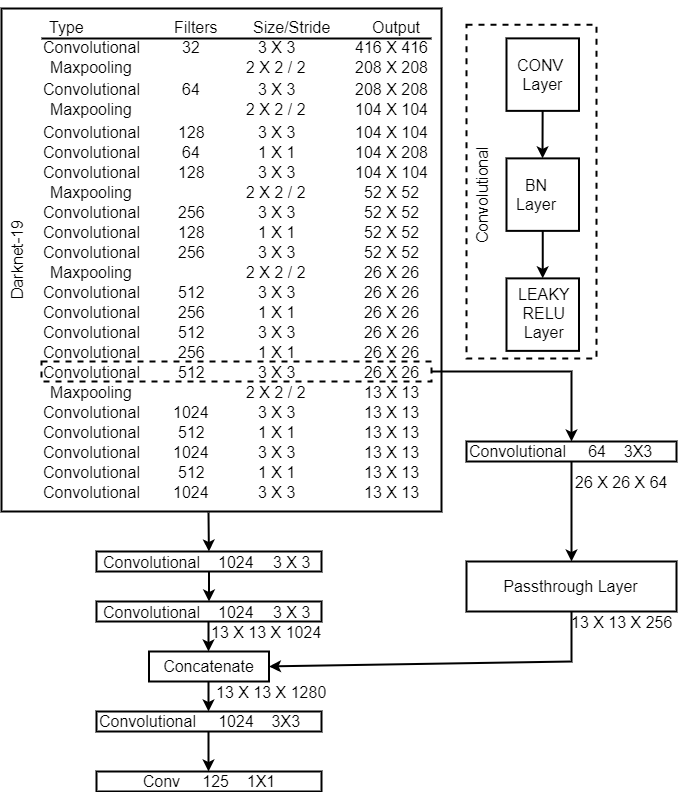


表6 Darknet-19

**分类训练.** 我们在ImageNet上包含1000各类别的分类数据集上训练网络160个epoch，初始学习率为0.1，多项式衰减率为10的4次幂，权重衰减为0.0005，以及角动量为0.9，骨干网络为Darknet网络框架。在训练过程中，我们使用标准的数据增强方法，包括随机裁剪，旋转，色调调整，改变饱和度以及曝光偏移。

如上所述，在对224×224的图像进行初步训练之后，我们以较大的尺寸448对网络进行了微调。对于此次微调，我们仍然使用上述参数进行训练，但是，仅训练10个epoch，并且以10^3的学习速率开始。 在这种更高的分辨率下，我们的网络可实现76.5％的top-1精度和93.3％的top-5精度。

**检测训练.** 为了进行检测，我们对网络进行了修改，去掉了最后的卷积层，以三个具有1024个 3 X 3 卷积核的卷积层代替，每一个3X3卷积层后面跟一个包含1X1卷积核的卷积层来获得我们检测所需要的输出尺寸。对于VOC数据集，我们预测5个候选框，每一个候选框预测5组坐标以及20个类别，因此需要125个卷积核。我们也在网络中最后一个3 X 3 X 512的卷积层与倒数第二个卷积层之间加入了passthrough层，因此，我们的模型能够利用更加细粒度的模型。

我们训练网络160个epoch，初始学习率为10^-3，在第60个迭代和第90个迭代时，学习率分别除以10。我们使用权重衰减系数为0.0005，动量为0.9，使用YOLO和SSD模型相似的数据增强（随机裁剪，色彩偏移等等）。我们在VOC与COCO数据集上采用相同的训练策略。

* 1. 更强

我们提出了一种分类和检测的联合训练机制。我们的模型利用标签图片去学习特定的检测信息，像边界框坐标和对象以及如何对对象进行分类。通过使用只有类标签的图片数据集来扩充我们模型能够检测的目标类别数量。

训练期间，我们混合检测和分类数据集中的图片。当我们的网络输入的是一张标记为分类的图片时，我们使用完整的损失函数进行反向传播。当网络输入的是一张分类图片时，我们仅对网络中的分类部分进行损失反向传播。

这个方法带来了几个挑战。检测数据集仅仅有一些普通的对象和标签，像“狗”，“船”等等。而分类数据集具有许多更广更细致的分类。ImageNet数据集中有超过一百种狗，包括“诺福克梗”、“约克郡梗”和“贝灵顿梗”。如果要对两个数据集进行训练，则需要一种一致的方式来合并这些标签。

大多数分类方法在所有可能的类别上使用softmax层来计算最终概率分布。使用softmax假定类是互斥的。 这带来了合并数据集的问题，例如，您不希望使用此模型将ImageNet和COCO合并，因为“诺福克梗”和“狗”类不是互斥的。

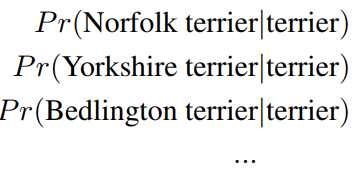
相反，我们可以使用一个多标签模型来合并数据集，不假定它们是互斥的。 这种方法忽略了我们对数据了解的所有结构，例如，所有COCO数据集中的类别都是互斥的。

**层次化分类.** ImageNet标签是从WordNet中提取的，WordNet是一个语言库，它构造概念以及它们之间的关系[12]。在WordNet中，“诺福克梗”和“约克郡梗”都是“梗”的下位词，“梗”是“猎狗”的一种，猎狗是“狗”的一种，狗是“犬”的一种，等等。大多数分类方法都假设标签的结构是扁平的，但是对于组合数据集，结构正是我们所需要的。

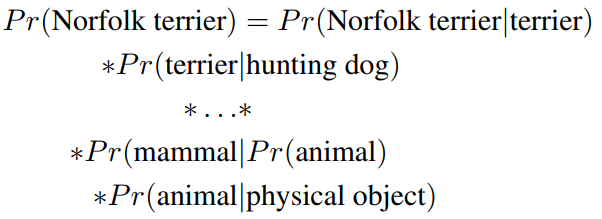
由于语言很复杂，WordNet的结构是有向图，而不是树。 例如，“狗”既是“犬”的类型，又是“家畜”的类型，它们都是WordNet中的同义词集。我们不使用完整的图结构，而是通过根据ImageNet中的概念构建层次树来简化问题。

为了构建这棵树，我们检查了ImageNet中的视觉名词，并查看了它们在WordNet有向图中到达根节点（在本例中为“物理对象”）的路径。许多同义词集在WordNet图中只有一条路径，因此首先我们将所有这些路径添加到树中。 然后，我们迭代检查剩下的概念，并添加使树长得尽可能小的路径。 因此，如果一个概念有两个到根节点的路径，一个路径会给树增加三条边，而另一个路径只会增加一条边，那么我们选择较短的路径。

最终的结果叫做WordTree，一个视觉概念的层次化模型。为了用WordTree执行分类，我们在给定同义集的情况下，针对该同义集的每个下位词的概率预测关于每个节点的条件概率。例如，在“梗”这个节点，我们预测：



如果要计算特定节点的绝对概率，我们只需沿着树到根节点的路径，对路径上的条件概率进行相乘。 因此，如果我们想知道图片是否为诺福克梗，我们可以计算：



出于分类目的，我们假设图像包含一个对象：Pr(physical object) = 1。

为了验证这个方法，我们在使用ImageNet数据集中1000个类构建的WordTree上训练Darknet-19模型。为了构建WordTree1k，我们添加了所有中间节点，这将标签空间从1000扩展到1369。在训练过程中，我们在树上传播ground truth真实标签，因此，如果一个图片被标记为“诺福克梗”，则还应该将其标记为“狗”和“哺乳动物”，等等。为了计算条件概率，我们的模型预测了一个具有1369个值的向量，我们计算所有属于同一概念的下位词的同义集的softmax，请参见图5。

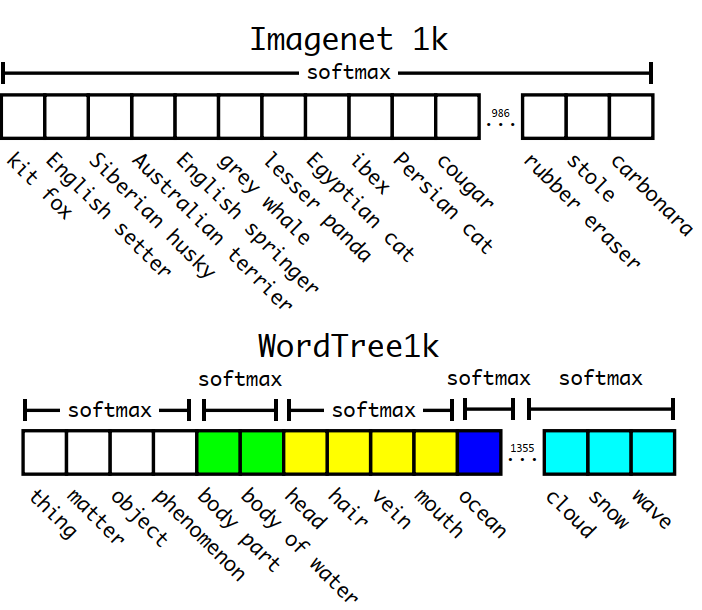


图5：在ImageNet和WordTree上的预测。 大多数ImageNet模型使用一个较大的softmax来预测概率分布。 使用WordTree，我们对同义词执行多个softmax操作。

使用与以前相同的训练参数，我们的分层Darknet-19达到71.9％的top-1准确性和90.4％的top-5准确性。 尽管添加了369个其他概念，并且我们的网络预测了一个树结构，但我们的准确性仅下降了一点。以这种方式执行分类也有一些好处。 对于新的或未知的对象类别，性能会正常降低。 例如，如果网络看到一条狗的图片，但不确定它是哪种类型的狗，它仍将以较高的置信度预测“狗”，但是仍然会在下义词中会散布较低的置信度。

该方法也可用于检测。 现在，我们不用假定每张图片都有一个对象，而是使用YOLOv2的对象预测器为我们提供Pr(physical object)的值。检测器预测一个边界框和概率树。我们向下遍历概率树，在每次分裂时采用最高的置信度的路径，直到达到某个阈值并预测为该对象类。

**数据集与WordTree结合.** 我们可以使用WordTree以合理的方式将多个数据集组合在一起。我们仅将数据集中的类别映射到树中的同义词集。 图6显示了使用WordTree组合ImageNet和COCO数据集中标签的示例。 WordNet非常多样化，因此我们可以对大多数数据集使用此技术。

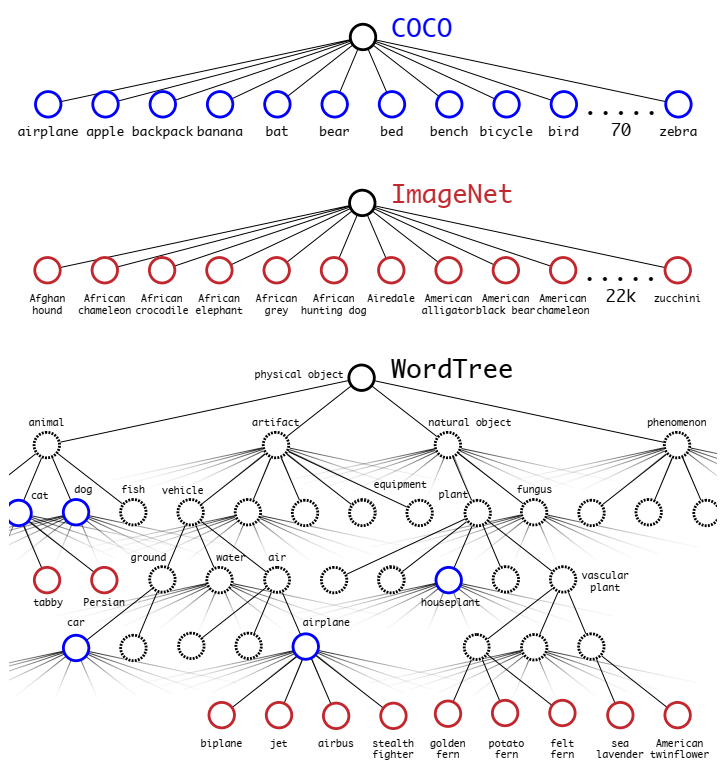


图6 利用层次化的WordTree来组合数据集。 使用WordNet概念图，我们可以构建关于视觉概念的层次树。 然后，我们可以通过将数据集中的类映射到树中的同义词集来将数据集合并在一起。 这是WordTree的简化视图，用于说明目的。

**分类检测联合训练.** 既然可以通过WordTree来组合我们的数据集，因此可以对模型进行分类和检测的联合训练。我们想训练一个超大型检测器，因此我们使用COCO检测数据集和ImageNet完整发行版中的前9000个类创建组合数据集。 为了对我们的模型进行评估，因此我们添加ImageNet检测挑战中尚未包含的所有类。 此数据集的相应WordTree具有9418个类别。 与COCO数据集相比，ImageNet是一个更大的数据集，因此我们通过对COCO进行上采样来平衡数据集，以使两个数据集的比例仅为4：1。

使用这个数据集，我们对YOLO9000进行训练。使用YOLO的基本结构，但是只用了3个先验锚框（不是前面所说的5个），以便于限制网络的输出大小。当网络输入的是一个目标检测图片时，正常的对误差进行反向传播。对于分类误差，仅仅传播在该标签相关的层级及其以上的误差。例如，如果标签是“狗”，我们会将所有的预测误差都分配到概念树的下部（“德国牧羊犬”以及“金毛犬”），因为标签确实没有相关的更详细的分类信息。

当网络输入的是一张分类图片时，模型仅仅反向传播分类误差。为此，只需要找到预测这个类别概率最高的边界框，然后在它的预测树上计算损失。我们还假设预测边界框与真实边框的交并比至少为0.3，在这个假设的基础上，来反向传播损失。

通过这种联合训练，YOLO9000 利用COCO数据集中的监测数据来学习发现图片中的目标，利用ImageNet中大量而广泛的类别图片来学习分类。

我们在ImageNet检测任务数据集上评估YOLO9000。 ImageNet的检测任务数据集与COCO数据集中有44个对象类别是相同的，这意味着在所有的测试图片中大多数都是分类数据，而不是检测数据。 YOLO9000总体获得19.7 mAP，在目标检测数据中不存在的156个对象类别上获得了16.0 mAP的性能。该mAP高于DPM所获得的结果，但是YOLO9000在半监督的情况下，使用结构不同的数据集进行了训练[4]。 它还同时实时检测9000个其他对象类别，全部都是实时检测。

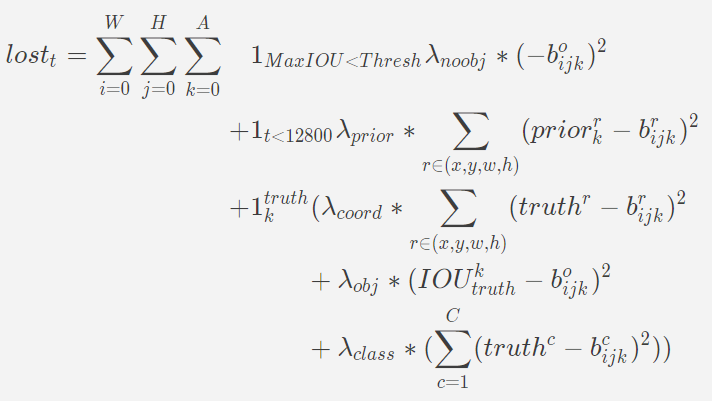
当我们在ImageNet上分析YOLO9000的性能时，我们会看到它很好地学习了新的动物种类，但在学习服装和设备等类别方面却步履维艰。新的动物种类更容易学习，因为目标预测可以从COCO中的动物进行很好的泛化。相反，COCO数据集中没有任何服装的边界框标签，只是对人进行了标注，因此YOLO9000很难为“太阳镜”或“游泳裤”等类别建模。

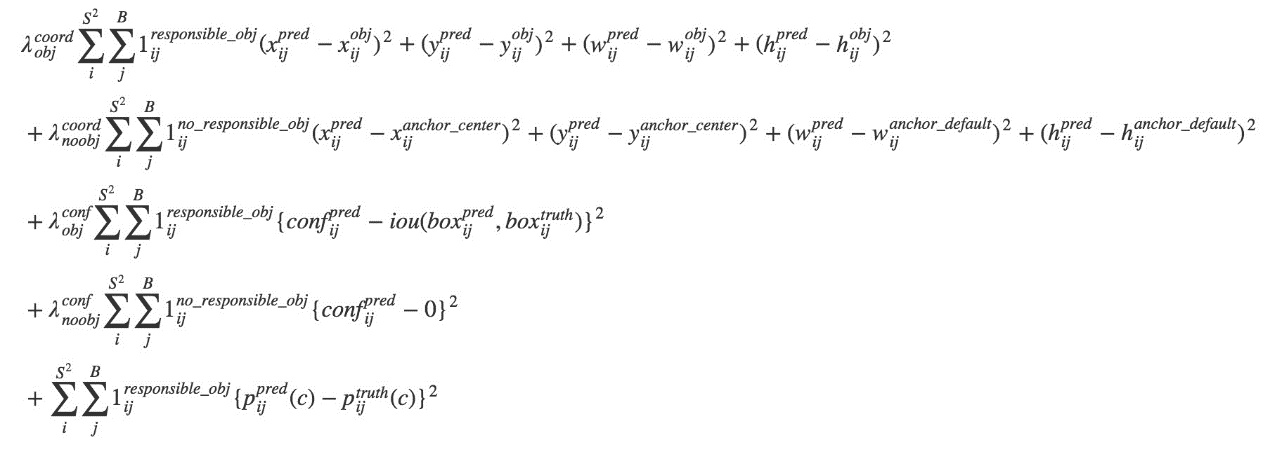
* 1. 损失函数

下式定义了在一个迭代中的损失计算函数，由三部分组成。第一项中，如果所有的IOU均小于阈值，即边界框内没有包含任何对象，对应的置信度损失。第二项中，由于预测边界框的坐标与先验框的坐标保持一致，此项表示前12800个迭代中的先验框与预测边界框的损失。第三项中，负责预测一个目标对象的边界框与该目标对象的真实边界框的位置误差，置信度误差以及分类误差。

1t < 12800意思是前12800次迭代计入误差。注意这里是与先验框的误差，而不是与真实对象边框的误差。可能是为了在训练早期使模型更快学会先预测先验框的位置。

λ 是对各个误差项的预设权重。





损失函数分为5项，下面分别解释：

第一项：负责预测物体的anchor的xywh损失。如果anchor负责预测物体，那么需要计算坐标的L2损失。

第二项：不负责预测物体的anchor的xywh损失。如果anchor不负责预测物体，那么需要在迭代的初期（比如iteration<12800）去计算坐标的L2损失。问题的关键是，anchor都不负责预测物体，那么它的预测目标是什么呢？答：预测目标是anchor的xywh。为什么要这么做？我的理解是，这么做可以让所有anchor的预测都接近anchor自身的xywh，这样当它有物体落入这个anchor的时候，anchor的预测不至于和目标差别太大，相应的损失也会比较小，训练起来会更加容易。

第三项：负责预测物体的anchor的confidence损失。负责预测物体的anchor需要计算confidence损失，confidence的目标就是预测的bbox和真实bbox的iou。

第四项：不负责预测物体的anchor的confidece损失。对那些不负责预测gt的anchor，需要计算每个anchor和所有gt box的IOU。如果算出来的最大IOU<0.6，相应的



*，*并且confidence的label就是0。但是，如果这个值大于0.6，相应的 

 ，也就是说，这个时候是不算这个anchor的confidence损失的。为什么要这么做呢？我的理解是，当anchor不负责预测物体的时候，如果它预测出来的结果和真实值差别很大的话，那代表它是没有物体的，那么这个时候就希望它的预测的confidence接近0。但是如果预测的结果和真实值比较接近的话，则不计算损失。

第五项：负责预测物体的anchor的类别损失。每个类别的输出概率0-1之间，计算的是L2损失。也就是说分类问题也把它当做了回归问题。另外需要注意的是，类别预测中是不需要预测背景的，因为confidence实际上就已经代表是否存在物体，类别就没必要去预测背景。

* 1. 结论

本文介绍了实时检测系统YOLOv2和YOLO9000。 YOLOv2是最先进的，并且在各种检测数据集中比其他检测系统要快。 此外，它可以在各种图像尺寸下运行，以在速度和精度之间提供良好的平衡。

YOLO9000是一个实时检测框架，可通过联合优化检测和分类来检测9000多个对象类别。 我们使用WordTree合并来自各种来源的数据，并使用联合优化技术在ImageNet和COCO上同时进行训练。YOLO9000是朝着缩小检测和分类数据集的差距迈出的重要一步。

我们的许多技术都在对象检测之外推广。 ImageNet的WordTree表示形式为图像分类提供了更丰富，更详细的输出空间。 使用分层分类的数据集组合在图像分类和图像分割领域中将很有用。 像多尺度训练这样的训练技术可以在各种视觉任务中提供帮助。

对于未来的工作，我们希望使用类似的技术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（在训练过程中为分类数据分配弱标签）来改善检测结果。 计算机视觉拥有大量的标记数据。 我们将继续寻找方法，将不同的数据源和数据结构整合在一起，以建立更强大的视觉世界模型。