**《吊顶检测系统》**

**环**

**境**

**配**

**置**

手

**册**

(仅供内部使用)

吊顶检测环境配置

# 目录

第1章 系统使用要求 [3](#_Toc145663958)

[1.1 软件要求 3](#_Toc145663959)

[1.2 硬件要求 3](#_Toc145663959)

第2章 环境搭建 [3](#_Toc145663958)

[2.1 下载、安装Anaconda 3](#_Toc145663959)

2.1.1 检验是否安装成功 [1](#_Toc145663958)1

[2.2 Anaconda创建、激活、删除等方法 1](#_Toc145663959)2

[2.2.1 查看当前存在的虚拟环境 1](#_Toc145663959)2

[2.2.2 创建虚拟环境 1](#_Toc145663959)2

[2.2.3 进入当前存在的虚拟环境 1](#_Toc145663959)5

[2.2.4 退出当前虚拟环境 1](#_Toc145663959)6

[2.2.5 删除虚拟环境 1](#_Toc145663959)6

[2.3 问题 1](#_Toc145663959)6

第3章 安装程序所需要的内容 [1](#_Toc145663958)7

[3.1 相机参数配置软件 1](#_Toc145663959)7

[3.1.1 下载安装程序 1](#_Toc145663959)7

[3.1.2 安装程序 1](#_Toc145663959)9

[3.1.3 软件连接相机 2](#_Toc145663959)8

[3.2 程序运行依赖安装 3](#_Toc145663959)0

[3.2.1 下载程序代码到本地 3](#_Toc145663959)0

[3.2.2 进入虚拟环境 3](#_Toc145663959)2

[3.2.3 安装依赖 3](#_Toc145663959)3

第4章 运行程序 [3](#_Toc145663958)7

第5章 调试代码 [4](#_Toc145663958)0

[5.1 下载pycharm 4](#_Toc145663959)0

[5.2 安装pycharm 4](#_Toc145663959)1

[5.3 破解pycharm 4](#_Toc145663959)4

[5.4 安装汉化插件 4](#_Toc145663959)9

[5.5 为pycharm添加解释器 5](#_Toc145663959)1

[5.6 运行、调试代码 5](#_Toc145663959)6

# **系统使用要求**

## **软件要求**

操作系统：Windows7、Windows10等。

编程语言：Python 3.6 或更高版本

## **硬件要求**

CPU：Intel i7及以上级别。

内存：16G以上。

硬件需求：

具备至少一个可用串口的设备，支持 MechEye 深度相机

其他为常规硬件配置，对显卡、声卡无特别要求。

# 环境搭建

**运行程序需要使用python作为开发语言，**

**推荐使用3.8、3.9版本python，此版本适配的依赖比较完善。**

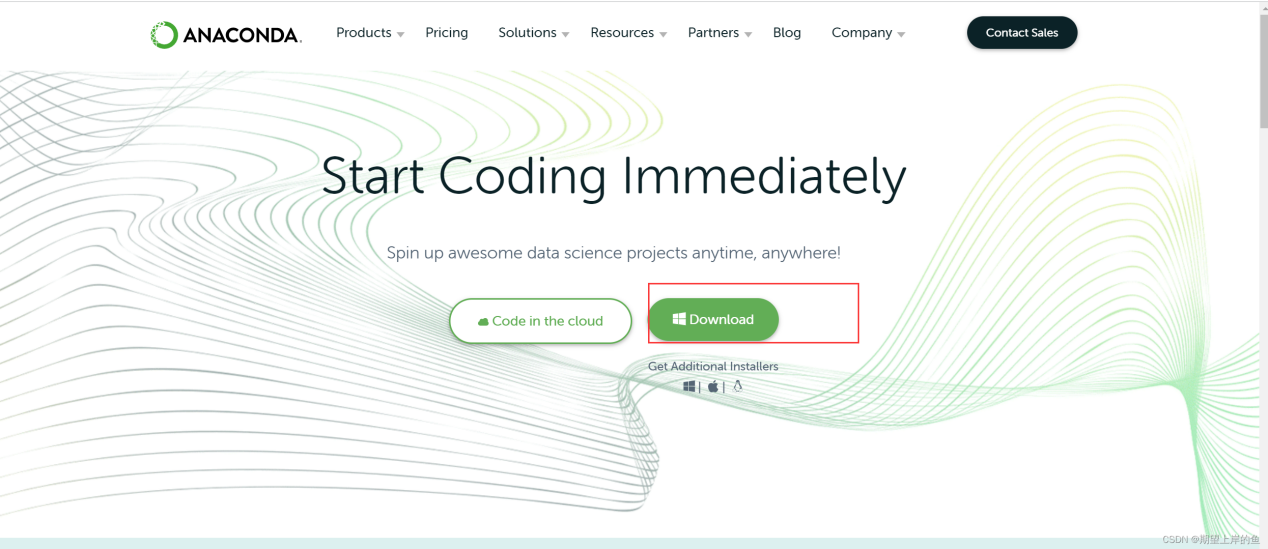
## 下载、安装anaconda3

官网下载

[Anaconda | The World’s Most Popular Data Science Platform]

<https://www.anaconda.com/>

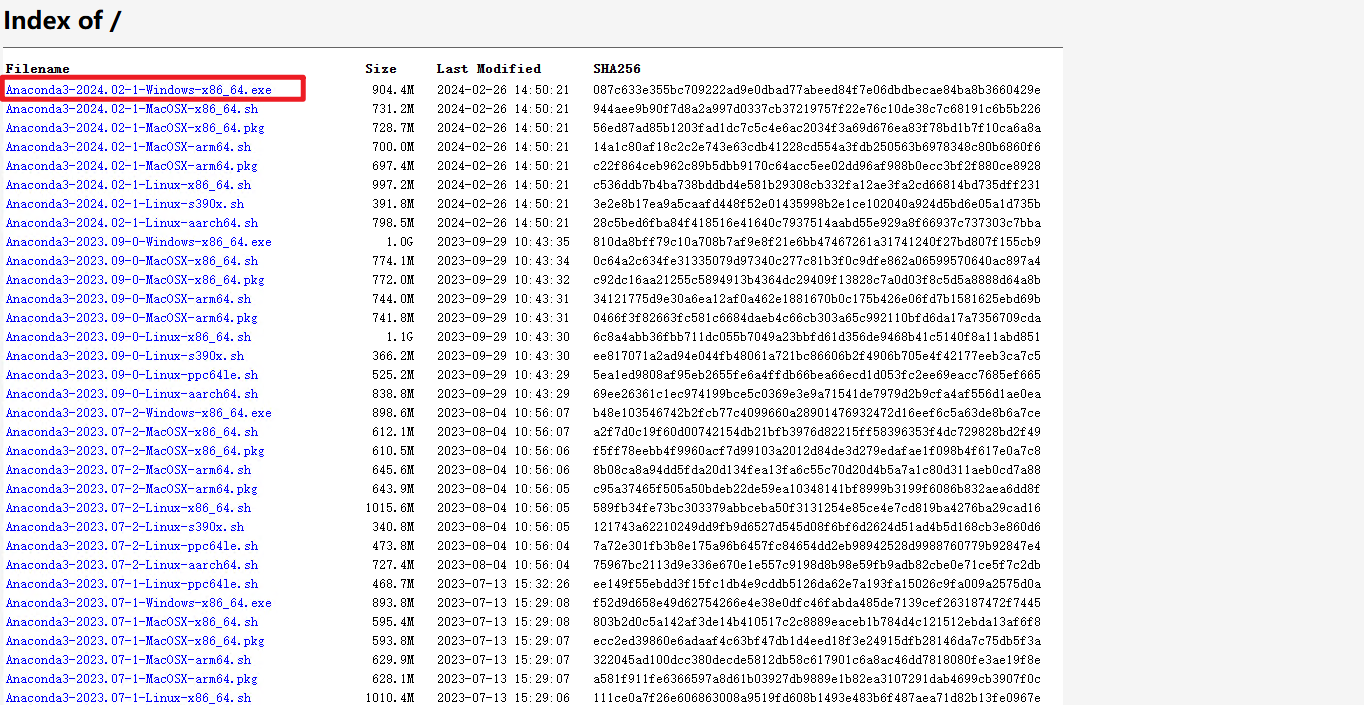
进去是这样的，直接点击"Download"即可。（这里我选择的是Windows环境且是64位）



清华镜像源下载

<https://repo.anaconda.com/archive/>

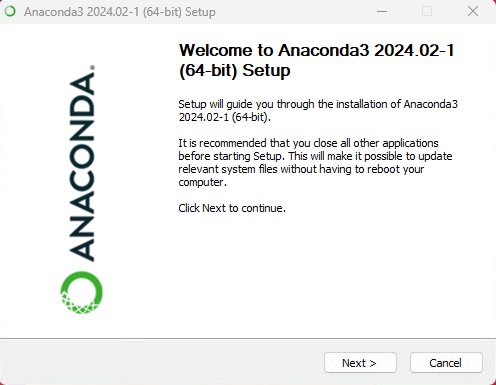
进去是这样的，直接点击想要下载的版本名称即可。（这里我选择的是Windows环境且是64位）



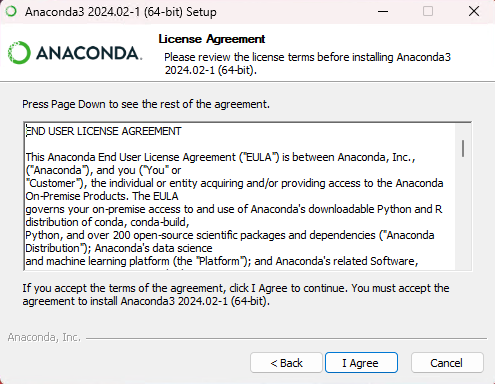
下载完成后双击“Anaconda3-2024.02-1-Windows-x86\_64.exe”运行安装程序

QOARX3IYAAQES

1. 点击“Next”按键

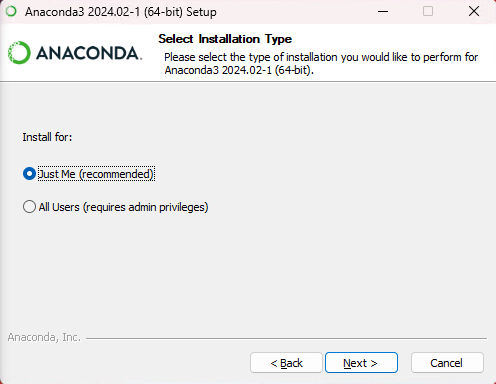


1. 点击“I Agree”按键

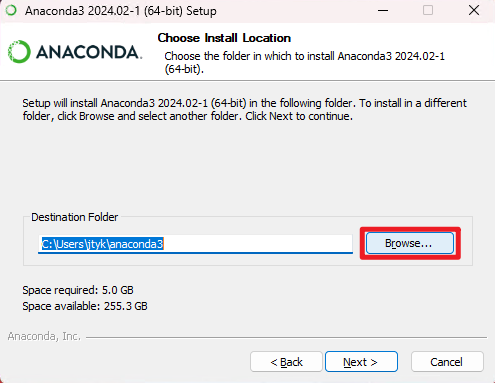


3、选择为谁安装：Just Me：只给当前用户安装，All Users：此电脑所有用户都可使用

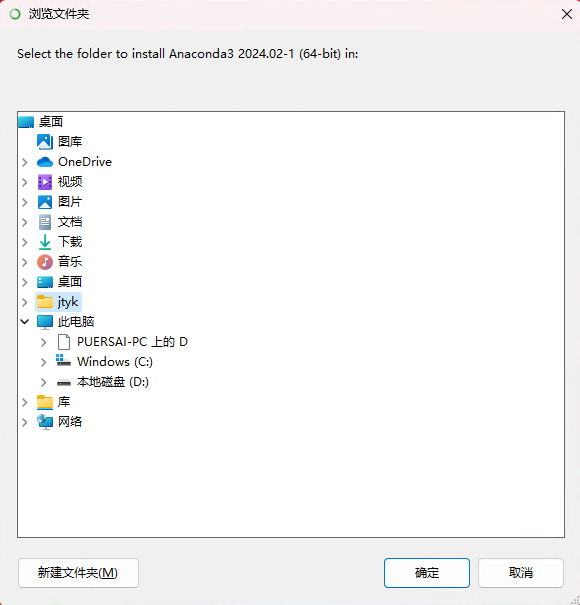
这里选择“Just Me”就可以，随后点击“Next”进行下一步



1. 在此步可改文件安装路径

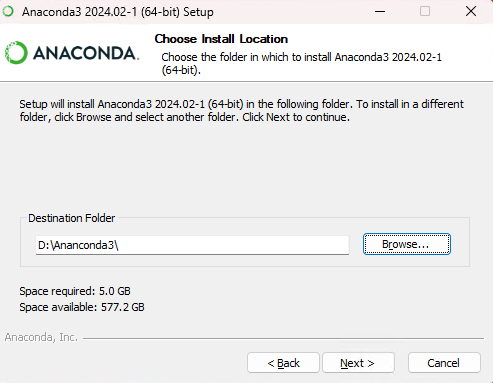


弹出选择框，选择自己需要的安装路径



这里我把路径修改成了“D:\Ananconda3\”，请根据自身情况设置

|  |
| --- |
| **注意：请记住软件安装的路径，如果出现问题，会用到此路径** |



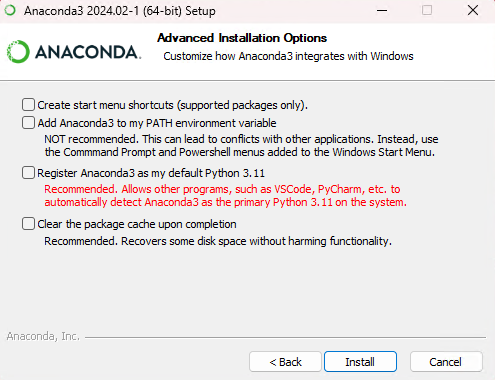
下图内容依次往下的内容为

🞎 创建开始菜单快捷方式

🞎 将Anaconda3添加到PATH环境变量中

🞎 注册Anaconda3作为我的默认Python 3.11

🞎 完成后清除包缓存



这里我选择了如下三个：

🞎 创建开始菜单快捷方式

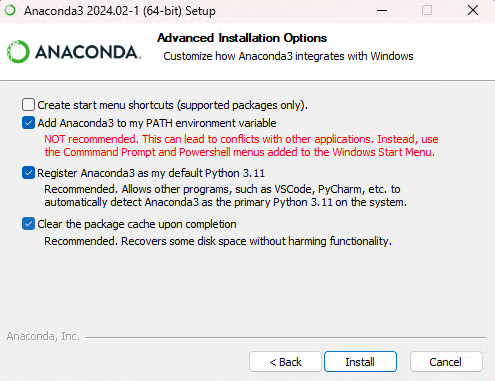
🗹 将Anaconda3添加到PATH环境变量中

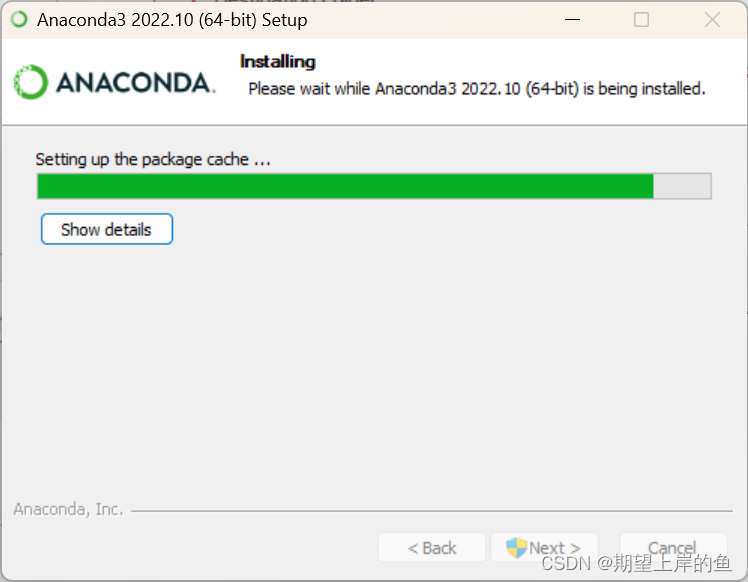
选择添加到系统变量的选项，可以直接在控制台用conda命令

🗹 注册Anaconda3作为我的默认Python 3.11

🗹 完成后清除包缓存

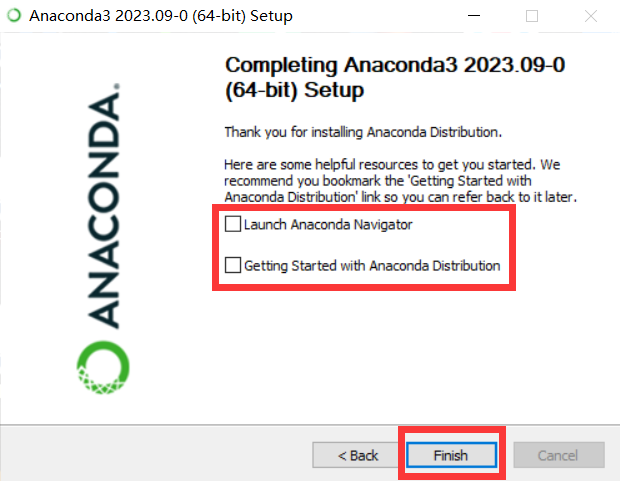
随后点击“Install”进行安装





最后取消勾选这两项，这个是推送广告的

点击“Finish”完成安装

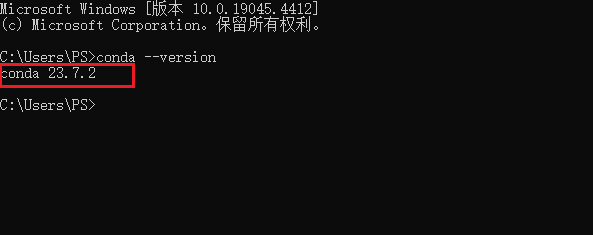


### 检验安装是否成功

同时按 win + r ，输入cmd，在弹出的命令行查看anaconda版本，输入 ：

|  |
| --- |
| conda --version |

出现下面内容为安装成功



## **anaconda3创建、激活、删除等方法**

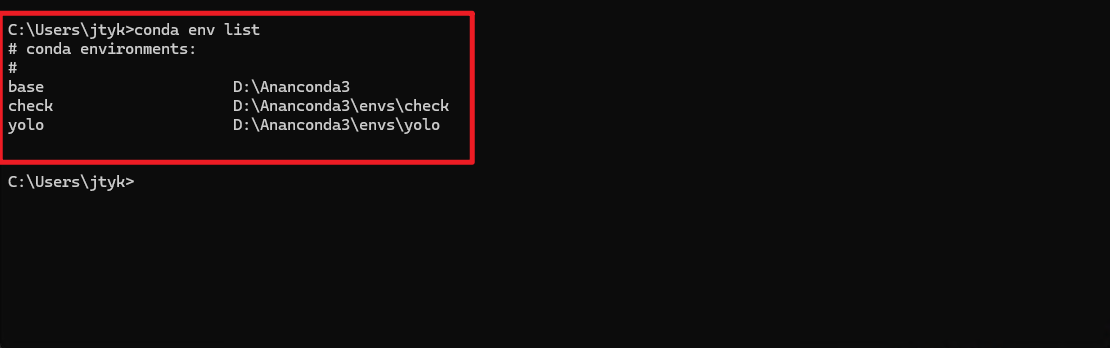
#### **查看当前存在的虚拟环境**

命令：

conda env list

此处我已经创建了虚拟环境，

第一次安装时只有base环境



#### **创建虚拟环境**

命令：

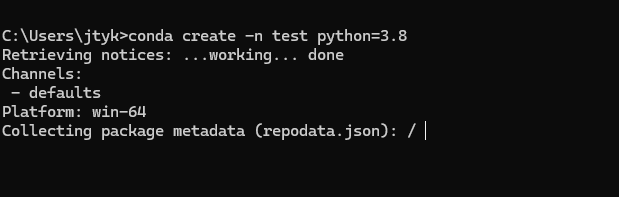
conda create -n 环境名 python=X.X

这里使用的命令是conda create -n test python=3.8，

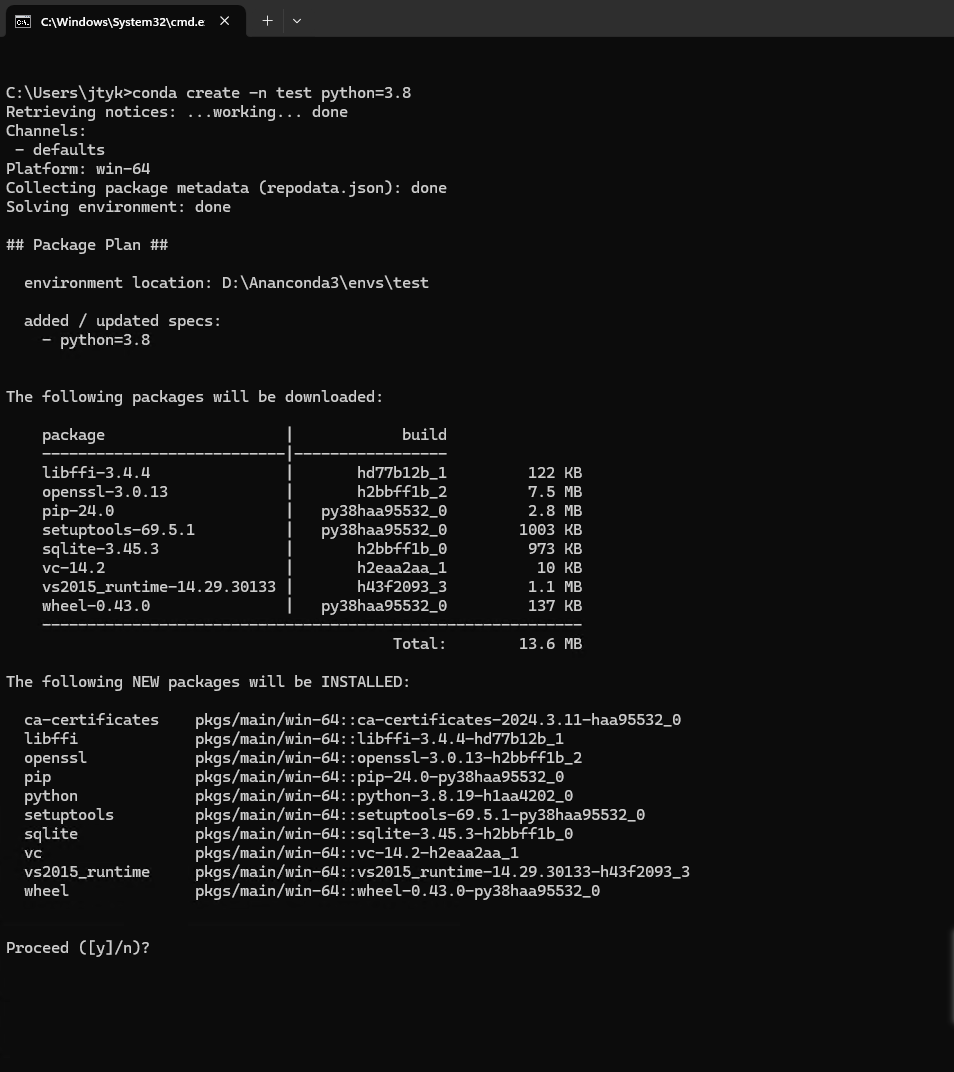
这里test是创建的环境名称，根据自身喜好设置

python版本推荐3.8版本，比较稳定

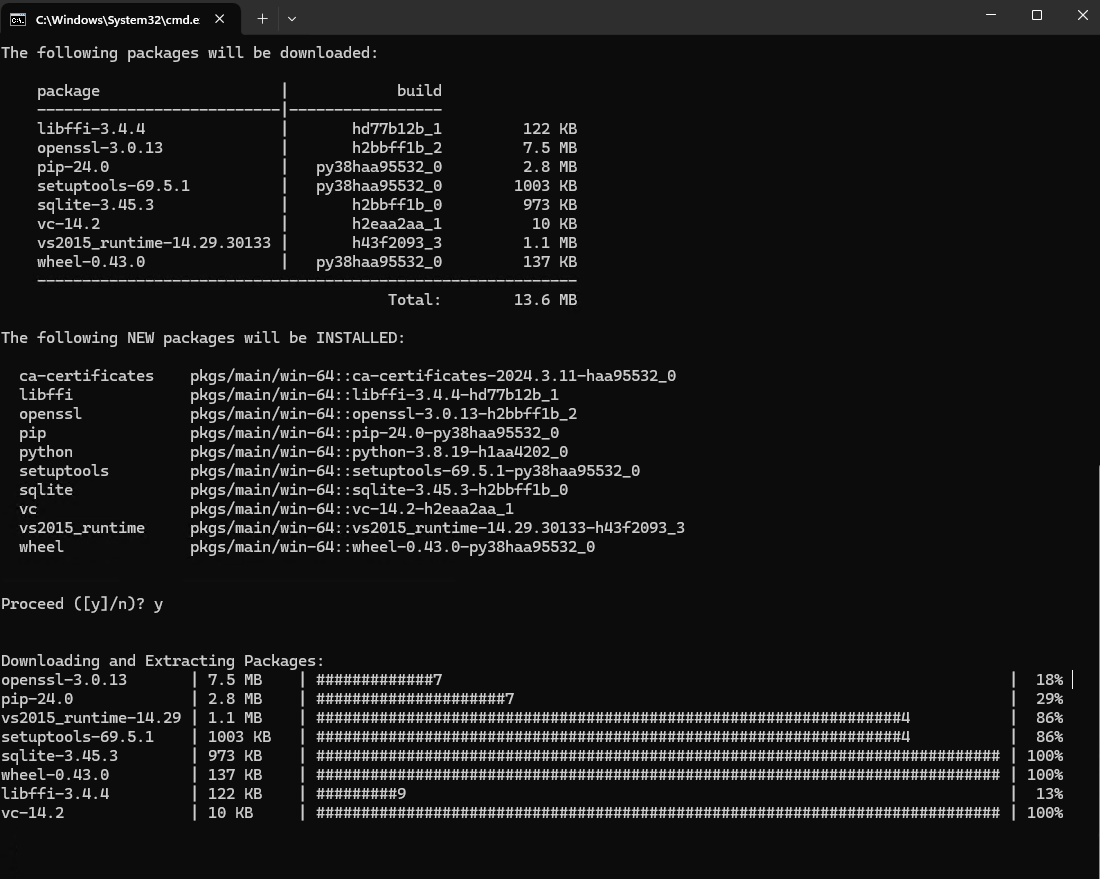
正在加载中



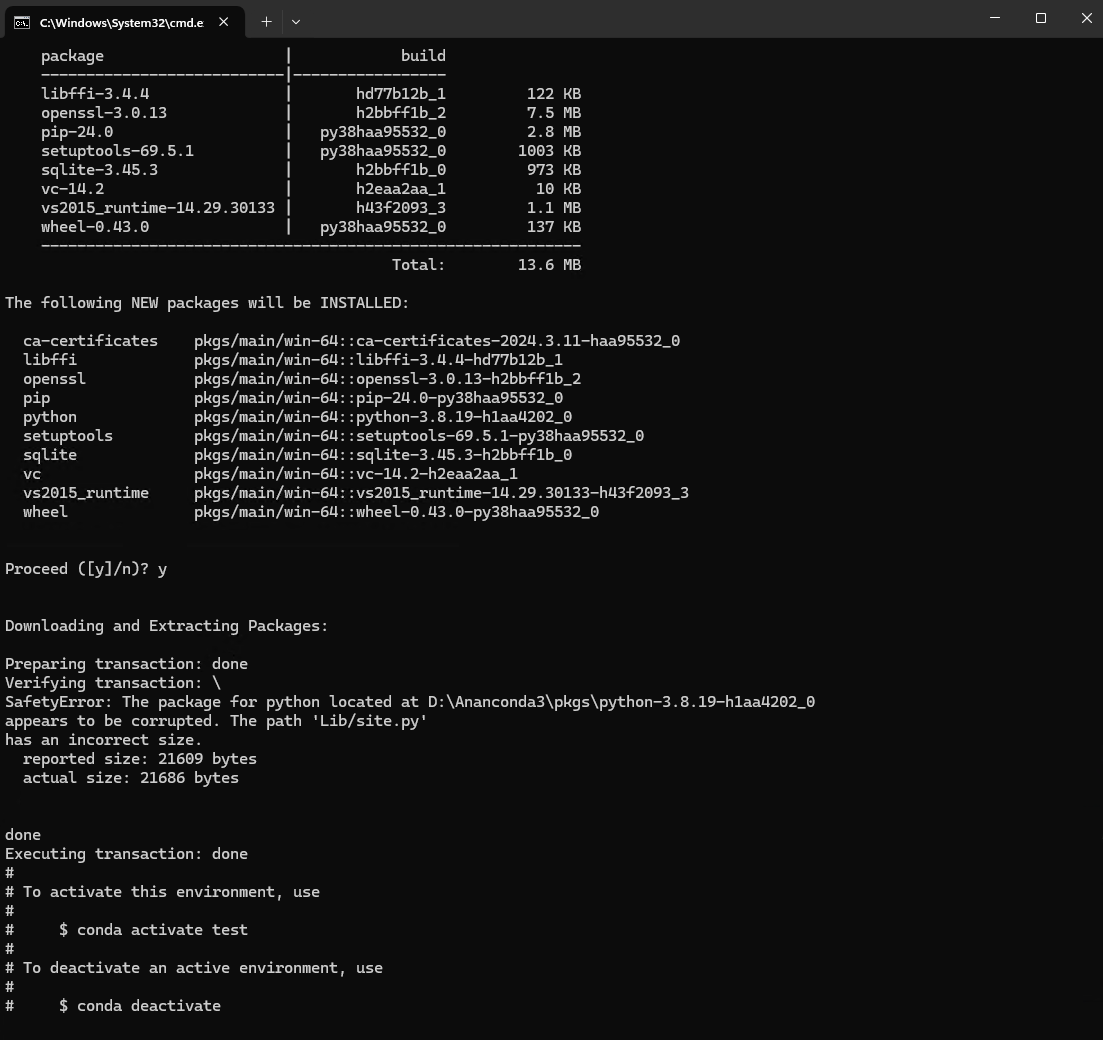
显示需要安装的内容，这里我们键盘输入Y随后Enter确认



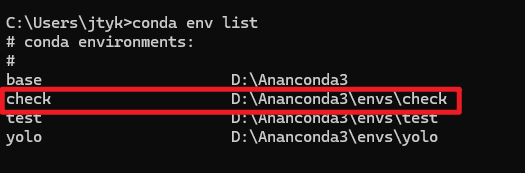
正在下载



创建虚拟环境完成



查看当前存在的虚拟环境，验证是否创建成功



#### **进入当前存在的虚拟环境**

命令：

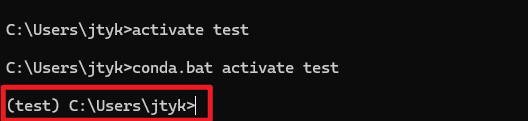
conda activate 环境名

或

activate 环境名

这两个命令根据anaconda的版本不同，命令会有所不同，一个不好用就使用另一个

前面出现（你的环境名称）的样式，说明成功进入虚拟环境



#### **退出当前虚拟环境**

命令：

conda deactivate 环境名

deactivate 环境名

#### **删除虚拟环境**

命令：

conda remove -n 环境名 --all

## **问题**

#### **conda不是内部或外部命令，也不是可运行的程序或批处理文件。**

此时conda没有添加到系统变量path中

计算机（右键）→属性→高级系统设置→（点击）环境变量

在**系统变量**里，找到并点击Path

输入下面的五个环境变量。需要Anaconda的安装路径以添加到环境变量中

你的Anaconda的安装路径\Anaconda3

你的Anaconda的安装路径\Anaconda3\Scripts

你的Anaconda的安装路径\Anaconda3\Library\bin

你的Anaconda的安装路径\Anaconda3\Library\mingw-w64\bin

你的Anaconda的安装路径\Anaconda3\Library\usr\bin

# 安装程序所需要的内容

## 相机参数配置软件

连接相机需要使用相机专用的软件，此软件还可以调整相机曝光时间、曝光度、激光强度等等

如果对相机感兴趣可以点击下方连接详细了解

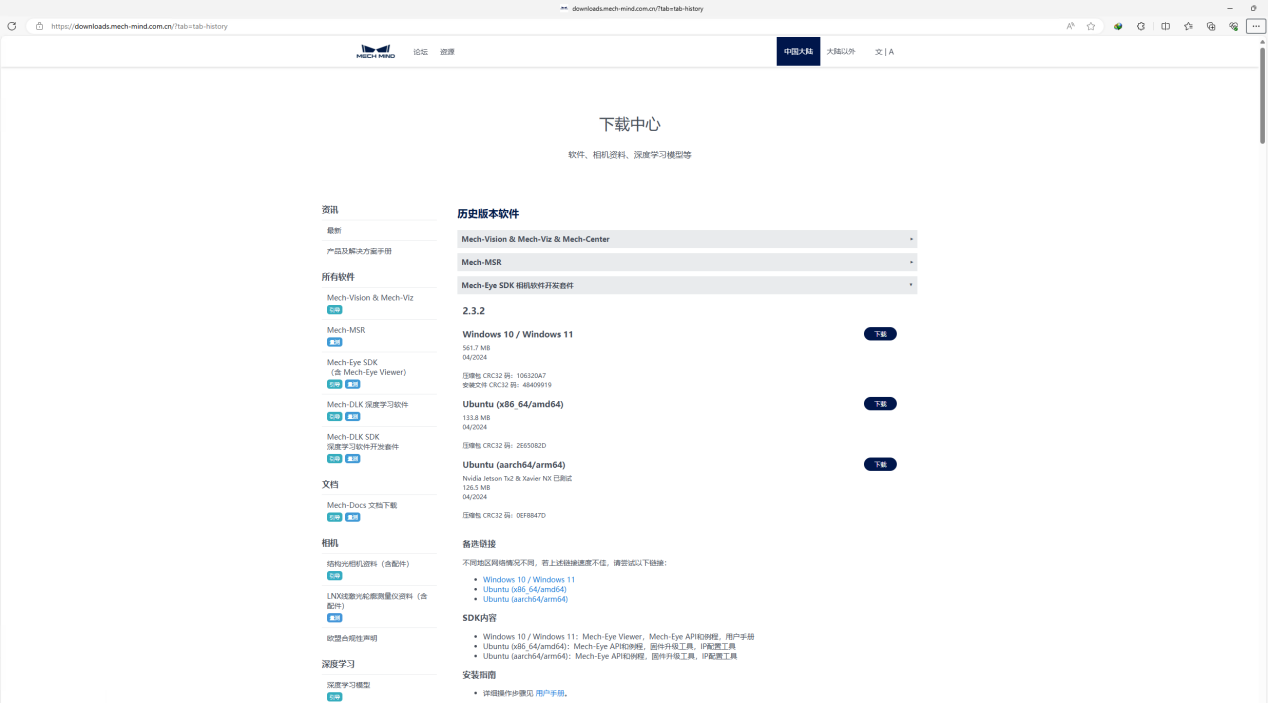
[欢迎使用Mech-Eye工业级3D相机！ (mech-mind.net)](https://docs.mech-mind.net/zh/eye-3d-camera/2.3.2/getting-started/index.html)

### 下载安装程序

相机软件下载地址：

[downloads.mech-mind.com.cn/?tab=tab-history](https://downloads.mech-mind.com.cn/?tab=tab-history)

打开连接，内容如图：



下滑找到2.1.0版本

如果找不到可以直接点击2.1.0版本下载连接：

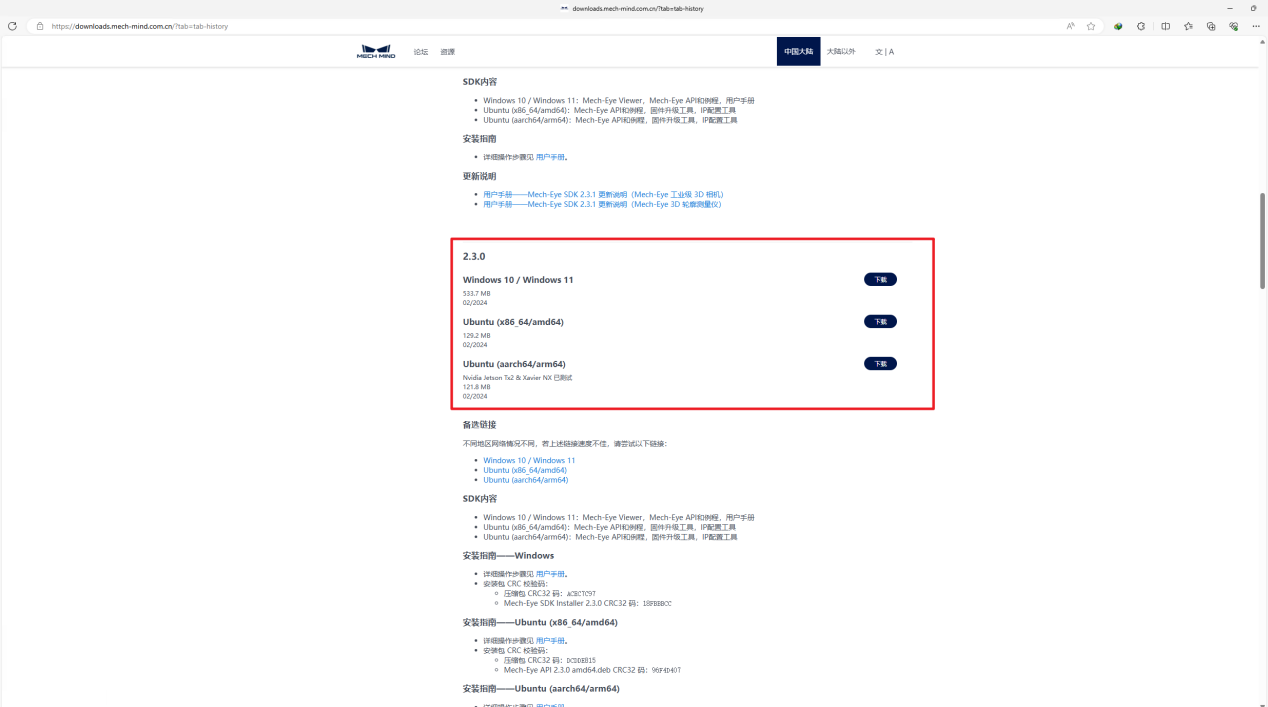
<https://mech-soft-1316409819.cos.ap-beijing.myqcloud.com/community_downloads/Mech-Eye%20SDK%202.1.0%20Renamed/Mech-Eye_SDK_Installer_2.1.0_Windows10%2611.zip?q-sign-algorithm=sha1&q-ak=AKID5U610udO8zzxkZPOFSaqTyIO3gBYkqrm&q-sign-time=1709624784%3B2573624844&q-key-time=1709624784%3B2573624844&q-header-list=host&q-url-param-list=&q-signature=86f7ee330b8dd148ee712b571d6bd1a987a6bb170>



接着下载2.3.0以上版本

如果找不到可以直接点击2.3.2版本下载连接：

<https://mech-soft-1316409819.cos.ap-beijing.myqcloud.com/community_downloads/Mech-Eye%20SDK%202.3.0/Mech-Eye%20SDK%20Installer%202.3.0.zip?q-sign-algorithm=sha1&q-ak=AKID5U610udO8zzxkZPOFSaqTyIO3gBYkqrm&q-sign-time=1709624784%3B2573624844&q-key-time=1709624784%3B2573624844&q-header-list=host&q-url-param-list=&q-signature=94d3c2f3b92bd43f3decfda613b3dd7051ae6c51>

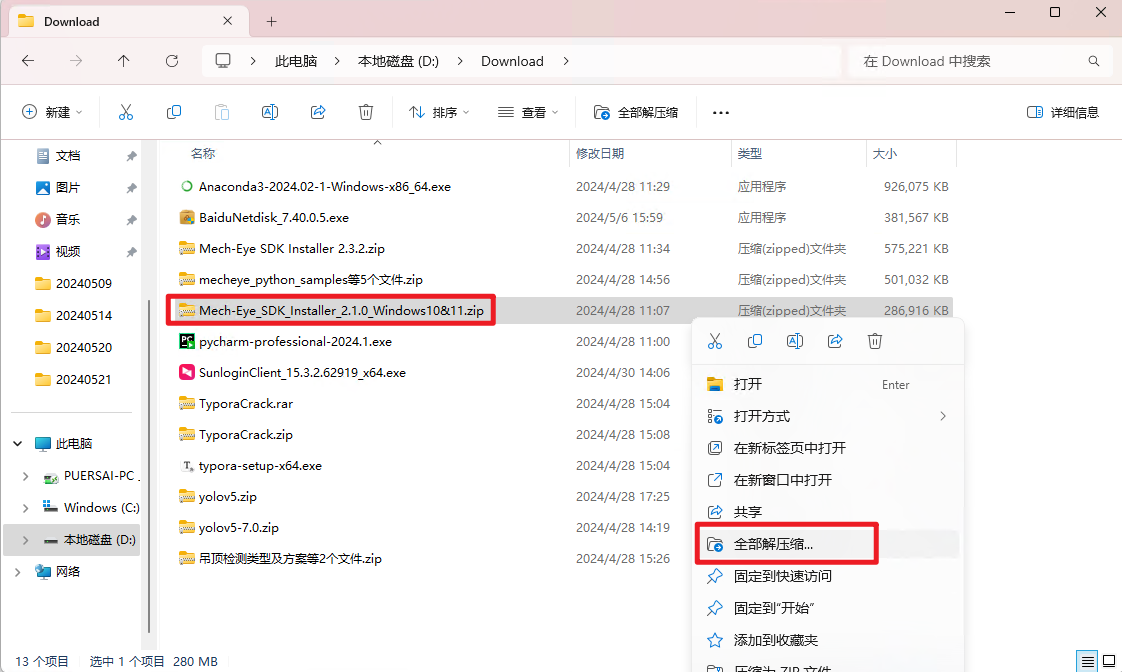


### 安装程序

**注意：在开始安装之前，可以预先创建一个文件夹用来设置程序安装路径 — Mech-Mind**

首先解压刚刚下载的2.1.0版本安装程序的压缩包：

Mech-Eye\_SDK\_Installer\_2.1.0\_Windows10&11.zip



双击2.1.0版本安装程序:



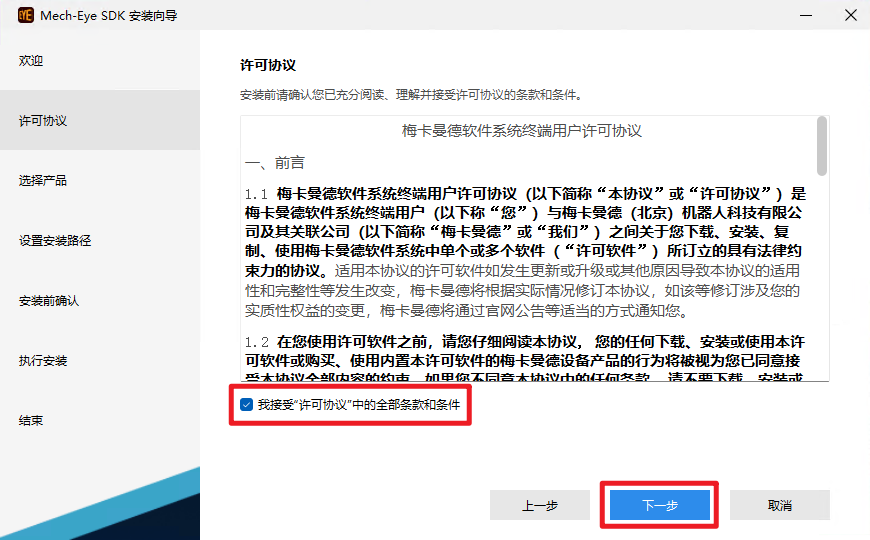
打开安装程序界面，如图：

点击“下一步”



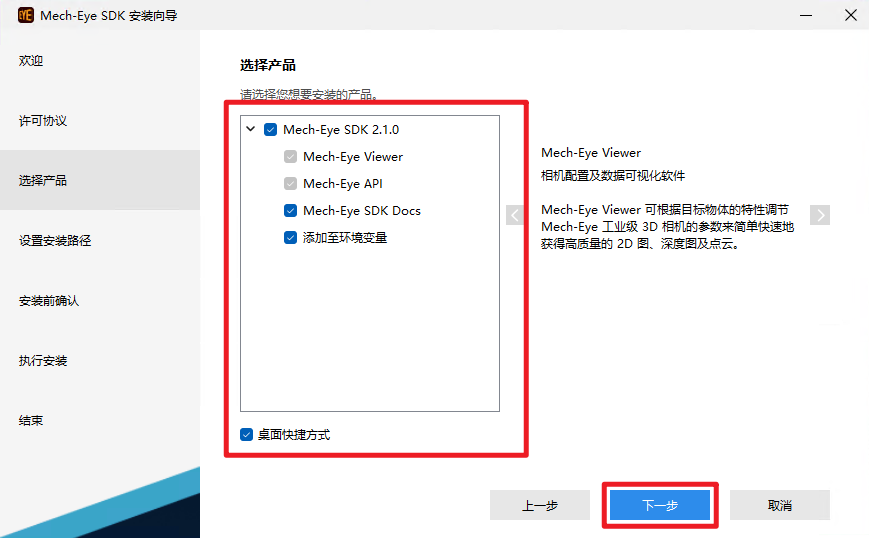
勾选“接受"许可协议"中的全部条款和条件”

点击“下一步”



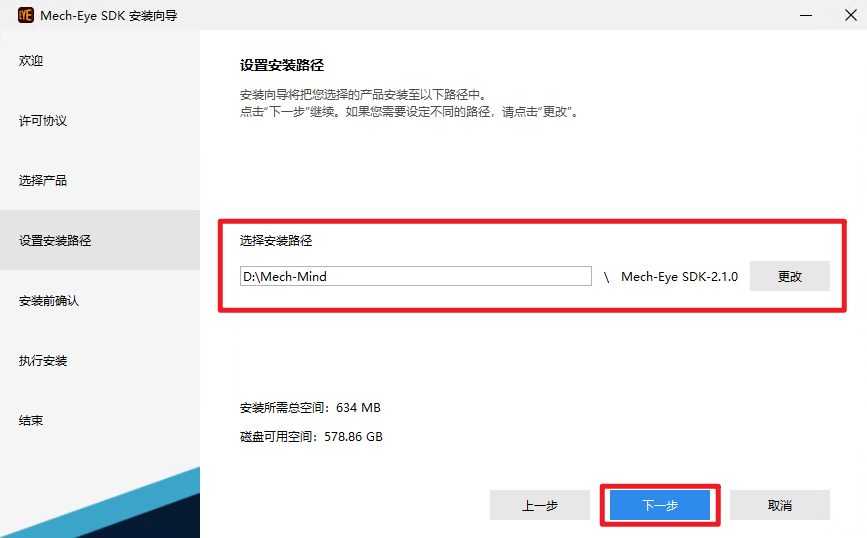
选择产品，默认勾选，不必操作

点击“下一步”



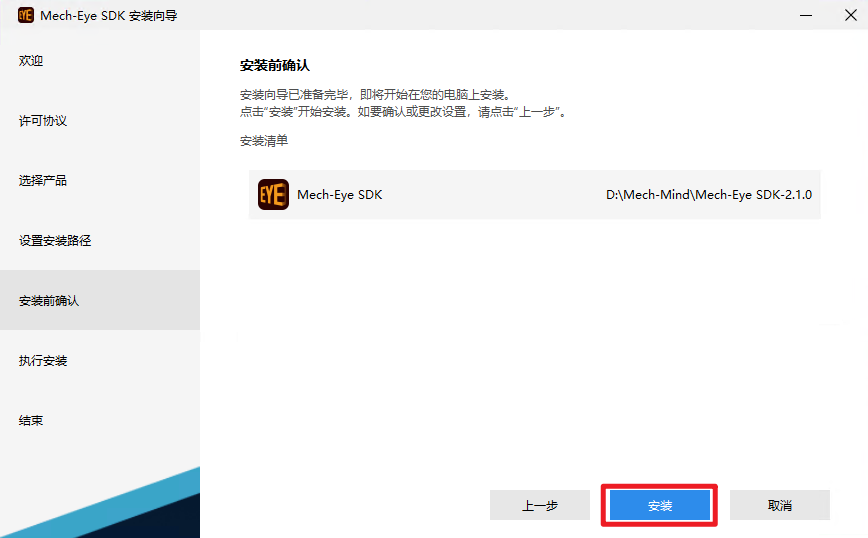
如果提前创建了Mech-Mind 文件夹，选择创建的文件夹路径进行安装

点击“下一步”

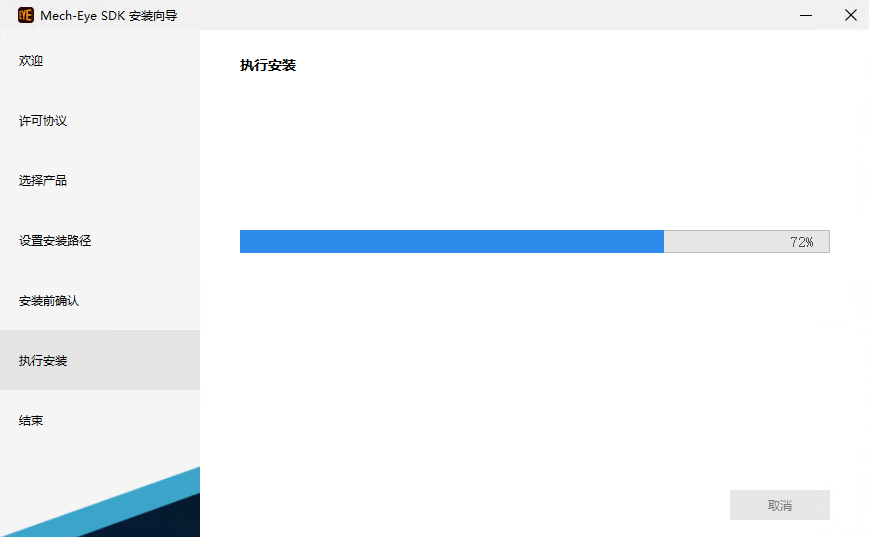


安装前确认，确认是2.1.0版本

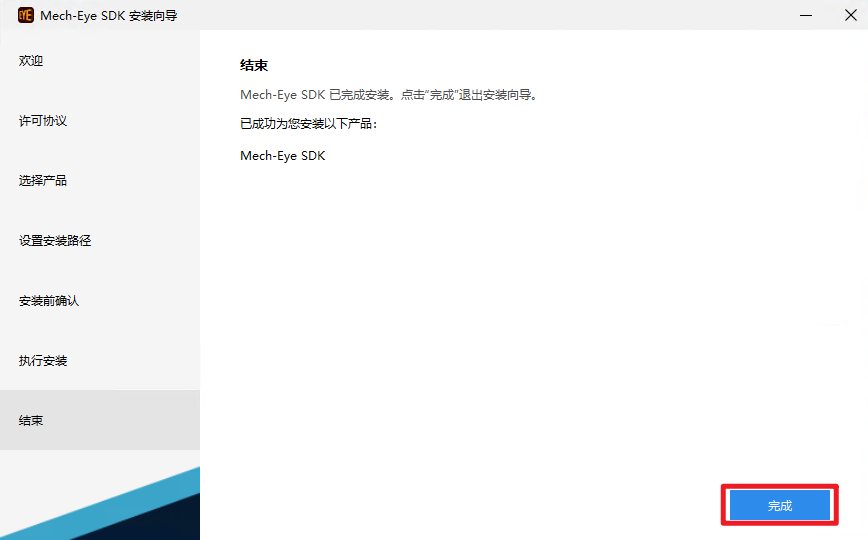
点击“安装”



正在安装.......



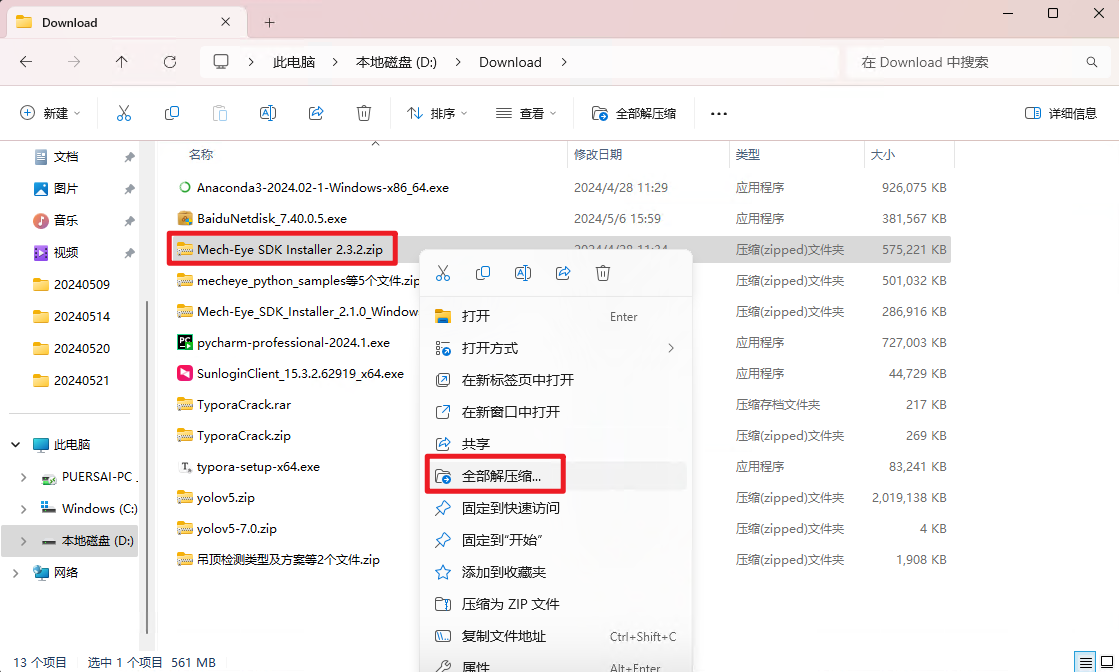
安装完成，点击“完成”退出安装程序



还需安装2.3.0以上版本相机程序软件：

解压刚刚下载的2.3.2版本安装程序的压缩包：

Mech-Eye SDK Installer 2.3.2.zip



双击2.3.2版本安装程序：

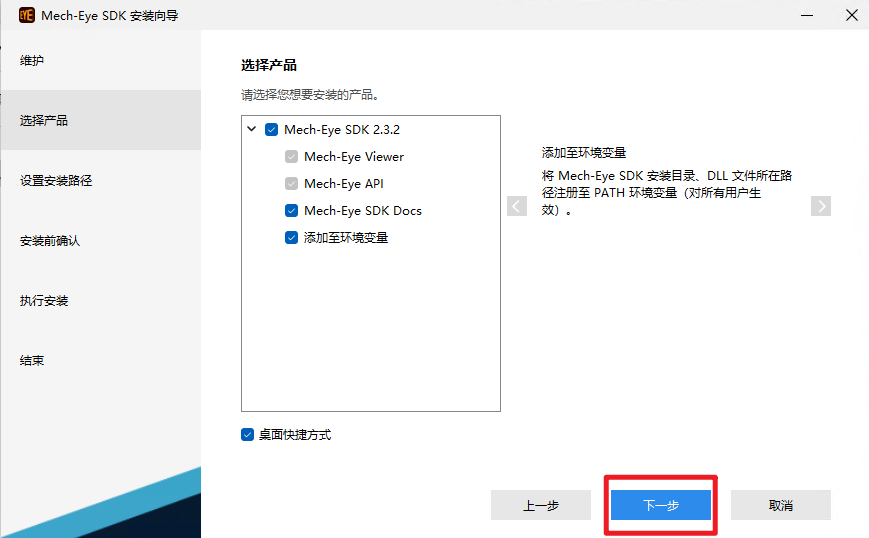


打开安装程序界面，如下图：

选择“升级，保留历史版本”选项，（保留2.1.0版本）

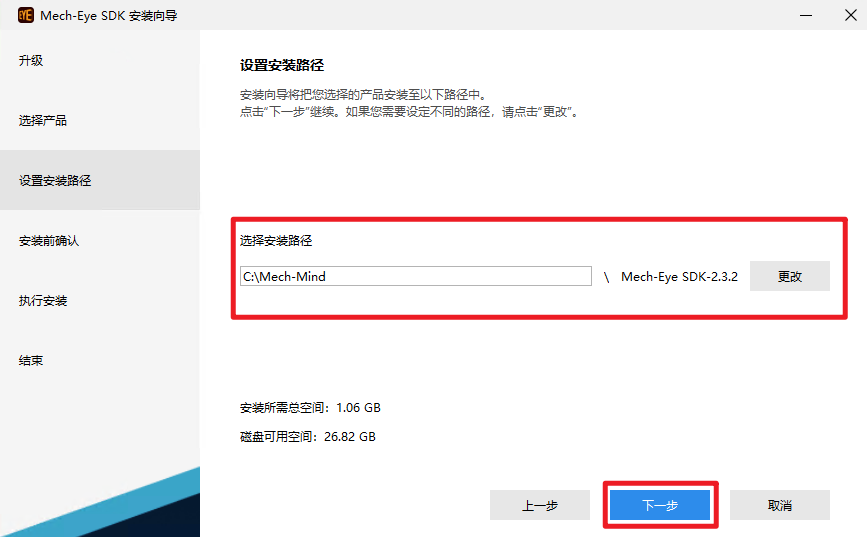


点击“下一步”



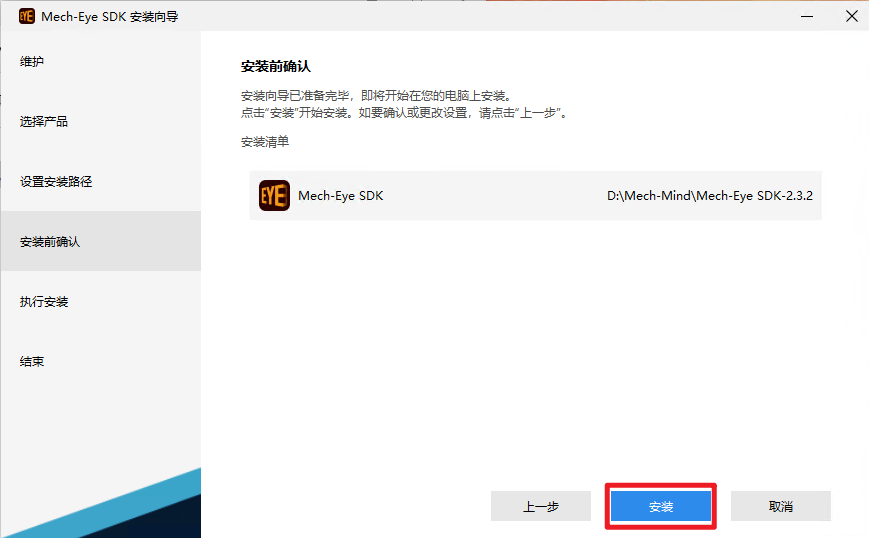
设置安装路径，可以选择和2.1.0版本放在同一文件夹内

点击“下一步”

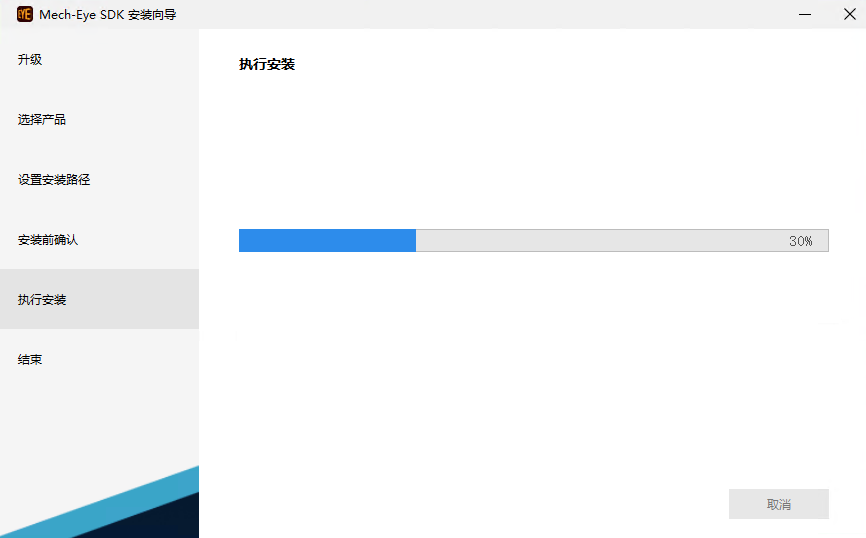


安装前确认，确认是2.3.0以上版本

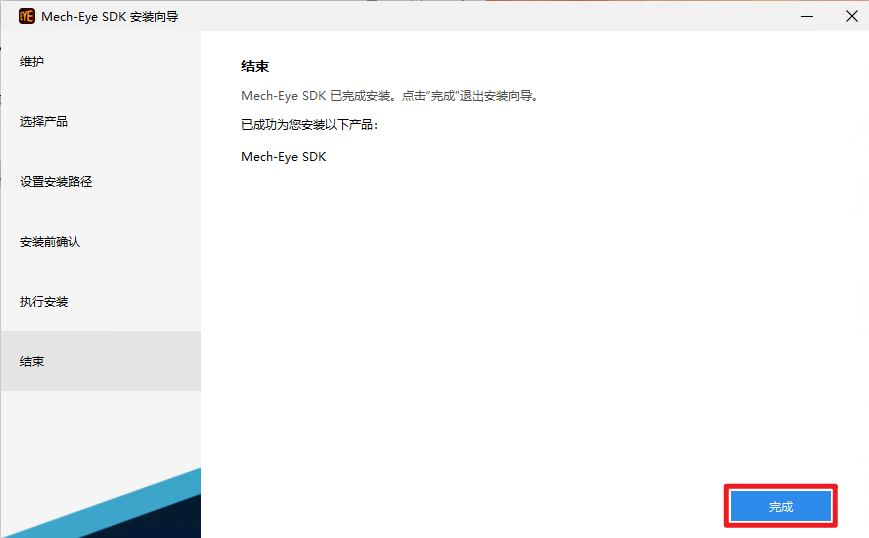
点击“安装”



正在安装.......



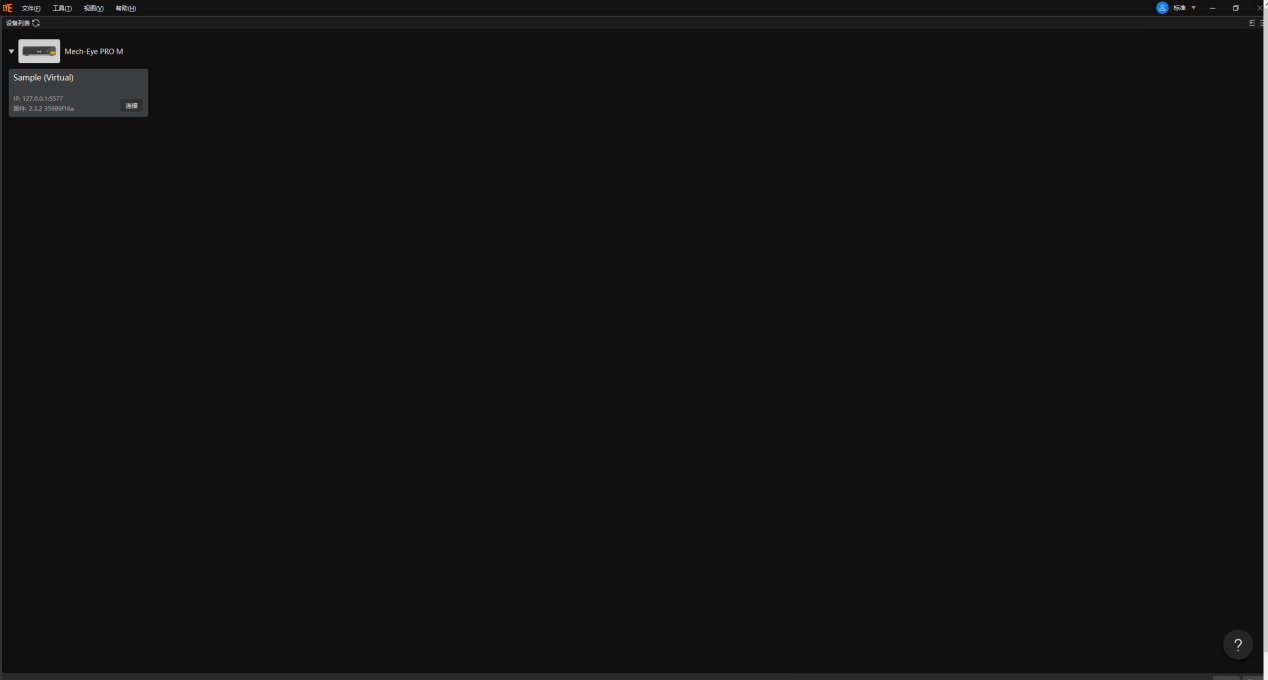
点击“完成”结束安装



双击应用程序，打开相机软件



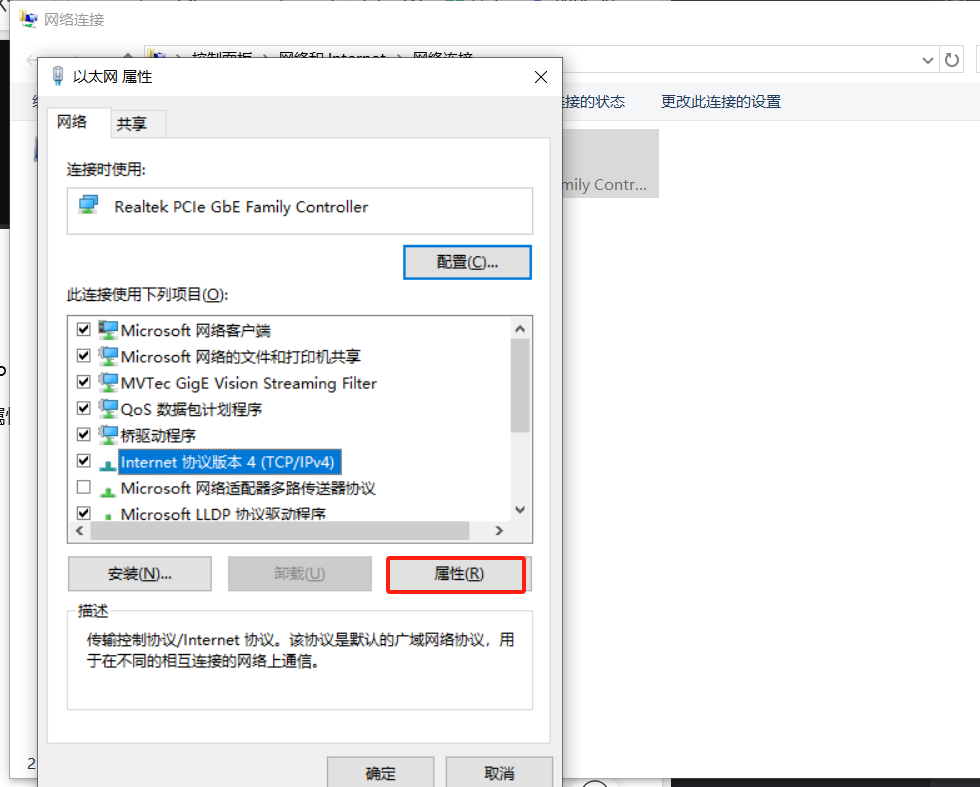
成功打开，如图：

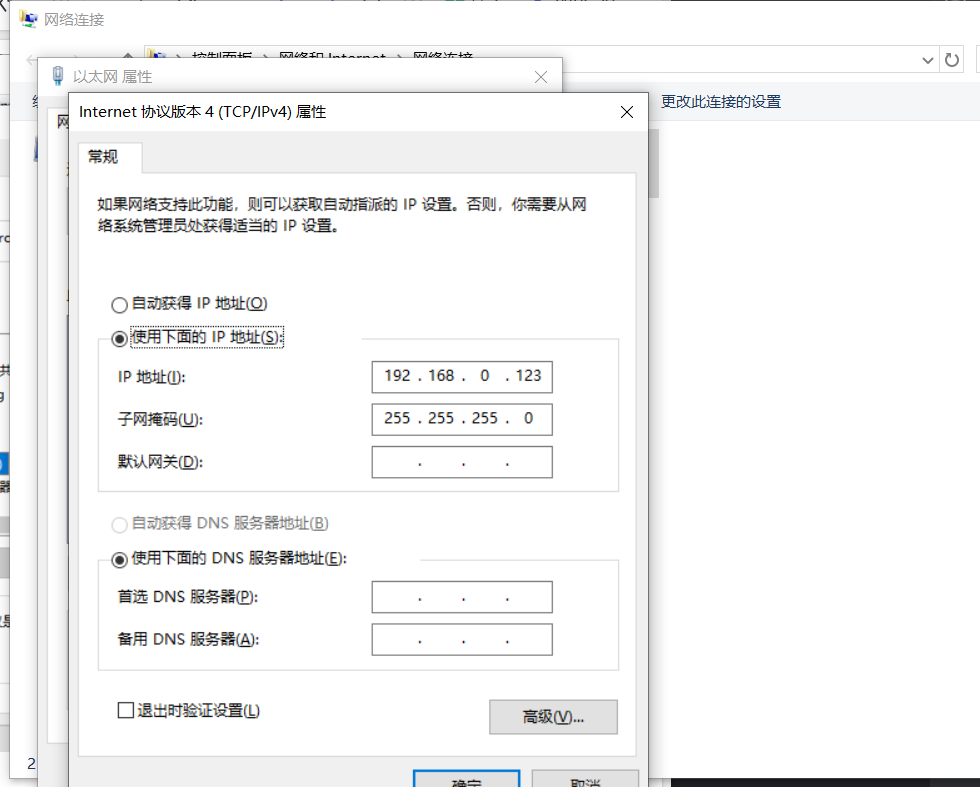


### 软件连接相机

用网线连接相机，需要配置相机IP地址

打开电脑网络适配器选项，选择属性，选择ipv4属性，IP地址需要和相机设置在同一网段下，子网掩码为255.255.255.0





## 程序运行依赖安装

### 下载程序代码到本地

XXXXXXXXXXXXXXXXXXXXXxxxxxxxxxxxxxxxxxxxXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXXXxxxxxxxxxxxxxxxxxxxXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXXXxxxxxxxxxxxxxxxxxxxXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXXXxxxxxxxxxxxxxxxxxxxXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXXXxxxxxxxxxxxxxxxxxxxXXXXXXXXXXXXXXXXXXXXX

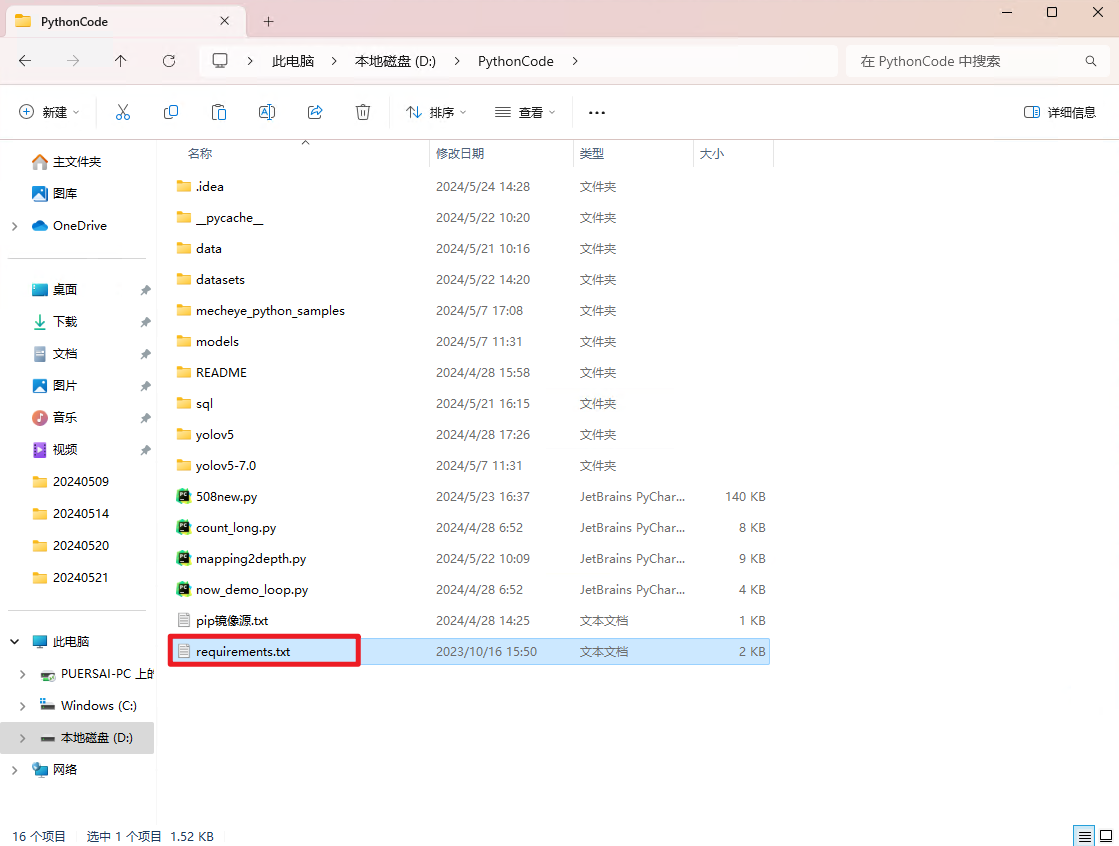
XXXXXXXXXXXXXXXXXXXXXxxxxxxxxxxxxxxxxxxxXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXXXxxxxxxxxxxxxxxxxxxxXXXXXXXXXXXXXXXXXXXXX

XXXXXXXXXXXXXXXXXXXXXxxxxxxxxxxxxxxxxxxxXXXXXXXXXXXXXXXXXXXXX

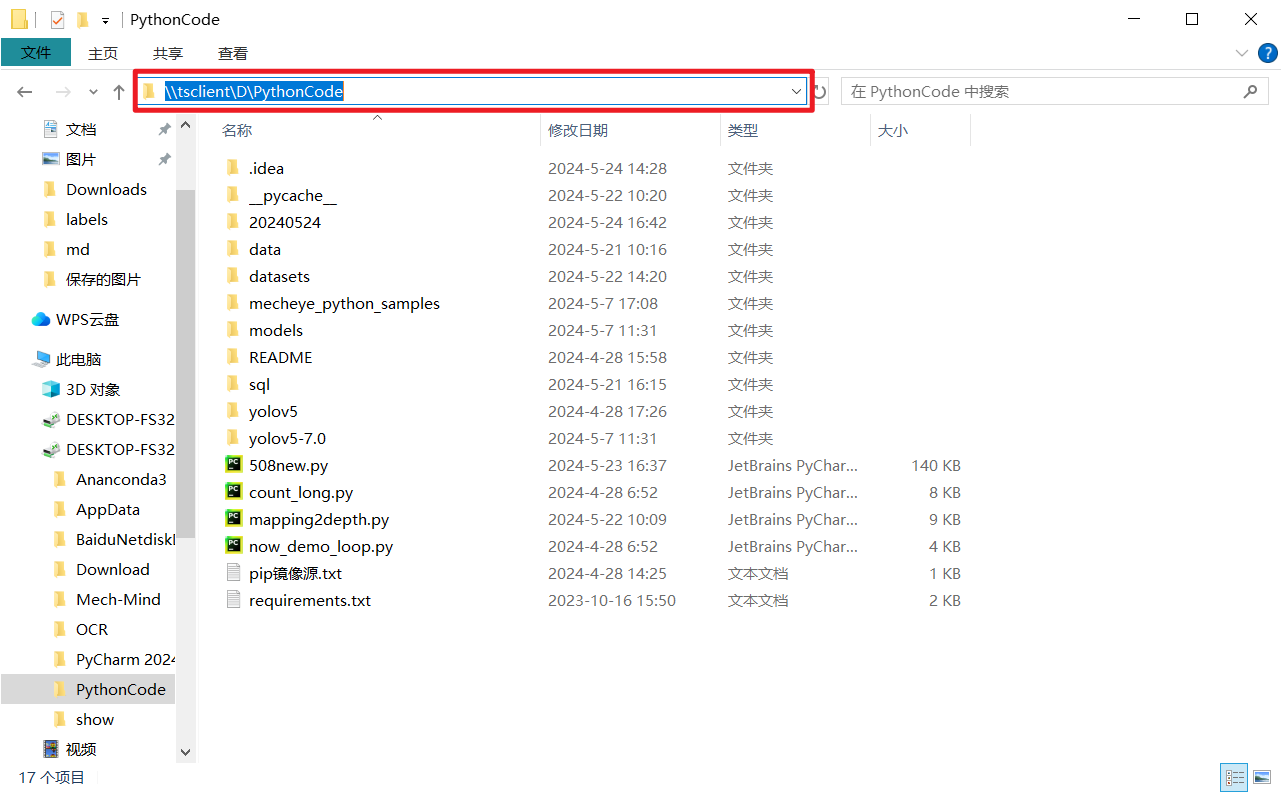
进入到程序代码文件夹内：

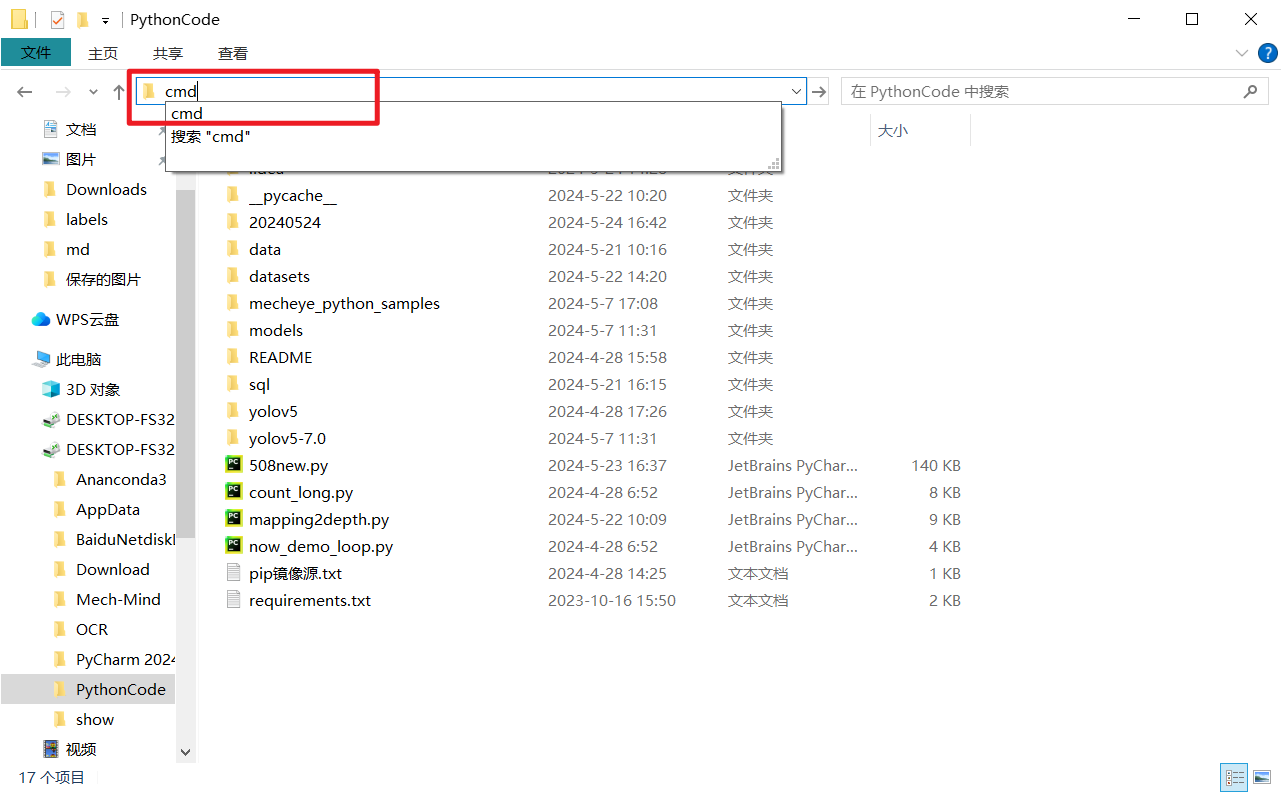
确认文件夹内存在“requirements.txt”



已经在requirements.txt内写好需要配置的依赖

在此处输入cmd，点击键盘“Enter”打开命令框，如图：





### 进入虚拟环境

如果已经按照上述步骤安装了Anaconda，进入虚拟环境进行依赖的安装：

在命令行内输入

conda activate 环境名

或

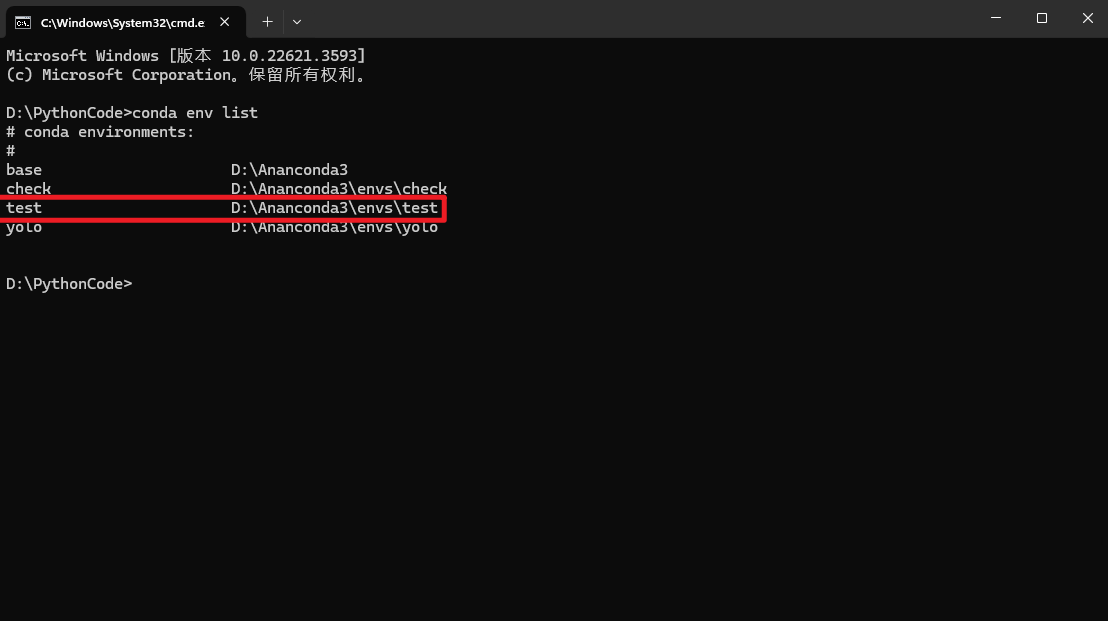
activate 环境名

进入虚拟环境

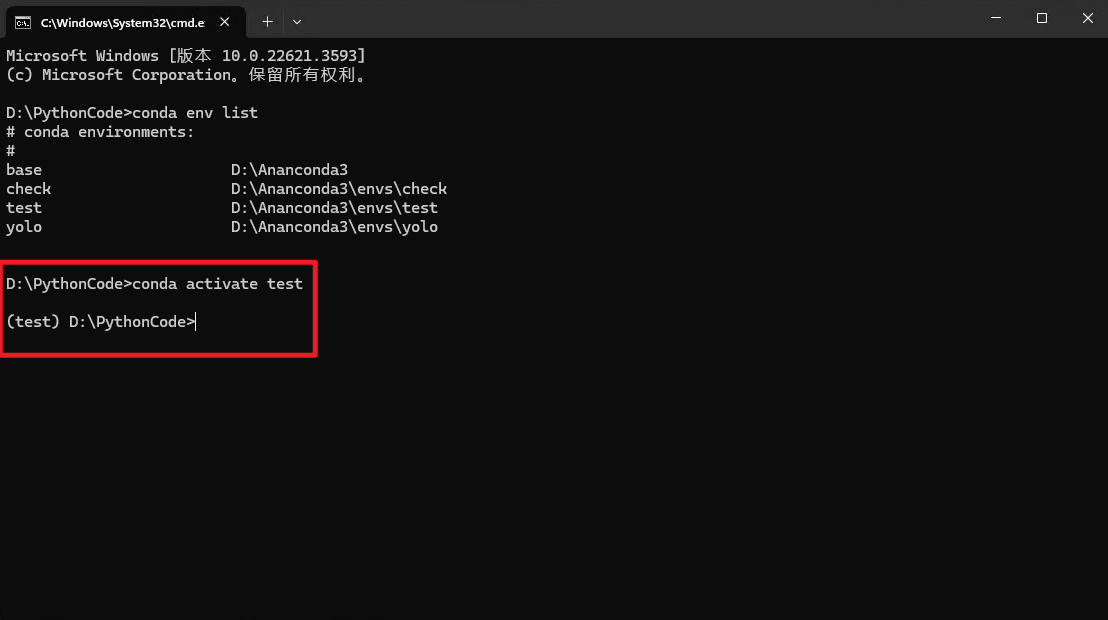
如果忘记了虚拟环境名称：

conda env list

找到刚刚创建的环境名称，如下图：



进入虚拟环境



### 安装依赖

输入：

pip install -r requirements.txt -i <https://pypi.tuna.tsinghua.edu.cn/simple>

-i 的意思是指定镜像源，因为有些下载会从国外下载会比较慢，这个源是国内的

1 清华大学（完全度和速度都很好，是一个优秀的pip镜像源）

<https://pypi.tuna.tsinghua.edu.cn/simple>

2 阿里云（完全度和速度也很好，是一个不错的选择）

<https://mirrors.aliyun.com/pypi/simple/>

3 网易（速度比较快，但是完全度有限）

<https://mirrors.163.com/pypi/simple/>

4 豆瓣（速度较快，但是完全度也有限）

<https://pypi.douban.com/simple/>

5 百度云（速度较快，但是完全度也有限）

<https://mirror.baidu.com/pypi/simple/>

6 中科大（速度较快，但完全度不如前面几个镜像源）

<https://pypi.mirrors.ustc.edu.cn/simple/>

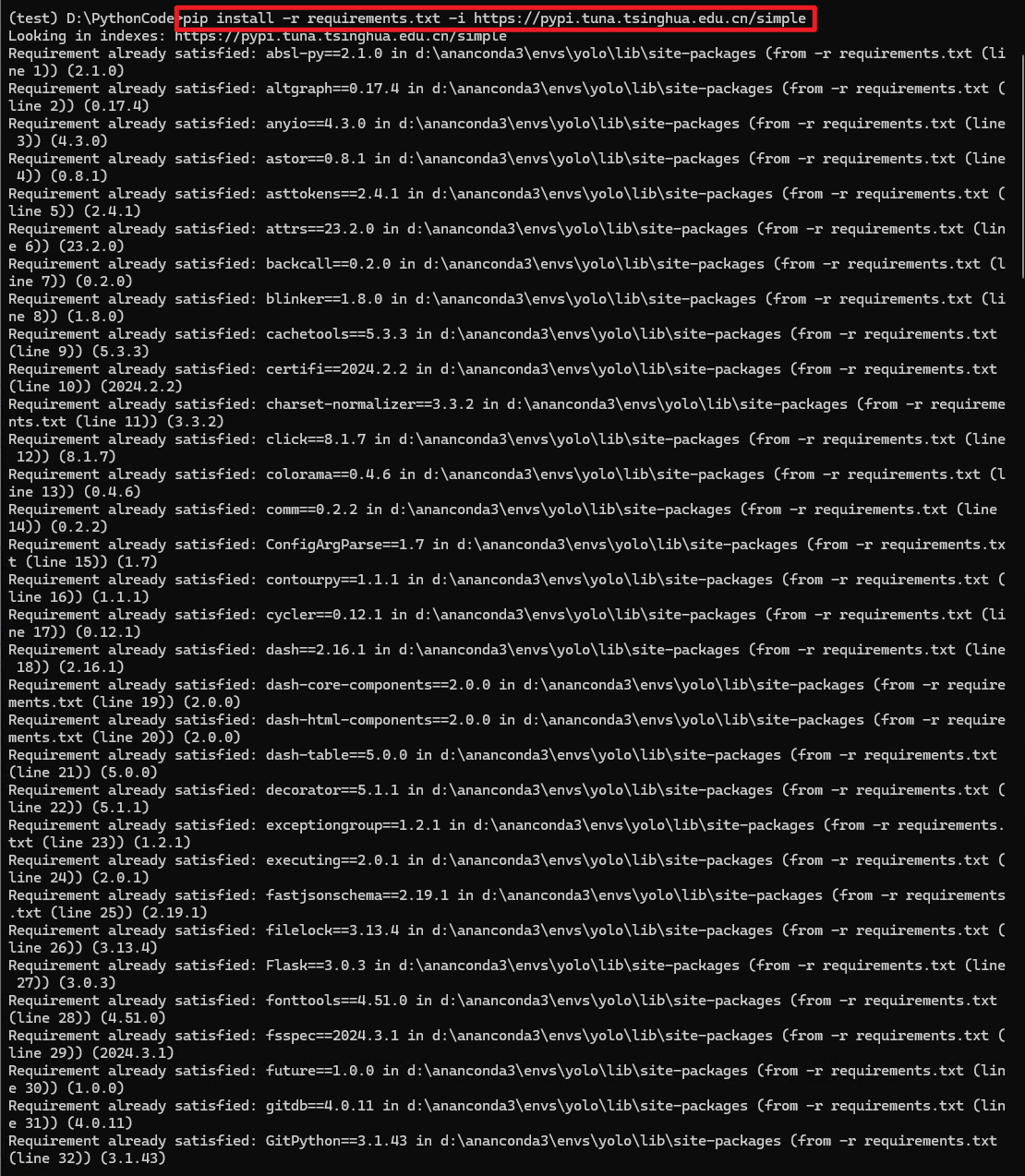
7 华为云（完全度和速度均中等）

<https://mirrors.huaweicloud.com/repository/pypi/simple/>

8 腾讯云（速度一般，完全度也一般）

<https://mirrors.cloud.tencent.com/pypi/simple/>

正在安装界面：



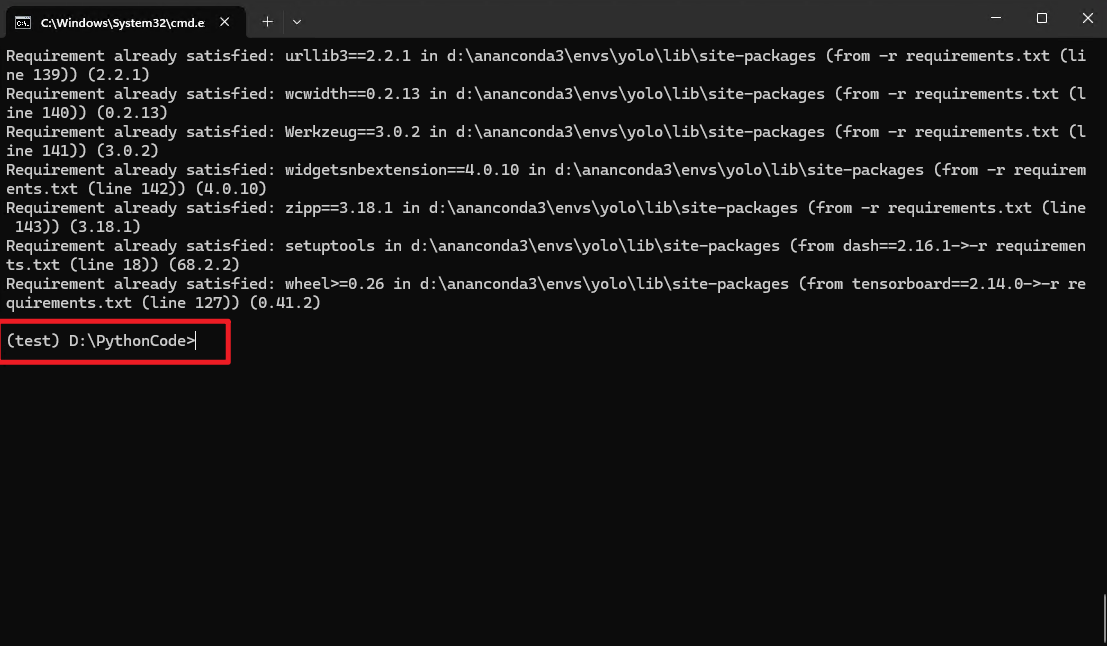
安装完成后就可以运行程序了

需要安装的这些依赖有：

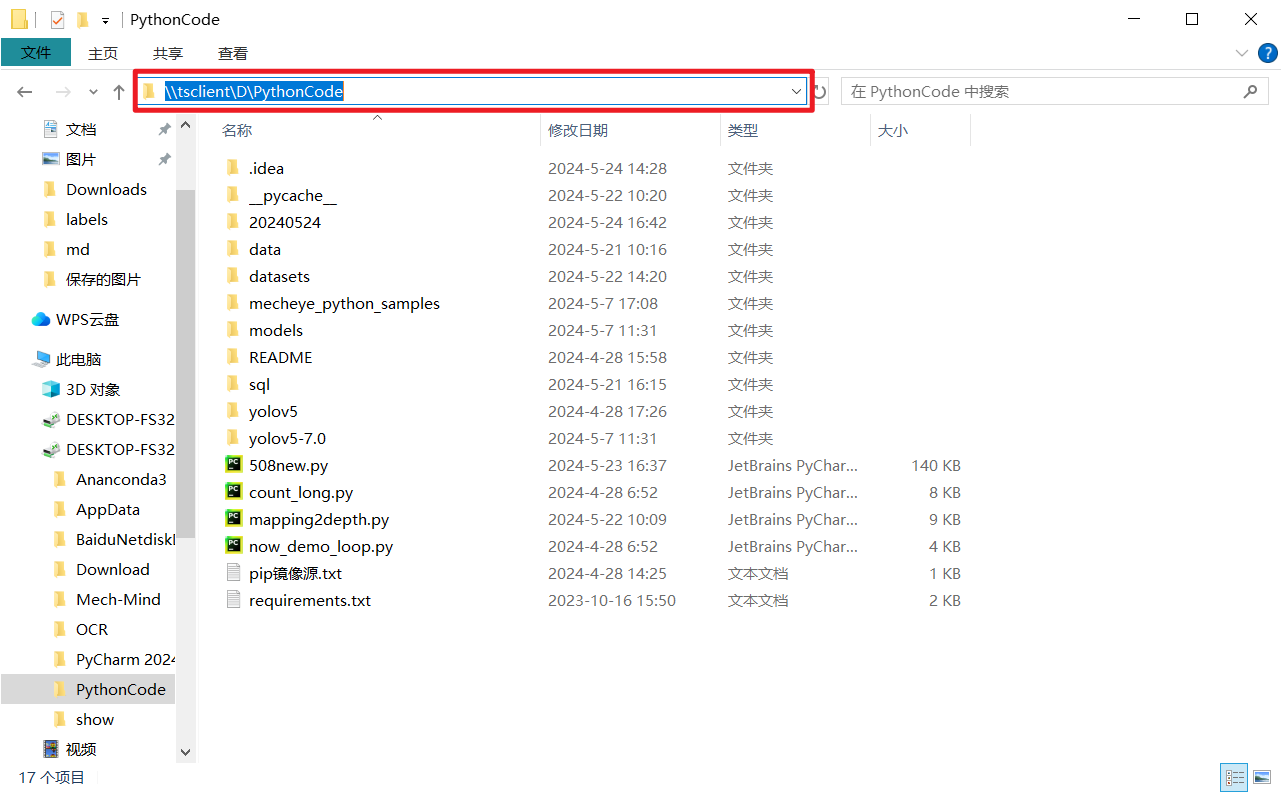
|  |  |  |  |
| --- | --- | --- | --- |
| absl-py==2.1.0 | packaging==24.0 | h11==0.14.0 | PyYAML==6.0.1 |
| altgraph==0.17.4 | paddlepaddle==2.6.1 | httpcore==1.0.5 | rapidfuzz==3.9.0 |
| anyio==4.3.0 | pandas==2.0.3 | httpx==0.27.0 | rarfile==4.2 |
| astor==0.8.1 | parso==0.8.4 | idna==3.7 | referencing==0.35.0 |
| asttokens==2.4.1 | pefile==2023.2.7 | importlib\_metadata==7.1.0 | requests==2.31.0 |
| attrs==23.2.0 | pickleshare==0.7.5 | importlib\_resources==6.4.0 | requests-oauthlib==2.0.0 |
| backcall==0.2.0 | pillow==10.3.0 | intel-openmp==2021.4.0 | retrying==1.3.4 |
| blinker==1.8.0 | pkgutil\_resolve\_name==1.3.10 | ipython==8.12.3 | rpds-py==0.18.0 |
| cachetools==5.3.3 | platformdirs==4.2.1 | ipywidgets==8.1.2 | rsa==4.9 |
| certifi==2024.2.2 | plotly==5.21.0 | iso8601==2.1.0 | scipy==1.10.1 |
| charset-normalizer==3.3.2 | prompt-toolkit==3.0.43 | itsdangerous==2.2.0 | seaborn==0.13.2 |
| click==8.1.7 | protobuf==3.20.2 | jedi==0.19.1 | serial==0.0.97 |
| colorama==0.4.6 | psutil==5.9.8 | Jinja2==3.1.3 | shapely==2.0.4 |
| comm==0.2.2 | pure-eval==0.2.2 | jsonschema==4.21.1 | shiboken6==6.6.3.1 |
| ConfigArgParse==1.7 | py-cpuinfo==9.0.0 | jsonschema-specifications==2023.12.1 | six==1.16.0 |
| contourpy==1.1.1 | pyasn1==0.6.0 | jupyter\_core==5.7.2 | smmap==5.0.1 |
| cycler==0.12.1 | pyasn1\_modules==0.4.0 | jupyterlab\_widgets==3.0.10 | sniffio==1.3.1 |
| dash==2.16.1 | pyclipper==1.3.0.post5 | kiwisolver==1.4.5 | soupsieve==2.5 |
| dash-core-components==2.0.0 | pycryptodome==3.20.0 | lmdb==1.4.1 | stack-data==0.6.3 |
| dash-html-components==2.0.0 | pygame==2.5.2 | Markdown==3.6 | sympy==1.12 |
| dash-table==5.0.0 | Pygments==2.17.2 | MarkupSafe==2.1.5 | tbb==2021.12.0 |
| decorator==5.1.1 | pyinstaller==6.6.0 | matplotlib==3.7.5 | tenacity==8.2.3 |
| exceptiongroup==1.2.1 | pyinstaller-hooks-contrib==2024.5 | matplotlib-inline==0.1.7 | tensorboard==2.14.0 |
| executing==2.0.1 | PyMuPDF==1.24.4 | MechEyeAPI==1.5.8.2.1.0 | tensorboard-data-server==0.7.2 |
| fastjsonschema==2.19.1 | PyMuPDFb==1.24.3 | mkl==2021.4.0 | termcolor==2.4.0 |
| filelock==3.13.4 | pyparsing==3.1.2 | mpmath==1.3.0 | thop==0.1.1.post2209072238 |
| Flask==3.0.3 | pyserial==3.5 | nbformat==5.10.4 | tifffile==2023.7.10 |
| fonttools==4.51.0 | PySide6==6.6.3.1 | nest-asyncio==1.6.0 | torch==2.3.0 |
| fsspec==2024.3.1 | PySide6\_Addons==6.6.3.1 | networkx==3.1 | torchvision==0.18.0 |
| future==1.0.0 | PySide6\_Essentials==6.6.3.1 | numpy==1.24.4 | tqdm==4.66.2 |
| gitdb==4.0.11 | python-dateutil==2.9.0.post0 | oauthlib==3.2.2 | traitlets==5.14.3 |
| GitPython==3.1.43 | pytz==2024.1 | open3d==0.18.0 | typing\_extensions==4.11.0 |
| google-auth==2.29.0 | PyWavelets==1.4.1 | opencv-python==4.9.0.80 | tzdata==2024.1 |
| google-auth-oauthlib==1.0.0 | pywin32==306 | opt-einsum==3.3.0 | ultralytics==8.2.4 |
| grpcio==1.62.2 | pywin32-ctypes==0.2.2 | wcwidth==0.2.13 | urllib3==2.2.1 |
| Werkzeug==3.0.2 | widgetsnbextension==4.0.10 | zipp==3.18.1 |  |

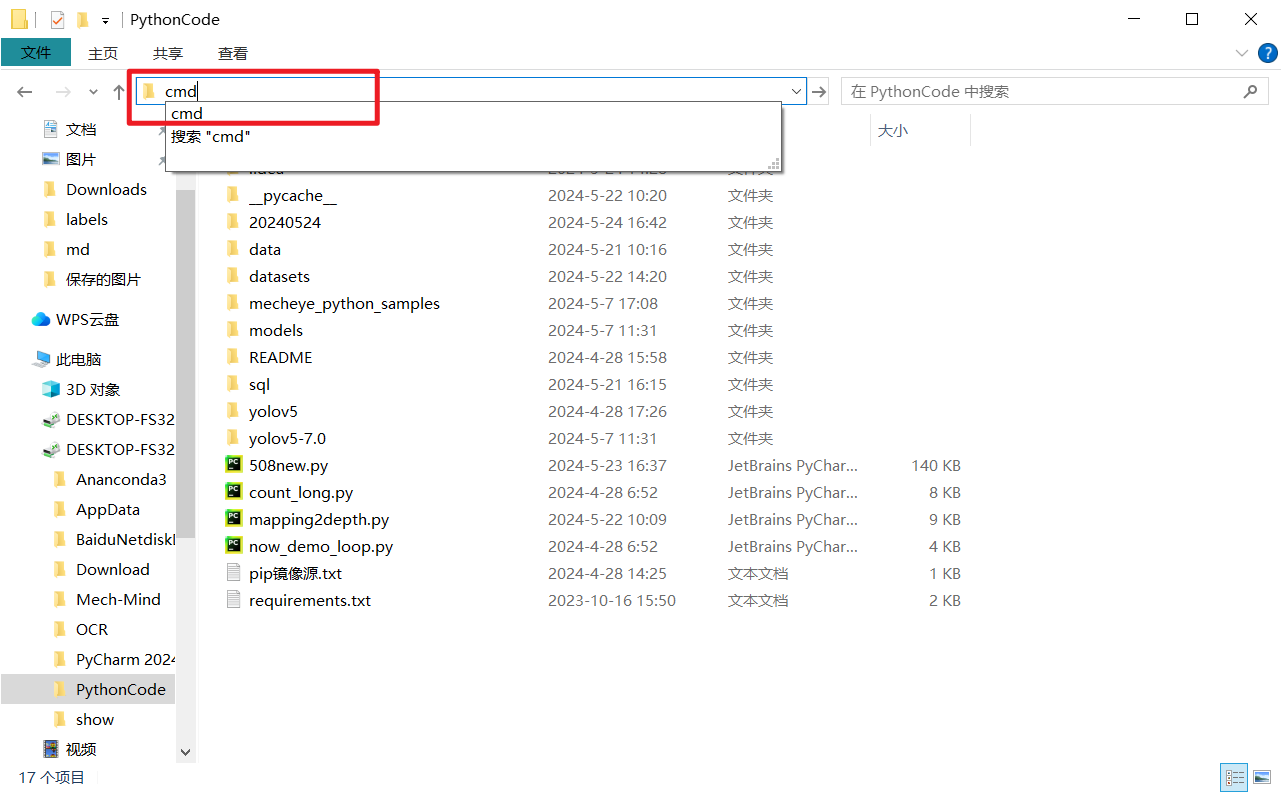
# 运行程序

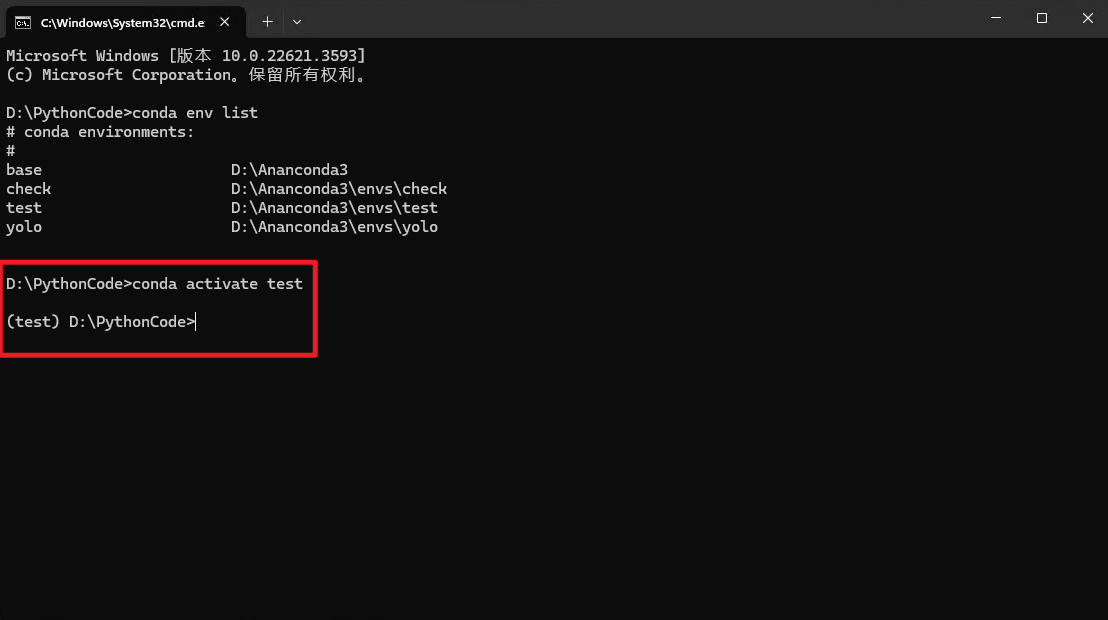
保持进入虚拟环境的状态、进入到代码文件夹内:



如果不是这个状态，需要进入



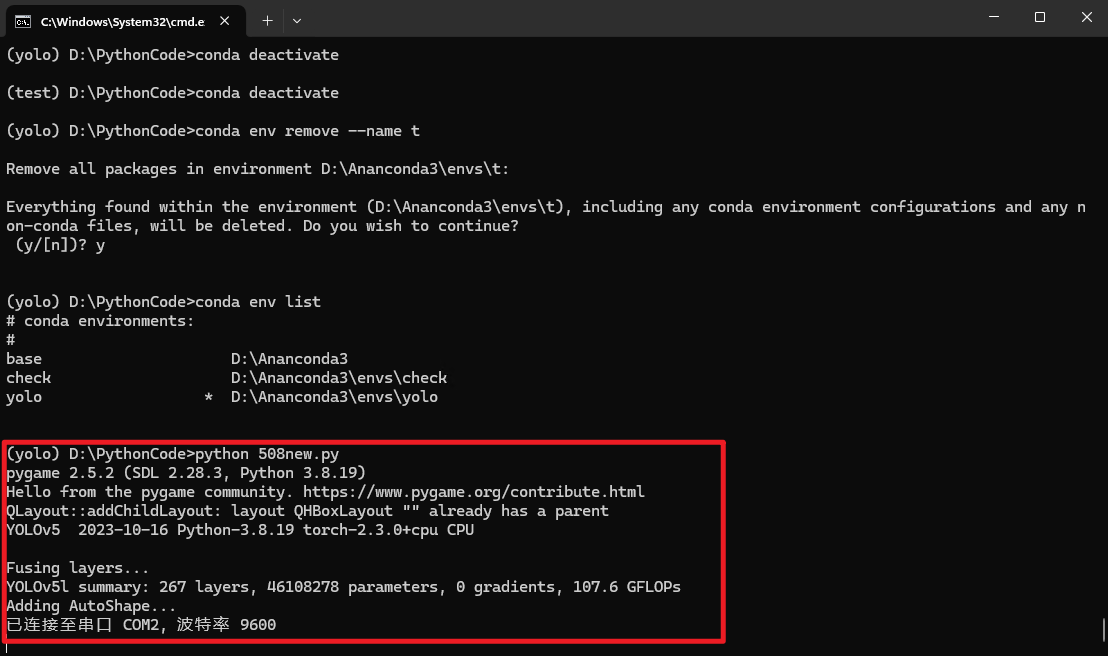




输入：

python 508new.py

运行程序





# 调试代码

## 下载pycharm

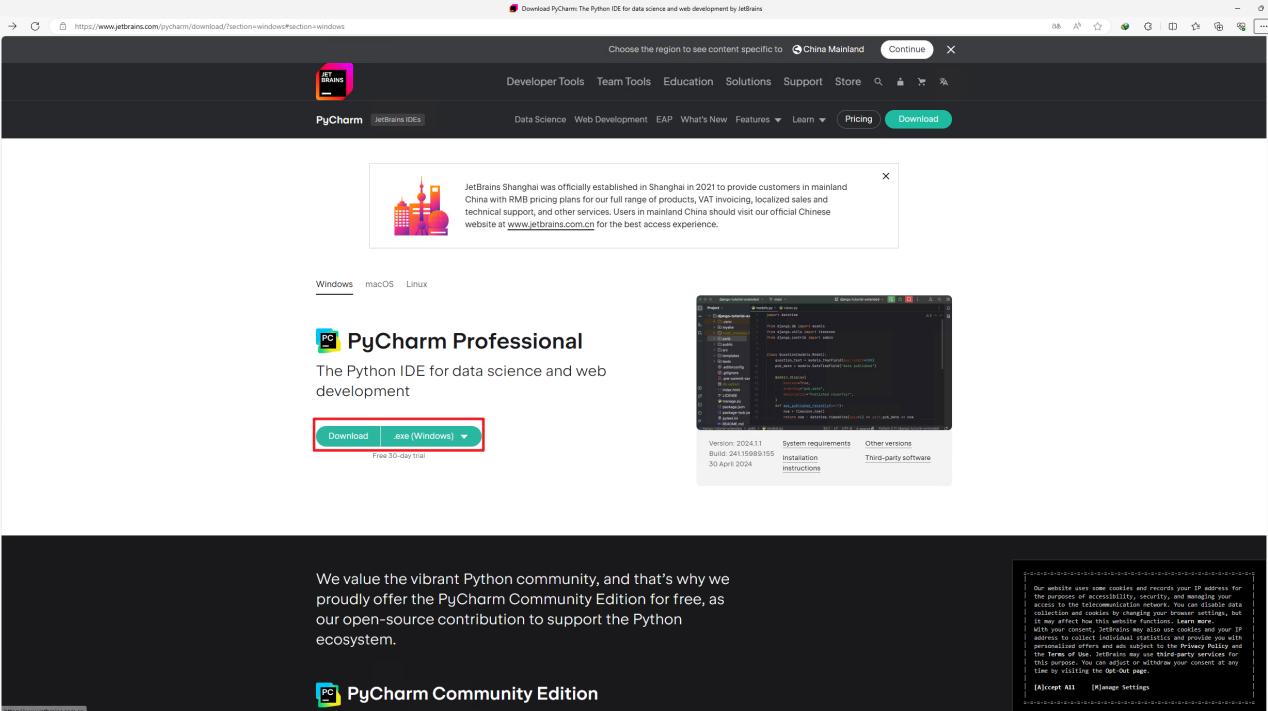
下载“pycharm”软件进行代码的调试

下载地址

<https://www.jetbrains.com/pycharm/download/#section=windows>

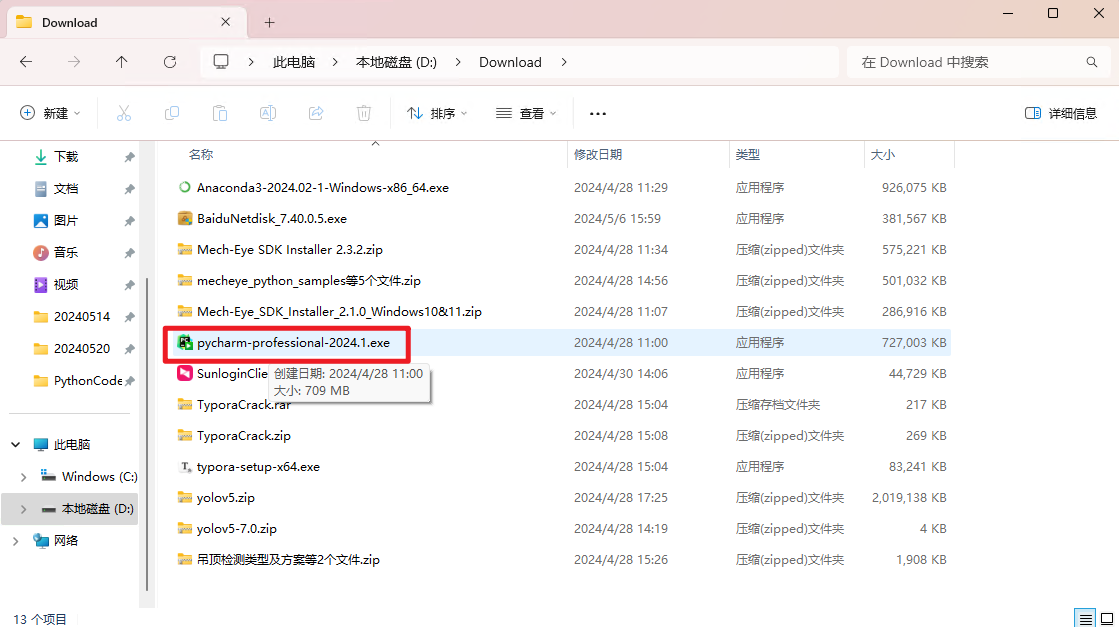
安装完成先退出，进行破解

进入网址，点击Download进行下载，如图：

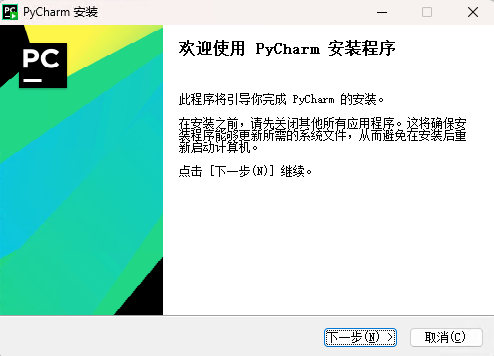


## 安装pycharm

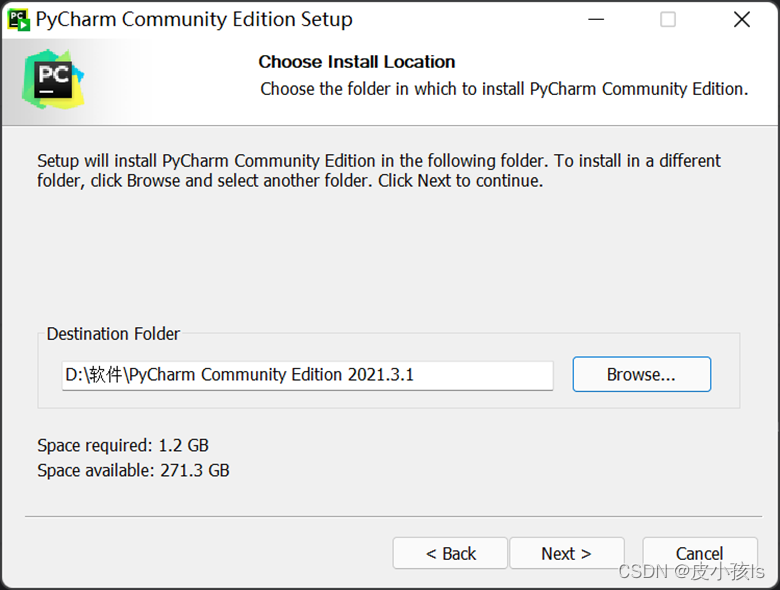
双击pycharm安装程序:



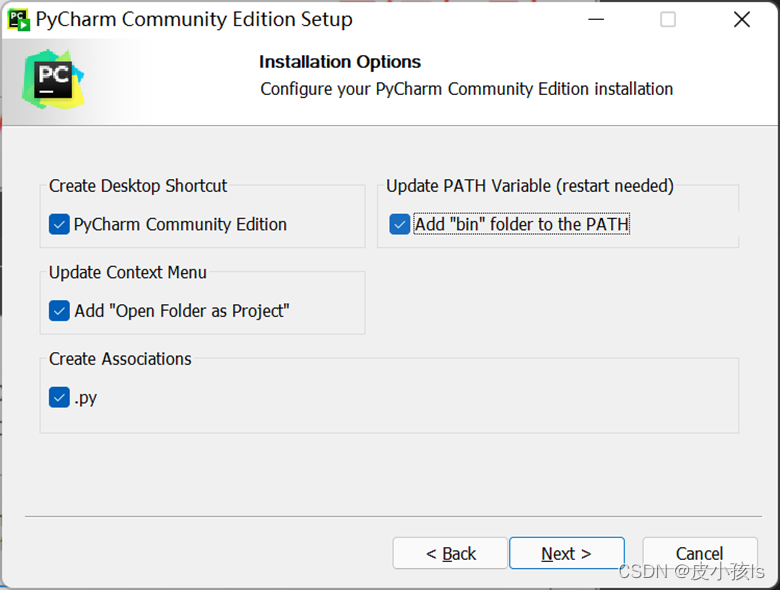
打开安装程序：



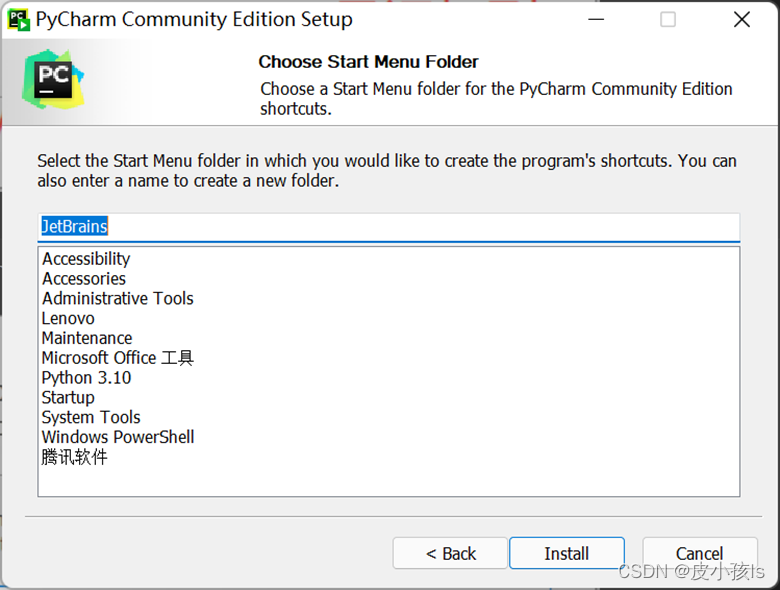
选择安装路径(可以先创建一个文件夹以进行安装):



可以将这些选项全部勾选上：



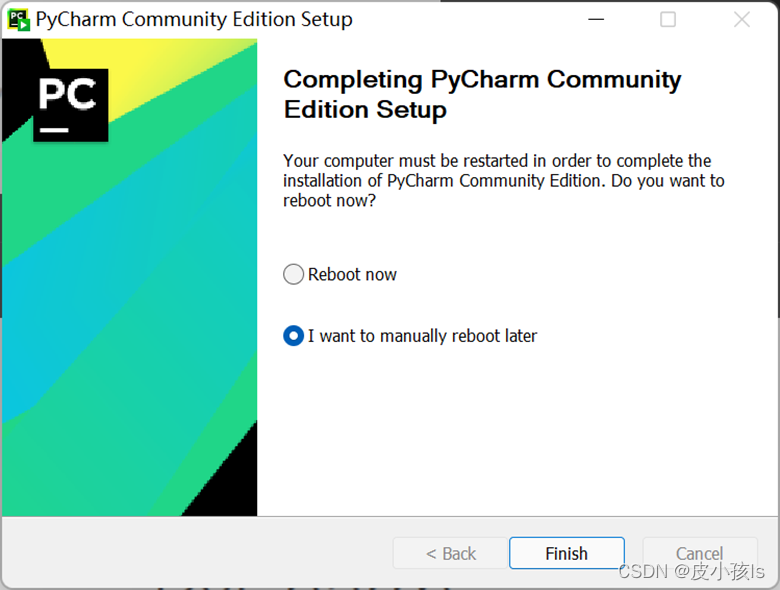
点击Install



安装完后点击Finish，

勾选 I want to manually reboot later:不立刻重新启动

如果想要立刻重新启动勾选：Reboot now



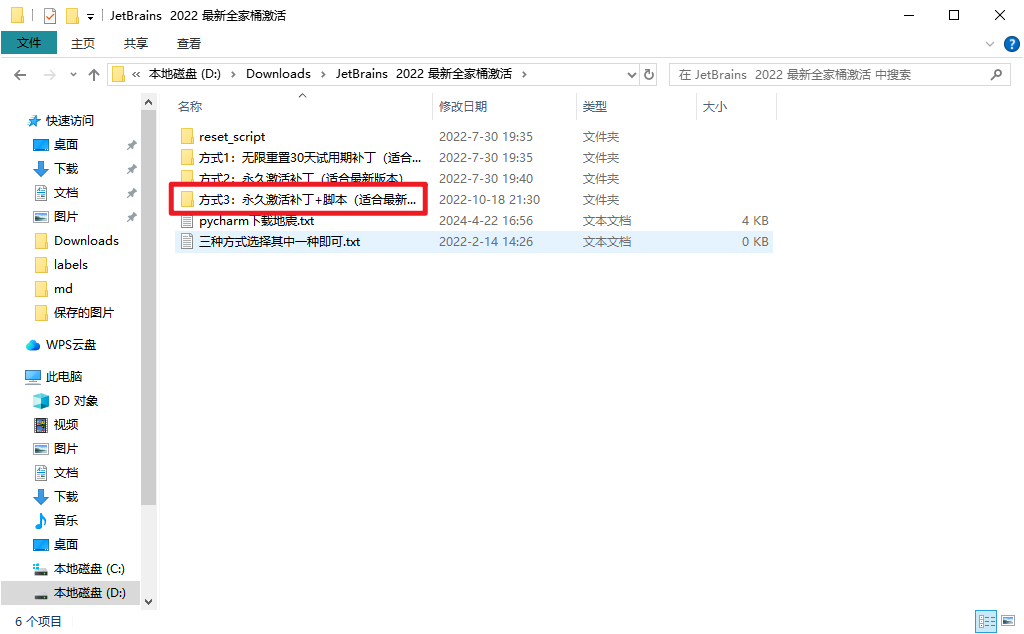
## **破解**

软件只能试用30天，需要破解，才能延长使用时间：

破解补丁：<https://pan.baidu.com/s/15R6CZ5Ylf2FFs4NrnncTtQ?pwd=3857>

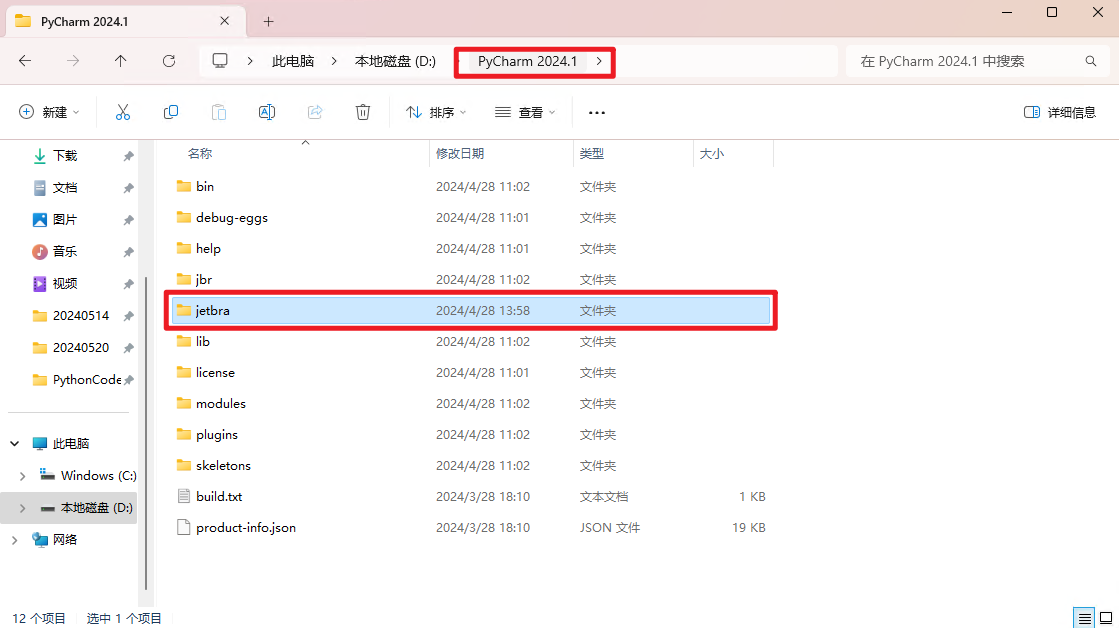
提取码：3857

下载完成后解压后打开文件夹：

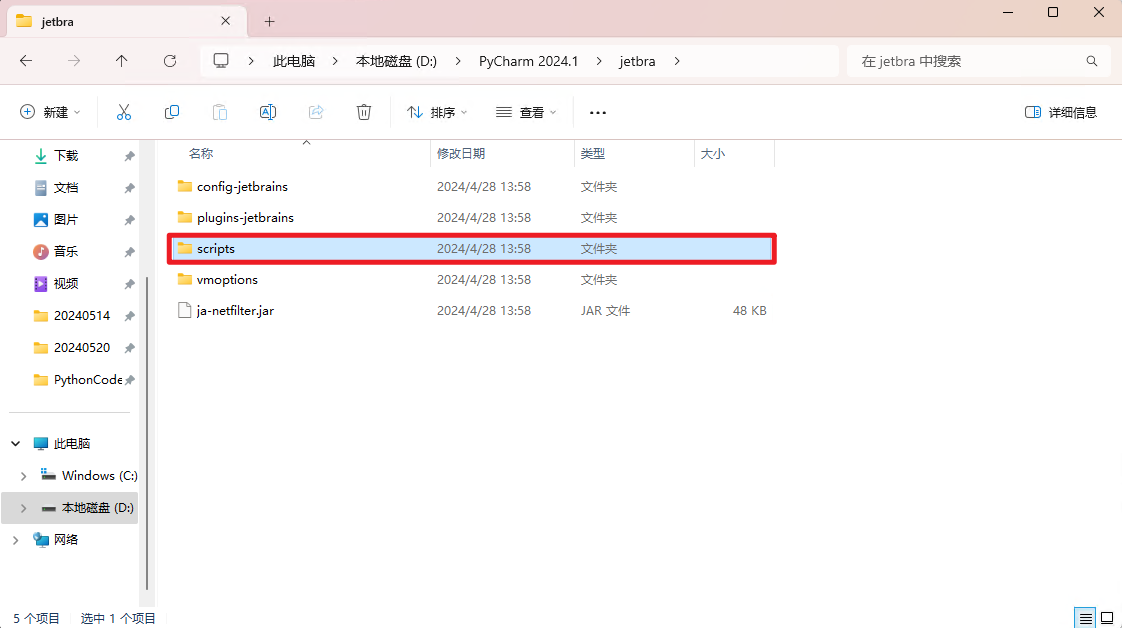


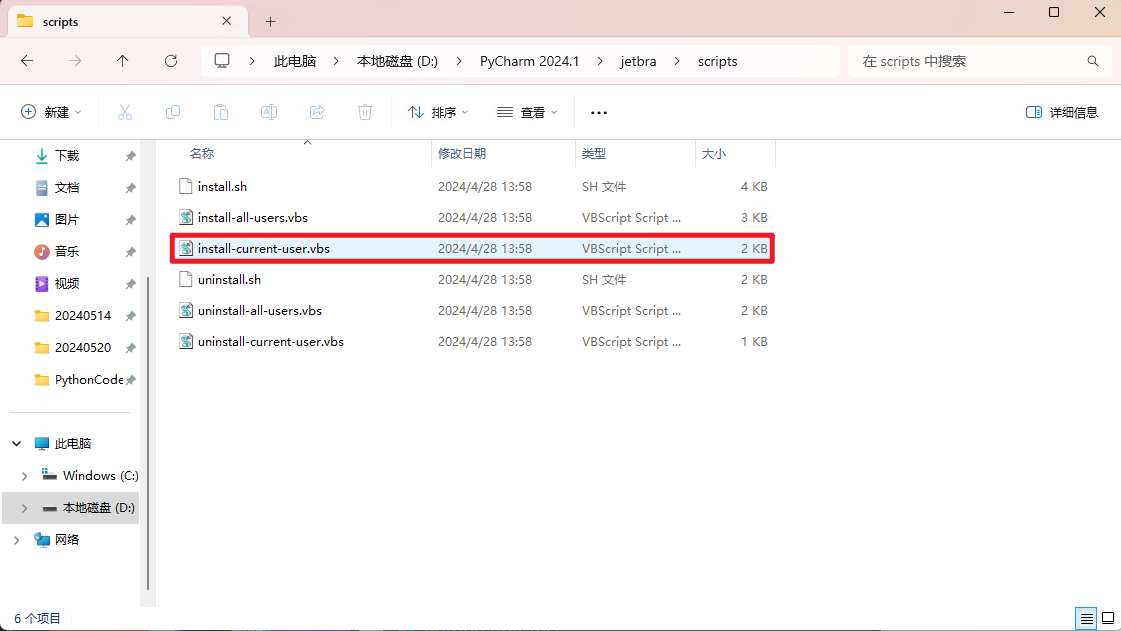
解压文件，点击【方式 3】文件夹 , 进入到文件夹 /jetbra补丁目录，

把/jetbra补丁目录移动到刚刚安装的pychram的安装路径下，

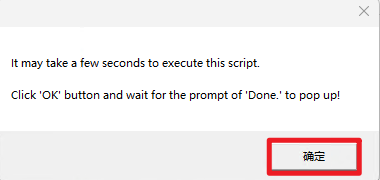


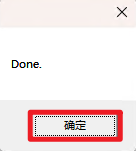
再点击进入 /scripts 文件夹，双击执行 install-current-user.vbs 破解脚本





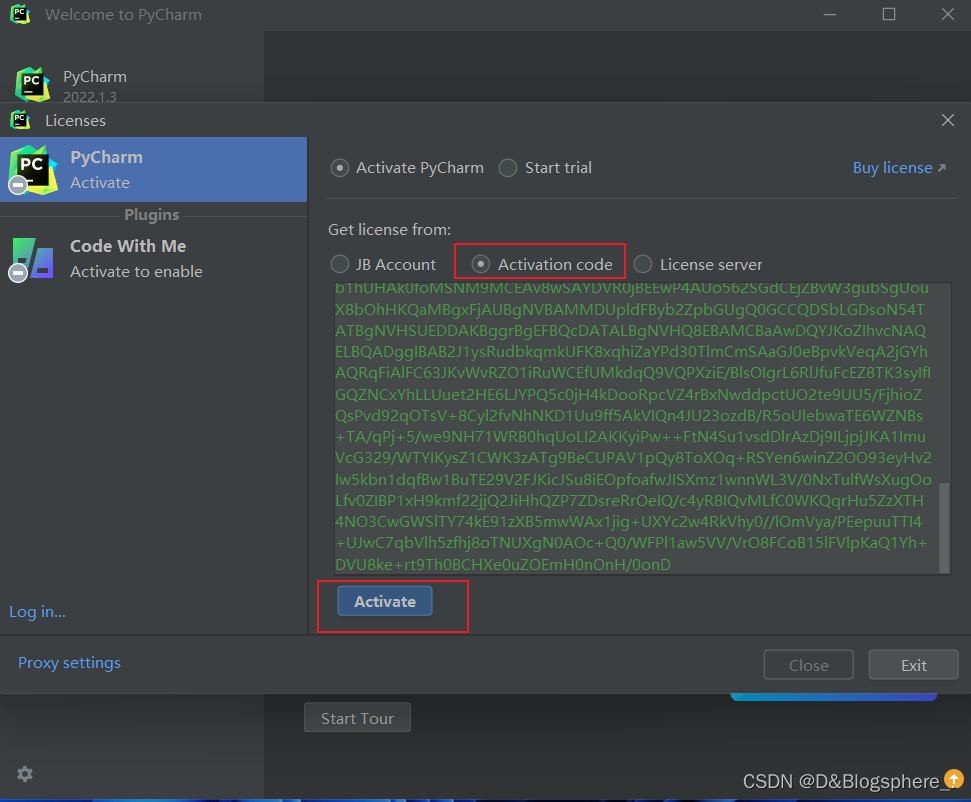
等到弹出对话框，点击done





重新打开 Pycharm 后，复制下面的激活码：

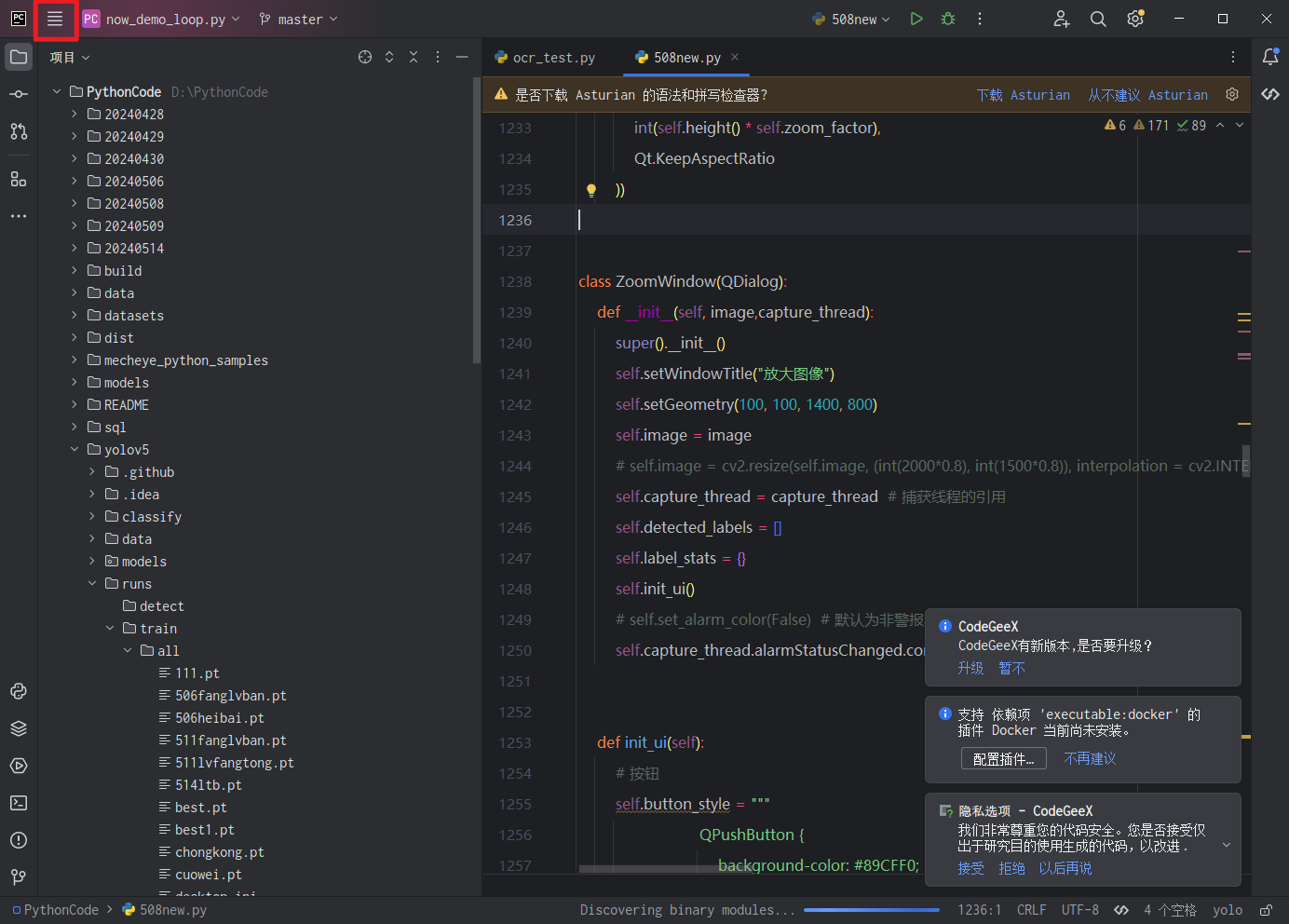
EUWT4EE9X2--FT9l1nyyF9EyNmlelrLP9rGtugZ6sEs3CkYIKqGgSi608LIamge623nLLjI8f6O4EdbCfjJcPXLxklUe1O/5ASO3JnbPFUBYUEebCWZPgPfIdjw7hfA1PsGUdw1SBvh4BEWCMVVJWVtc9ktE+gQ8ldugYjXs0s34xaWjjfolJn2V4f4lnnCv0pikF7Ig/Bsyd/8bsySBJ54Uy9dkEsBUFJzqYSfR7Z/xsrACGFgq96ZsifnAnnOvfGbRX8Q8IIu0zDbNh7smxOwrz2odmL72UaU51A5YaOcPSXRM9uyqCnSp/ENLzkQa/B9RNO+VA7kCsj3MlJWJp5Sotn5spyV+gA==-



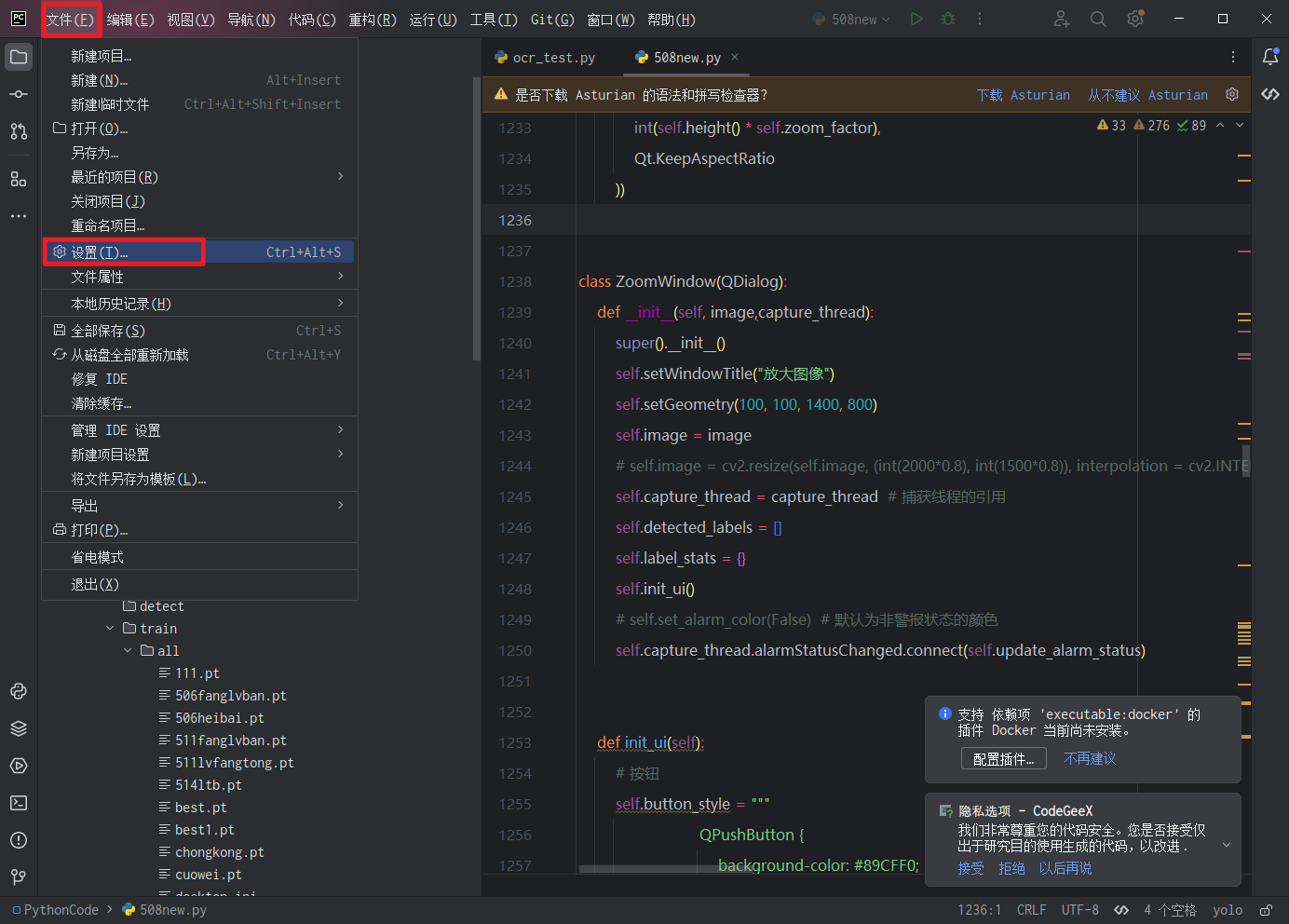
## 安装汉化插件

如果英语很不错可以选择不安装汉化插件：

点击左上角的四道杠“亖”图标，如图：



依次点击“文件”、“设置”，如图：



点击“插件”，

在中间的搜索框内输入“chinese”，选择第二个，

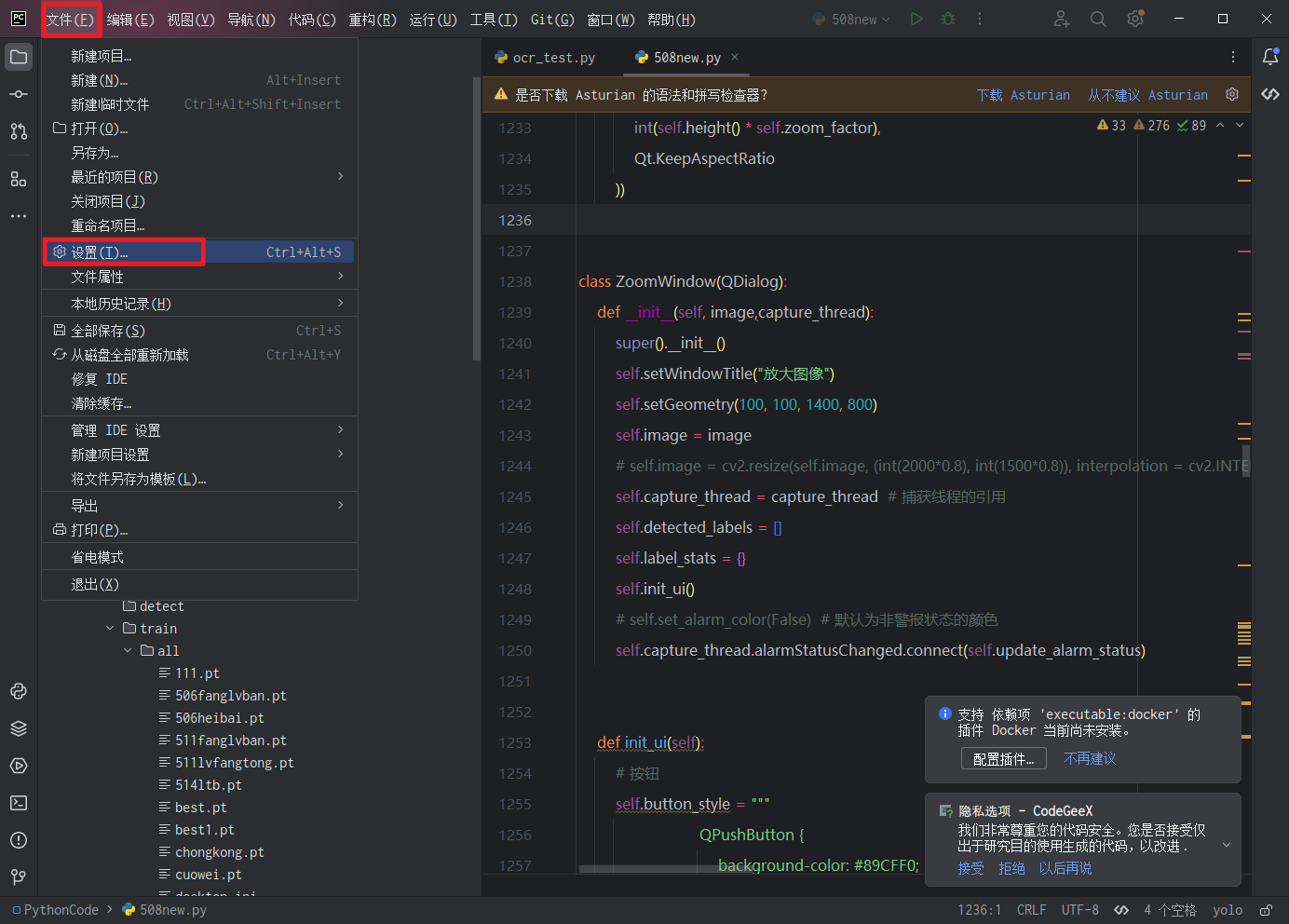
在右侧点击安装/install

安装完成后需要重启软件生效



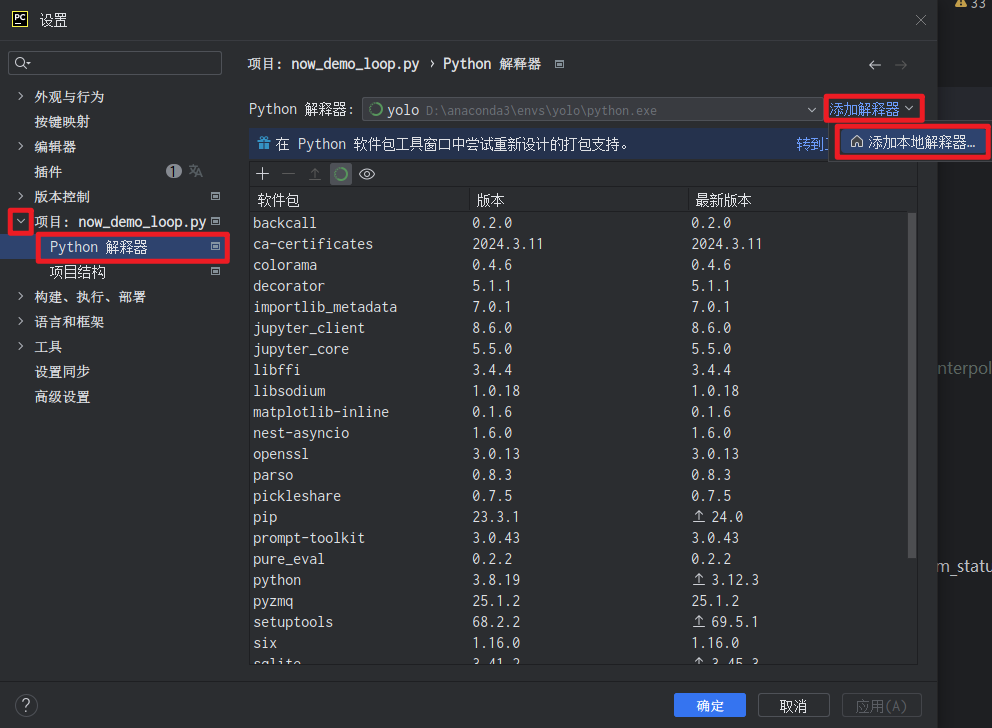
## **为pycharm添加解释器**

依次点击“文件”、“设置”，如图：



点击左侧“项目”左侧的小三角，展开内容，点击“Python解释器”

点击右侧“添加解释器”，选择添加本地解释器



左侧点击“Virtualenv环境”

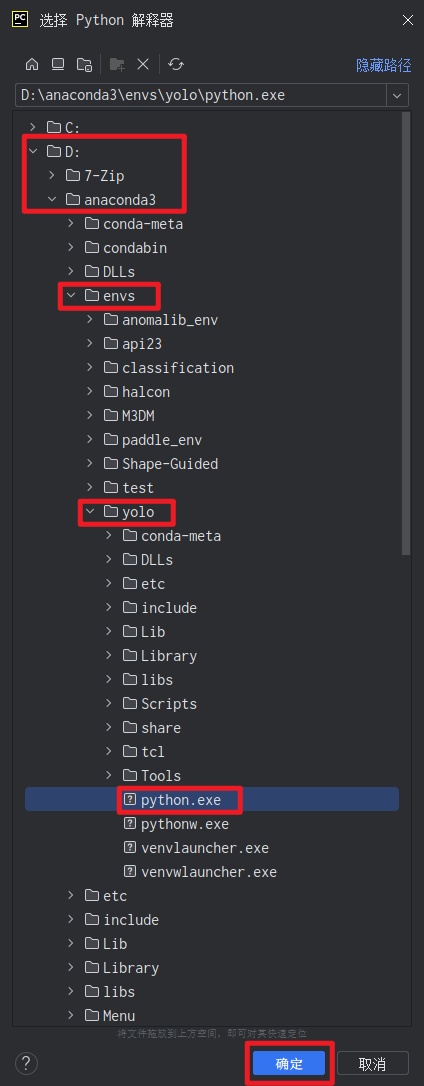
勾选“现有”选项

点击右侧“. . ."按键

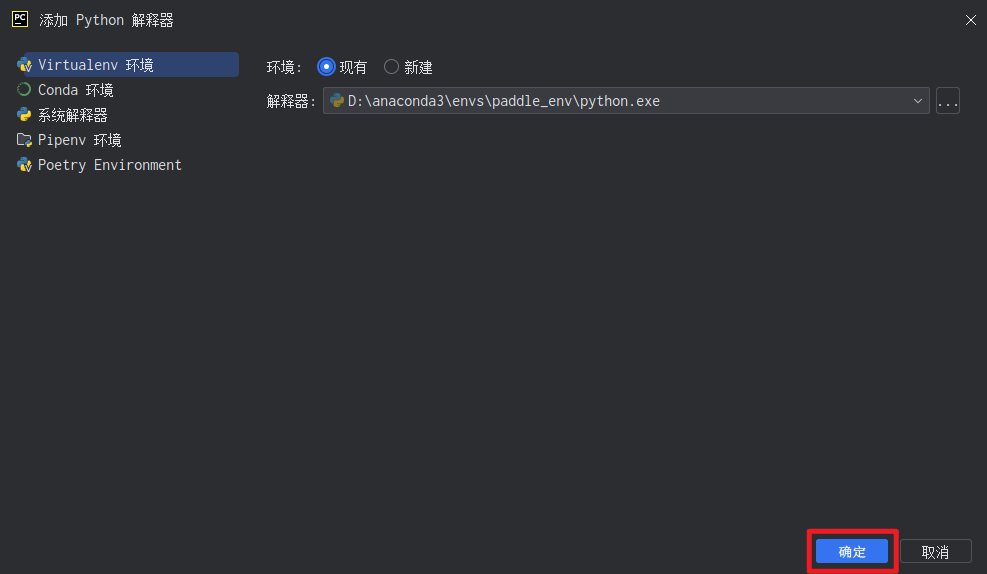


选择anaconda安装路径下的“envs”文件夹

选择刚刚创建的虚拟环境名称，以（环境名称为yolo）为例：选择名为“yolo”文件夹下的“python.exe”文件，最后点击“确定”

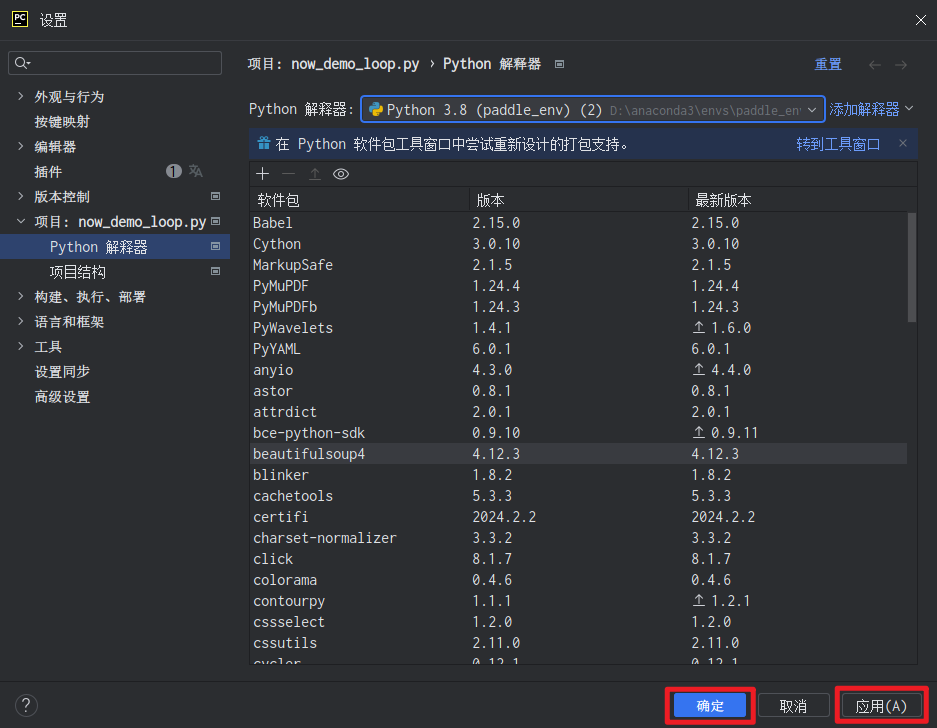


点击“确定”



依次点击“应用(A)”、“确认”

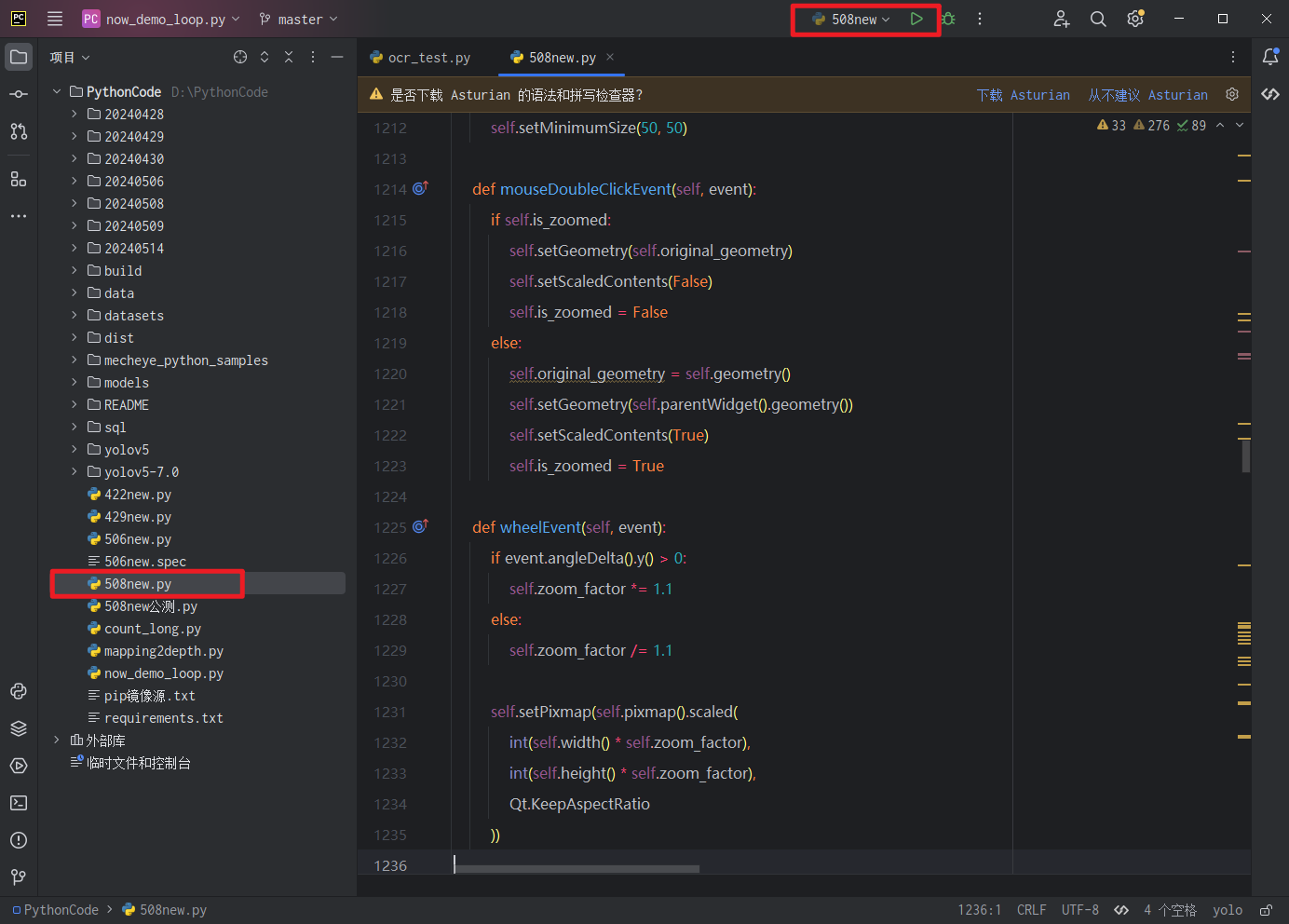
或直接点击“确认”



## 运行、调试代码

左侧框内选择文件，双击打开

点击顶部三角形状可运行，确保三角前面的文件名是要运行的文件



右侧区域为编辑代码的窗口，可在此处编辑代码

