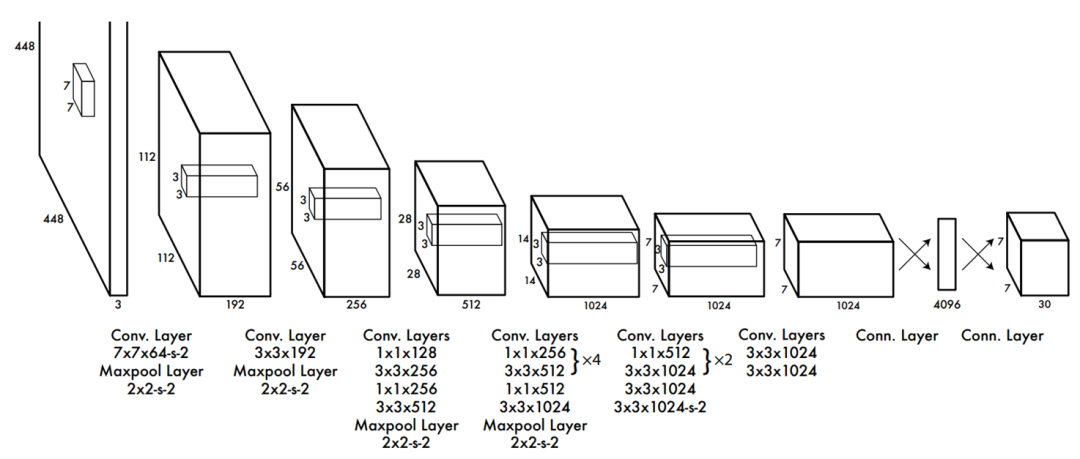
**深度学习之目标检测算法YOLO发展历程**

目标检测是计算机模拟人眼，用来解决图像中有什么对象，它们在哪里，以及如何交互等问题的算法。不论在智能交通，工业监测，还是医学领域，目标检测都有很多应用。在2016年以前，基于深度学习的目标检测算法主要通过遍历性的分类任务来实现，如DPM、R-CNN等。这些算法遍历地越精确，检测器的精确程度则越高，但相应的代价是消耗巨大的时间和空间成本。且由于每个类别都是单独训练的，因此我们很难对这一类算法进行优化。YOLO算法将目标识别重新架构成了一个回归问题，它可以直接通过图像像素预测出边界坐标和类概率。也就是说，在YOLO体系中，你只需要对一个图片观察一次（you only look once）就能预测出这个对象是什么以及它在何处[1]。

**YOLOv1**

**基本架构：**

YOLOv1模型为卷积神经网络，该网络包含24个卷积层，后跟2个全连接层。1×1 卷积层交替出现，减少前一层的特征空间。在 ImageNet 分类任务上，使用一半的分辨率（224×224 输入图像）预训练卷积层，然后将分辨率加倍以进行检测。



**基本理念：**

YOLOv1将不同的检测目标统一到了同一个神经网络中。它使用整个图像的特征来预测每一个边界框，且能够预测图像的所有类别的所有边界框。这意味着YOLO网络可以对完整图像和图像中的所有对象进行全局推理。它的设计能够实现在保持较高的平均精度的同时快速地进行端到端训练。

**优点：**

1. YOLO速度非常快，因为回归问题没有复杂的流程（pipeline）。

2. YOLO可以基于整幅图像进行预测。与基于窗口滑动和区域协议的技术不同的是，YOLO在训练和测试时可以看到整个图像，所以它能够隐式地编码有关类及其外观的关联信息。这解决了Fast R-CNN中由于无法看到全局空间而出现背景识别错误率较高的问题。

3. YOLO能够学习到物体的通用表示（generalizable representations），泛化能力好。因此，当训练集和测试集类型不同时，YOLO的表现比DPM和 R-CNN好得多，应用于新领域也很少出现崩溃的情况。

**缺点：**

1. YOLOv1对边界框预测施加了很强的空间约束，因为每个网格单元只能预测一个类的两个框。这种空间约束限制了模型可以预测的附近物体的数量，即难以处理成群出现的小物体，例如成群的鸟。

2. 由于模型学习从数据中预测边界框，因此它很难泛化到具有新的或不寻常的纵横比或配置的对象。

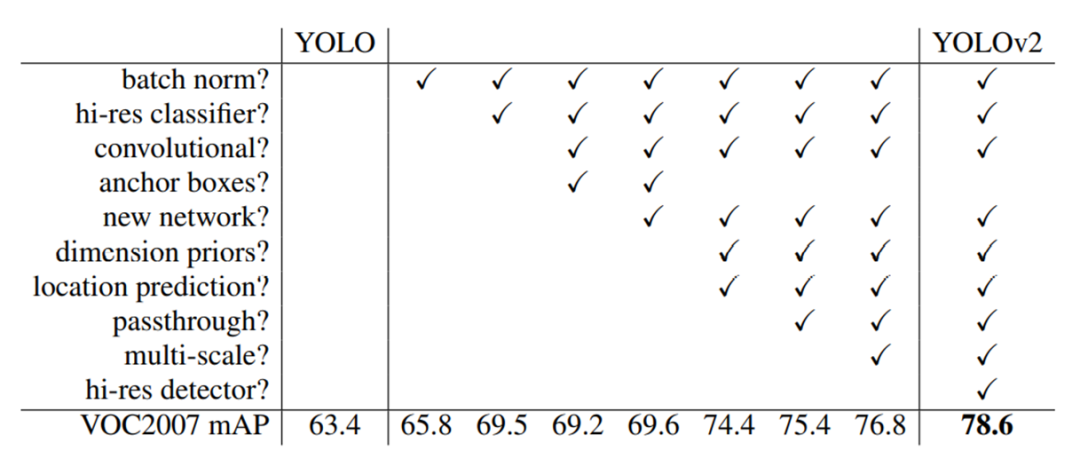
3. 无论边界框的大小都用损失函数近似为检测性能，物体IOU误差和小物体IOU误差对网络训练中loss贡献值接近，但对于大边界框来说，小损失影响不大，而对于小边界框，小错误对IOU影响较大。这种处理方式降低了物体检测的定位准确性。

**性能：**

YOLO的背景错误比 Fast R-CNN 少得多。它在VOC 2007上有很好的表现，并且在应用于艺术品时，其 AP 的衰减比其他方法少。与 DPM 一样，YOLO 将对象的大小和形状，以及对象和其通常出现位置之间的关系进行建模。尽管不同模型在艺术品和自然图像的像素级别上有很大不同，但它们在对象的大小和形状方面相似，因此 YOLO 仍然可以预测良好的边界框以及进行检测。

**YOLOv2**

由于YOLOv1相对于最先进的检测系统某些方面尚存在不足。例如，YOLOv1的误差分析与Fast R-CNN相比，产生了大量的定位错误。并且与基于区域提议的方法相比，YOLOv1的召回率相对较低。YOLOv2简化了网络结构，使得图像特征（representation）更容易学习，在保持分类准确性的同时提高了召回率和定位。YOLOv2适用于各种尺寸的图像，且平衡了运行速度和准确性。



图：YOLOv2改进过程[2]

同时，目前拥有标签的检测数据集十分有限，使得检测任务受到了限制，但是标记检测图像的代价又十分高昂。因此YOLOv2在改进YOLOv1的基础上，提出了一种联合训练算法，称为YOLO9000。它是一个通过联合优化检测和分类来检测9000多个物体类别的实时框架，可以利用标记的检测图像来精确定位，同时使用分类图像来增加其标签量和鲁棒性。**YOLOv2相比于YOLOv1更快，更精确，更稳定。**

**YOLOv3**

YOLOv3主要是在YOLOv1和YOLOv2的基础上对模型架构进行了调整。其网络结构解析如下：

1. backbone部分由YOLOv2时期的Darknet-19进化至Darknet-53，加深了网络层数，引入了Resnet中的跨层加和操作。这一变动相比之前提高了精度，处理速度有所下降，但比同精度的ResNet快很多。

2. YOLOv3中，只有卷积层，通过调节卷积步长控制输出特征图的尺寸。所以对于输入图片尺寸没有特别限制。

3. YOLOv3借鉴了金字塔特征图思想，小尺寸特征图用于检测大尺寸物体，而大尺寸特征图检测小尺寸物体。第一层特征图的输出维度恒定为8\*8\*255。

4. 上采样层(upsample)能够将小尺寸特征图通过插值等方法，生成大尺寸图像，不改变特征图的通道数。

YOLO的整个网络，吸取了Resnet、Densenet、FPN的精髓，可以说是融合了目标检测当时业界最有效的全部技巧。

性能：

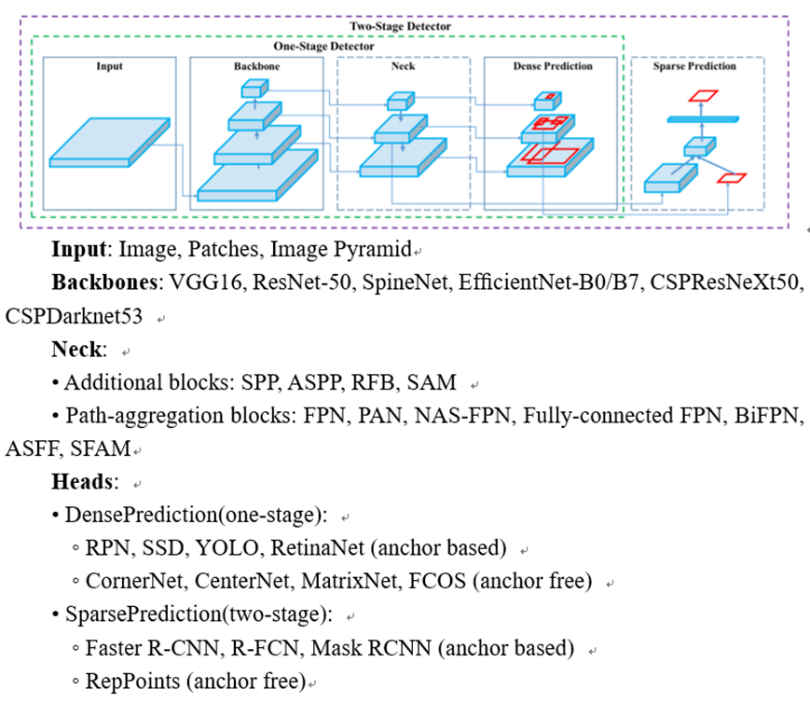
YOLOv3精度与SSD相比略有小优，与Faster R-CNN相比略有逊色，几乎持平，比RetinaNet差一些，但是速度是SSD、RetinaNet、Faster R-CNN的至少2倍以上，在工业界具有极高的应用价值。

**性能：**

**YOLOv3相比于YOLOv2相对更大一些，但是会精确性更高。**

**YOLOv4**

**模型架构：**



**主要贡献：**

1. 设计了强大而高效的检测模型，任何人都可以用1080 Ti和2080 Ti训练这个超快而精准的模型。

2. 验证了很多近几年SOTA的深度学习目标检测训练技巧。

3. 修改了很多SOTA的方法， 让它们对单GPU训练更加高效，例如CBN，PAN，SAM等。

**性能：**

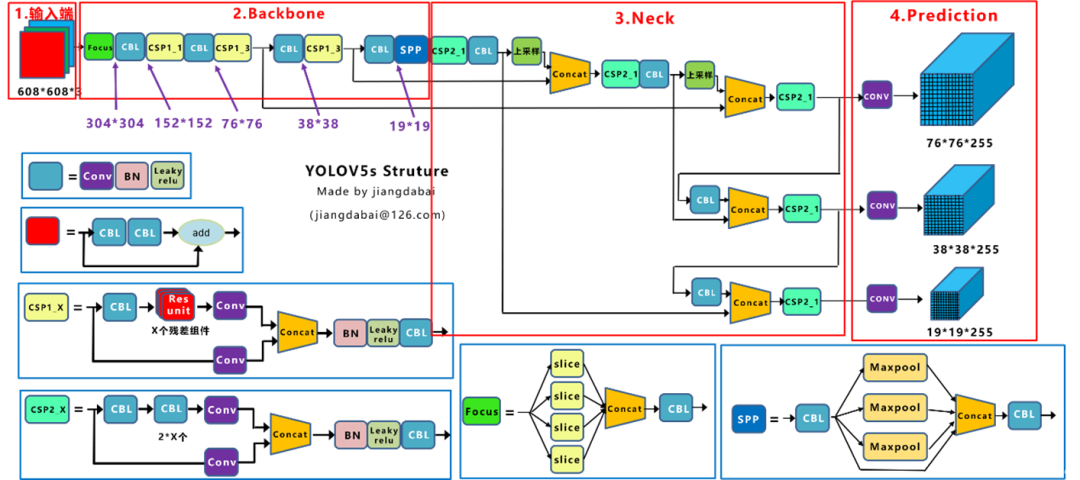
YOLOv4提出了一个最先进的目标检测器，它比当时所有可替代检测器速度更快（FPS），准确性更高（AP）。这个检测器可以仅在一块8-16GB的GPU上进行训练，这使得它可以被广泛的使用。

**YOLOv5**

YOLOv5的整体模型和YOLOv4差别很小，主要特点是权重文件非常小，灵活度比YOLOv4更高，可以搭载在配置更低的移动设备上。而且yolov5的速度较yolov4更快，准确度更高。其它一些改变包括引入了自动anchor重计算策略，采用了最新版本的pytorch进行混合精度以及分布式训练，采用了模型权重指数滑动平均的ema策略（比赛常用策略）以及支持onnx格式导出等。在不久前的小麦检测比赛中期，不少使用YOLOv5的参赛者名次位于前列，而且经常实时的第一名方案就是基于YOLOv5（尽管后面因为YOLOv5的license 是GPL 协议（GPL 3.0）而被禁用），可见其性能表现相当好。

模型架构：

Yolov5代码中有四种网络，和之前的Yolov3，Yolov4中的cfg文件不同，它们都是以yaml的形式来呈现的。四个文件的内容基本一致，只有最上方的depth\_multiple（用于控制网络的深度）和width\_multiple（用于控制网络的宽度）两个参数不同。其整体架构如下图所示：



图：YOLOv5网络架构（Input：Mosaic数据增强、自适应锚框计算；Backbone：Focus结构，CSP结构；Neck：FPN+PAN结构；Prediction：GIOU\_Loss）

**性能：**

Yolov5s网络最小，速度最少，AP精度也最低。如果检测的内容以大目标为主，追求速度，是个不错的选择。其他的三种网络，在此基础上，不断加深加宽，AP精度也不断提升，但速度的消耗也在不断增加。可以根据实际需求，适配相应的网络。

YOLOv6是由美团推出的，所做的主要工作是为了更加适应GPU设备，将2021年的RepVGG结构引入到了YOLO。思路比较简单。

**YOLOv6**

YOLOv6检测算法的思路类似YOLOv5（backbone+neck）+YOLOX（head），此处不再赘述。

**主要改动：**

1.骨干网络由CSPDarknet换为了EfficientRep

2.Neck是基于Rep和PAN构建了Rep-PAN

3.检测头部分模仿YOLOX，进行了解耦操作，并进行了少许优化。

**网络结构**

骨干网络: EfficientRep

Neck: FPN+RepPAN

检测头：类似YOLOX

tricks1：引入RepVGG

按照RepVGG的思路，为每一个3x3的卷积添加平行了一个1x1的卷积分支和恒等映射分支，然后在推理时融合为3x3的结构，这种方式对计算密集型的硬件设备会比较友好。

tricks2：骨干网络EfficientRep

把backbone中stride=2的卷积层换成了stride=2的RepConv层,并且也将CSP-Block修改为了RepBlock。

tricks3：Neck中引入Rep

为了进一步降低硬件上的耗时，将PAN中的CSP-Block替换为RepBlock, 从而生成了Rep-PAN结构。

tricks4：对检测头解耦并重新设计了高效的解耦头为了加快收敛速度和降低检测头复杂度，YOLOv6模仿YOLOX对检测头进行了解耦，分开了目标检测中的边框回归过程和类别分类过程。 由于YOLOX的解耦头中，新增了两个额外的3x3卷积，会在一定程度增加运算的复杂度。鉴于此，YOLOv6基于Hybrid Channels的策略重新设计出了一个更高效的解耦头结构。在不怎么改变精度的情况下降低延时，从而达到了速度与精度的权衡。

**优势：**

对耗时做了进一步的优化，进一步提升YOLO检测算法性能。

**YOLOv7**

YOLOv7是YOLOv4团队的续作，主要是针对模型结构重参化和动态标签分配问题进行了优化。

**思路**

YOLOv7检测算法的思路是与YOLOv4、v5类似。

**主要改动：**

1.提出了计划的模型结构重参化。

2.借鉴了YOLOv5、Scale YOLOv4、YOLOX，“拓展”和“复合缩放”方法，以便高效的利用参数和计算量。

3.提出了一种新的标签分配方法。

**网络结构**

在YOLOv4、YOLOv5、YOLOv6基础上通过添加了以下tricks进行了进一步的升级改造。

tricks1：高效的聚合网络

E-ELAN采用expand、shuffle、merge cardinality结构，实现在不破坏原始梯度路径的情况下，提高网络的学习能力。在体系结构方面，E-ELAN只改变了计算模块中的结构，而过渡层的结构则完全不变。作者的策略是利用分组卷积来扩展计算模块的通道和基数，将相同的group parameter和channel multiplier用于计算每一层中的所有模块。然后，将每个模块计算出的特征图根据设置的分组数打乱成G组，最后将它们连接在一起。此时，每一组特征图中的通道数将与原始体系结构中的通道数相同。最后，作者添加了G组特征来merge cardinality。

tricks2：模型缩放

类似于YOLOv5、Scale YOLOv4、YOLOX，一般是对depth、width或者module scale进行缩放，实现扩大或缩小baseline的目的。

tricks3：引入了卷积重参化并进行了改进采用梯度传播路径来分析不同的重参化模块应该和哪些网络搭配使用。同时分析出RepConv中的identity破坏了ResNet中的残差结构和DenseNet中的跨层连接，因此作者做了改进，采用没有Identity连接的RepConv结构进行卷积重参数化。下图是设计的用于PlainNet和ResNet的计划重参数卷积。

tricks4：引入了辅助训练模块-coarse-to-fine（由粗到细）引导标签分配策略

常用的方式是图（c）所示，即辅助头和引导头各自独立，分别利用ground truth和它们（辅助头、引导头）各自的预测结果实现标签分配。YOLOV7算法中提出了利用引导头的预测结果作为指导，生成从粗到细的层次标签，将这些层次标签分别用于辅助头和引导头的学习，如下图（d）和（e）所示。

**优势**

参数量和计算量大幅度减少，但性能仍能保持少量的提升。