

# 环境部署

## 一、创建新环境 (conda)

### 1. 创建虚拟环境

进入Anaconda Prompt后创建新的虚拟环境，环境名为rtdenv(可自定义)

```
1 | conda create -n rtdenv python=3.11
```

### 2. 进入虚拟环境

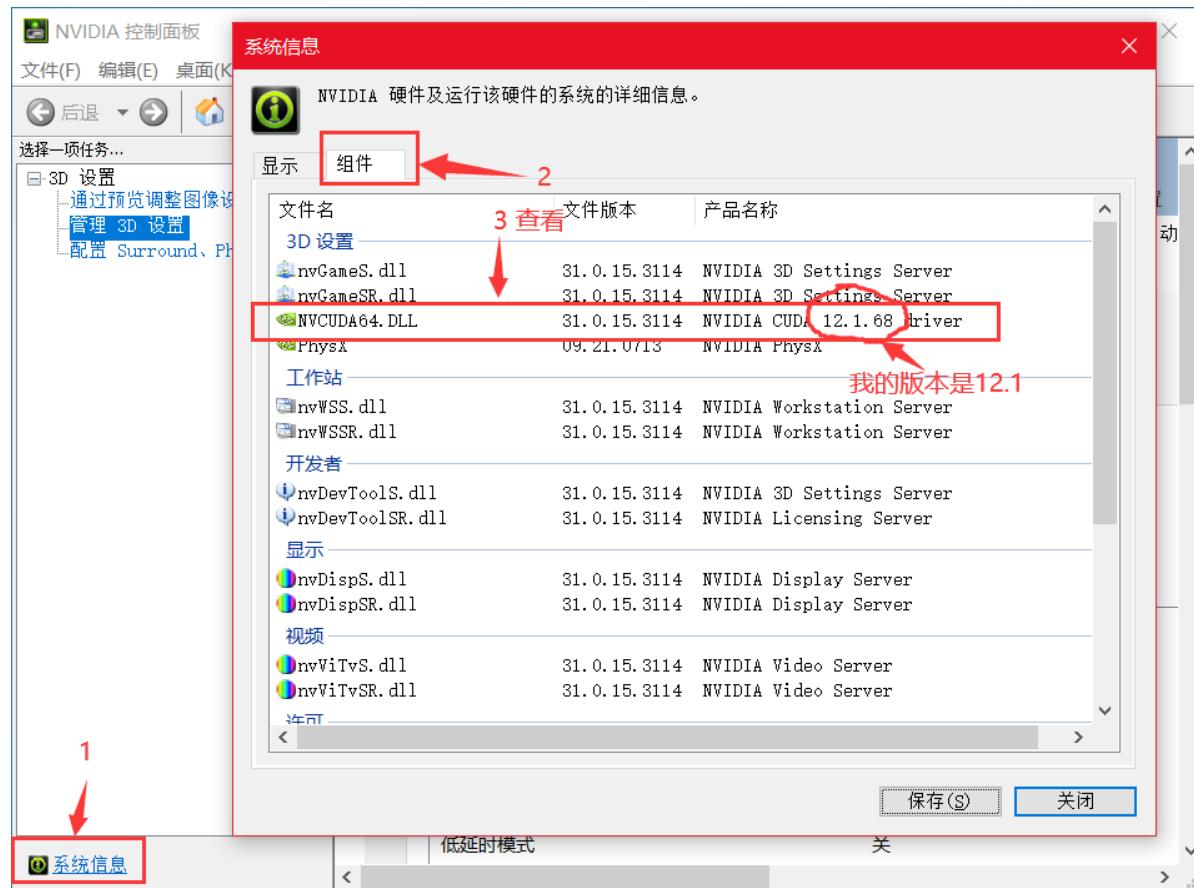
```
1 | conda activate rtdenv
```

```
(base) C:\Users\Tek44>conda activate rtdenv  
(rtdenv) C:\Users\Tek44>
```

## 二、根据cuda版本与python版本下载pytorch

### 1. 查看cuda版本 (A卡可跳过安装cuda与cudnn步骤)

桌面右键打开NVIDIA控制面板，根据图示顺序点击查看cuda版本 (系统信息->组件)



也可以在cmd中输入nvidia-smi查看

NVIDIA-SMI 531.14			Driver Version: 531.14			CUDA Version: 12.1		
GPU	Name	TCC/WDDM	Bus-Id	Disp. A	Volatile	Uncorr.	ECC	
Fan	Temp	Pwr:Usage/Cap		Memory-Usage	GPU-Util	Compute M.	MIG M.	
0	NVIDIA GeForce GTX 1660 Ti	WDDM 1W / N/A	00000000:01:00.0	Off 394MiB / 6144MiB	0%	N/A Default	N/A	
<hr/>								
Processes:								
GPU	GI	CI	PID	Type	Process name	GPU Memory Usage		
ID	ID					N/A		
0	N/A	N/A	23136	C+G	D:\Typora\Typora.exe	N/A		

## 2. 如果无cuda，安装cuda（上一步中检查有cuda可跳过此步）

### (1) 安装显卡驱动

进入nvidia官网<https://www.nvidia.cn/Download/index.aspx?lang=cn>，根据显卡版本选择下载对应驱动并安装

## 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 GTX 16 Series (Notebooks)
产品家族:	GeForce GTX 1660 Ti
操作系统:	Windows 10 64-bit
下载类型:	Studio 驱动程序 (SD)
语言:	Chinese (Simplified)

搜索

- 安装完毕后可在cmd输入nvidia -smi检查

### (2) 下载cuda

CUDA下载地址: <https://developer.nvidia.com/cuda-toolkit-archive>

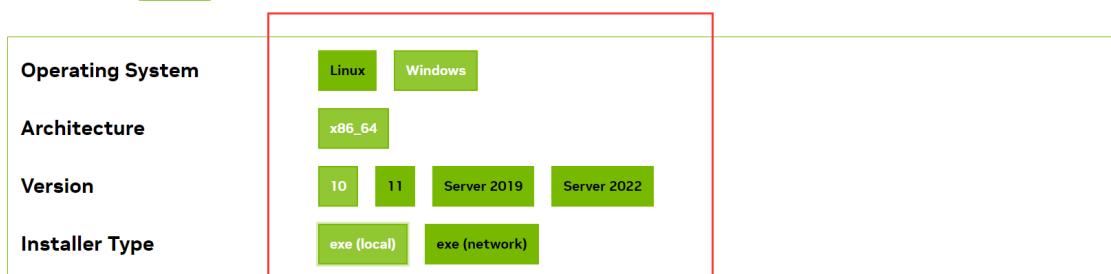
- 进入网址后选择CUDA版本，此处以选择12.1.1版本为例(可以安装更低如11.8)
- 选择参数并开始下载

因为安装包过大邮件无法发送，故这里没有提供已下载文件

# CUDA Toolkit 12.1 Downloads

## 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](#).



## Download Installer for Windows 10 x86\_64

The base installer is available for download below.

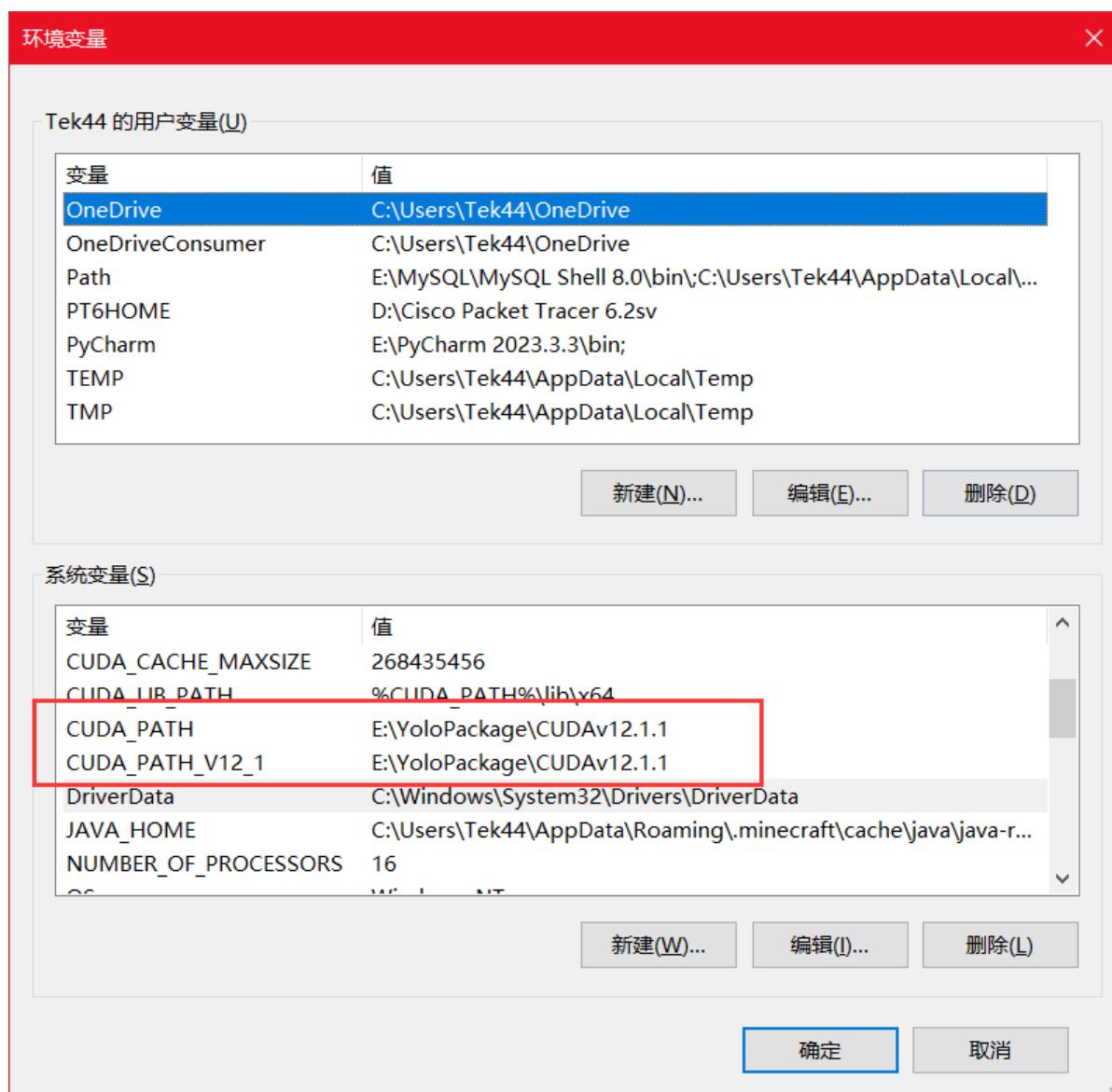
> Base Installer

Download (3.1 GB)

- 下载完获得文件：cuda\_12.1.1\_531.14\_windows.exe

执行该文件进行安装，安装中的自定义安装选项可全勾选，注意要记住安装目录位置。

## (3) cuda安装完毕后设置环境变量



此处我在安装时选择了自定义目录，环境变量中选择的CUAV12.1.1目录如下

名称	修改日期	类型	大小
bin	2024/1/30 12:21	文件夹	
compute-sanitizer	2024/1/29 20:47	文件夹	
Development	2024/1/29 15:33	文件夹	
extras	2024/1/29 20:47	文件夹	
include	2024/1/30 12:21	文件夹	
lib	2024/1/29 20:47	文件夹	
libnvvvm-samples	2024/1/29 20:47	文件夹	
libnvvp	2024/1/29 20:47	文件夹	
nvml	2024/1/29 20:47	文件夹	
nvvm	2024/1/29 20:47	文件夹	
res	2024/1/29 20:47	文件夹	
src	2024/1/29 20:47	文件夹	
tools	2024/1/29 20:47	文件夹	
CUDA_Toolkit_Release_Notes.txt	2023/4/4 8:41	文本文档	79 KB
DOCS	2023/4/4 8:41	文件	1 KB
EULA.txt	2023/4/4 8:41	文本文档	61 KB
LICENSE	2023/4/4 9:21	文件	61 KB
README	2023/4/4 8:41	文件	1 KB
version.json	2023/4/17 22:56	JSON File	3 KB

### 3. 安装cudnn

cudnn下载地址: <https://developer.nvidia.com/rdp/cudnn-archive>

(1) 根据cuda版本选择下载版本, 此处以选择8.9.0的cuda12.X版本为例 (注意此处需要登录nvidia账号才可下载, 可以直接注册新账号登录)

env-build文件夹中提供了已下载的文件, 如果与演示版本一致可以直接使用安装, 跳过登录下载。

文件名: cudnn-windows-x86\_64-8.9.0.131\_cuda12-arcnive.zip

Download cuDNN v8.9.1 (May 5th, 2023), for CUDA 11.x

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

- 下载后得到文件: cudnn-windows-x86\_64-8.9.0.131\_cuda12-arcnive.zip

#### (2) 安装cudnn文件

打开刚才下载的zip压缩包, 然后将bin, include, lib中的文件复制粘贴到cuda的文件夹下(既环境变量中设置的文件夹)

注意: 对整个文件夹bin, include, lib选中后进行复制粘贴

## 4. 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

```
C:\Users\Tek44>nvcc -V
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2023 NVIDIA Corporation
Built on Mon_Apr_3_17:36:15_Pacific_Daylight_Time_2023
Cuda compilation tools, release 12.1, V12.1.105
Build cuda_12.1.r12.1/compiler.32688072_0
```

```
C:\Users\Tek44>
```

## 5. 虚拟环境中安装pytorch (A卡Compute PlatForm选项选择CPU)

### (1) 官网获取下载命令

地址: <https://pytorch.org/get-started/locally/>

根据下载版本选择对应的选项，下方获取命令

如果与演示版本一致可以直接使用以下命令安装:

```
1 | conda install pytorch torchvision torchaudio pytorch-cuda=12.1 -c pytorch
   -c nvidia
```

The screenshot shows the PyTorch download page with various configuration options. The 'Compute Platform' section is highlighted with a red box, specifically the 'CUDA 12.1' button. The 'Run this Command:' box also contains a command with a red border.

PyTorch Build	Stable (2.3.1)	Preview (Nightly)			
Your OS	Linux	Mac	Windows		
Package	Conda	Pip	LibTorch	Source	
Language	Python	C++ / Java			
Compute Platform	CUDA 11.8	CUDA 12.1	CUDA 12.4	ROCM 6.0	CPU
Run this Command:	<pre>conda install pytorch torchvision torchaudio pytorch-cuda=12.1 -c pytorch -c nvidia</pre>				

获取命令：

```
1 | conda install pytorch torchvision torchaudio pytorch-cuda=12.1 -c pytorch -
   -c nvidia
```

## (2) 将获取的命令输入在虚拟环境中运行完成pytorch的安装

注意已切换至新建立的虚拟环境！

```
(rtdenv) C:\Users\Tek44>conda install pytorch torchvision torchaudio pytorch-cuda=12.1 -c pytorch -c nvidia
```

等待一段时间即完成安装

## 三、虚拟环境中下载其余包

虚拟环境安装完pytorch后即可安装其他包，其目录在 `env-build/requirements.txt` 中

### 1. 在虚拟环境中进入env-build文件夹

注意已切换至新建立的虚拟环境！

```
1 | cd /d (env-build文件夹的绝对路径)
```

```
(rtdenv) C:\Users\Tek44>cd /d E:\YoloPackage\env-build  
(rtdenv) E:\YoloPackage\env-build>
```

### 2. 批量安装其余包

```
1 | pip install -r requirements.txt
```

```
(rtdenv) C:\Users\Tek44>pip install -r requirements_loading.txt  
Collecting django==5.0.4 (from -r requirements_loading.txt (line 1))  
  Using cached Django-5.0.4-py3-none-any.whl.metadata (4.1 kB)  
Collecting ultralytics==8.1.16 (from -r requirements_loading.txt (line 2))  
  Downloading ultralytics-8.1.16-py3-none-any.whl.metadata (40 kB)  
          ----- 40.4/40.4 kB 148.4 kB/s eta 0:00:  
Collecting channels==3.0.5 (from -r requirements_loading.txt (line 3))  
  Using cached channels-3.0.5-py3-none-any.whl.metadata (1.3 kB)
```

等待一段时间即完成安装

## 配置调整

准备：

- 上述步骤建立的环境命令台进入Real\_Time\_DetectWEB文件夹（manage.py同目录）

```
(rtdenv) E:\YoloPackage>cd RTD-web-pack/Real_Time_DetectWEB
(rtdenv) E:\YoloPackage\RTD-web-pack\Real_Time_DetectWEB>
此电脑 > 新加卷 (E:) > YoloPackage > RTD-web-pack > Real_Time_DetectWEB
```

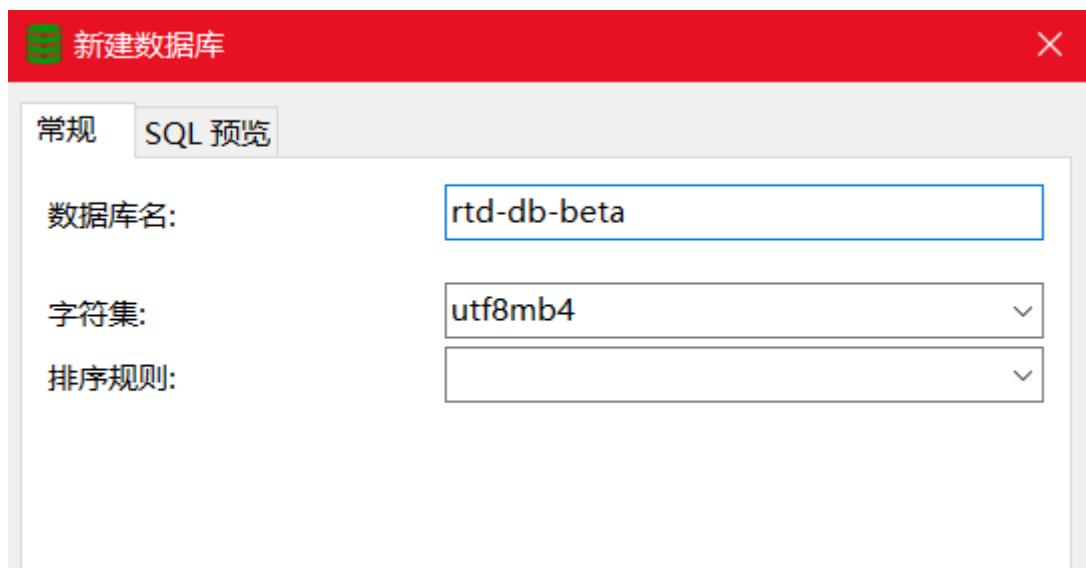
名称	修改日期	类型	大小
.git	2024/6/11 17:16	文件夹	
.idea	2024/6/11 17:16	文件夹	
media	2024/6/11 17:16	文件夹	
Real_Time_DetectWEB	2024/6/11 17:16	文件夹	
RTDweb	2024/6/11 17:16	文件夹	
static	2024/4/26 15:00	文件夹	
manage.py	2024/4/4 15:28	JetBrains PyCharm	1 KB
README.md	2024/4/4 21:39	Markdown File	1 KB
yolov8s.pt	2024/4/11 14:01	PT 文件	22,045 KB

- Pycharm或其他方式打开项目代码准备修改部分配置 (Real\_Time\_DetectWEB即为项目文件夹)
- 准备mysql

## 一、数据库配置

### 1. 新建mysql数据库 (Navicat演示)

任意新建数据库，并记录下数据库的用户名，密码，端口



### 2. 打开项目代码，更改数据库设置

找到 Real\_Time\_DetectWEB/Real\_Time\_DetectWEB/settings.py 文件修改其中的数据库四项配置内容

```

    项目 ̗  ⊕  ◁  ×  :  —  settings.py x

    v Real_Time_DetectWEB E:\YoloPackage\RTDweb
      > media
      > Real_Time_DetectWEB
        _init_.py
        asgi.py
        routings.py
        settings.py
        urls.py
        wsgi.py
      > RTDweb
        static
        manage.py
      README.md
      yolov8s.pt
    > 外部库
    临时文件和控制台

  74 # Database
  75 # https://docs.djangoproject.com/en/4.1/ref/settings/#databases
  76
  77 DATABASES = {
  78     'default': {
  79         'ENGINE': 'django.db.backends.mysql',
  80         'NAME': 'r_t_d_data', 新建的数据库名
  81         'USER': 'root', 数据库的用户名
  82         'PASSWORD': '123456', 数据库的密码
  83         'HOST': 'localhost',
  84         'PORT': 3306 数据库的端口
  85     }
  86 }
  87
  88 # Password validation
  89 # https://docs.djangoproject.com/en/4.1/ref/settings/#auth-password-validators
  90
  91 AUTH_PASSWORD_VALIDATORS = [
  92     {
  93         'NAME': 'django.contrib.auth.password_validation.UserAttributeSimilarityValidator',
  94     },
  95     {
  96         'NAME': 'django.contrib.auth.password_validation.MinimumLengthValidator',
  97     },
  98     {
  99         'NAME': 'django.contrib.auth.password_validation.CommonPasswordValidator',
  100    },
  101    {
  102        'NAME': 'django.contrib.auth.password_validation.NumericPasswordValidator',
  103    }
  104]

```

修改示范：

```

    v DATABASES = {
    v     'default': {
    v         'ENGINE': 'django.db.backends.mysql',
    v         'NAME': 'rtd-db-beta', // 修改部分
    v         'USER': 'root123',
    v         'PASSWORD': '0000000',
    v         'HOST': 'localhost',
    v         'PORT': 3307
    v     }
    v }

```

### 3. 初始化数据库

conda命令台中执行以下命令，完成数据库表的初始化创建

```

1 | python manage.py makemigrations #一步
2 | python manage.py migrate      #二步

```

```

(rtdenv) E:\YoloPackage\RTD-web-pack\Real_Time_DetectWEB>python manage.py makemigrations
No changes detected

(rtdenv) E:\YoloPackage\RTD-web-pack\Real_Time_DetectWEB>python manage.py migrate
Operations to perform:
  Apply all migrations: RTDweb, admin, auth, contenttypes, sessions
Running migrations:
  Applying RTDweb.0001_initial... OK
  Applying RTDweb.0002_alter_originimg_name... OK
  Applying RTDweb.0003_alter_originimg_img_path... OK
  Applying RTDweb.0004_originimg_is_detect... OK

```

## 二、运行程序

配置完毕后即可运行程序，conda命令台中执行如下指令或者使用编译器运行项目

```
1 | python manage.py runserver
```

```
(rtdenv) E:\YoloPackage\RTD-web-pack\Real_Time_DetectWEB>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
June 11, 2024 - 17:51:21
Django version 5.0.4, using settings 'Real_Time_DetectWEB.settings'
Starting ASGI/Channels version 3.0.5 development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

加载结束进入<http://127.0.0.1:8000/>打开网页。

- **CTRL+BREAK可终止程序**

**注意事项：**

视频监测与实时监测功能需要提供监测模型，可以上传提供的预训练模型 `yolov8s.pt`，其位置与本word同目录。