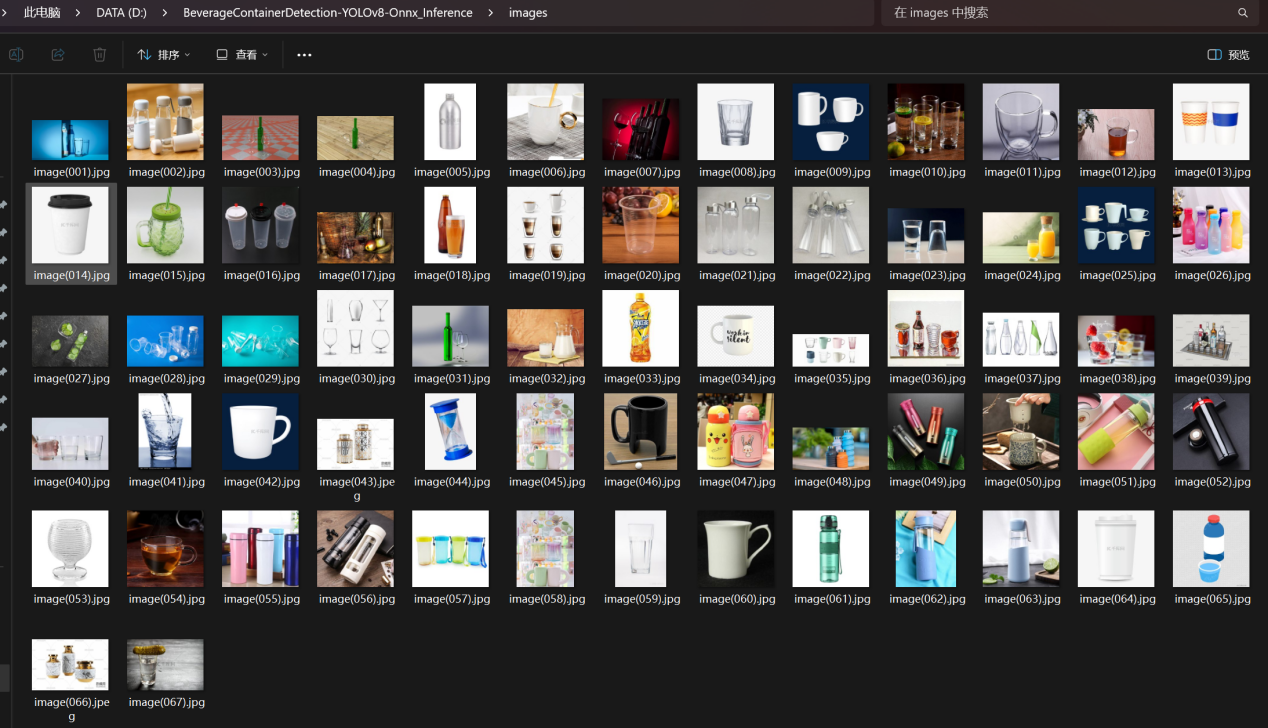
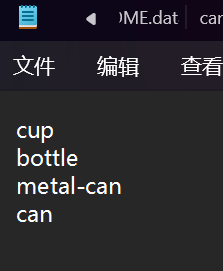
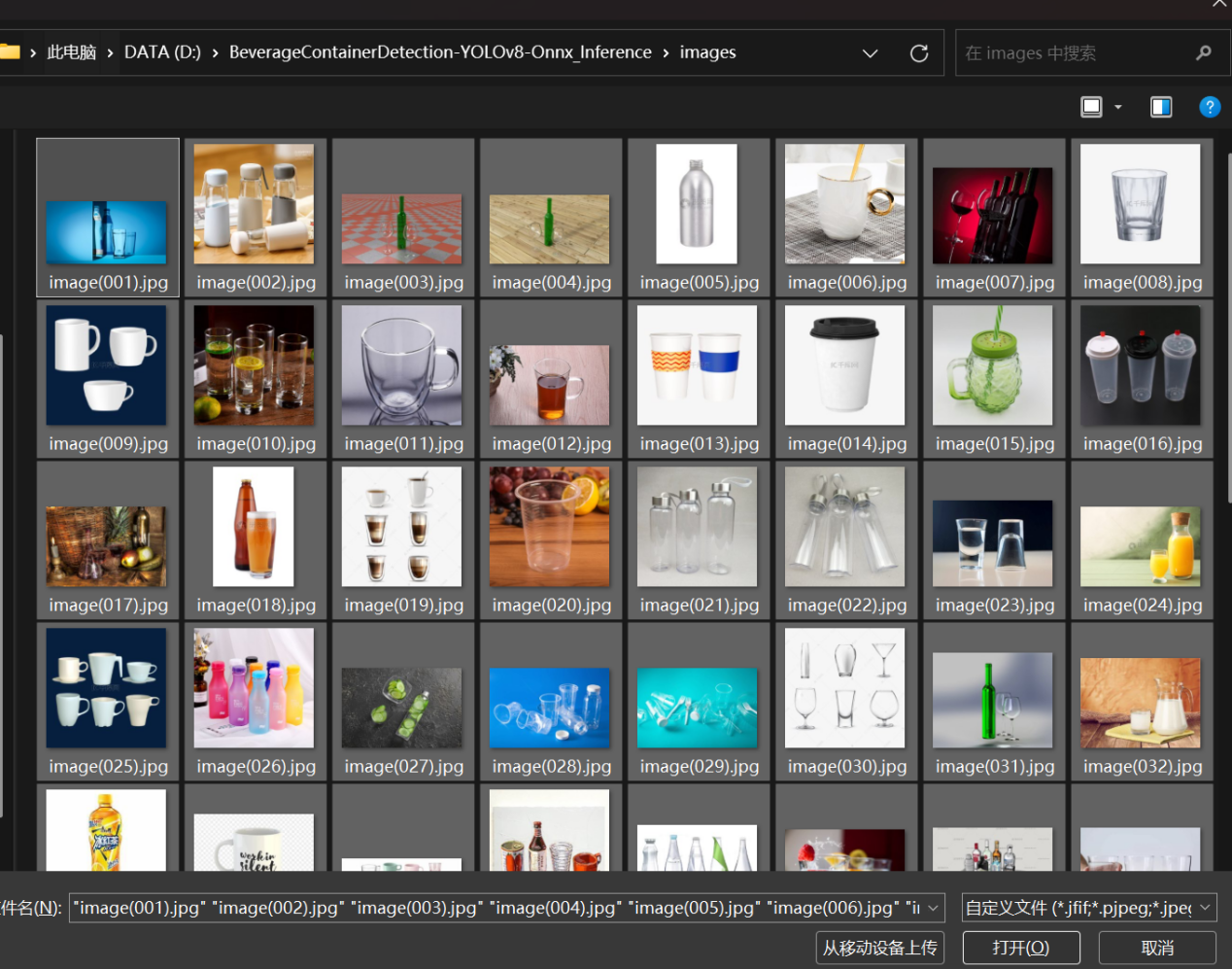
## 一、数据集准备

**数据标注：**选择ybat工具来进行数据标注

1. 新建一个文件夹images并将所收集的人体图片放入该文件夹中，如图2所示：



1. 打开ybat-master文件夹，确保里面存在名为classes.txt的文本文件，该文件定义了本任务中所需识别物体的名称，可以自行修改想要识别物体的种类，格式如下：  
   
2. 使用浏览器打开ybat-master文件夹中的ybat.html文件，点击选择文件，使用Ctrl+A选中images文件夹中的所有图片并添加进来，如图3所示。

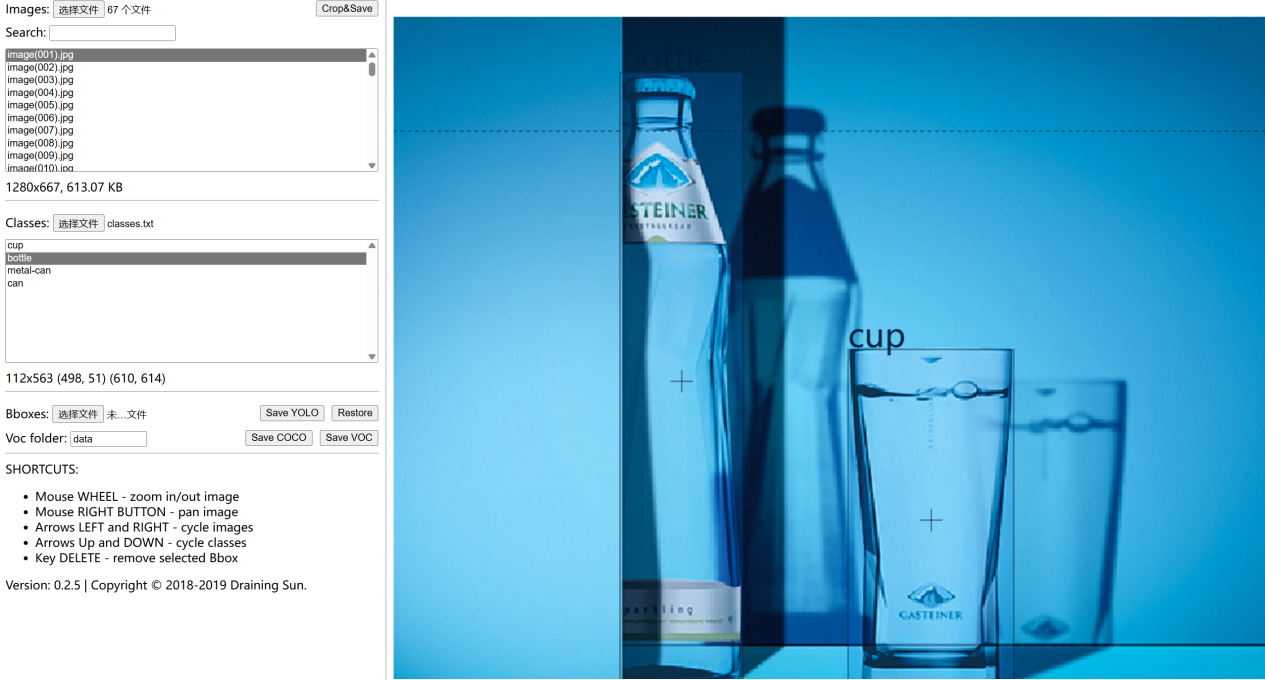


在classes处点击选择ybat-master文件夹中的classes.txt文件。

做完这一步之后可以看到待标注的图片和类别，如图所示：



在标注每一个目标时先在左侧选择要标注的class，然后在使用鼠标框选目标，注意每标注一个目标都需要先选择对应的class，否则无法框选下一个目标。框选目标时应该尽可能完美的包围住目标，不应使候选框和目标之间有较大空隙或不能完全框住目标（这很大程度上影响识别效果）。标注错误时，应使用鼠标点击选中需要删除的框，然后点击键盘的delete键，并非backspace键。以下是标注示例：



标注完一张图片之后在左侧选择下一张图片，或者使用方向键切换图片。

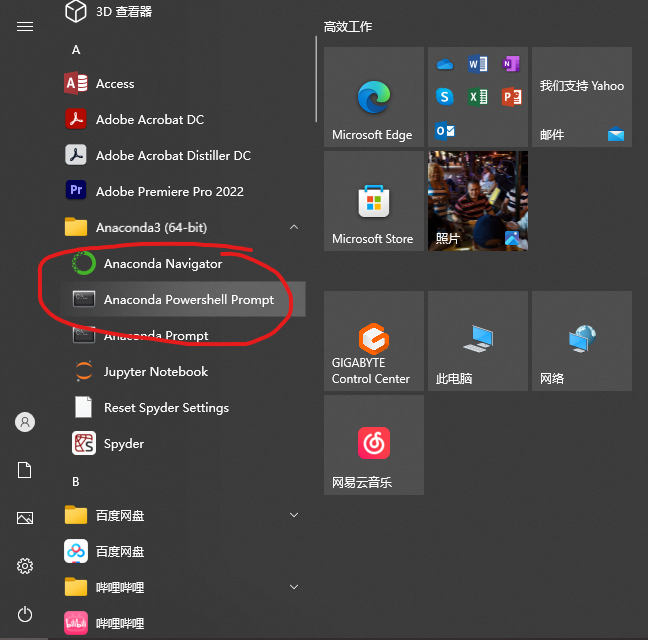
1. 标注完成之后点击Save YOLO将标注保存为yolo格式，此时浏览器将会下载一个压缩包，将压缩包解压并改名为labels。将images文件夹和labels文件夹放置到与本手册同级目录的code文件夹中。

## 二、环境配置

为了更加方便的进行实验我们使用Anaconda来进行实验环境的快速配置。你也可以选择自行配置原生python环境。

1下载并安装Anaconda。[Download Now | Anaconda](https://www.anaconda.com/download/success)

2 从开始菜单中找到并打开Anaconda Powershell Prompt



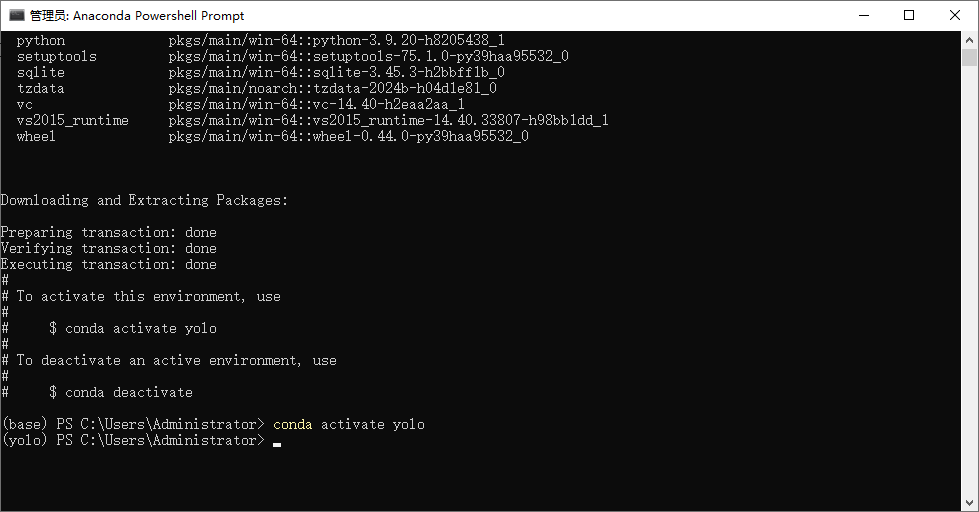
3 输入以下命令来创建名为yolo的python环境：

conda create -n yolov8 python=3.9 -y

并按回车键运行（每输入一次命令都需要输入回车）

4 进入创建好的环境：

conda activate yolo



安装依赖包：

（适用于只有cpu的电脑）pip install ultralytics==8.2.0

（适用于支持且安装了cuda的nvidia显卡的电脑，如果失败说明cuda版本太低或者没有安装cuda，需要在torch官网下载与你cuda版本相同的torch或只使用cpu，参考链接[YOLOv8超详细环境搭建以及模型训练（GPU版本）\_yolov8 gpu-CSDN博客](https://blog.csdn.net/2401_85556416/article/details/141394730)）↓

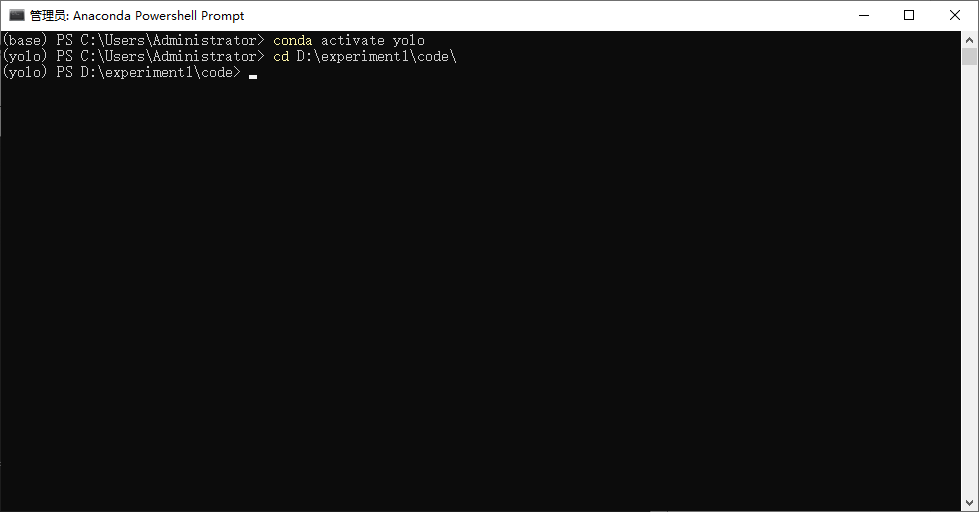
pip install torch==1.13.1+cu116 torchvision==0.14.1+cu116 torchaudio==0.13.1 --extra-index-url <https://download.pytorch.org/whl/cu116>

pip install ultralytics==8.2.0

pip install numpy==1.26.4

（实测torch 1.13.1 gpu版本需要numpy版本1.26.4左右，否则会报错，当你安装其他版本的torch时可能也会出现这种问题，请自行寻找适合的numpy版本。这不是我们的错误，而是torch官方的疏忽）

5使用cd命令定向到code文件夹中，关于cd命令的使用需自行查阅。



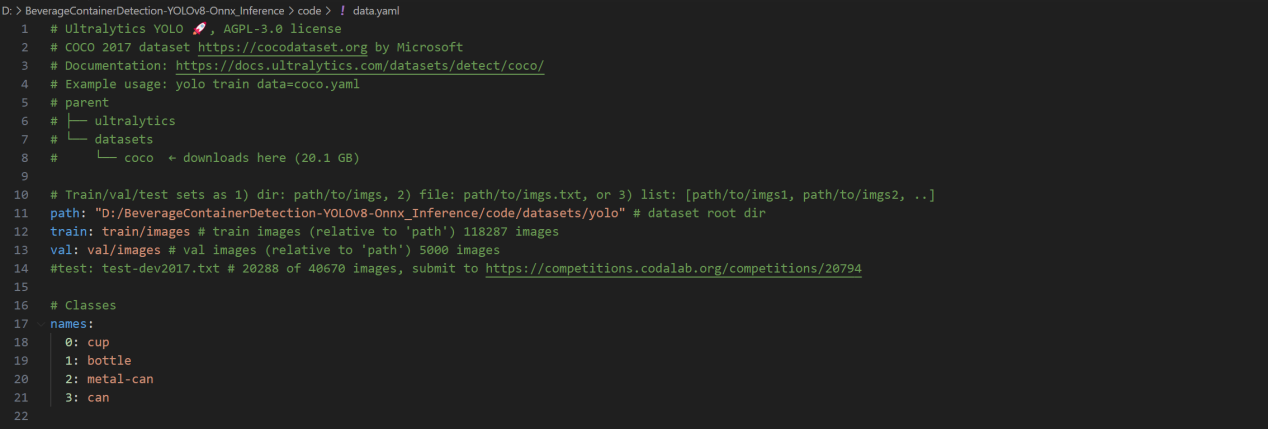
6 确保code文件夹中存在上述步骤中处理好的images和labels文件夹，运行

python split.py

来将数据划分为训练集和测试集。运行结束后将在code文件夹下出现datasets文件夹。

## 三、训练和评估

1 数据集配置：打开code中的data.yaml文件，将path改为datasets文件夹在你电脑上的绝对位置路径中不能有中文，如果你的配置与本文档上述步骤不同，也请修改相应的信息：



2 训练：

在刚才打开的命令行中运行

python run.py --model yolov8s.pt --epochs 100 --workers 0 --batch 16 --size 640

参数说明：

model 模型名字

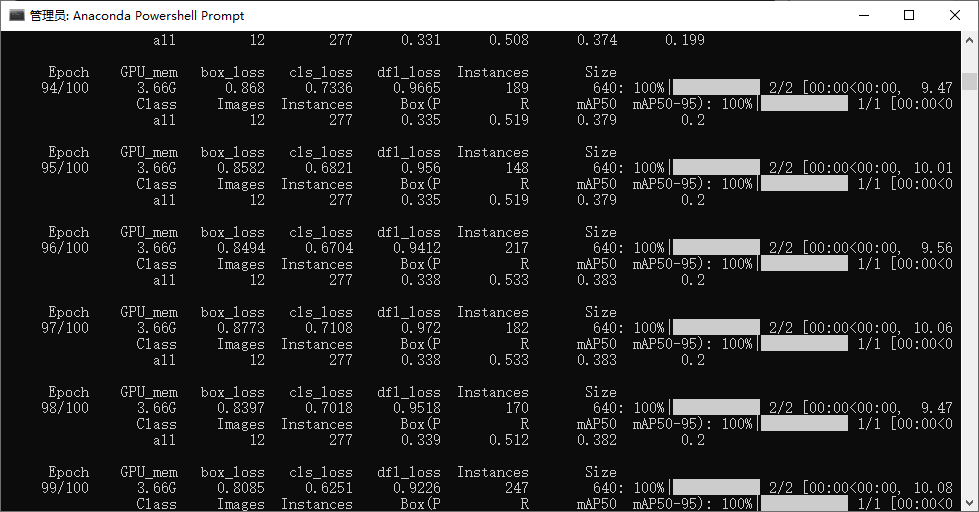
epochs 训练轮次

workers 加载数据的cpu线程数

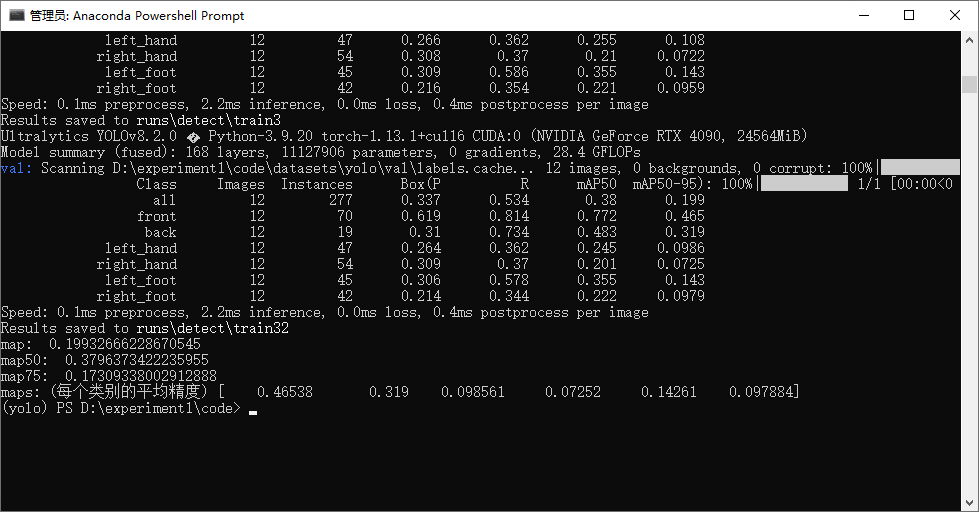
batch 批处理大小，一次处理多少张数据，如果显存爆了请调低此值

size 训练和验证时的分辨率大小

如果报错先查看是否激活了名为yolo的虚拟环境并cd到了code文件夹中，出现以下界面说明正在训练：



训练完成后将会自动执行验证评测，训练的信息将会保存到runs/detect/trainxx 文件夹中

3 验证：训练之后会自行验证，你会得到map，ap50，ap70和每个类别的精度指标。

此外，我们还提供了验证脚本用于再次验证训练好的模型：

python val.py --model ./runs/detect/trainxx/weights/best.pt

（xx是你最后一次执行的序号，在runs文件夹中自行查看）

4 预测

我们提供了脚本用于使用训练好的模型进行预测：

python predict.py --model ./runs/detect/trainxx/weights/best.pt --image path/to/your/image

python predict.py --model ./runs/detect/train2/weights/best.pt --image D:\cups\_detection\_project\yolov8\code\picture.jpg

由于数据量较少模型仍存在错检，漏检等问题。

5 调参和优化

想要得到等好的效果，超参数（包括训练的超参数和数据增强）的调整可能会很重要。Yolov8提供的超参数选项和如何设置超参数可以在这里查看：[Train - Ultralytics YOLO Docs](https://docs.ultralytics.com/modes/train/" \l "train-settings)

**四、模型推理**

在工作目录中输入指令 python inference\_camera.py ，能够启动电脑摄像头对模型.pt文件进行实时目标检测  
  
**五、模型转换为onnx形式进行部署**  
1、输入 cd deploy 进入工作目录

1. 环境配置  
   在环境中安装onnx runtime库，打开test\_requirements.py ,按照指示安装需要的库，然后运行该文件检查onnx和onnx runtime是否安装完成
2. Onnx格式转换  
   输入指令 python change\_onnx.py 完成转换，注意更改自己的文件路径，转换完成后在输入 python test\_onnx.py 验证转换完成
3. 使用onnx runtime进行推理  
   打开inference.py文件，根据注释修改路径，运行该脚本即可在本地摄像头使用onnx进行模型部署