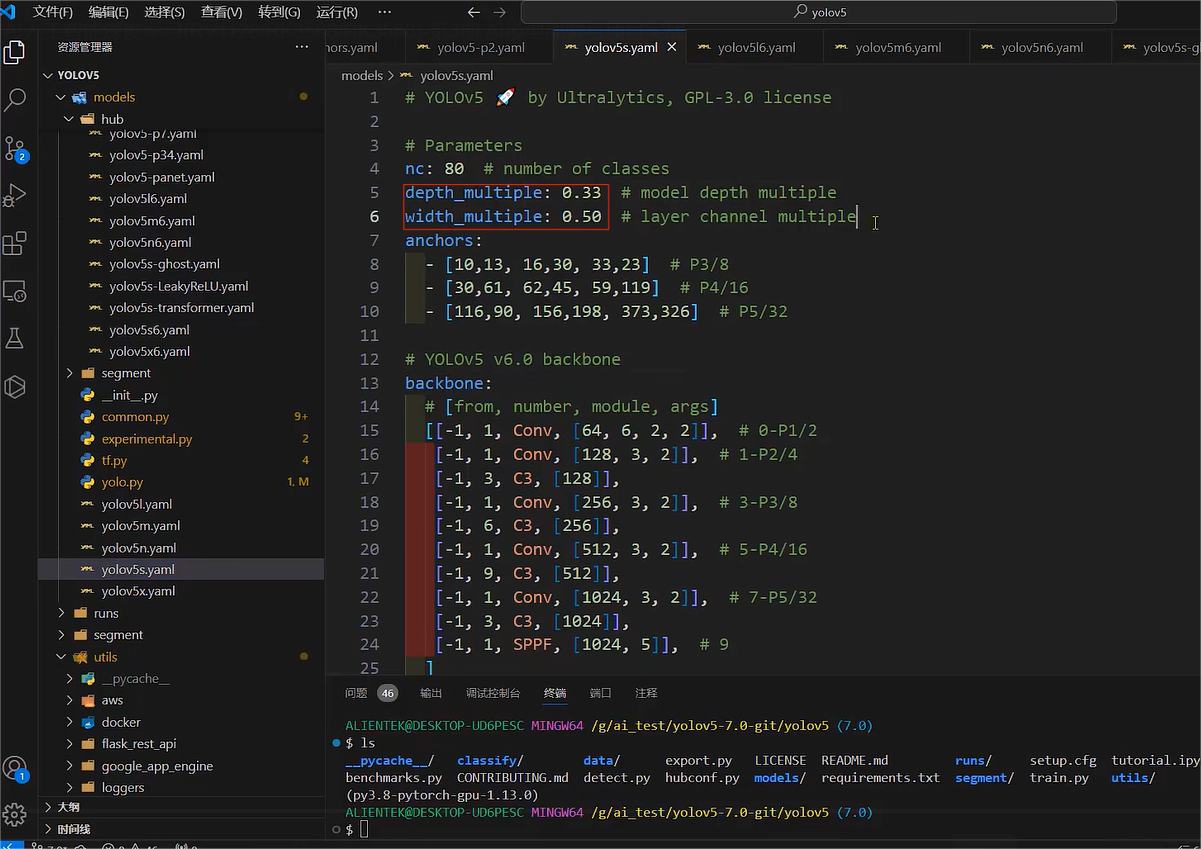
本节以YoLov5s.yaml这个配置文件来了解Yolov5模型的网络架构：



在前面的课程里我们在介绍这些参数的时候，提到上面两个参数是用于控制模型的深度和通道宽度的。

Depth\_multiple是YOLOv5模型的深度因子，控制模型的深度（即网络层数 1/3 = 0.33，相当于把网络的深度缩小了三倍）。

width\_multiple是YOLOv5模型的宽度因子，控制模型的通道宽度（即通道数，这里值为0.5表示将通道宽度缩小2倍）。



这几个配置文件不同之处主要就是这几个参数值不一样。其它参数都是一样的。

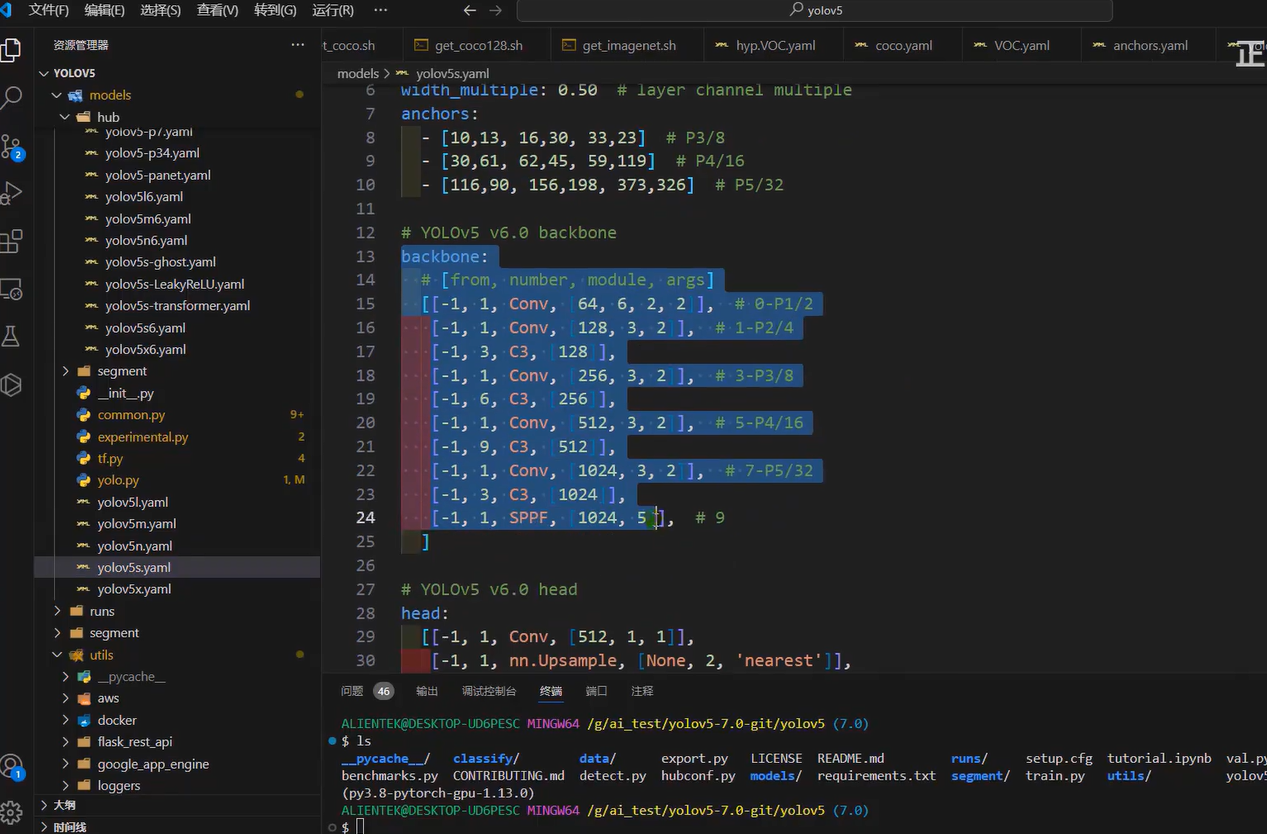
里面的anchors、backbone、head、这些参数配置都是一样的。

说明这几个不同版本的模型，他们在模型的架构上是一样的，只是在深度和宽度上做了调整。（小模型（5n、5s）其参数是比较小的，5l、5x的值就比较大了）这两个参数就是用于调整模型的大小和复杂度的。下面就具体讲解其如何控制模型的宽度和复杂度的。

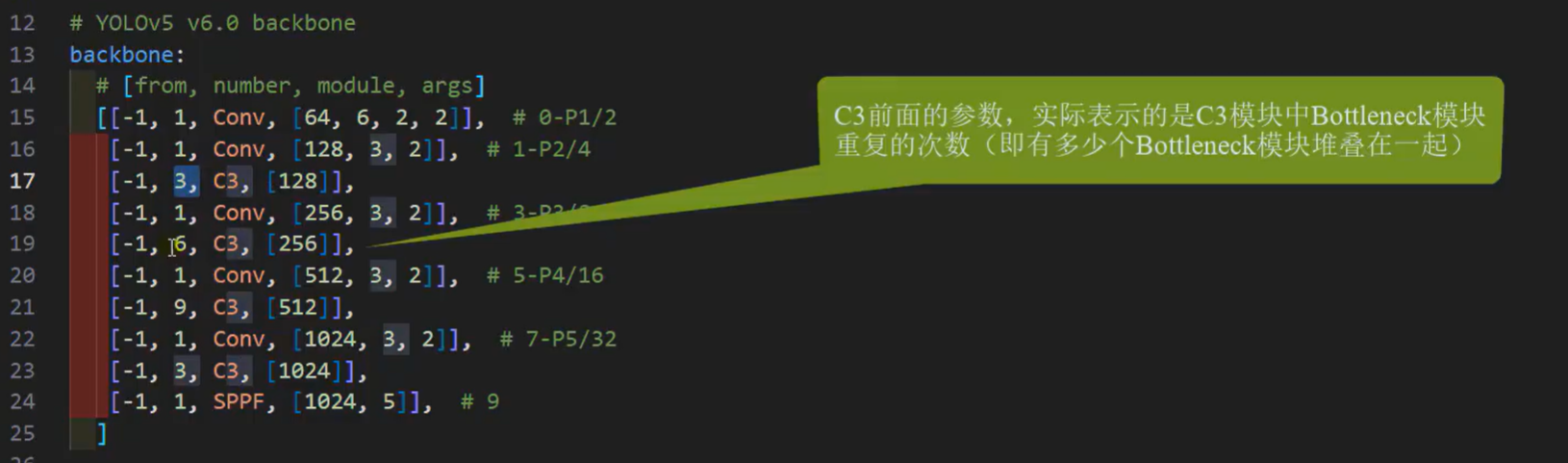
首先是模型的深度因子，用于控制模型的深度，模型的层数越多，网络就越深。参数量就越大，计算量就越大。推理的时候就越耗内存，对硬件的要求越高。如果模型的层数越少，网络就越浅，参数就越小。计算量就越小。推理的时候就越节省内存和带宽。但这样就降低了模型的性能。这个参数用于控制模型的深度，具体来说就是控制模块的重复次数：

/\*

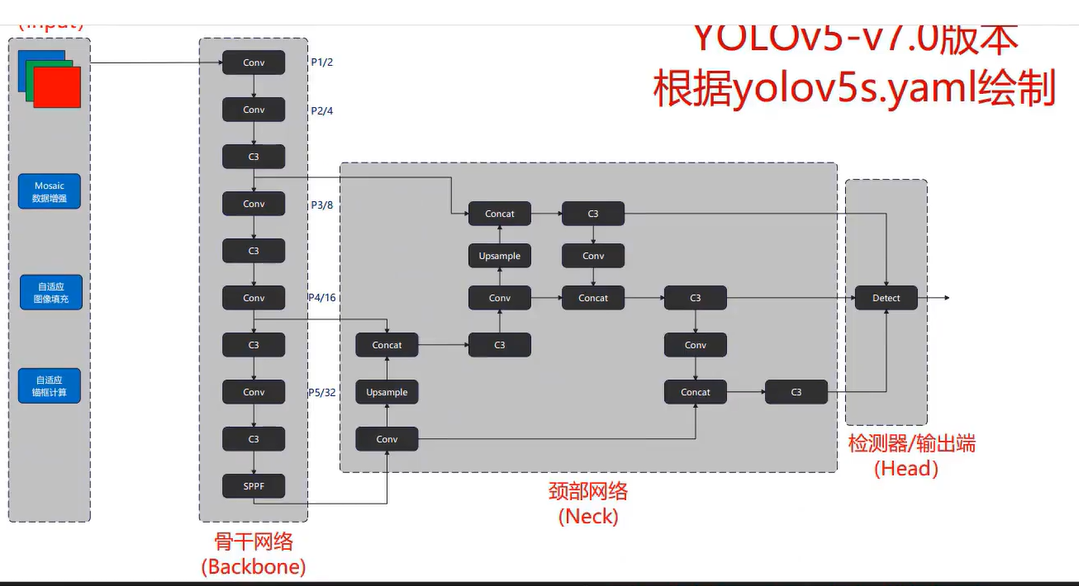
以以下文件为例子：



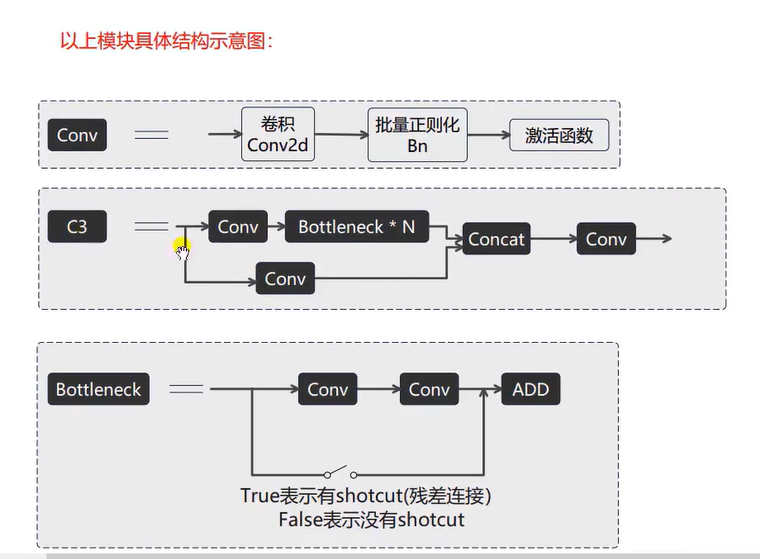
C3模块的参数这里，



也就是：



C3模块实际的样子如下：



Depth\_multiple(模型的深度因子)这个参数就表示C3里的Bottleneck这个模块重复的次数。N就表示Bottleneck重复了多少次。也就是C3模块中有多少个Bottleneck堆叠在一起。

C3前面是3，所以表示Bottleneck重复3次，但实际上并不是重复了3次，而是一次。因为经过前面Depth\_multiple参数的配置（0.33）这里就表示1/3，那么C3前面的3乘以 1/3，约等于1。所以上面那一层的C3模块中Bottleneck相当于重复了1次。

也就是说 Bottleneck实际的重复的次数 = C3前面的参数 \* 模型的深度因子。

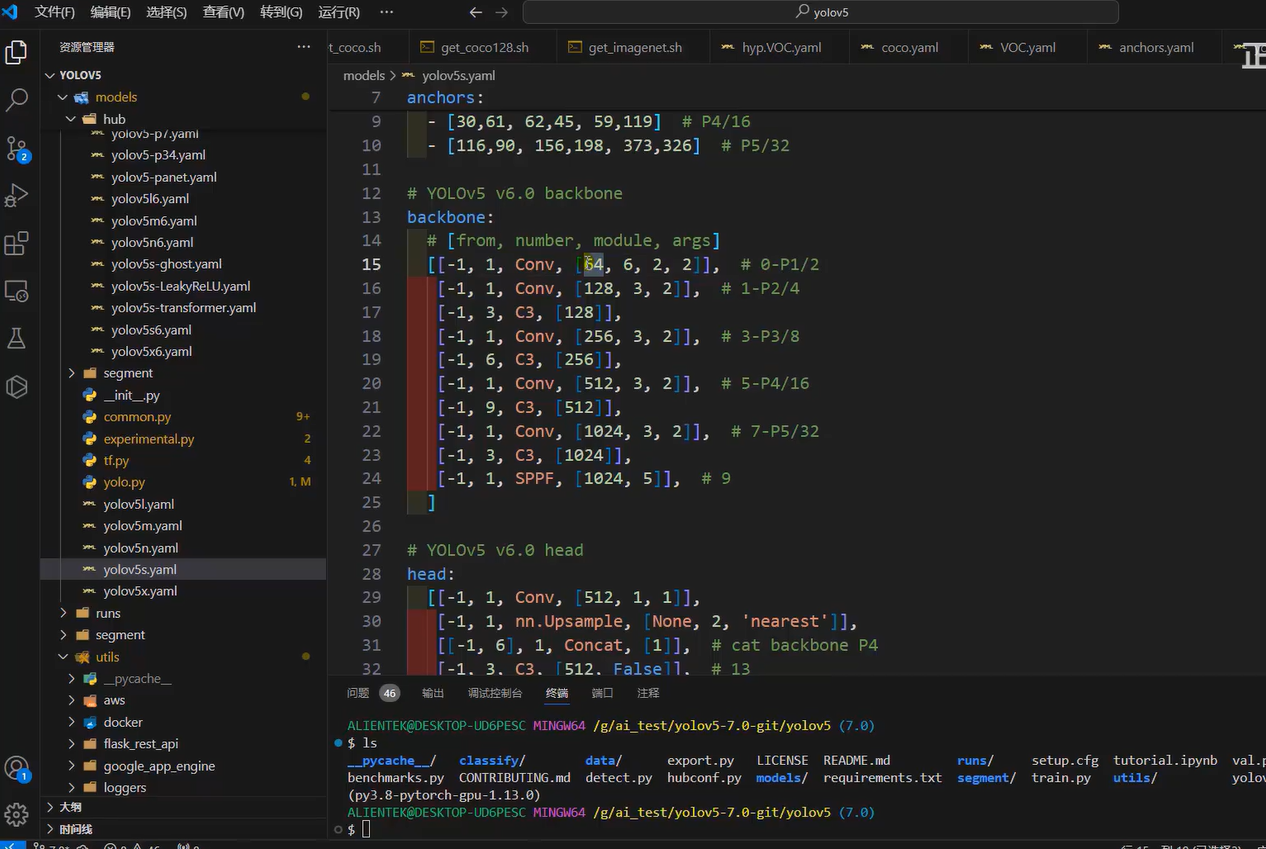
所以模型的深度因子值越大，网络就越深，网络越深就有助于模型学习更复杂的特征表示，但模型的参数量就越大了，模型文件就越大了，就增加了计算量。推理的时候就比较消耗内存

\*/

再来看模型的宽度因子width\_multiple，模型的宽度也就是通道数。增加卷积层的通道数，可使模型能够捕捉更丰富的特征信息；但也增加了模型的参数量，也增加了计算成本。

增加通道数以后，也增加了模型的参数量，增加了计算成本。减小卷积层的通道数，可以减少参数量，但可能影响模型的表示能力。

这里模型的宽度因子值为0.5，再回到骨干网络部分：



这里输出的通道数是64，但实际上输出的应该是32（64\*0.5）。

总的来说，这两个参数可以影响模型的参数量，影响模型的表达，影响计算复杂度和内存的占用。例如根据计算资源、推理的速度、精度要求等，来调节这两个参数，从而定制模型，如果模型要部署在资源有限的场景中，如移动设备/边缘设备就需要调小这两个参数因子。

下一节根据这些问价画出YOLOv5的网络架构。