

下载[源码包](#)，解压

```
1 wget
   https://github.com/ultralytics/yolov5/archive/refs/tags/v
   6.1.tar.gz
2 tar -zxvf yolov5-6.1.tar.gz
```

进入解压目录，安装依赖

```
1 cd yolov5-6.1
2 pip install -r requirements.txt
```

下载预训练模型 [yolov5s6.pt](#)，在项目根目录新建 **weights** 文件夹，将下载的预训练模型放入。

使用 **labellmg** 标注自己的数据集，在项目根目录新建 **datasets** 文件夹，然后将数据集放入，**datasets** 文件夹的结构如下，每一个数据集都包含三个文件夹 **train**，**test**，**valid**，其中 **test** 文件夹不是必须的，每个文件夹都包含两个文件夹 **images** 和 **labels**，分别放置图片和标签。

```
1 datasets
2 |— multi-class-denoise
3 |   |— test
4 |   |   |— images
5 |   |   |— labels
6 |   |— train
7 |   |   |— images
8 |   |   |— labels
9 |   |— valid
10 |   |— images
```

```

11 |         └─ labels
12 | └─ one-class-denoise
13 |     └─ test
14 |         └─ images
15 |         └─ labels
16 |     └─ train
17 |         └─ images
18 |         └─ labels
19 |     └─ valid
20 |         └─ images
21 |         └─ labels
22 | └─ one-class-noise
23 |     └─ test
24 |         └─ images
25 |         └─ labels
26 |     └─ train
27 |         └─ images
28 |         └─ labels
29 |     └─ valid
30 |         └─ images
31 |         └─ labels

```

在根目录下的 **data** 文件夹下新建 **xxx.yaml** 训练配置文件，用以指定数据集的位置以及要识别的类别数目，一个参考如下 **data/one-class-denoise.yaml** :

```
1 path: ../datasets/one-class-denoise # dataset root dir
2
3 train: train/images # train images (relative to 'path')
4 val: valid/images # val images (relative to 'path')
5 test: test/images # test images (optional)
6
7 # Classes
8 nc: 1 # number of classes
9 names: ['obj'] # class names
```

然后就可以愉快的训练了

```
1 python3 train.py --img 640 --batch 16 --epochs 200 --data
  ./data/one-class-denoise.yaml --cfg ./models/yolov5s.yaml
  --weights ./weights/yolov5s6.pt
```

- `--img` : 指定图片大小
- `--batch` : 指定批量大小
- `--epochs` : 指定训练轮次
- `--data` : 指定训练配置文件地址
- `--cfg` : 指定使用的模型
- `--weights` : 指定使用的预训练权重

训练的结果放在 `runs/train/` 下面, 可以通过 `results.csv` 观察到训练结果, 或者通过 `tensorboard` 观察结果

```
1 tensorboard --logdir runs/train --port 6009
```

然后在浏览器通过 `localhost:6009` 即可实时查看到训练结果。

在 `runs/train/exp/weights` 下保存着训练好的模型，有两个模型 `best.pt` 以及 `last.pt`，分别表示训练过程中最好的模型以及最后一个模型。我们随后便可使用训练好的模型进行推理

```
1 python3 detect.py --source datasets/one-class-  
  denoise/test/images/ --weights  
  ./runs/train/exp/weights/best.pt
```

- `--source`：指定要推理的图像来源，可以是具体的图片，也可以是视频，也可以存放图像的文件夹地址
- `--weights`：指定要使用的模型参数，使用指定训练好的 `best.pt` 或者 `last.pt`