

# YOLOv8实战TensorRT部署-Windows

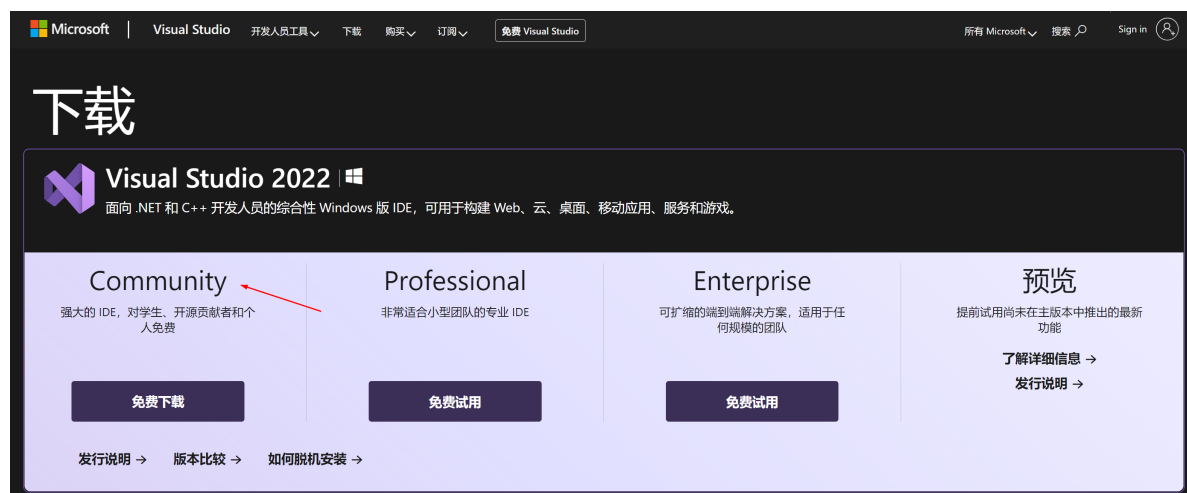
课程演示环境：Windows10, cuda 11.8, cudnn8.9

## 1 软件安装

### 1) 安装Visual Studio 2022

下载Visual Studio 社区版

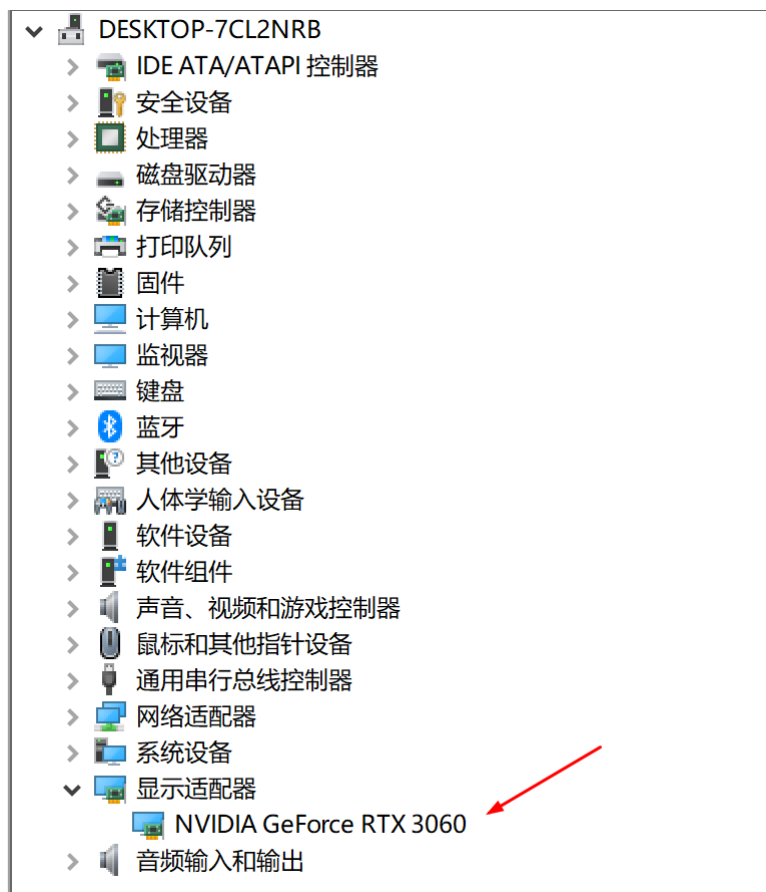
下载链接：<https://visualstudio.microsoft.com/zh-hans/downloads/>



注意：安装时可勾选“Python开发”和“C++开发”

### 2) 下载和安装nvidia显卡驱动

首先要在设备管理器中查看你的显卡型号，比如在这里可以看到我的显卡型号为RTX 3060。



NVIDIA 驱动下载: <https://www.nvidia.cn/Download/index.aspx?lang=cn>

下载对应你的英伟达显卡驱动。

## NVIDIA 驱动程序下载

在下方的下拉列表中进行选择，针对您的 NVIDIA 产品确定合适的驱动。

产品类型:	GeForce	▼
产品系列:	GeForce RTX 30 Series	▼
产品家族:	GeForce RTX 3060	▼
操作系统:	Windows 10 64-bit	▼
下载类型:	Studio 驱动程序 (SD)	▼
语言:	English (US)	▼

?

下载之后就是简单的下一步执行直到完成。

完成之后，在cmd中输入执行：

```
nvidia-smi
```

如果输出下图所示的显卡信息，说明你的驱动安装成功。

命令提示符

```
C:\Users\Bai>nvidia-smi
Tue Apr 18 22:58:11 2023
```

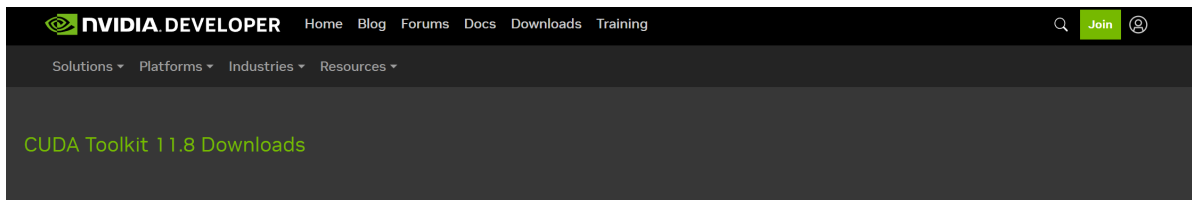
NVIDIA-SMI 531.61				Driver Version: 531.61			CUDA Version: 12.1		
GPU	Name	TCC/WDDM	Bus-Id	Disp. A	Volatile	Uncorr.	ECC		
Fan	Temp	Perf	Pwr:Usage/Cap	Memory-Usage	GPU-Util	Compute	M. MIG		
0	NVIDIA GeForce RTX 3060	WDDM	00000000:01:00.0	On			N/A		
0%	40C	P8	15W / 170W	1495MiB / 12288MiB	7%	Default	N/A		

注：图中的 CUDA Version是当前Driver版本能支持的最高的CUDA版本

### 3) 下载CUDA

CUDA用的是11.8版本

cuda下载链接：[https://developer.nvidia.com/cuda-downloads?target\\_os=Windows&target\\_arch=x86\\_64&target\\_version=10&target\\_type=exe-local](https://developer.nvidia.com/cuda-downloads?target_os=Windows&target_arch=x86_64&target_version=10&target_type=exe-local)



Home

#### Select Target Platform

Click on the green buttons that describe your target platform. Only supported platforms will be shown. By downloading and using the software, you agree to fully comply with the terms and conditions of the [CUDA EULA](#).

Operating System	Linux	Windows			
Architecture	x86_64				
Version	10	11	Server 2016	Server 2019	Server 2022
Installer Type	exe (local)	exe (network)			

下载后得到文件：cuda\_11.8.0\_522.06\_windows.exe

执行该文件进行安装。

### 4) 安装cuda

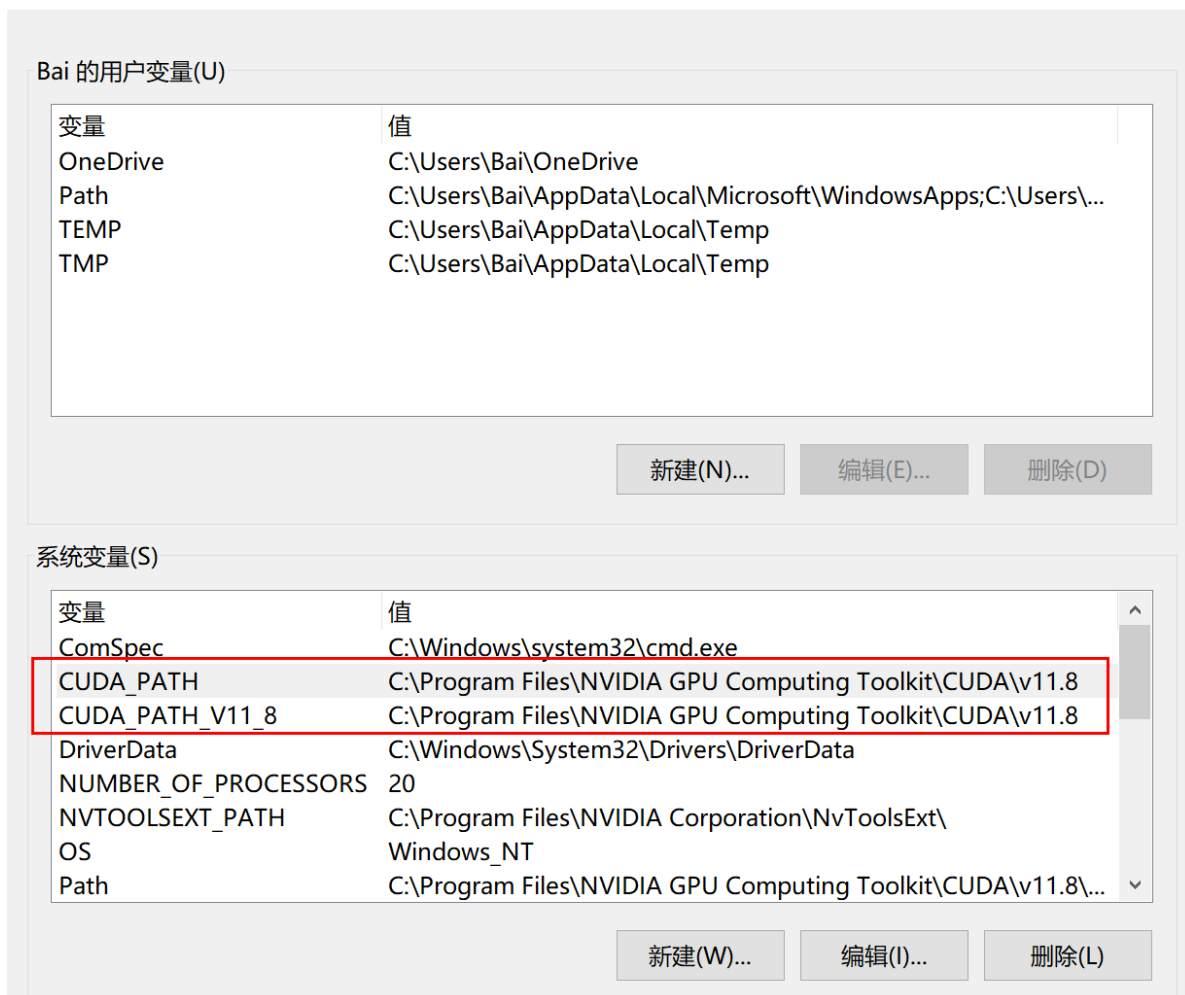
(1) 将cuda运行安装，建议默认路径



安装时可以勾选Visual Studio Integration

## (2) 安装完成后设置环境变量

环境变量



看到系统中多了CUDA\_PATH和CUDA\_PATH\_V11\_8两个环境变量。

## 5) 下载cuDNN

cuda下载地址: <https://developer.nvidia.com/cudnn>

需要有NVIDIA账号

注意: cudnn版本要和cuda版本匹配

## cuDNN Download

NVIDIA cuDNN is a GPU-accelerated library of primitives for deep neural networks.

☒ I Agree To the Terms of the [cuDNN Software License Agreement](#)

Note: Please refer to the [Installation Guide](#) for release prerequisites, including supported GPU architectures and compute capabilities, before downloading.

For more information, refer to the cuDNN Developer Guide, Installation Guide and Release Notes on the [Deep Learning SDK Documentation](#) web page.

[Download cuDNN v8.9.0 \(April 11th, 2023\), for CUDA 12.x](#)

[Download cuDNN v8.9.0 \(April 11th, 2023\), for CUDA 11.x](#)

### Local Installers for Windows and Linux, Ubuntu(x86\_64, armsbsa)

[Local Installer for Windows \(Zip\)](#)

[Local Installer for Linux x86\\_64 \(Tar\)](#)

[Local Installer for Linux PPC \(Tar\)](#)

[Local Installer for Linux SBSA \(Tar\)](#)

[Local Installer for Debian 11 \(Deb\)](#)

[Local Installer for Ubuntu18.04 x86\\_64 \(Deb\)](#)

[Local Installer for Ubuntu20.04 x86\\_64 \(Deb\)](#)

[Local Installer for Ubuntu22.04 x86\\_64 \(Deb\)](#)

[Local Installer for Ubuntu20.04 aarch64sbsa \(Deb\)](#)

[Local Installer for Ubuntu22.04 aarch64sbsa \(Deb\)](#)

[Local Installer for Ubuntu20.04 cross-sbsa \(Deb\)](#)

[Local Installer for Ubuntu22.04 cross-sbsa \(Deb\)](#)

下载后得到文件: cudnn-windows-x86\_64-8.9.0.131\_cuda11-archive.zip

## 6) 安装cuDNN

### 复制cudnn文件

对于cudnn直接将其解开压缩包, 然后需要将bin,include,lib中的文件复制粘贴到cuda的文件夹下

C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v11.8

注意: 对整个文件夹bin,include,lib选中后进行复制粘贴

## 7) CUDA安装测试

最后测试cuda是否配置成功:

打开CMD执行:

```
nvcc -V
```

即可看到cuda的信息

```
Microsoft Windows [版本 10.0.19045.2006]
(c) Microsoft Corporation。保留所有权利。

C:\Users\Bai>nvcc -V
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2022 NVIDIA Corporation
Built on Wed_Sep_21_10:41:10_Pacific_Daylight_Time_2022
Cuda compilation tools, release 11.8, V11.8.89
Build cuda_11.8.r11.8/compiler.31833905_0
```

## 8) 安装Anaconda

Anaconda 是一个用于科学计算的 Python 发行版，支持 Linux, Mac, Windows, 包含了众多流行的科学计算、数据分析的 Python 包。

1) 下载安装包

Anaconda下载Windows版: <https://www.anaconda.com/>

2) 然后安装anaconda

## 9) 安装pytorch

创建虚拟环境，环境名字可自己确定，这里本人使用mypytorch作为环境名：

```
conda create -n mypytorch python=3.9
```

Anaconda Prompt

```
(base) C:\Users\Bai>conda create -n mypytorch python=3.9
```

安装成功后激活mypytorch环境：

```
conda activate mypytorch
```

在所创建的mypytorch环境下安装pytorch, 执行命令：

```
conda install pytorch torchvision torchaudio pytorch-cuda=11.8 -c pytorch -c nvidia
```

注意：11.8处应为自己电脑上的cuda版本号

**离线安装：**

下载网址: <https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/pytorch/win-64/>

安装pytorch2.0版本: pytorch-2.0.0-py3.9\_cuda11.8\_cudnn8\_0.tar.bz

```
conda install --offline pytorch-2.0.0-py3.9_cuda11.8_cudnn8_0.tar.bz
```

## 2 YOLOv8项目克隆和安装

## 1) 克隆YOLOv8并安装

安装Git软件 (<https://git-scm.com/downloads>) , 克隆项目到本地 (如d:)

项目repo网址: <https://github.com/ultralytics/ultralytics>

在 Git CMD窗口中执行:

```
git clone https://github.com/ultralytics/ultralytics
```

在mypytorch虚拟环境下执行:

```
cd ultralytics
```

```
pip install -e .
```

## 3) 下载预训练权重文件

下载yolov8预训练权重文件, 并放置在新建立的weights文件夹下

例如: D:\ultralytics\ultralytics\weights

## 百度网盘下载链接:

链接: [https://pan.baidu.com/s/1q\\_wSVCQDch70T4hl2sidWA](https://pan.baidu.com/s/1q_wSVCQDch70T4hl2sidWA)

提取码: pqjg

## 4) 安装测试

预测图片:

```
yolo predict model=D:/ultralytics/ultralytics/weights/yolov8s.pt  
source=D:/ultralytics/ultralytics/assets/bus.jpg
```

批量预测图片:

```
yolo predict model=D:/ultralytics/ultralytics/weights/yolov8s.pt  
source=D:/ultralytics/ultralytics/assets
```

预测图片并存储推理结果:

```
yolo predict model=D:/ultralytics/ultralytics/weights/yolov8s.pt  
source=D:/ultralytics/ultralytics/assets/bus.jpg save_txt
```

预测摄像头:

```
yolo predict model=D:/ultralytics/ultralytics/weights/yolov8s.pt source=0 show
```

命令参数说明: <https://docs.ultralytics.com/modes/predict/>

## 3 TensorRT安装

参考[官网安装教程](#)

## 1) 下载安装包:

1. Go to: <https://developer.nvidia.com/tensorrt> (需要Nvidia账户)
2. 点击 **立即下载**(Download Now)
3. 选择合适的TensorRT版本 本人选择: TensorRT 8.X
4. Select the check-box to agree to the license terms.
5. Click the package you want to install. Your download begins.  
本人选择: TensorRT 8.6 GA for x86\_64 Architecture  
本人使用的版本: TensorRT-8.6.1.6.Windows10.x86\_64.cuda-11.8.zip

## 2) 配置环境变量

1. 将下载的压缩文件拷贝进来解压
2. 解压得到TensorRT-8.6.1.6的文件夹, 将里边的lib绝对路径添加到环境变量Path中, 即  
D:\TensorRT-TensorRT-8.6.1.6\lib
3. 将TensorRT解压位置 D:\TensorRT-TensorRT-8.6.1.6\lib下的所有dll文件复制到C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v11.8\bin目录下

## 3) 安装pycuda

如果要使用python接口的tensorrt, 则需要安装pycuda

在mypytorch虚拟环境下执行

```
pip install pycuda
```

## 4) 测试TensorRT示例代码

### 1. 配置VS2022

用VS2022打开sampleMNIST示例sln文件 (D:\TensorRT-8.6.1.6\samples\sampleOnnxMNIST)

提示: 安装C++桌面开发

- a. 将D:\TensorRT-8.6.1.6\lib加入 项目->属性->VC++目录->可执行文件目录
- b. 将D:\TensorRT-8.6.1.6\lib加入 VC++目录->库目录
- c. 将D:\TensorRT-8.6.1.6\include加入C/C++ --> 常规 --> 附加包含目录
- d. 将nvinfer.lib、nvinfer\_plugin.lib、nvonnxparser.lib和nvparsers.lib加入链接器->输入->附加依赖项  
D:\TensorRT-8.6.1.6\lib\\*.lib

2. 编译后可执行得到测试结果

生产->生成解决方案

错误: 严重性 代码 说明 项目 文件 行 禁止显示状态

错误 MSB8036 找不到 Windows SDK 版本 10.0.17134.0。请安装所需版本的 Windows SDK, 或者在  
项目属性页中或通过右键单击解决方案并选择“重定解决方案目标”来更改 SDK 版本。 sample\_onnx\_mnist  
G:\Program Files\Microsoft Visual  
Studio\2022\Community\MSBuild\Microsoft\VC\v170\Microsoft.Cpp.WindowsSDK.targets 46

菜单中选择 项目->重定解决方案目标



### 重定向项目

以下项目使用早期版本的 Visual C++ 平台工具集。可升级项目以面向最新的 Microsoft 工具集。还可从安装在计算机上的项中选择目标 Windows SDK 版本。

Windows SDK 版本: 10.0 (最新安装的版本) ▾

平台工具集: 升级到 v143 ▾

..\sampleOnnxMNIST\sample\_onnx\_mnist.vcxproj

确定

取消

## 4 安装OpenCV

下载opencv4.7: <https://opencv.org/releases/>

接着只需要将其解压缩，然后配置环境变量就行了。

运行exe（其实是解压），将压缩包解压到相应目录，如：

D:\opencv\build\x64\vc16\bin

在系统变量 Path 的末尾添加：D:\opencv\build\x64\vc16\bin

重启电脑

## 5 YOLOv8的TensorRT加速

### 1) 下载tensorrtx项目文件

<https://github.com/wang-xinyu/tensorrtx>

从课程网盘下载tensorrtx.zip文件，并解压

### 百度网盘下载链接：

链接: [https://pan.baidu.com/s/1q\\_wSVCQDch70T4hl2sidWA](https://pan.baidu.com/s/1q_wSVCQDch70T4hl2sidWA)

提取码: pqjg

### 2) 生成yolov8s.wts文件

下载yolov8权重文件到ultralytics/ultralytics/weights目录下

// 例如 <https://github.com/ultralytics/assets/releases/yolov8n.pt>

拷贝gen\_wts.py文件到ultralytics/ultralytics目录下

```
cp tensorrtx/yolov8/gen_wts.py ultralytics/ultralytics
```

在ultralytics/ultralytics路径下执行

```
python gen_wts.py
```

可在gen\_wts.py中修改需要转换的权重文件

```
pt_file = "./weights/yolov8s.pt"
wts_file = "./weights/yolov8s.wts"
```

// a file 'yolov8s.wts' will be generated.

**拷贝文件yolov8s.wts文件到tensorrtx/yolov8/workspace目录下**

注意：workspace目录自己创建；另外创建tensorrtx/yolov8/build目录

### 3) 修改CMakeLists.txt

修改D:\tensorrtx\yolov8下的CMakeLists.txt文件，修改后的CMakeLists.txt见网盘。

**注意：**使用时需要根据自己电脑上的软件位置和GPU架构做相应的修改。

GPU架构官网查询：<https://developer.nvidia.com/cuda-gpus>

### 4) 修改include/config.h

## 5) 编译tensorrtx/yolov8

#### (1) 安装cmake

下载地址<https://cmake.org/>

#### (2) 执行cmake-gui来配置project

#### (3) 点击 Configure并设置环境

#### (4) 点击Finish,等待Configure done

#### (5) 点击Generate并等待Generate done

#### (6) 点击Open Project

注意：使用Release模式；需要调试可使用Debug模式

#### (7) 生成解决方案

## 6 执行TensorRT加速后的yolov8命令(C++)

从build\Release下拷贝yolov8.exe和myplugins.dll到D:/tensorrtx/yolov8/workspace目录下，

**创建yolov8引擎文件**

// serialize model to plan file i.e. yolov8s.engine; yolov8.exe -s [.wts] [.engine] [n/s/m/l/x]

```
yolov8.exe -s yolov8s.wts yolov8s.engine s
```

**图像文件推理**

// deserialize plan file and run inference

后处理用gpu

```
yolov8.exe -d yolov8s.engine D:/tensorrtx/yolov8/workspace/images/bus.jpg g
```

后处理用cpu

```
yolov8.exe -d yolov8s.engine D:/tensorrtx/yolov8/workspace/images/bus.jpg c
```

#### 图像文件夹推理

```
yolov8.exe -d yolov8s.engine D:/tensorrtx/yolov8/workspace/images g
```

#### 视频文件推理

```
yolov8.exe -d yolov8s.engine D:/tensorrtx/yolov8/workspace/video/driving.mp4 g
```

## 7 执行TensorRT加速后的yolov8命令(Python)

#### 创建yolov8引擎文件这一步同c++

从build\Release下拷贝myplugins.dll到D:/tensorrtx/yolov8/workspace目录下,

拷贝yolov8\_trt.py到D:/tensorrtx/yolov8/workspace目录下

在mypytorch虚拟环境下执行

#### 图像文件推理

```
python yolov8_trt.py yolov8s.engine D:/tensorrtx/yolov8/workspace/images/bus.jpg
```

#### 图像文件夹推理

```
python yolov8_trt.py yolov8s.engine D:/tensorrtx/yolov8/workspace/images
```

#### 视频文件推理

```
python yolov8_trt.py yolov8s.engine D:/tensorrtx/yolov8/workspace/video/driving.mp4
```

## 8 INT8量化加速

1. 准备校准图片 ( calibration images ) , 可以从你的训练集随机选择 1000张图片。对于coco, 可以从百度网盘下载校准图片集 `coco_calib.zip`
2. 解压 `coco_calib.zip` 到 `tensorrtx/yolov8/workspace`
3. 在文件 `include/config.h` 中设置 `USE_INT8`  
然后, 使用 `camke-gui` 重新编译  
File->Delete Cache
4. serialize the model and test  
D:/tensorrtx/yolov8/workspace目录下, 执行

```
yolov8.exe -s yolov8s.wts yolov8s.engine s
```

```
yolov8.exe -d yolov8s.engine D:/tensorrtx/yolov8/workspace/images/bus.jpg g
```



# 声明

本课程的数据集、程序文件以及课件的演示文稿、视频由讲师白勇拥有知识产权的权利。只限于学习者在本课程使用，不得在课程范围外向任何第三方散播。任何其他人或机构不得盗版、复制、仿造。

