

主讲人：正点原子团队

硬件平台：正点原子ATK-DLRV1126开发板

版权所有：广州市星翼电子科技有限公司

资料下载：www.openedv.com/docs/index.html

教学平台：www.yuanzige.com

天猫店铺：zhengdianyuanzi.tmall.com

技术论坛：www.openedv.com/forum.php

公众平台：正点原子



■ YOLOv5模型训练参数介绍

YOLOv5模型训练命令参数介绍

python train.py命令后的参数说明

命令参数	说明
--weights	<p>指定预训练模型权重的路径，使用该参数表示采用迁移学习的方式来训练模型，若不指定该参数，则表示从头开始训练，从头开始训练耗时比较长。命令使用示例如下：</p> <pre>python train.py --weights yolov5s.pt</pre> <ol style="list-style-type: none">1、若从头开始训练，可使用“--weights”参数指定一个空字符串：''，或者将default默认值设置为空字符串：''2、默认值是YOLOv5工程根目录下的yolov5s.pt文件。若不指定该参数，则采用默认的值进行训练，若YOLOv5工程根目录下没有yolov5s.pt文件，系统会自动下载该模型文件3、若在命令中既指定“--weights”参数，又自定义了default默认值，则模型使用的是命令行“--weights”参数指定的预训练权重文件来训练
--cfg	<p>指定模型配置文件的路径，默认值为空''（即没有默认值）。命令使用示例如下：</p> <pre>python train.py --cfg models/yolov5s.yaml</pre> <ol style="list-style-type: none">1、在模型配置文件中指定了模型的结构参数、锚框参数、模型的深度因子和宽度因子，以及物体的类别nc2、若已使用“--weights”参数指定了预训练权重文件，则训练时将直接使用预训练权重文件中保存的模型结构，此时可以不使用该参数3、若同时指定了“--data”参数和“--cfg”参数，若在模型配置文件和数据集配置文件中都指定了nc，且值不一样，系统会自动计算数据集中的类别数，然后确定真正的nc值4、“--weights”参数和“--cfg”参数不能都同时指定为空，必须要有一个，否则代码会报错（一般，“--weights”参数默认值是yolov5s.pt文件）

YOLOv5模型训练命令参数介绍

python train.py命令后的参数说明

命令参数	说明
--data	<p>指定数据集配置文件的路径，数据集配置文件中指定了数据集的路径、类别名、和一些预处理操作。默认值是YOLOv5工程根目录下的data/coco128.yaml。命令使用示例如下：</p> <pre>python train.py --data data/coco128.yaml</pre> <p>1、若未检查到数据集，代码会自动下载coco128数据集 2、YOLOv5工程中的数据集配置文件中一般有自动下载数据集的代码，所以会自动下载数据集</p>
--hyp	<p>指定超参数配置文件的路径。默认值为YOLOv5工程根目录下的data/hyps/hyp.scratch-low.yaml文件。命令使用示例如下：</p> <pre>python train.py --hyp data/hyps/hyp.scratch-low.yaml</pre>
--epochs	<p>指定训练的轮数，1个epoch表示使用训练集中的全部数据训练1次，默认为100轮，打印的信息显示0~99轮。命令行用法：</p> <pre>python train.py --epochs 5</pre>
--batch-size	<p>指定训练批量的大小，表示每批次输入的数据量，即每次从训练集中选取batch-size个样本进行训练，默认配置为default=16，即每批次输入16个。命令行用法：</p> <pre>python train.py --batch-size 32</pre> <p>1、--epochs 和 --batch-size 是不一样的，一轮（一个epoch）意味着整个数据集被模型遍历一次，在遍历过程中，模型会处理数据集中的一小批样本（称为一个batch），并进行一次前向传播和反向传播，以更新其权重。</p> <p>例如，若数据集有1000个样本，batch-size大小设置为100，在一个epoch中模型会进行10步（或迭代）来处理完整个数据集</p> <p>2、batch-size大小需要根据GPU显存以及内存资源来合理设置</p>

YOLOv5模型训练命令参数介绍

python train.py命令后的参数说明

命令参数	说明
--imgsz	<p>指定训练时输入图片的大小，默认为640*640大小。默认值为640。命令用法：</p> <pre>python train.py --imgsz 640</pre> <p>1、图片尺寸越大，产生的计算量越大，训练的速度越慢</p> <p>2、图片尺寸最好能被32整除（若不能被32整除，系统会自动设置一个临近的且能被32整除的值来进行训练）</p>
--rect	<p>指定采用矩形尺寸还是方形尺寸的图片来进行训练（宽和高相等的为方形尺寸，如640*640的方形尺寸，宽高不相等的为矩形尺寸），默认采用方形尺寸的图片来进行训练，若要采用矩形尺寸，可指定该参数，命令如下：</p> <pre>python train.py --rect</pre> <p>1、方形尺寸训练：参与训练的图片变为指定的方形尺寸后，再进行训练，若设置的尺寸不是能被32整除，系统会自动设置一个临近的且能被32整除的方形尺寸值来进行训练</p> <p>2、矩形训练：参与训练的图片在保持原高宽比的基础上进行 resize 和填充，且填充到32的最小整数倍，然后再进行训练</p>
--resume	<p>指定使用哪个训练过的模型来进行断点续训（断点续训是指从上一个训练任务中断的地方继续训练，直至训练完毕），该选项默认是关闭的，命令用法：</p> <pre>python train.py --weights /path/last.pt</pre>

YOLOv5模型训练命令参数介绍

python train.py命令后的参数说明

命令参数	说明
--nosave	是否只保存最后一轮训练的权重文件，默认保存最后一轮训练得到的权重文件（last.pt）和在验证集上loss最低的权重文件（best.pt）。 若指定只保存最后一轮训练的 .pt 权重文件，参考命令： python train.py --nosave
--noval	是否只在最后一个epoch训练后才进行测试，默认是在每一个epoch训练之后都会进行测试，若只对最后一次训练进行验证，命令用法如下： python train.py --noval
--noautoanchor	是否禁用自动计算锚框的功能，默认是开启了自动计算锚框的功能，若要关闭自动计算锚框的功能，参考命令： python train.py --noautoanchor 1、在YOLOv5中，在开始训练前，先自动计算数据集标注信息针对已有默认的锚框（针对的是COCO数据集，且图像为640*640大小的锚框）的最佳召回率，当最佳召回率大于等于0.98的时候，则不需要更新已有的锚框，直接使用已有的锚框进行训练，若最佳召回率小于0.98的时候，则需要重新计算和使用数据集中的锚框 2、一般建议使用自动计算锚框的功能，好处是：简化训练过程、提升训练速度和精度、提高模型泛化能力
--noplots	是否保存绘图文件，默认保存绘图文件，若不想保存，指定该参数即可，参考命令： python train.py --noplots

YOLOv5模型训练命令参数介绍

python train.py命令后的参数说明

命令参数	说明
--evolve	<p>是否使用遗传算法来更新（锚框）数据，默认是不使用。YOLOv5默认采用的是在COCO数据集上得到的锚框数据，若要开启该功能，命令如下（一般不建议开启，这会耗费大量的资源和时间，命令中的n表示遗传算法运行的迭代次数，若命令后不指定n，则默认采用const=300）：</p> <pre>python train.py --evolve n</pre>
--bucket	<p>若要在谷歌云盘中下载或者上传数据，可以指定该参数，该参数很少用到。若要指定，参考命令：</p> <pre>python train.py --bucket gsutil bucket</pre>
--cache	<p>是否将数据集缓存到内存，训练时直接从内存读取数据，可以加快数据加载和训练速度，默认值为 None，若要开启该参数，命令参考：</p> <pre>python train.py --cache ram</pre> <pre>python train.py --cache disk</pre>
--image-weights	<p>是否对数据集图片进行加权训练，开启该参数后，上一轮训练效果不好的图片，会在下一轮训练中增加一些权重来加权训练，该功能默认是关闭的，若要开启，命令如下（需要搭配--rect参数使用）：</p> <pre>python train.py --image-weights --rect</pre>

YOLOv5模型训练命令参数介绍

python train.py命令后的参数说明

命令参数	说明
--device	<p>指定训练使用的设备（GPU 或 CPU），默认参数为None，即代码自动检测设备，存在如下几种情况：</p> <p>1、若设备无GPU，则只能使用CPU训练：python train.py --device cpu</p> <p>2、若设备有单GPU，则默认使用GPU训练，也可以在代码中指定：python train.py --device 0</p> <p>3、若设备有多个GPU，当使用单个GPU训练时：python train.py --device 0（0表示使用第 1 个 GPU 训练）</p> <p>4、若设备有多个GPU，当使用多个GPU训练时：python train.py --device 0, 1, 2（表示使用第 1、2、3 个 GPU 训练）</p> <p>5、不指定该参数时，代码会自动选择设备，有GPU则优先使用GPU进行训练，无GPU则使用CPU进行训练</p>
--multi-scale	<p>是否启用多尺度训练（在不同的图像尺寸上训练模型，以便模型能够学习并适应不同尺度的目标），默认是关闭的，若要开启，命令示例：</p> <p>python train.py --multi-scale</p>
--single-cls	<p>指定训练数据集是单类别还是多类别，默认值为 False，即多类别。若使用单类别训练，参考命令：</p> <p>python train.py --single-cls</p>
--optimizer	<p>指定训练过程中使用 什么优化器，在代码中可以指定SGD、Adam和AdamW优化器，默认使用SGD优化器。命令行用法：</p> <p>python train.py --optimizer SGD</p>

YOLOv5模型训练命令参数介绍

python train.py命令后的参数说明

命令参数	说明
--sync-bn	<p>是否开启跨卡同步批量归一化，默认是关闭的，即默认使用传统的批量归一化，若要开启该功能，参考命令：</p> <pre>python train.py --sync-bn</pre> <p>1、开启后，所有 GPU 上计算的均值和方差是一致的，可提高模型训练的稳定性和效果，但可能会增加训练时间和硬件要求</p> <p>2、传统的批量归一化存在的缺点：每个 GPU上计算的均值和方差可能会不一样，导致模型训练不稳定</p>
--workers	<p>指定Dataloader加载数据的线程数，默认采用8个线程加载和处理数据，若要修改默认值，参考命令：</p> <pre>python train.py --workers 10</pre> <p>若改为0，表示不使用多线程，仅使用主线程加载和处理数据</p>
--project	<p>指定训练得到的模型的保存主路径名称，默认保存在“runs/train”下，若要指定其它主路径名称，命令用法，如：</p> <pre>python train.py --project 'models'</pre> <p>得到的模型保存的主路径是在YOLOv5工程根目录的“models”下</p>
--name	<p>指定训练得到的模型的保存子目录名称，默认是“exp”，例如，第一次训练得到的结果会保存在“runs/train/exp”下，第二次训练得到的结果保存在“runs/train/exp2”下，第三次训练得到的结果保存在“runs/train/exp3”，依次类推。若要修改子目录名称，命令用法：</p> <pre>python train.py --name 'get'</pre> <p>第一次训练得到的模型保存在YOLOv5工程根目录的“models/get”下</p> <p>第二次训练得到的模型保存在YOLOv5工程根目录的“models/get2”下</p>

YOLOv5模型训练命令参数介绍

python train.py命令后的参数说明

命令参数	说明
--exist-ok	训练得到的结果保存的子目录是否自动保存在exp、exp2、exp3.....目录下，还是选择覆盖上一次同名目录。例如，若上一次训练得到的结果保存在“runs/train/exp2”下，再次训练时，指定该参数以后，训练得到的结果会保存在“runs/train/exp2”下，且训练结果会覆盖原先“runs/train/exp2”文件夹中保存的所有结果，若未指定该参数，训练得到的结果会保存在“runs/train/exp3”下。 命令使用示例： python train.py --exist-ok
--quad	是否使用 quad dataloader 数据加载器并行地从磁盘中读取和处理多个图像，以提高数据加载的效率，该功能默认是关闭的，若要开启，命令参考： python train.py --quad
--cos-lr	是否使用余弦退火学习率调度策略动态调整学习率，默认是关闭的，若要开启，参考命令： python train.py --cos-lr
--label-smoothing	训练时是否使用标签平滑策略，以防止过拟合。默认值epsilon=0.0，若将标签平滑策略使用的epsilon设置为0.1，参考命令： python train.py --label-smoothing 0.1
--patience	训练使用Early Stopping策略（早停机制），防止过拟合，默认值为100，即若连续训练100个epoch，验证精度都没有得到提升，则认为训练已经过拟合，然后停止训练。若要指定早停机制的epochs数，参考命令： python train.py --patience 300

YOLOv5模型训练命令参数介绍

python train.py命令后的参数说明

命令参数	说明
--freeze	<p>训练时使用的冻结训练策略，默认值为0，即不冻结任何层，命令用法如下：</p> <pre>python train.py --freeze n</pre> <p>其中的n，表示前n层（层数是从第0层开始，若n=2，则表示第0和第1层），n的值分为如下情况：</p> <ol style="list-style-type: none">1、若n的值为'0'或'-1'，表示不冻结任何层，即更新所有层的权重参数2、若n的值大于1，表示冻结前n层，这前n层的权重参数在训练过程中保持不变，只更新后n层的权重参数
--save-period	<p>每训练n个epoch保存一次训练权重，若要指定训练多少次就保存权重，命令用法：</p> <pre>python train.py --save-period n</pre> <ol style="list-style-type: none">1、若n>0，表示每训练n个epoch保存一次训练权重2、若n<0（默认值为-1），表示关闭save-period，即只保存best.pt和last.pt权重
--seed	<p>该参数用于设置随机数生成器的种子（seed），确保每次实验时，这些随机过程（权重初始化、数据打乱、随机采样、Dropout正则化等）都使用相同的随机数序列，确保每次实验时得到一致的实验结果，方便比较不同的模型结构、超参数或数据集对性能的影响，确保实验的可复现性。命令用法：</p> <pre>python train.py --seed n</pre> <p>其中，n是一个整数，默认的随机数生成器的种子为0</p>

YOLOv5模型训练命令参数介绍

python train.py命令后的参数说明

命令参数	说明
--local_rank	<p>是否使用分布式训练，默认值为-1，即默认关闭分布式训练。若有3个GPU，使用分布式训练时，可以这么设置：</p> <pre>python train.py --local_rank 1,2</pre> <p>表示使用第 2 张和第 3 号张GPU 进行分布式训练（第0张未使用），其中：第一个进程将使用第 2 张 GPU，第二个进程将使用第 3 张 GPU</p>
--entity	<p>该参数用的较少，大概就是用于指定要训练的特定实体，如一个特定的用户、项目、类别等，具体用法本人未搞清楚</p>
--upload_dataset	<p>是否上传数据集，默认值为False，即不上传数据集。命令用法：</p> <ol style="list-style-type: none">1、python train.py --upload_dataset的后面未指定其它参数（且默认值const=True），表示上传整个数据集2、python train.py --upload_dataset的后面指定了参数'val'（且默认值const=True），则表示上传验证集
--bbox_interval	<p>在训练的过程中，每隔多少个epoch记录一次带有边界框的图片。命令用法：</p> <pre>python train.py --bbox_interval n</pre> <ol style="list-style-type: none">1、当n<=0时（默认值为-1），表示关闭--bbox_interval2、当>0时，表示每隔n个epoch记录一次带有边界框的图片
--artifact_alias	<p>该参数的具体用法本人尚未搞清楚</p>



版权所有：广州市星翼电子科技有限公司
天猫店铺：<https://zhengdianyuanyi.tmall.com>