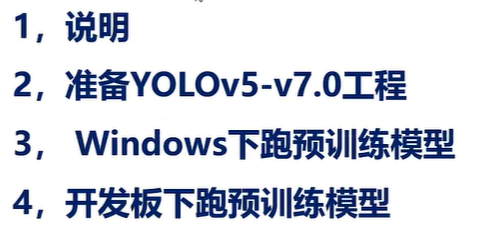
本节开始就进行实战：讲解怎么训练一个YOLOv5-v7.0的工程。训练完成后，怎么部署到RV1126开发板上。RK3568/88等开发板可以借鉴这个课程

/\*



本讲课程的内容，我们先按照顺序逐个讲解。

\*/

/\*

一、有几点注意事项需要进行说明。



1. YOLO的版本说明：

YOLO有许多版本，根据不同的研究方向，版本的划分情况有不同的说法。例如现在普遍认为YOLO有8个版本。从YOLOv1到v8共8个版本。后续可能还有YOLOv9等。

随着时间的推移，逐渐衍生出了YOLO的不同并行体。这些并行体在保持YOLO的核心思想上针对特定的应用场景/需求进行定制。一般就是在主要的版本上进行微调/改进。然后得到新的YOLO衍生版本。如YOLOX就是旷视科技开源的，以YOLOv3为主体经过修改得到的。其它还有YOLOv3-SPP1、YOLOv3-SPP3等等。这些都是YOLO的各种衍生版本/并行体。

对于衍生版本的了解就需要查看论文和文献了，去了解其设计思想和应用场景等。目前主要使用的就是v3和v5版本。因为这两个版本具有高效的检测性能和出色的设计精度。为用户提供了丰富的训练选项。对用户来说比较易于扩展和改进，方便用于工业应用中。

/\*了解即可，重点使用YOLOv5

YOLOV3采用Darknet-53的网络作为骨干网络，也就是采用Darknet-53网络作为特征提取器。Darknet-53是一个全卷积网络，网络共有53层的卷积层和池化层。能够提取出图像中丰富的特征。YOLOv3同v1、v2相比，其采用多尺度、锚框（Ancher）的训练。可以在3个不同尺度上进行预测。采用多尺度训练在预测的时候提高了对小目标的检测能力。

总的来说YOLOv3采用的策略能够提高模型的精度和鲁棒性（指系统在面临内部结构和外部环境变化时，保持其性能和功能稳定的能力‌。具体来说，一个具有鲁棒性的系统能够在各种干扰、噪声、故障等不利因素的影响下，仍然能够正常运行，并且保持较好的性能表现）。

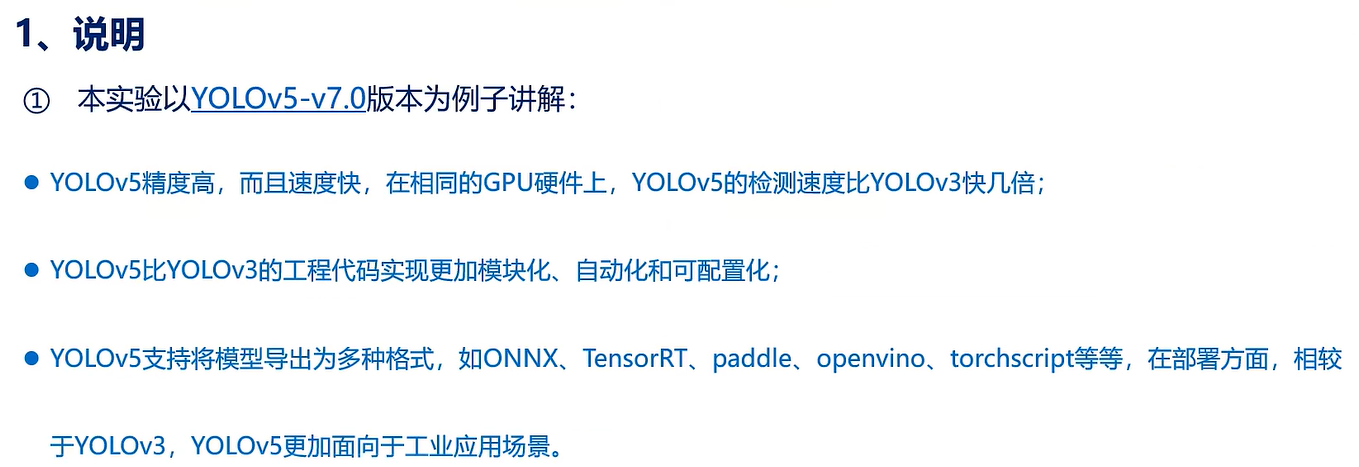
\*/

/\*

YOLOv5对比于v3，其架构上进行了更多的优化，极大提高了网络的速度和精度。此外，还有很多高级的训练选项，如采用数据增强（如马赛克数据增强）、自适应锚框策略、混合精度训练等等。它采用了一些高级的训练选项。就可以提高模型的精度，提高了模型的泛化能力。可以使得YOLOv5模型适合更加复杂的场景。

YOLOv5的精度高，且速度快，在相同的GPU硬件上YOLOv5的检测速度比v3快了几倍。

YOLOv5提供了更加自动化的训练和部署流程，用户可以通过简单修改一下配置文件和模型的参数，不需要深入修改代码就可以完成模型的训练。关于这点在后面去训练自己的数据集的时候，就可以明显体会了。此外YOLOv5支持把模型导出成各种格式。便于在不同平台上实现部署模型。在部署方面可以说YOLOv5更加面向于工业的应用场景。所以我们的课程是基于YOLOv5来讲解的。



\*/

1. YOLOv5提供了不同大小的模型，

如 YOLOv5n，n表示nano；

YOLOv5s，s表示small；

YOLOv5m，m表示medium，

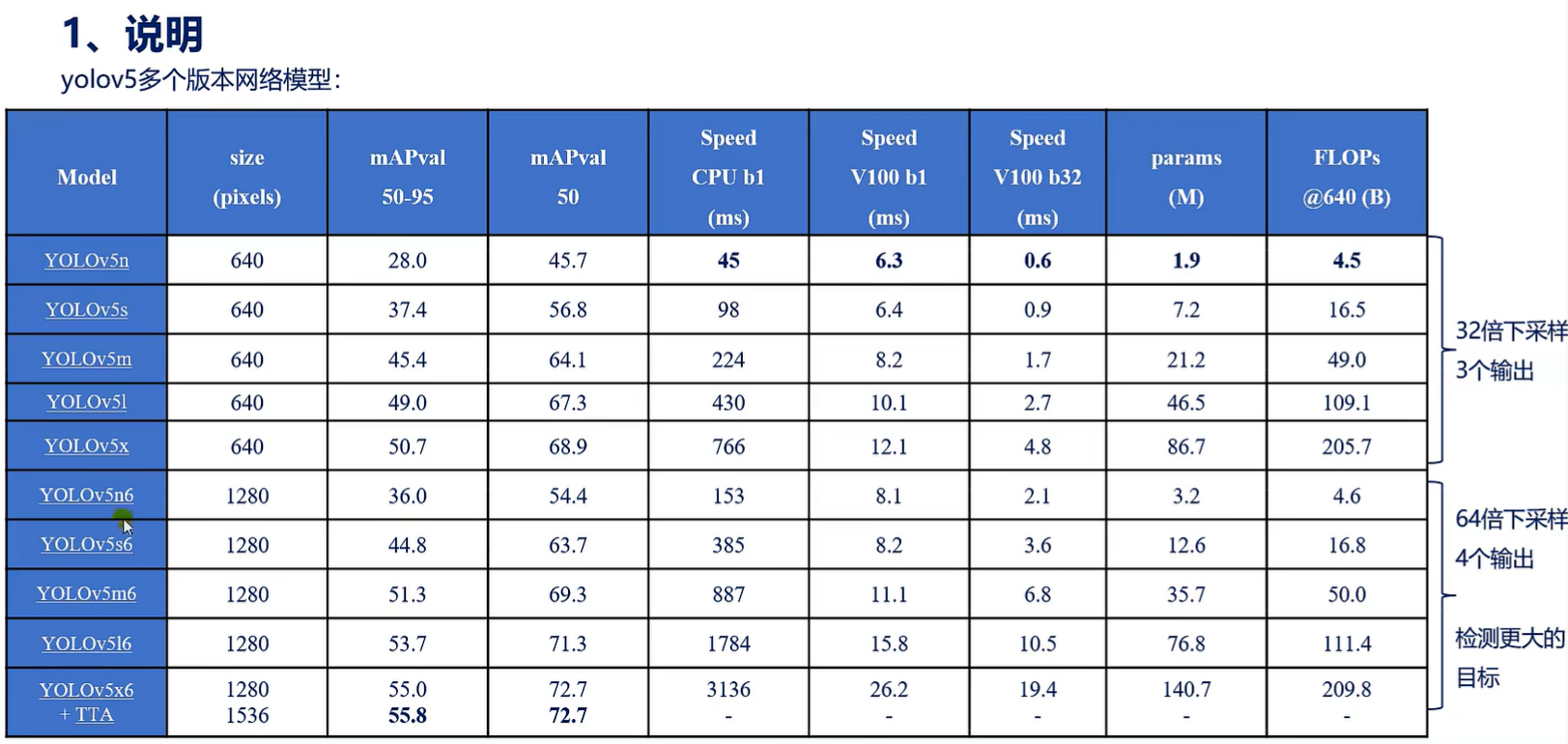
YOLOv5l，l表示large，

YOLOv5x，x表示extra large

这5个版本。这5个版本按照模型从小到大的顺序来排列。也就是说YOLOv5n是最小的模型，具有最少的卷积层和参数量。其次就是YOLOv5s模型比较小，YOLOv5x是最大的模型，具有最多的卷积层和参数量。从v5n到v5x网络深度和网络宽度不断地扩大，计算的复杂度也不断增加。但其检测速度和准确度方面出现明显的差异。YOLOv5n的检测速度是最快的，但检测的准确性较低。而v5x的准确性最高，但检测速度最慢。V5m和v5l在检测速度和准确度都是相对较好的。所以如果要考虑在资源受限的设备上部署YOLO，如移动设备/边缘设备那么就可以考虑使用YOLOv5n/YOLOv5s。而v5x比较适用于需要高精度且计算资源充足的场景。v5n和v5s比较时候部署在资源受限的移动设备/嵌入式终端上。所以rv1126主要就是使用v5s这张。



前面提到：YOLOv5有许多衍生版本的网络模型，下表就是从github网页中拷贝来的：



左边第一列就是YOLOv5的预训练模型，右边就是预训练模型对应的一些参数介绍了，棋子YOLOv5的nsmlx这些模型就是在输入尺寸为640\*640的图片上训练得到的，最终就是对输入图像进行32倍的下采样。并输出3个特征层作为预测。然后下面这几个YOLOv5的n6、s6、m6、l6、x6这些预训练的模型是输入尺寸为1280\*1280的图像上训练得到的，那么这些模型最终会对输入图像进行64倍的下采样。并输出4个特征层作为预测，那后面的这几个相较于前面的几个就可以检测更大的目标了。YOLOv5的这些版本，他们的区别主要就是在于网络的深度和宽度不同。关于网络的深度和宽度在后面的分析YOLOv5的模型配置文件的时候再细讲。

我们这里是学习，要考虑速度且把模型部署在开发板上，所以在后面的课程里，用训练模型的权重就使用YOLOv5s。也就是采用迁移学习的方式来训练模型，而不是从0开始训练模型。也就是采样YOLO官方训练好的权重，在其基础上训练我们的数据集。在其基础上学习。

采用迁移学习有许多好处：

可以解决数据不足的问题。官方的预训练模型权重是基于大量的数据训练得到的。如果我们数据量比较少，采用迁移学习，可以利用原来模型中的大量的数据来增强模型的泛化能力，避免过拟合，也就间接节省了我们的时间,降低了数据的标注成本。（总所周知，在准备数据集的时候，是需要花费大量的时间的，特别是在图片上标注出数据，一张图片需要人工的标注，这是一个耗时且需要大量人力的工作。需要人为调参，训练精度不足又需要重新调参。）

迁移学习可以提高学习的效率，节省时间。通过迁移已有的知识/模型，迁移学习可以显著的减少在新任务上学习的时间。使得模型能够更快地适应新的环境和新的数据。

迁移学习就是将已经学到的知识从原来的领域迁移到目标领域。从而加快学习的速度，提高学习的效果。可以解决数据不足的问题。



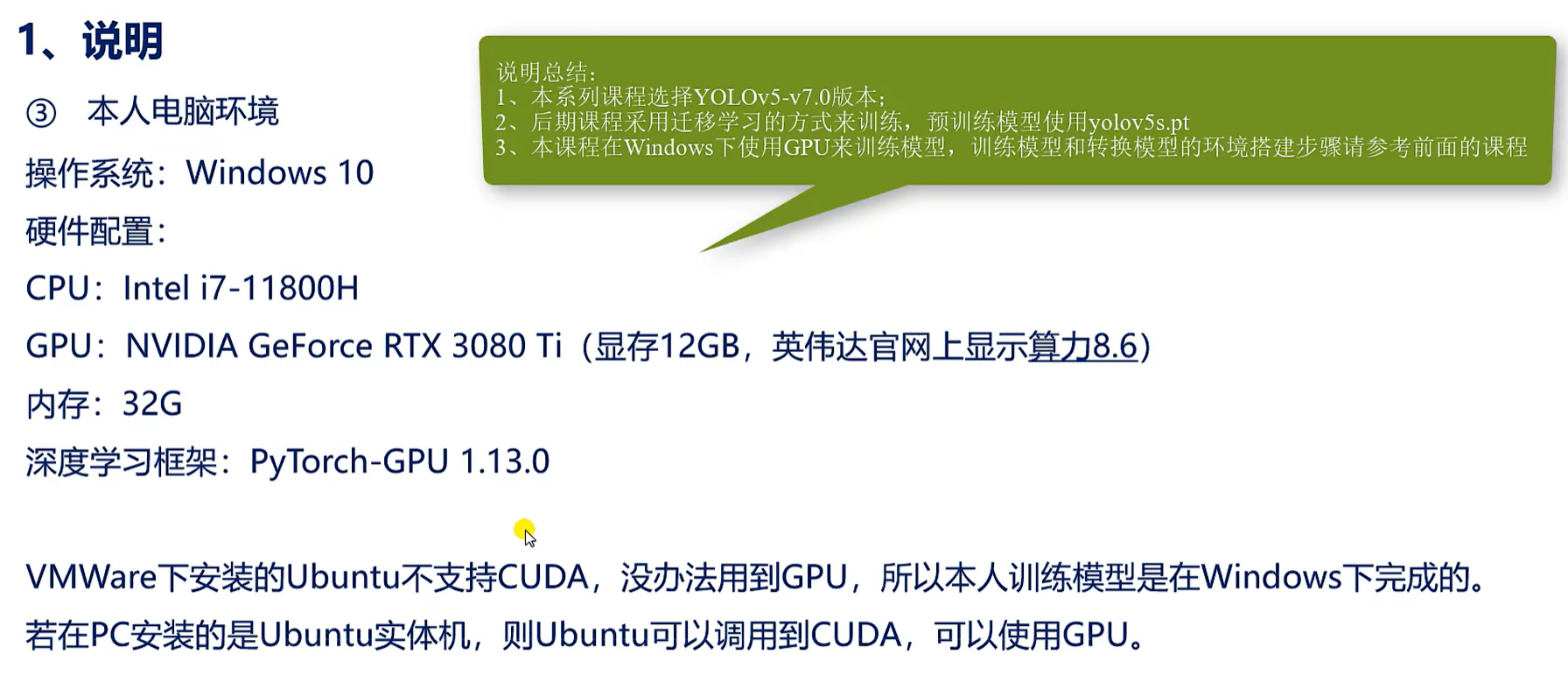
\*/

/\*

<https://github.com/ultralytics/yolov5?tab-readme-ov-file>

我们打开github上的链接，可以看到YOLOv5有10个版本。不同的版本，工程代码有一些不同，由于版本太多了，我们就以最新的v7.0版本来训练和部署模型。所以需要注意，我们使用的是YOLOv5的7.0版本。

本课程他是在Win下训练的：



\*/

下一节就下载YOLOv5的工程，并搭建好YOLOv5的开发环境。