

# 训练

在 1.创建自己的数据集 中我们创建了自己的数据集,假设数据集的路径为 `${dataset_path}` ,那么其中的路径有:

- 训练集: `${dataset_path}/train`
- 验证集: `${dataset_path}/val`

例如:

- `${dataset_path}` = `E:/dataset`
- 训练集路径: `E:/dataset/train`
- 验证集路径: `E:/dataset/val`

其中的图像和标签是按照1.中最后所说那样。

## 1.配置格式文件

在当前目录下创建一个 `data` 文件夹,在 `data` 文件夹下创建一个 `data.yaml` 文件,



打开 `data.yaml` 文件,并且添加以下内容

```
## 训练集路径
train: ${dataset_path}/train
## 验证集路径
val: ${dataset_path}/val
## 标签数量
nc: 6
## 标签的类别
names: ["cr", "in", "pa", "ps", "rs", "sc"]
```

注意:上述内容是学长训练自己数据的示范,训练自己的数据还需要从头填写你自己的内容。

## 2.训练开始

在yolo源码的目录下,运行命令

```
## 激活虚拟环境,如果已经激活就可以不运行
activate myyolo
```

### 2.1 cpu训练

```
## 开始训练
yolo train model=./yolov8n.pt data=data/data.yaml epochs=150 batch=4 imgsz=320 device=cpu
```

其中参数解释如下:

- train : 训练模式
  - model : 预训练权重路径
  - data : 训练设置的yaml文件路径
  - epochs : 训练的轮数
  - batch : 一次前向计算的图像张数
  - imgsz : 模型输入图像的尺寸可填:[640,320],主要是为了减小运行内存,无论填哪个都可以允许,模型会在训练时将你的数据resize到适合训练的尺寸。
  - device : 训练设备,可选:cpu,gpu(0,1,2,3)
- 开始训练后可以看到如下内容

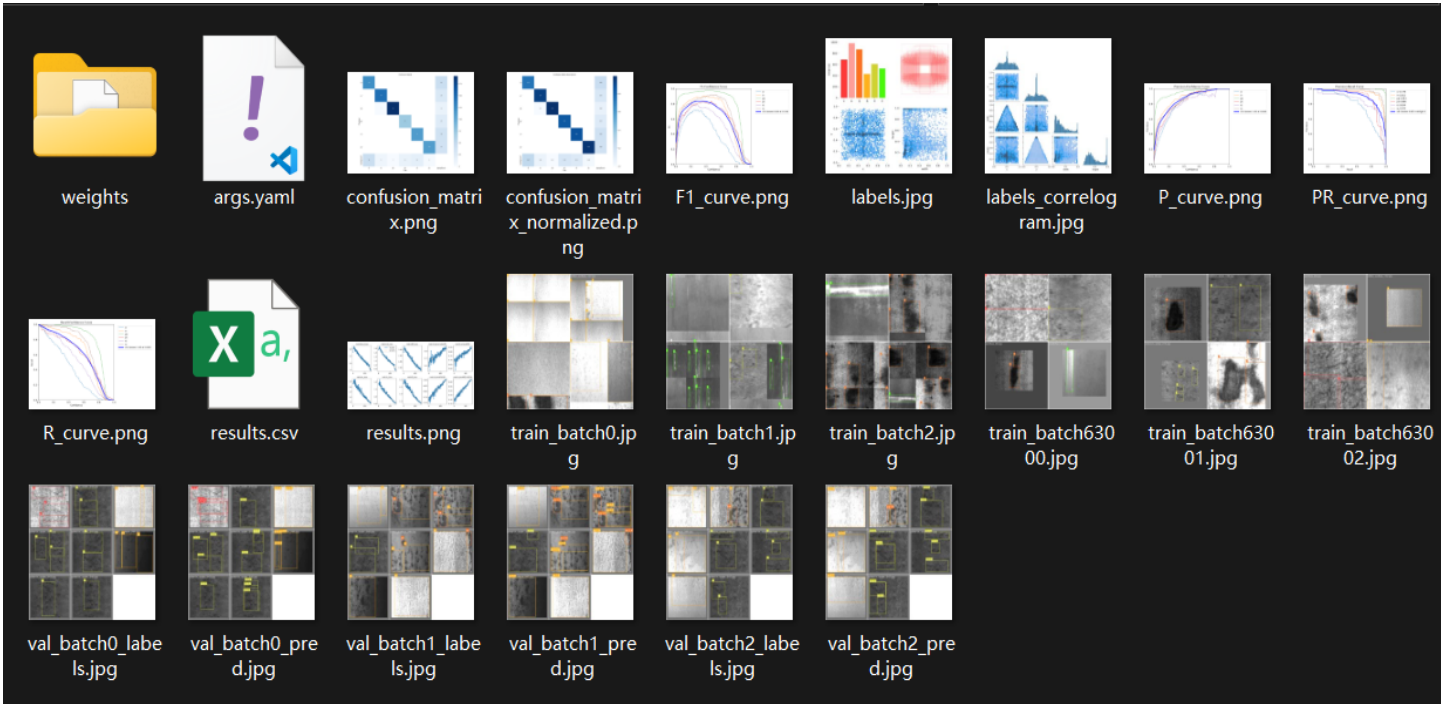
```
(myyolo) E:\code\python\ultralytics>yolo train model=../yolov8n.pt data=data/data.yaml epochs=150 batch=4 imgsz=320 device=cpu
ultralytics YOLOv8.0.237 Python-3.8.18 torch-2.1.2-cpu CPU (Intel Core(TM) i7-10750H 2.60GHz)
Engine/trainer: task=detect, mode=train, model=../yolov8n.pt, data=data/data.yaml, epochs=150, time=None, patience=50, batch=4, imgsz=320, save=True, save_period=1, cache=False, device=cpu, workers=8, project=None, name=train2, exist_ok=False, pretrain=True, optimizer=auto, verbose=True, seed=0, deterministic=True, single_cls=False, rect=True, cos_lr=False, close_mosaic=10, resume=False, amp=True, fraction=1.0, profile=False, freeze=None, multi_scale=False, overlap_mask=True, mask_ratio=4, dropout=0.0, val=True, split=val, save_json=False, save_hybrid=False, conf=None, iou=0.7, max_det=300, half=False, dnn=False, plots=True, source=None, vid_stride=1, stream_buffer=False, visualize=False, augment=False, agnostic_nms=False, classes=None, retina_masks=False, embed=None, show=False, save_frames=False, save_txt=False, save_conf=False, save_crop=False, show_labels=True, show_conf=True, show_boxes=True, line_width=None, format=torchscript, keras=False, optimizer=False, int8=False, dynamic=False, simplify=False, opset=None, workspace=4, msa=False, lr0=0.01, lr=0.01, momentum=0.937, weight_decay=0.0005, warmup_epochs=10, warmup_momentum=0.8, warmup_bias_lr=0.1, box=7.5, cls=0.5, dfl=1.5, pose=12.0, kobj=1.0, label_smoothing=0.0, nbs=64, hsv_h=0.015, hsv_s=0.7, hsv_v=0.4, degrees=0.0, translate=0.1, scale=0.5, shear=0.0, perspective=0.0, flipud=0.0, flipud=0.0, mosaic=0.5, mixup=0.0, copy_paste=0.0, auto_augment=RandAugment, erasing=0.4, crop_fraction=1.0, cfg=None, tracker=botssort.yaml, save_dir=runs\detect\train2
Overriding model.yaml nc=80 with nc=6

  from n  params  module  arguments
  0  -1  1  464  ultralytics.nn.modules.conv.Conv  [3, 16, 3, 2]
  1  -1  1  4672  ultralytics.nn.modules.conv.Conv  [16, 32, 3, 2]
  2  -1  1  7268  ultralytics.nn.modules.block.C2f  [32, 32, 1, True]
  3  -1  1  18560  ultralytics.nn.modules.conv.Conv  [32, 64, 3, 2]
  4  -1  2  49664  ultralytics.nn.modules.block.C2f  [64, 64, 2, True]
  5  -1  1  73984  ultralytics.nn.modules.conv.Conv  [64, 128, 3, 2]
  6  -1  2  197632  ultralytics.nn.modules.block.C2f  [128, 128, 2, True]
  7  -1  1  295424  ultralytics.nn.modules.conv.Conv  [128, 256, 3, 2]
  8  -1  1  468288  ultralytics.nn.modules.block.C2f  [256, 256, 1, True]
  9  -1  1  164688  ultralytics.nn.modules.block.SPPF  [256, 256, 5]
  10  -1  1  0  torch.nn.modules.upsampling.Upsample  [None, 2, 'nearest']
  11  [-1, 6] 1  0  ultralytics.nn.modules.conv.Concat  [1]
  12  -1  1  188224  ultralytics.nn.modules.block.C2f  [384, 128, 1]
  13  -1  1  0  torch.nn.modules.upsampling.Upsample  [None, 2, 'nearest']
  14  [-1, 4] 1  0  ultralytics.nn.modules.conv.Concat  [1]
  15  -1  1  37248  ultralytics.nn.modules.block.C2f  [192, 64, 1]
  16  -1  1  36992  ultralytics.nn.modules.conv.Conv  [64, 64, 3, 2]
  17  [-1, 12] 1  0  ultralytics.nn.modules.conv.Concat  [1]
  18  -1  1  123648  ultralytics.nn.modules.block.C2f  [192, 128, 1]
  19  -1  1  147712  ultralytics.nn.modules.conv.Conv  [128, 128, 3, 2]
  20  [-1, 9] 1  0  ultralytics.nn.modules.conv.Concat  [1]
  21  -1  1  493856  ultralytics.nn.modules.block.C2f  [384, 256, 1]
  22  [15, 18, 21] 1  752480  ultralytics.nn.modules.head.Detect  [6, [64, 128, 256]]
Model summary: 225 layers, 3012018 parameters, 3012002 gradients, 8.2 GFLOPs

Transferred 319/355 items from pretrained weights
Freezing layer 'model.22.dfl.conv.weight'
train: Scanning E:\dataset\medal\mine\train\labels... 1880 images, 4 backgrounds, 0 corrupt: 100%|██████████| 1880/1880 [00:00<
train: WARNING ⚠ E:\dataset\medal\mine\train\images\crazing_120.jpg: 1 duplicate labels removed
train: WARNING ⚠ E:\dataset\medal\mine\train\images\inclusion_62.jpg: 1 duplicate labels removed
train: WARNING ⚠ E:\dataset\medal\mine\train\images\patches_198.jpg: 1 duplicate labels removed
train: New cache created: E:\dataset\medal\mine\train\labels.cache
val: Scanning E:\dataset\medal\mine\val\labels.cache... 366 images, 1 backgrounds, 0 corrupt: 100%|██████████| 366/366 [00:00<
Plotting labels to runs\detect\train2\labels.jpg...
optimizer: 'optimizer=auto' found, ignoring 'lr0=0.01' and 'momentum=0.937' and determining best 'optimizer', 'lr0' and 'momentum' automatically...
optimizer: Adam(lr=0.001, momentum=0.9) with parameter groups 57 weight(decay=0.0), 64 weight(decay=0.0005), 63 bias(decay=0.0)
150 epochs...

Epoch  GPU_mem  box_loss  cls_loss  dfl_loss  Instances  Size
1/150   2.014      3.938      1.844      12          320: 13%|██████████| 60/450 [00:20<02:11, 2.97s/epoch]
```

在途中我们可以看到训练保存的路径,以及训练的进度。训练结束后我们到保存目录下可以看到如下内容



其中 weights 文件夹是最重要的,里面

- **best.pt** : 训练中loss最小那一轮的权重文件
- **last.pt** : 训练最后一轮的权重文件

而其他文件,比如 args.yaml 记录训练的一些参数设置,其他的图像记录了损失曲线,指标曲线。测试, .pt 文件里面包含了模型权重和一些其他的功能函数,可以直接用,比如

```
yolo predict model=best.pt source=图像路径
```

可以直接得到预测的结果;如果转为 .onnx 文件,我们还需要改一些源码,这个会在之后的内容讨论。

## 2.2 gpu训练

首先要配置 cuda + cudnn ,这里推荐网址:

[https://blog.csdn.net/qq\\_41946216/article/details/129476960?](https://blog.csdn.net/qq_41946216/article/details/129476960?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522170471383716800225544505%2522%252C%2522scm%2522%253A%25220140713.130102334..%2522%257D&request_id=170471383716800225544505&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduend~default-1-129476960-null-null.142^v99^pc_search_result_base4&utm_term=windows11配置cuda&spm=1018.2226.3001.4187)

[ops\\_request\\_misc=%257B%2522request%255Fid%2522%253A%2522170471383716800225544505%2522%252C%2522scm%2522%253A%25220140713.130102334..%2522%257D&request\\_id=170471383716800225544505&biz\\_id=0&utm\\_medium=distribute.pc\\_search\\_result.none-task-blog-2~all~sobaiduend~default-1-129476960-null-null.142^v99^pc\\_search\\_result\\_base4&utm\\_term=windows11配置cuda&spm=1018.2226.3001.4187](https://blog.csdn.net/qq_41946216/article/details/129476960?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522170471383716800225544505%2522%252C%2522scm%2522%253A%25220140713.130102334..%2522%257D&request_id=170471383716800225544505&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduend~default-1-129476960-null-null.142^v99^pc_search_result_base4&utm_term=windows11配置cuda&spm=1018.2226.3001.4187)

同时虚拟环境 myyolo 中 cpu 版本的 torch,torchvision 删除,如下图

```
Developer Command Prompt 1 x Developer Command Prompt x + v
matplotlib 3.7.4
mpmath 1.3.0
networkx 3.1
numpy 1.24.4
onnx 1.15.0
opencv-python 4.9.0.80
packaging 23.2
pandas 2.0.3
pillow 10.2.0
pip 23.3.1
protobuf 4.25.1
psutil 5.9.7
py-cpuinfo 9.0.0
pyparsing 3.1.1
python-dateutil 2.8.2
pytz 2023.3.post1
PyYAML 6.0.1
requests 2.31.0
scipy 1.10.1
seaborn 0.13.1
setuptools 68.2.2
six 1.16.0
sympy 1.12
thop 0.1.1.post2209072238
torch 2.1.2
torchvision 0.16.2
tqdm 4.66.1
typing_extensions 4.9.0
tzdata 2023.4
ultralytics 8.0.237
urllib3 2.1.0
wheel 0.41.2
zipp 3.17.0

(myyolo) E:\code\python\ultralytics>pip uninstall torch torchvision
```

查询 cuda 的版本,

```
(myyolo) E:\code\python\ultralytics>nvidia-smi
Mon Jan 8 19:39:23 2024

+-----+
| NVIDIA-SMI 537.58                  Driver Version: 537.58          CUDA Version: 12.2          |
+-----+-----+
| GPU   Name                TCC/WDDM  Bus-Id  Disp.A  Volatile Uncorr. ECC |
| Fan   Temp   Perf          Pwr:Usage/Cap  Memory-Usage  GPU-Util  Compute M. |
|              MIG M. |
+-----+-----+
|  0  NVIDIA GeForce GTX 1650 Ti  WDDM  00000000:01:00.0 Off  |
| N/A   35C    P0             13W / 50W    0MiB / 4096MiB  0%      Default |
|                                   N/A      |
+-----+-----+

+-----+
| Processes: |
| GPU   GI    CI          PID    Type    Process name                        GPU Memory |
|      ID    ID              |              | Usage |
+-----+-----+
| No running processes found |
+-----+

(myyolo) E:\code\python\ultralytics>
```

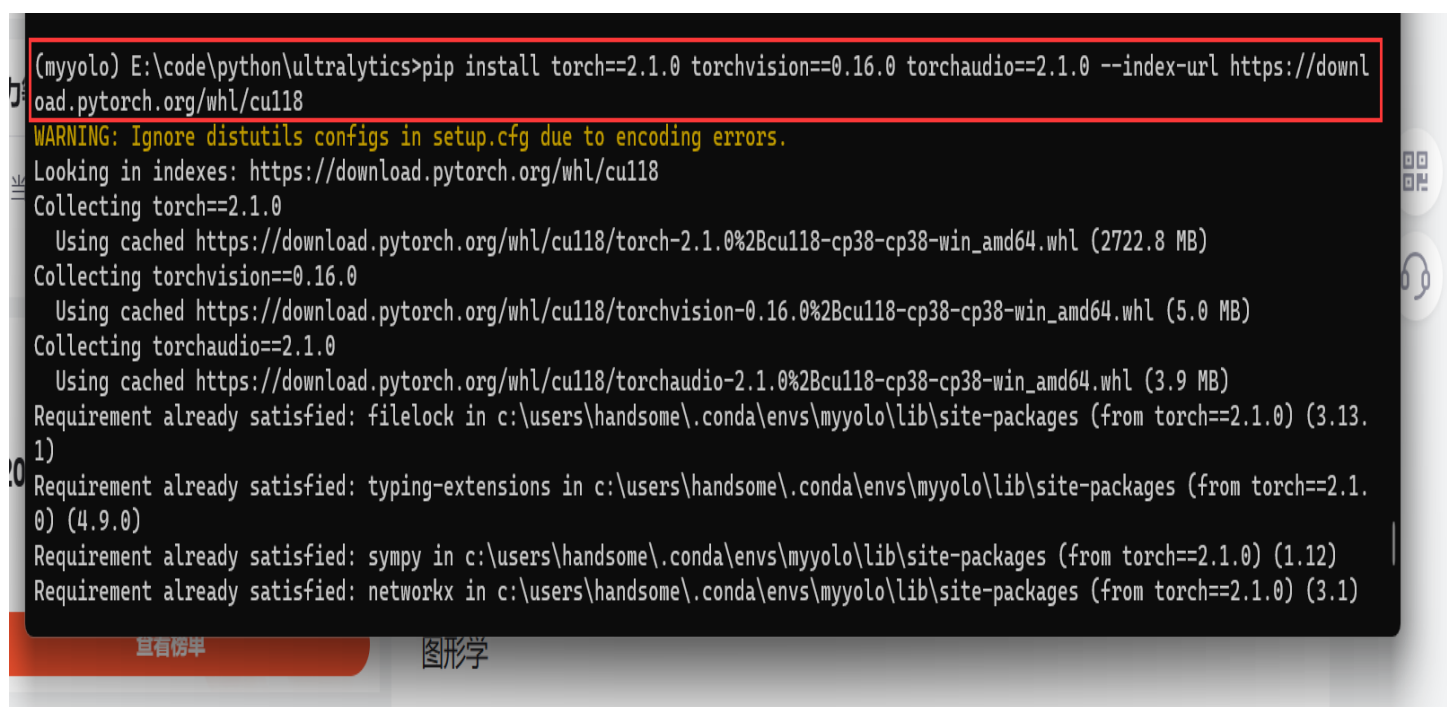
去 torch 官网安装对应版本或者较低版本的 torch,torchvision ,网址 : <https://pytorch.org/get-started/previous-versions/> ,如图学长安装适合 cuda-11.8 的 torch,torchvision

Linux and Windows

```
# ROCM 5.6 (Linux only)
pip install torch==2.1.0 torchvision==0.16.0 torchaudio==2.1.0 --index-url https://download.pytorch.org/whl/
# CUDA 11.8
pip install torch==2.1.0 torchvision==0.16.0 torchaudio==2.1.0 --index-url https://download.pytorch.org/whl/
# CUDA 12.1
pip install torch==2.1.0 torchvision==0.16.0 torchaudio==2.1.0 --index-url https://download.pytorch.org/whl/
# CPU only
pip install torch==2.1.0 torchvision==0.16.0 torchaudio==2.1.0 --index-url https://download.pytorch.org/whl/
```

v2.0.1

将命令复制粘贴到命令行中运行



```
(myyolo) E:\code\python\ultralytics>pip install torch==2.1.0 torchvision==0.16.0 torchaudio==2.1.0 --index-url https://download.pytorch.org/whl/cu118
WARNING: Ignore distutils configs in setup.cfg due to encoding errors.
Looking in indexes: https://download.pytorch.org/whl/cu118
Collecting torch==2.1.0
  Using cached https://download.pytorch.org/whl/cu118/torch-2.1.0%2Bcu118-cp38-cp38-win_amd64.whl (2722.8 MB)
Collecting torchvision==0.16.0
  Using cached https://download.pytorch.org/whl/cu118/torchvision-0.16.0%2Bcu118-cp38-cp38-win_amd64.whl (5.0 MB)
Collecting torchaudio==2.1.0
  Using cached https://download.pytorch.org/whl/cu118/torchaudio-2.1.0%2Bcu118-cp38-cp38-win_amd64.whl (3.9 MB)
Requirement already satisfied: filelock in c:\users\handsome\.conda\envs\myyolo\lib\site-packages (from torch==2.1.0) (3.13.1)
Requirement already satisfied: typing-extensions in c:\users\handsome\.conda\envs\myyolo\lib\site-packages (from torch==2.1.0) (4.9.0)
Requirement already satisfied: sympy in c:\users\handsome\.conda\envs\myyolo\lib\site-packages (from torch==2.1.0) (1.12)
Requirement already satisfied: networkx in c:\users\handsome\.conda\envs\myyolo\lib\site-packages (from torch==2.1.0) (3.1)
```

然后就是等它下完,下完以后查看是否为 cuda 适配的,如图在命令行中运行 pip list

```
Developer Command Prompt | X Developer Command Prompt X + v
mpmath 1.3.0
networkx 3.1
numpy 1.24.4
onnx 1.15.0
opencv-python 4.9.0.80
packaging 23.2
pandas 2.0.3
pillow 10.2.0
pip 23.3.1
protobuf 4.25.1
psutil 5.9.7
py-cpuinfo 9.0.0
pyparsing 3.1.1
python-dateutil 2.8.2
pytz 2023.3.post1
PyYAML 6.0.1
requests 2.31.0
scipy 1.10.1
seaborn 0.13.1
setuptools 68.2.2
six 1.16.0
sympy 1.12
thop 0.1.1.post2209072238
torch 2.1.0+cu118
torchaudio 2.1.0+cu118
torchvision 0.16.0+cu118
tqdm 4.66.1
typing_extensions 4.9.0
tzdata 2023.4
ultralytics 8.0.237
urllib3 2.1.0
wheel 0.41.2
zipp 3.17.0

(myxolo) E:\code\python\ultralytics>
```

如果后面跟着 `cuxxx` 就说明是安装上 `cuda` 版本的 `torch` 了,现在用 `gpu` 训练,到 `yolov8` 的源码目录,运行命令

```
yolo train model=./yolov8n.pt data=data/data.yaml epochs=150 batch=4 imgsz=320 device=0
```

注意这里只用 `device` 发生了变化,如果有多个 `gpu` 那么, `device=0,1,2...` 也可以, `gpu` 训练在这里只有加速训练的作用,并不会对最后的权重 `best.pt`, `last.pt` 产生什么其他的影响.测试:

```
yolo predict model=模型路径 source='图像路径'
```