

# 深度学习--环境配置

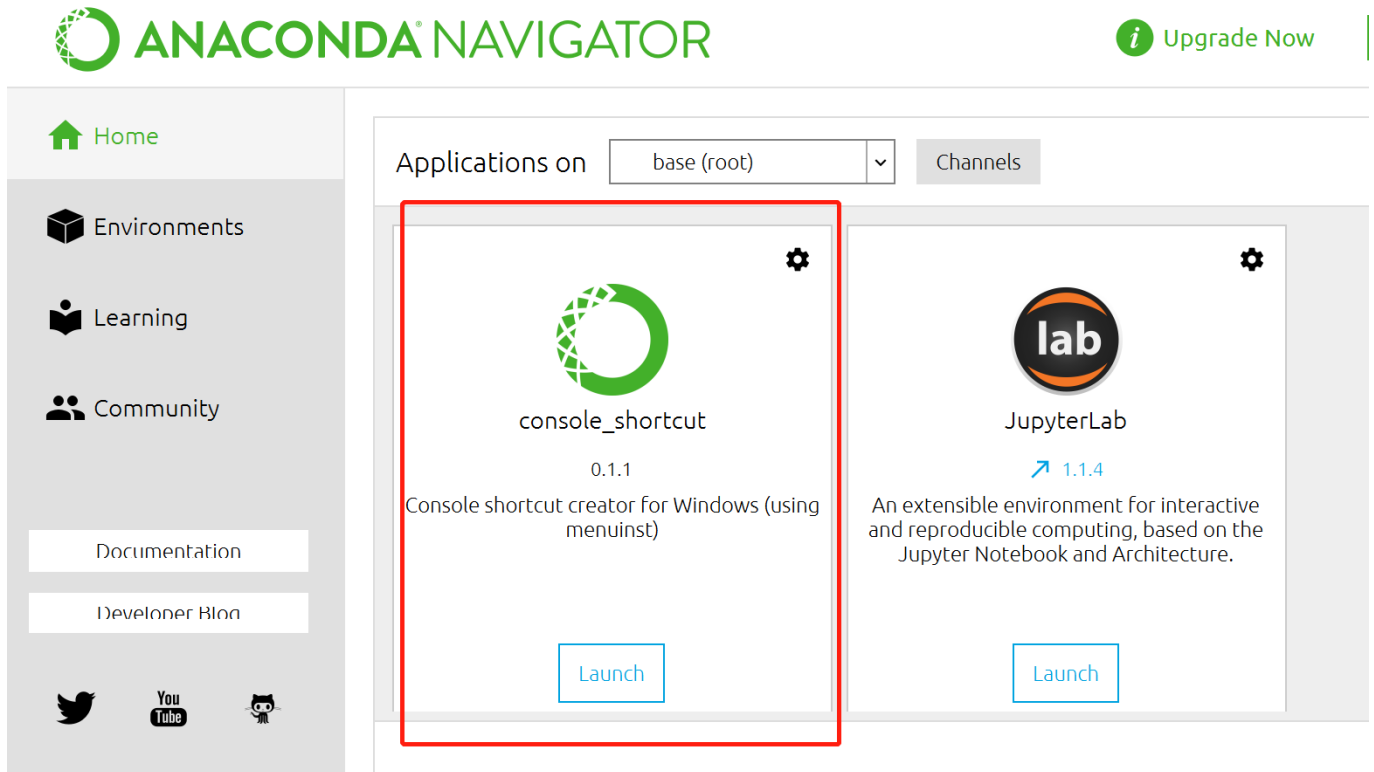
zqy

**注意:** (1) 在Part 2部署运行环境的时候需要先cd到项目的根目录下, 如E:\Git-local, 然后再进行conda env list等操作

(2) cmd不要用win+r打开, 要在anaconda中打开

Anaconda Navigator

File Help



## Part 1 开发环境搭建 (硬件+软件)

要有NVIDIA的显卡, 因为AMD的无法安装CUDA; NVIDIA显卡的配置最好GTX 980+, 显存8G以上; 或者可以用平台提供的免费算力 (用起来麻烦, 但是偶尔跑个算法还是ok的)

### Step 1. 安装cuda

cuda是英伟达推出的运算平台 我们要根据显卡的型号来选择cuda的版本

如何查看电脑NVIDIA的版本: 桌面右键打开NVIDIA控制面板, 然后在系统信息中查看 笔者NVIDIA的版本是:  
511.79



然后在英伟达的官网下载对应的cuda版本 英伟达官网：[docs.nvidia.com/cuda/cuda-toolkit-release-notes/index.html](https://docs.nvidia.com/cuda/cuda-toolkit-release-notes/index.html)

docs.nvidia.com/cuda/cuda-toolkit-release-notes/index.html

**DEVELOPER ZONE** **CUDA TOOLKIT DOCUMENTATION**

CUDA Toolkit v11.6.2  
Release Notes

- 1. CUDA 11.6 Release Notes
  - 1.1. CUDA Toolkit Major Component Versions
  - 1.2. General CUDA
  - 1.3. CUDA Compilers
  - 1.4. CUDA Developer Tools
  - 1.5. Resolved Issues
  - 1.6. Depreciated Features
  - 1.7. Known Issues
- 2. CUDA Libraries

applications compiled against a particular version of the CUDA SDK continue to work on subsequent (later) driver releases.

More information on compatibility can be found at <https://docs.nvidia.com/cuda/cuda-c-best-practices-guide/index.htm#and-upgrades>.

**Note:** Starting with CUDA 11.0, the toolkit components are individually versioned, and the toolkit itself is versioned as sh

The minimum required driver version for CUDA minor version compatibility is shown below. CUDA minor version compatib in <https://docs.nvidia.com/deploy/cuda-compatibility/index.html>

**Table 2. CUDA Toolkit and Minimum Required Driver Version for CUDA Minor Version Compatibility**

CUDA Toolkit	Minimum Required Driver Version for CUDA Minor Version Compatibility*	
	Linux x86_64 Driver Version	Windows x86_64 Driver Version
CUDA 11.6.x	>=450.80.02	>=452.39
CUDA 11.5.x	>=450.80.02	>=452.39
CUDA 11.4.x	>=450.80.02	>=452.39
CUDA 11.3.x	>=450.80.02	>=452.39
CUDA 11.2.x	>=450.80.02	>=452.39
CUDA 11.1 (11.1.0)	>=450.80.02	>=452.39
CUDA 11.0 (11.0.3)	>=450.36.06**	>=451.22**

\* Using a Minimum Required Version that is different from Toolkit Driver Version could be allowed in compatibility mode - Compatibility Guide for details.

\*\* CUDA 11.0 was released with an earlier driver version, but by upgrading to Tesla Recommended Drivers 450.80.02 (Linux) minor version compatibility is possible across the CUDA 11.x family of toolkits.

这里安装cuda 11.1.0,打开网址：[developer.nvidia.com/cuda-11.1.0-download-archive](https://developer.nvidia.com/cuda-11.1.0-download-archive) 根据电脑的操作系统下载，得到exe文件

## CUDA Toolkit 11.1.0

**Please Note:** Due to an incompatibility issue, we advise users to defer updating to Linux Kernel 5.9+ until mid-November when an NVIDIA Linux GPU driver update with Kernel 5.9+ support is expected to be available.

**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 Server 2019 Server 2016

**Installer Type**

exe (local) exe (network)

**Download Installer for Windows 10 x86\_64**

The base installer is available for download below.

> **Base Installer** Download (3.1 GB) 

Installation Instructions:

1. Double click cuda\_11.1.0\_456.43\_win10.exe

文件有点大，需要等待一下 下载后直接运行exe文件，一直点下去ok就行，一直安装...



检测cuda是否安装：打开cmd，输入nvcc -V,若出现cuda版本，则ok

```
Microsoft Windows [版本 10.0.19043.928]
(c) Microsoft Corporation。保留所有权利。

C:\Users\MC>nvcc -V
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2020 NVIDIA Corporation
Built on Tue_Sep_15_19:12:04_Pacific_Daylight_Time_2020
Cuda compilation tools, release 11.1, V11.1.74
Build cuda_11.1.rel1gpu_driver.455TC455_06.29069683_0

C:\Users\MC>
```

安装完cuda后，可以在cmd中使用nvidia-smi指令查看显卡使用情况

```
C:\Users\MC>nvidia-smi
Fri Apr 22 10:44:49 2022

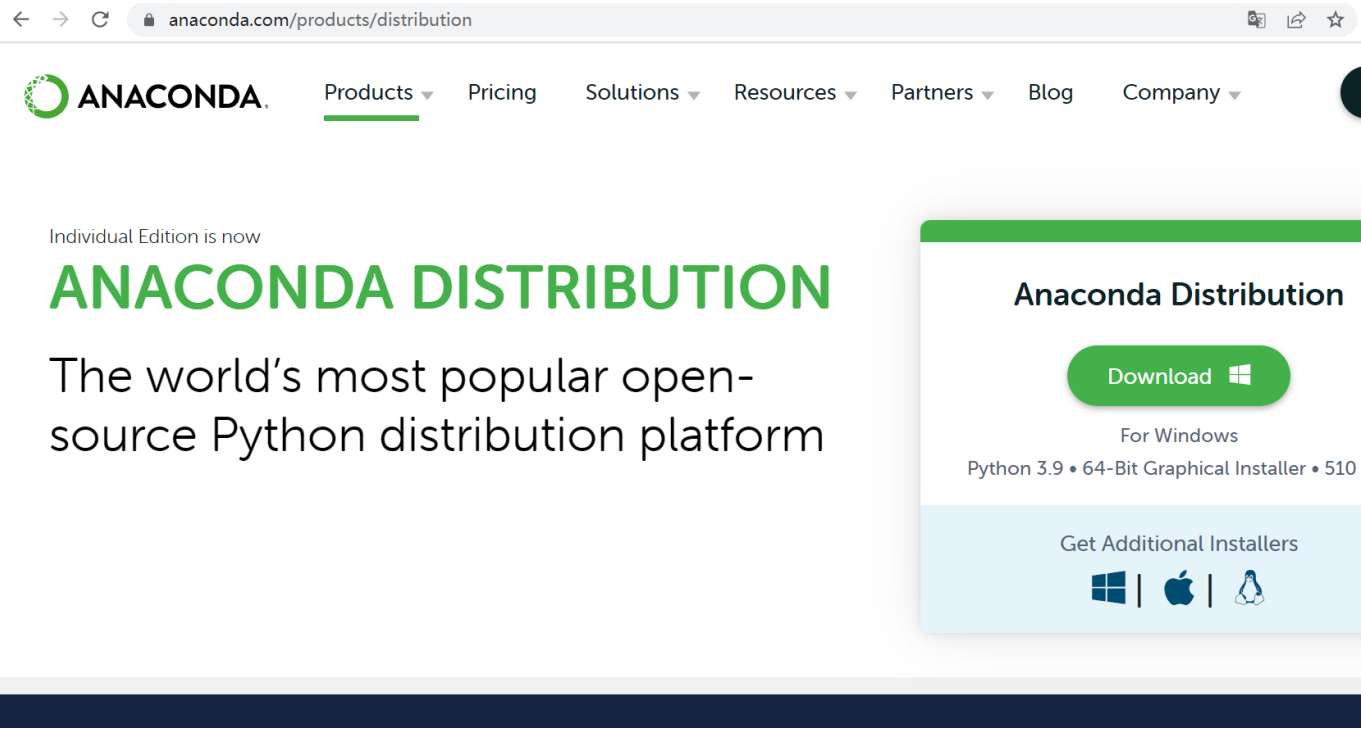
+-----+
| NVIDIA-SMI 511.79                Driver Version: 511.79          CUDA Version: 11.6         |
+-----+-----+
| GPU Name           TCC/WDDM    Bus-Id        Disp.A   Volatile Uncorr. ECC  |
| Fan  Temp  Perf  Pwr:Usage/Cap      Memory-Usage  GPU-Util  Compute M. |
|              |              |              |      |              |              |
+-----+-----+
|   0  NVIDIA GeForce ... WDDM      00000000:01:00:0 Off   0%          N/A   |
| N/A   42C   P8     8W /  N/A      42MiB /  4096MiB             Default  |
|              |              |              |      |              |              |
+-----+-----+

Processes:
+-----+
| GPU  GI  CI           PID  Type  Process name                        GPU Memory |
|   ID  ID  ID                                   Usage   |
+-----+
|    0   N/A N/A       17444   C+G   E:\vpn\Acc\Acc. exe                 N/A       |
+-----+
```

至此，cuda安装完成

Step 2.安装Anaconda 3

Anaconda是python的包管理器和环境管理器，安装包的版本根据电脑系统选择就ok 下载地址：  
<https://www.anaconda.com/products/distribution>



如果电脑下载很慢，也可以去清华大学开源软件镜像站下载 网址：

https://mirrors.tuna.tsinghua.edu.cn/anaconda/archive/?C-M&O-D

清华大学开源软件镜像站

HOMEEVENTSBLOGRSSPODCAST

Index of /anaconda/archive/

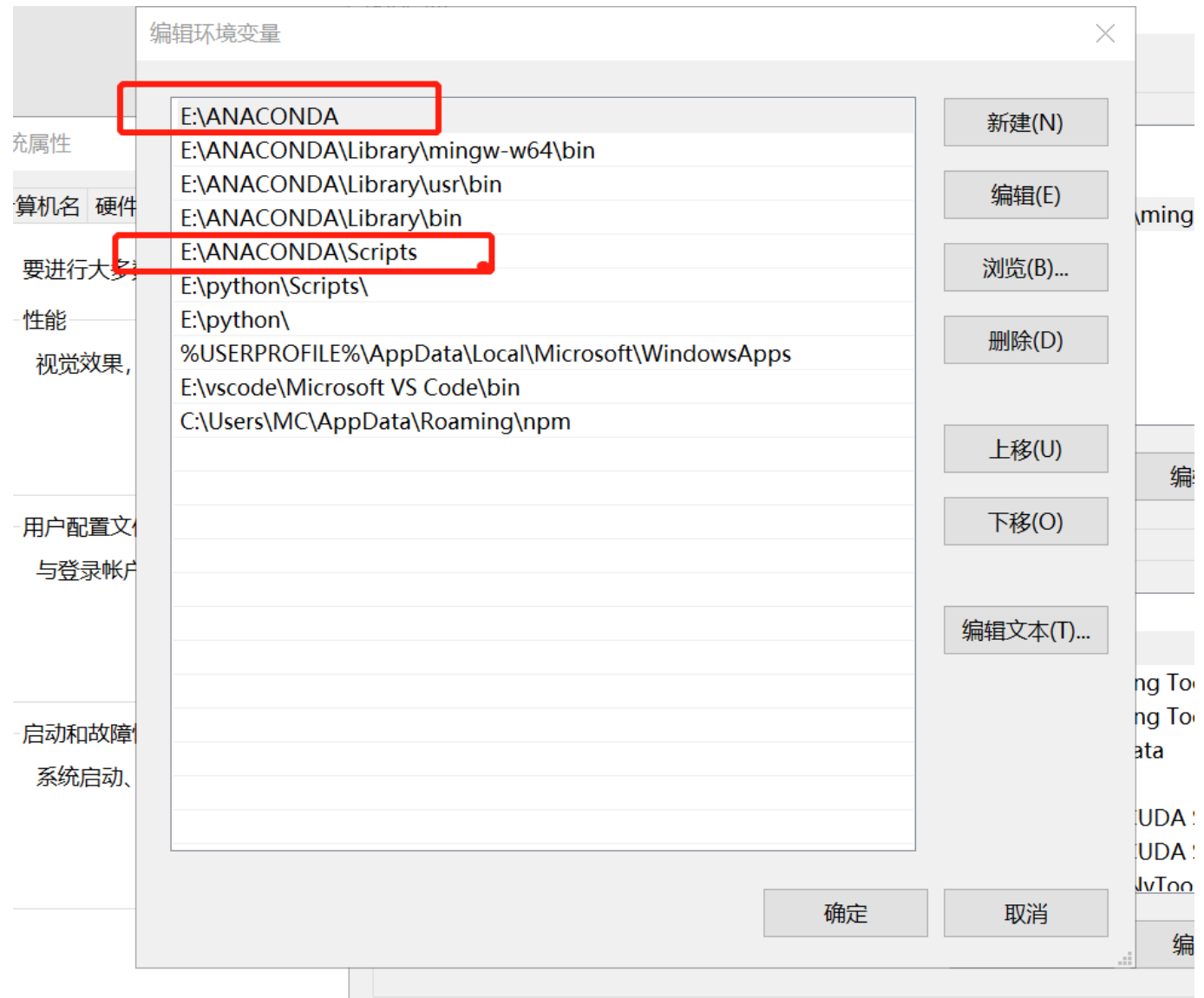
Last Update: 2022-04

File Name ↓	File Size ↓	Date ↓
Parent directory/	-	-
Anaconda-1.4.0-Linux-x86.sh	220.5 MiB	2013-07-04 01:47
Anaconda-1.4.0-Linux-x86_64.sh	286.9 MiB	2013-07-04 17:26
Anaconda-1.4.0-MacOSX-x86_64.sh	156.4 MiB	2013-07-04 17:40
Anaconda-1.4.0-Windows-x86.exe	210.1 MiB	2013-07-04 17:48
Anaconda-1.4.0-Windows-x86_64.exe	241.4 MiB	2013-07-04 17:58
Anaconda-1.5.0-Linux-x86.sh	238.8 MiB	2013-07-04 18:10
Anaconda-1.5.0-Linux-x86_64.sh	306.7 MiB	2013-07-04 18:22
Anaconda-1.5.0-MacOSX-x86_64.sh	166.2 MiB	2013-07-04 18:37
Anaconda-1.5.0-Windows-x86.exe	236.0 MiB	2013-07-04 18:45
Anaconda-1.5.0-Windows-x86_64.exe	280.4 MiB	2013-07-04 18:57
Anaconda-1.5.1-MacOSX-x86_64.sh	166.2 MiB	2013-07-04 19:11

下载后傻瓜式安装，一直点下去就ok

配置环境变量，在系统变量中添加anaconda的根目录和Scripts目录，笔者目录如下:

- anaconda根目录：E:\ANACOND
- Scripts目录：E:\ANACONDA\Scripts



Conda 简介：Conda 是一个开源的软件包管理系统和环境管理系统，用于安装多个版本的软件包及其依赖关系，并在它们之间轻松切换。Conda 是为 Python 程序创建的，适用于 Linux, OS X 和 Windows，也可以打包和分发其他软件，是目前流行的Python环境管理工具

配置好后，在cmd中运行`conda env list`就能查看到我们的虚拟环境了

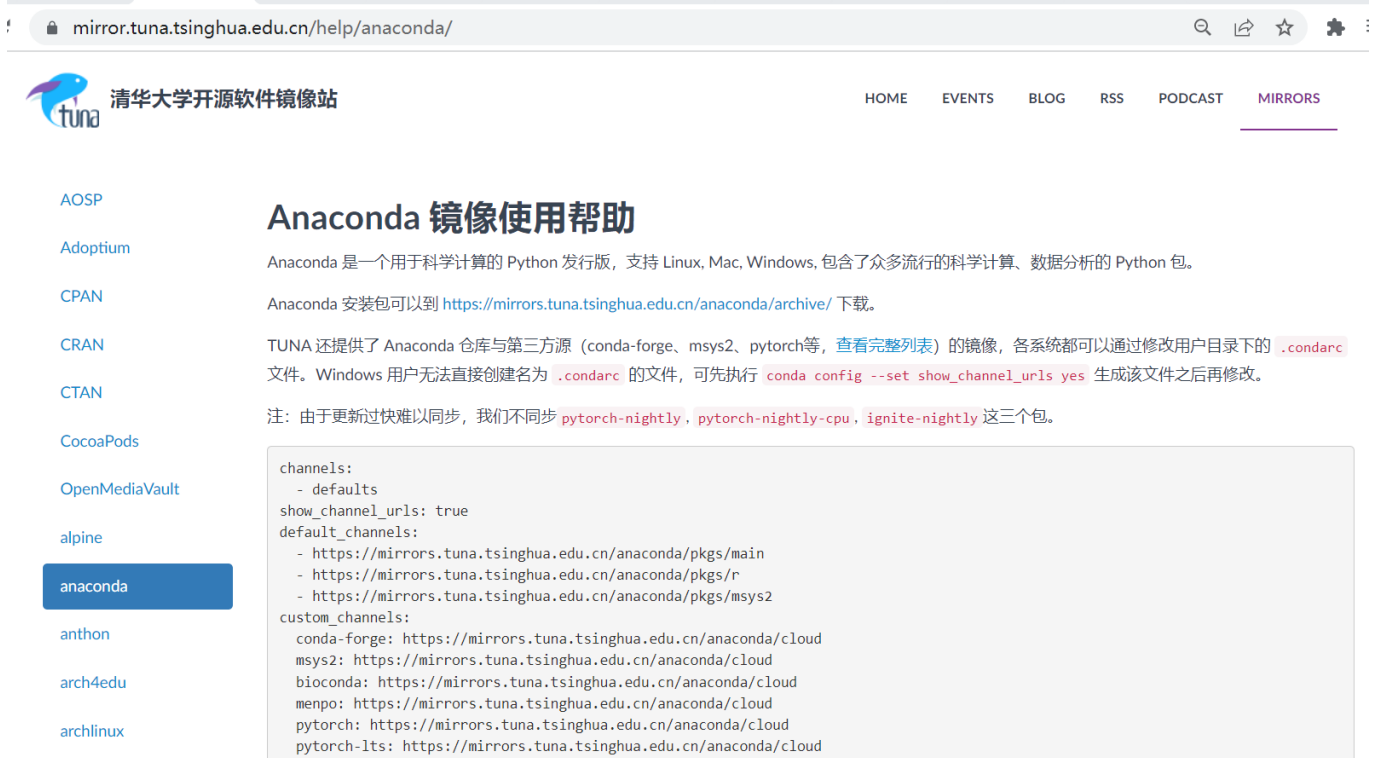
```
C:\Windows\system32\cmd.exe

Microsoft Windows [版本 10.0.19043.928]
(c) Microsoft Corporation。保留所有权利。

C:\Users\MC>conda env list
# conda environments:
#
base                                 * E:\ANACONDA

C:\Users\MC>
```

安装好后，我们需要先更换安装包的下载源 [mirror.tuna.tsinghua.edu.cn/help/anaconda](https://mirror.tuna.tsinghua.edu.cn/help/anaconda)



conda默认使用的是国外的源地址，下载速度很慢 修改用户目录下的`condarc`文件，添加如下内容并保存，这样就可以了,具体操作：**a.先运行`conda config --set show_channel_urls yes`**然后在用户目录下就会生成`condarc`文件，以记事本格式打开，添加以下内容并保存 注意：下面的用`http`,不要用`https`，不然会报错

```
channels:
  - defaults
show_channel_urls: true
default_channels:
  - http://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
  - http://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/r
```



```
- http://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgms/msys2
custom_channels:
conda-forge: http://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
msys2: http://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
bioconda: http://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
menpo: http://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
pytorch: http://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
pytorch-lts: http://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
simpleitk: http://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud
```

然后创建一个新的环境，用于后续的算法环境的搭建，输入指令`conda create -n zqy jupyter notebook cudnn` `cudnn`是用于深度神经网络的GPU加速库，`jupyter notebook`是好用的交互式编辑器

**cuDNN概述** (1) NVIDIA CUDA® 深度神经网络库 (cuDNN) 是GPU加速的用于深度神经网络的原语库。cuDNN为标准例程提供了高度优化的实现，例如向前和向后卷积，池化，规范化和激活层。(2) 全球的深度学习研究人员和框架开发人员都依赖cuDNN来实现高性能GPU加速。它使他们可以专注于训练神经网络和开发软件应用程序，而不必花时间在底层GPU性能调整上。cuDNN的加快广泛使用的深度学习框架，包括Caffe2, Chainer, Keras, MATLAB, MxNet, PyTorch和TensorFlow。已将cuDNN集成到框架中的NVIDIA优化深度学习框架容器，访问NVIDIA GPU CLOUD了解更多信息并开始使用。(3) `cudnn`加速框架



这里都会安装到新的虚拟环境中，然后点y,开始安装第三方依赖库第一次下载可能会很慢，慢慢等待

```
C:\Windows\system32\cmd.exe - conda create -n zqy jupyter notebook cudnn
sip anaconda/pkgs/main/win-64::sip-4.19.13-py39hd77b12b_0
six anaconda/pkgs/main/noarch::six-1.16.0-pyhd3eb1b0_1
soupsieve anaconda/pkgs/main/noarch::soupsieve-2.3.1-pyhd3eb1b0_0
sqlite anaconda/pkgs/main/win-64::sqlite-3.38.2-h2bbff1b_0
stack_data anaconda/pkgs/main/noarch::stack_data-0.2.0-pyhd3eb1b0_0
terminado anaconda/pkgs/main/win-64::terminado-0.13.1-py39haa95532_0
testpath anaconda/pkgs/main/noarch::testpath-0.5.0-pyhd3eb1b0_0
tornado anaconda/pkgs/main/win-64::tornado-6.1-py39h2bbff1b_0
traitlets anaconda/pkgs/main/noarch::traitlets-5.1.1-pyhd3eb1b0_0
typing-extensions anaconda/pkgs/main/noarch::typing-extensions-4.1.1-hd3eb1b0_0
typing_extensions anaconda/pkgs/main/noarch::typing_extensions-4.1.1-pyh06a4308_0
tzdata anaconda/pkgs/main/noarch::tzdata-2022a-hda174b7_0
vc anaconda/pkgs/main/win-64::vc-14.2-h21ff451_1
vs2015_runtime anaconda/pkgs/main/win-64::vs2015_runtime-14.27.29016-h5e58377_2
wcwidth anaconda/pkgs/main/noarch::wcwidth-0.2.5-pyhd3eb1b0_0
webencodings anaconda/pkgs/main/win-64::webencodings-0.5.1-py39haa95532_1
wheel anaconda/pkgs/main/noarch::wheel-0.37.1-pyhd3eb1b0_0
widgetsnbextension anaconda/pkgs/main/win-64::widgetsnbextension-3.5.2-py39haa95532_0
wincertstore anaconda/pkgs/main/win-64::wincertstore-0.2-py39haa95532_2
winpty anaconda/pkgs/main/win-64::winpty-0.4.3-4
zlib anaconda/pkgs/main/win-64::zlib-1.2.12-h8cc25b3_1
```

Proceed ([y]/n)? y

Downloading and Extracting Packages

ipython-8.2.0	1011 KB	#####	100%
typing_extensions-4.	28 KB	#####	100%
qt-5.9.7	72.5 MB	#####	100%
winpty-0.4.3	678 KB	#####	100%
prometheus_client-0.	47 KB	#####	100%
webencodings-0.5.1	20 KB	#####	100%
ipykernel-6.9.1	200 KB	#####	100%
argon2-cffi-bindings	36 KB	#####	100%
cudnn-8.2.1	428.9 MB	##5	3%

如果出现Conda - Downloaded bytes did not match Content-Lengt错误，参考以下文章：

[https://blog.csdn.net/qq\\_46092061/article/details/119141556](https://blog.csdn.net/qq_46092061/article/details/119141556)

如何手动安装cudnn <https://www.zhangshilong.cn/work/36621.html>

conda下载超时怎么办：[https://blog.csdn.net/Arthur\\_Holmes/article/details/105095088](https://blog.csdn.net/Arthur_Holmes/article/details/105095088)

(可能要等好久) 第三方库安装完后，使用active指令切换到名为xxx的虚拟环境中，每个虚拟环境都是独立的，互不影响，这样子可以配置多个开发环境，避免冲突问题 Windows: **activate zqy** Linux: **source**

activate zqy

```
C:\Windows\system32\cmd.exe
Microsoft Windows [版本 10.0.19043.928]
(c) Microsoft Corporation。保留所有权利。

C:\Users\MC>conda env list
# conda environments:
#
base                  *  E:\ANACONDA
myenv_test01          E:\ANACONDA\envs\myenv_test01
zqy                   E:\ANACONDA\envs\zqy

C:\Users\MC>activate zqy


(zqy) C:\Users\MC>
```

## Part 2.算法部署与运行

以超分辨率重建Ai算法为例 github：https://github.com/xinntao/Real-ESRGAN

← → ↻ [github.com/xinntao/Real-ESRGAN](https://github.com/xinntao/Real-ESRGAN)

☰ README.md



Real-ESRGAN  
PRACTICAL RESTORATION

[English](#) | [简体中文](#)

downloads 356k

pypi v0.2.4.0

issues 161 open

issues 86 closed

license BSD-3-Clause

PyLint passing

PyPI Publish passing

🔥🔥🔥 Add small video models for anime videos (针对动漫视频的小模型). Please see [anime video models](#).

1. [Colab Demo](#) for Real-ESRGAN [Open in Colab](#)
2. [Colab Demo](#) for Real-ESRGAN (anime videos) [Open in Colab](#)
3. Portable [Windows](#) / [Linux](#) / [MacOS](#) executable files for Intel/AMD/Nvidia GPU. You can find more information [here](#). The ncnn implementation is in [Real-ESRGAN-ncnn-vulkan](#).

Real-ESRGAN aims at developing Practical Algorithms for General Image/Video Restoration.  
We extend the powerful ESRGAN to a practical restoration application (namely, Real-ESRGAN), which is trained with pure synthetic data.

🙏 Real-ESRGAN needs your contributions. Any contributions are welcome, such as new features/models/typo fixes/suggestions/maintenance, etc. See [CONTRIBUTING.md](#). All contributors are list [here](#).

## 1. 算法环境搭建

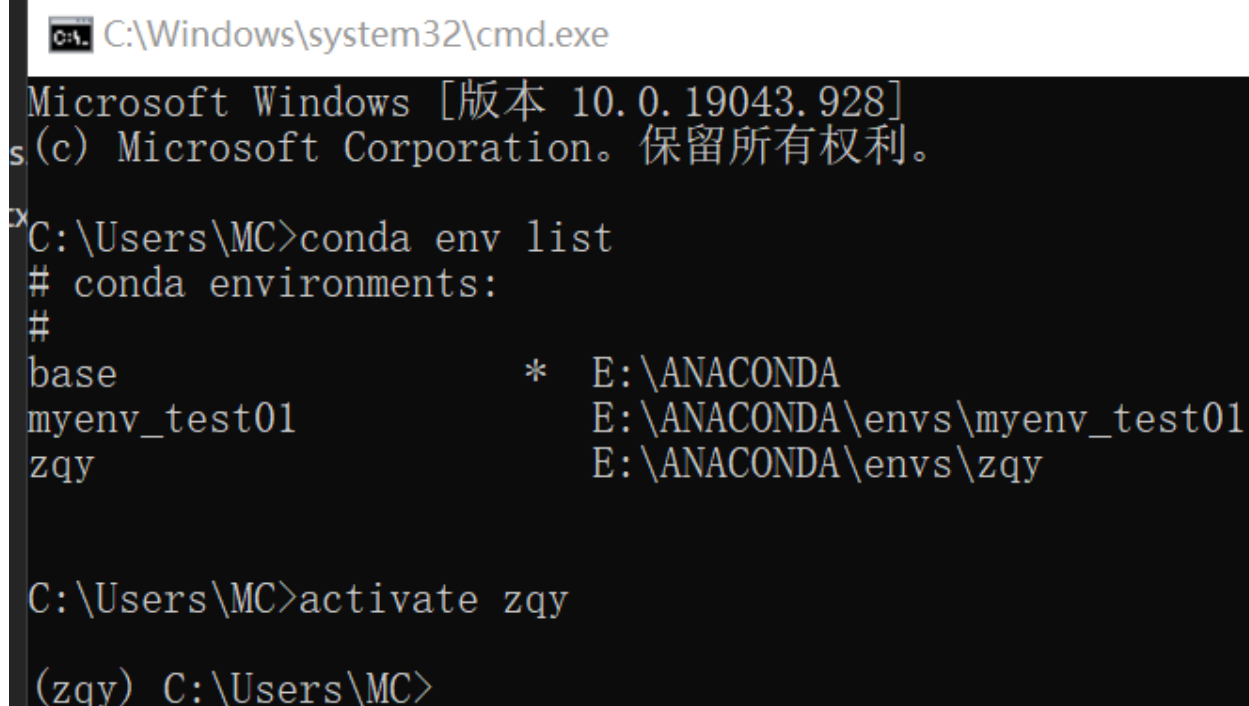
### Step 1. 下载代码包

在<https://github.com/xinntao/Real-ESRGAN>点击download下载 打开requirements.txt文件，里面详细说明了需要的第三方库

```
facexlib>=0.2.0.3 gfpgan>=0.2.1 numpy opencv-python Pillow torch>=1.7 (深度学习框架)
torchvision tqdm
```

### Step 2. 安装pytorch

输入`conda env list`显示环境列表 输入`activate xxx` 切换到要安装的虚拟环境中



```
C:\Windows\system32\cmd.exe
Microsoft Windows [版本 10.0.19043.928]
(c) Microsoft Corporation。保留所有权利。

C:\Users\MC>conda env list
# conda environments:
#
base                  *  E:\ANACONDA
myenv_test01          E:\ANACONDA\envs\myenv_test01
zqy                   E:\ANACONDA\envs\zqy

C:\Users\MC>activate zqy

(zqy) C:\Users\MC>
```

然后打开pytorch官网<https://pytorch.org/get-started/locally/>,根据我们的情况选择所需要的版本

PyTorch. Note that LibTorch is only available for C++.

Additional support or warranty for some PyTorch Stable and LTS binaries are available through the [PyTorch Enterprise Support Program](#).

PyTorch Build	Stable (1.11.0)	Preview (Nightly)	LTS (1.8.2)	
Your OS	Linux	Mac	Windows	
Package	Conda	Pip	LibTorch	Source
Language	Python	C++ / Java		
Compute Platform	CUDA 10.2	CUDA 11.3	ROCm 4.5.2 (beta)	CPU

Run this Command:

```
conda install pytorch torchvision torchaudio cudatoolkit=11.3 -c pytorch
```

然后复制命令`conda install pytorch torchvision torchaudio cudatoolkit=11.3 -c pytorch`安装pytorch 这里建议不要本地安装，不容易找bug

`C:\Windows\system32\cmd.exe - conda install pytorch torchvision torchaudio cudatoolkit=11.3 -c pytorch`

```
mkl_fft          anaconda/pkgs/main/win-64::mkl_fft-1.3.1-py39h277e83a_0
mkl_random       anaconda/pkgs/main/win-64::mkl_random-1.2.2-py39hf11a4ad_0
numpy            anaconda/pkgs/main/win-64::numpy-1.21.5-py39h7a0a035_1
numpy-base      anaconda/pkgs/main/win-64::numpy-base-1.21.5-py39hca35cd5_1
pillow           anaconda/pkgs/main/win-64::pillow-9.0.1-py39hdc2b20a_0
pyopenssl       anaconda/pkgs/main/noarch::pyopenssl-22.0.0-pyhd3eb1b0_0
pysocks         anaconda/pkgs/main/win-64::pysocks-1.7.1-py39haa95532_0
pytorch         pytorch/win-64::pytorch-1.11.0-py3.9_cuda11.3_cudnn8_0
pytorch-mutex   pytorch/noarch::pytorch-mutex-1.0-cuda
requests        anaconda/pkgs/main/noarch::requests-2.27.1-pyhd3eb1b0_0
tk              anaconda/pkgs/main/win-64::tk-8.6.11-h2bbff1b_0
torchaudio      pytorch/win-64::torchaudio-0.11.0-py39_cul13
torchvision     pytorch/win-64::torchvision-0.12.0-py39_cul13
urllib3         anaconda/pkgs/main/win-64::urllib3-1.26.9-py39haa95532_0
win_inet_pton   anaconda/pkgs/main/win-64::win_inet_pton-1.1.0-py39haa95532_0
xz              anaconda/pkgs/main/win-64::xz-5.2.5-h62dcd97_0
zstd            anaconda/pkgs/main/win-64::zstd-1.4.9-h19a0ad4_0
```

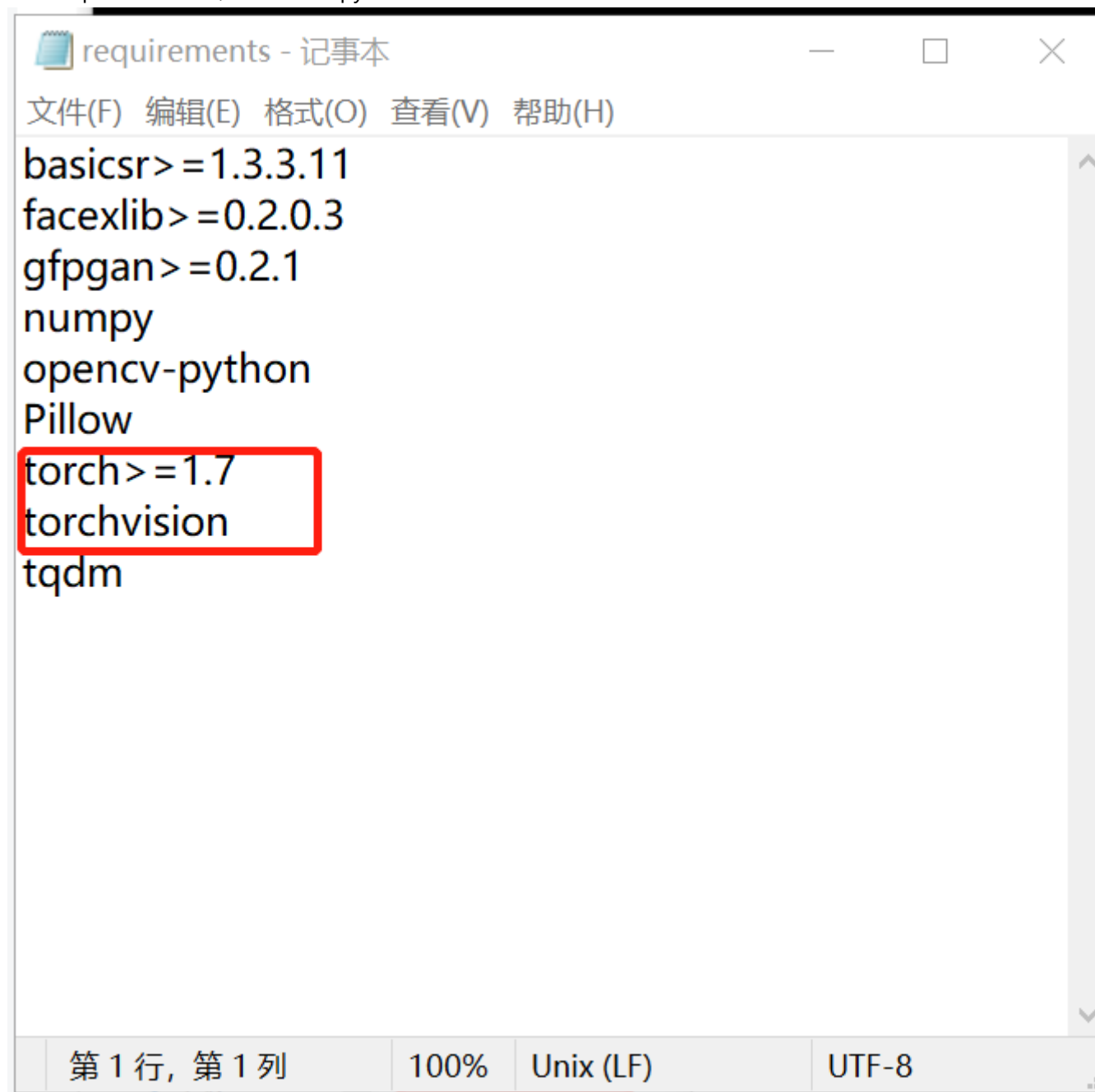
Proceed ([y]/n)? y

Downloading and Extracting Packages

Package	Size	Progress	Percentage
torchaudio-0.11.0	3.1 MB	#####	100%
zstd-1.4.9	478 KB	#####	100%
charset-normalizer-2	35 KB	#####	100%
cryptography-36.0.0	1.0 MB	#####	100%
torchvision-0.12.0	9.1 MB	#####	100%
pysocks-1.7.1	55 KB	#####	100%
libtiff-4.2.0	786 KB	#####	100%
requests-2.27.1	54 KB	#####	100%
libwebp-1.2.2	658 KB	#####	100%
tk-8.6.11	3.3 MB	#####	100%
pillow-9.0.1	921 KB	#####	100%
numpy-1.21.5	25 KB	#####	100%
xz-5.2.5	244 KB	#####	100%
lz4-c-1.9.3	132 KB	#####	100%
pytorch-1.11.0	1.23 GB	#####7	43%

### Step 3.安装requirements.txt中的其他依赖

打开requirements.txt,将安装好的pytorch和torchvision删除掉



```
requirements - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
basicsr>=1.3.3.11
facexlib>=0.2.0.3
gfpgan>=0.2.1
numpy
opencv-python
Pillow
torch>=1.7
torchvision
tqdm

第 1 行, 第 1 列    100%    Unix (LF)    UTF-8
```

输入命令：`python -m pip install -r .\requirements.txt` 下载太慢的话用`pip install -r requirements.txt -i https://pypi.tuna.tsinghua.edu.cn/simple/ --default-timeout=5000` 速度会快很多,参照[换源+修改下载设置时间](#)



```
Downloading https://pypi.tuna.tsinghua.edu.cn/packages/8a/c4/d15f1e627ff
78 kB)
| 78 kB 360 kB/s
collecting addict
Downloading https://pypi.tuna.tsinghua.edu.cn/packages/6a/00/b08f23b7d7e
kB)
collecting future
Downloading https://pypi.tuna.tsinghua.edu.cn/packages/45/0b/38b06fd9b92
829 kB 312 kB/s
collecting lmbd
Downloading https://pypi.tuna.tsinghua.edu.cn/packages/90/fd/788b49864b3
(106 kB)
| 106 kB 128 kB/s
collecting pyyaml
Downloading https://pypi.tuna.tsinghua.edu.cn/packages/08/f4/ffa743f8601
(151 kB)
| 151 kB 327 kB/s
Requirement already satisfied: requests in e:\anaconda\envs\zqy\lib\site-p
collecting scikit-image
Downloading https://pypi.tuna.tsinghua.edu.cn/packages/ec/c3/e151d1f03e7
amd64.whl (12.6 MB)
| 2.0 MB 177 kB/s eta 0:01:00
```

(注意，千万把VPN关掉，不然会报一堆错误)

```
C:\Windows\system32\cmd.exe - conda install pytorch torchvision torchaudio cudatoolkit=11.3 -c pytorch - python -m
resp = conn.urlopen(
File "E:\ANACONDA\envs\zqy\lib\site-packages\pip\_vendor\urllib3\connectionpool.py", line 11
self._prepare_proxy(conn)
File "E:\ANACONDA\envs\zqy\lib\site-packages\pip\_vendor\urllib3\connectionpool.py", line 11
conn.connect()
File "E:\ANACONDA\envs\zqy\lib\site-packages\pip\_vendor\urllib3\connection.py", line 3
conn = self._connect_tls_proxy(hostname, conn)
File "E:\ANACONDA\envs\zqy\lib\site-packages\pip\_vendor\urllib3\connection.py", line 5
return ssl_wrap_socket(
File "E:\ANACONDA\envs\zqy\lib\site-packages\pip\_vendor\urllib3\util\ssl_.py", line 45
ssl_sock = _ssl_wrap_socket_impl(sock, context, tls_in_tls)
File "E:\ANACONDA\envs\zqy\lib\site-packages\pip\_vendor\urllib3\util\ssl_.py", line 49
return ssl_context.wrap_socket(sock)
File "E:\ANACONDA\envs\zqy\lib\ssl.py", line 500, in wrap_socket
return self.sslsocket_class._create(
File "E:\ANACONDA\envs\zqy\lib\ssl.py", line 997, in _create
raise ValueError("check_hostname requires server_hostname")
ValueError: check_hostname requires server_hostname
```

## 解决pip install -r requirements.txt很慢的方案

```
(zqy) E:\Github_local_test\Code\Real-ESRGAN-master>python -m pip install -r .\requirements.txt
Collecting basicsr>=1.3.3.11
  Downloading basicsr-1.3.5.tar.gz (161 kB)
    |#####| 161 kB 65 kB/s
Collecting facexlib>=0.2.0.3
  Downloading facexlib-0.2.2-py3-none-any.whl (59 kB)
    |#####| 59 kB 26 kB/s
Collecting gfpgan>=0.2.1
  Downloading gfpgan-1.3.2-py3-none-any.whl (47 kB)
    |#####| 47 kB 15 kB/s
Requirement already satisfied: numpy in e:\anaconda\envs\zqy\lib\site-packages (from -r .\requirements.txt (line 4)) (1.21.5)
Collecting opencv-python
  Using cached opencv_python-4.5.5.64-cp36-abi3-win_amd64.whl (35.4 MB)
Requirement already satisfied: Pillow in e:\anaconda\envs\zqy\lib\site-packages (from -r .\requirements.txt (line 6)) (9.0.1)
Collecting tqdm
  Downloading tqdm-4.64.0-py2.py3-none-any.whl (78 kB)
    |#####| 78 kB 12 kB/s
Collecting addict
  Downloading addict-2.4.0-py3-none-any.whl (3.8 kB)
Collecting future
  Downloading future-0.18.2.tar.gz (829 kB)
    |#####| 829 kB 61 kB/s
Collecting lmbd
  Downloading lmbd-1.3.0-cp39-cp39-win_amd64.whl (106 kB)
    |#####| 106 kB 94 kB/s
Collecting pyyaml
  Downloading PyYAML-6.0-cp39-cp39-win_amd64.whl (151 kB)
    |#####| 151 kB 19 kB/s
Requirement already satisfied: requests in e:\anaconda\envs\zqy\lib\site-packages (from basicsr>=1.3.3.11->-r .\requirements.txt)
Collecting scikit-image
  Downloading scikit_image-0.19.2-cp39-cp39-win_amd64.whl (12.6 MB)
    |#####| 6.5 MB 60 kB/s eta 0:01:43
```

至此，算法依赖环境搭建完成

```
72606
Successfully built basicsr filterpy future
Installing collected packages: pyasn1, zipp, rsa, pyasn1-modules, oauthlib, cachetools, requests-oauthlib, numpy, kiwisolver, importlib-metadata, google-auth, fonttools, cyclr, werkzeug, tifffile, tensorboard-plugin-wit, tensorboard-data-server, scipy, PyWavelets, protobuf, networkx, matplotlib, markdown, llvmlite, imageio, grpcio, google-auth-oauthlib, abl-py, yapf, tqdm, tb-nightly, scikit-image, pyyaml, opencv-python, numba, lmbd, future, filterpy, addict, facexlib, basicsr, gfpgan
  Attempting uninstall: numpy
    Found existing installation: numpy 1.21.5
    Uninstalling numpy-1.21.5:
```

## 2.算法部署与运行

README\_CN.md中记录了运行的方法



快速上手

**普通图片**

下载我们训练好的模型: [RealESRGAN\\_x4plus.pth](#)

```
wget https://github.com/xinntao/Real-ESRGAN/releases/download/v0.1.0/RealESRGAN_x4plus.pth -P experiments/pretrained_models
```

推断!

```
python inference_realesrgan.py -n RealESRGAN_x4plus -i inputs --face_enhance
```

结果在 `results` 文件夹

**动画图片**

先下载好训练完成的模型[RealESRGAN\\_x4plus.pth](#)地址



将模型放到工程目录的experiments/pretrained\_models目录下

脑 > 本地磁盘 (E:) > Github\_local\_test > Code > Real-ESRGAN-master > experiments > pretrained\_models

名称	修改日期	类型	大小
README	2022/2/23 4:44	Markdown 源文件	1 KB
RealESRGAN_x4plus.pth	2022/4/22 16:48	PTH 文件	65,470 KB

输入python inference\_realesrgan.py -n RealESRGAN\_x4plus -i inputs --face\_enhance测试一下，结果会在results文件夹中 会出现以下错误

```
(zqy) E:\Github_local_test\Code\Real-ESRGAN-master>python inference_realesrgan.py -n RealESRGAN_x4plus -i inputs --face_enhance
Traceback (most recent call last):
  File "E:\Github_local_test\Code\Real-ESRGAN-master\inference_realesrgan.py", line 7, in <module>
    from realesrgan import RealESRGANer
  File "E:\Github_local_test\Code\Real-ESRGAN-master\realesrgan\__init__.py", line 6, in <module>
    from .version import *
ModuleNotFoundError: No module named 'realesrgan.version'
```

说明包导入有问题，看下本地目录，找到realesrgan文件夹，打开\_\_init\_\_.py文件 把from.version import \*注释掉

此电脑 > 本地磁盘 (E:) > Github\_local\_test > Code > Real-ESRGAN-master > realesrgan

名称	修改日期	类型	大小
__pycache__	2022/4/24 14:56	文件夹	
archs	2022/4/24 14:56	文件夹	
data	2022/4/24 14:56	文件夹	
models	2022/4/24 14:56	文件夹	
weights	2022/2/23 4:44	文件夹	
__init__	2022/2/23 4:44	Python File	1 KB
train	2022/2/23 4:44	Python File	1 KB
utils	2022/2/23 4:44	Python File	12 KB

\* \_\_init\_\_ - 记事本

文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

```
# flake8: noqa
from .archs import *
from .data import *
from .models import *
from .utils import *
# from .version import *
```

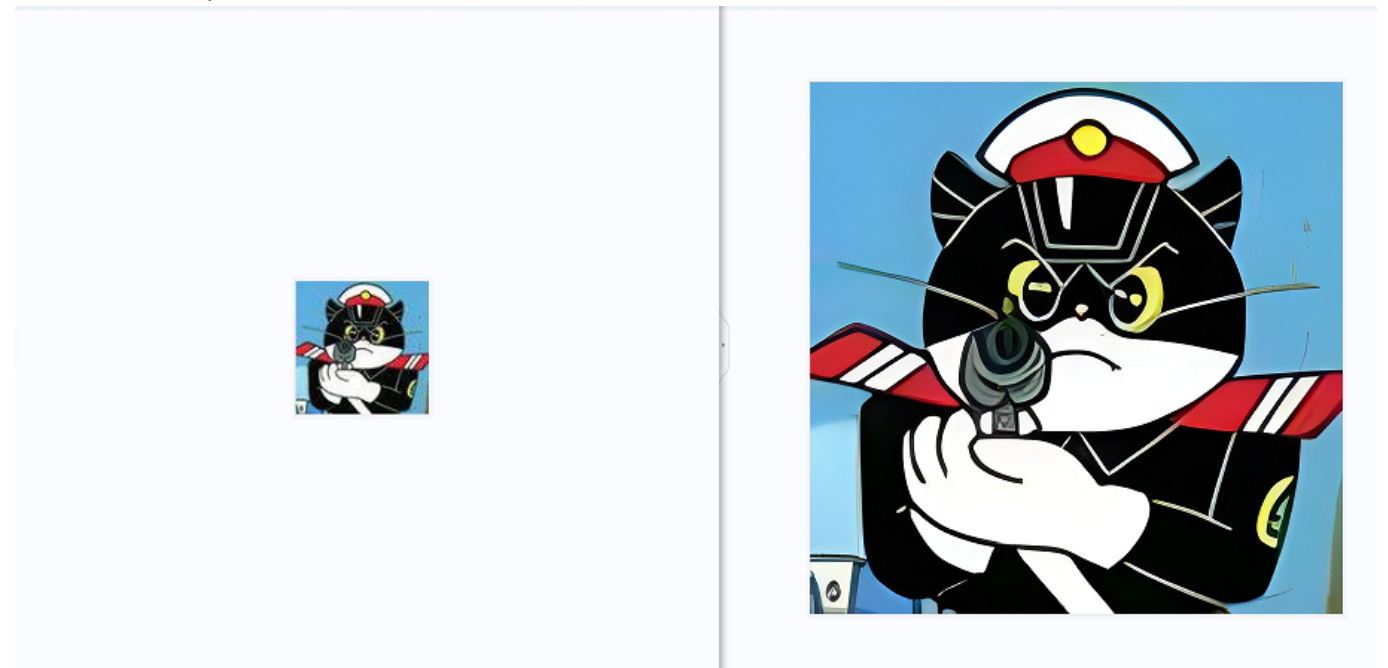
等待下载其余资源后，即可以运行

```
Testing 0 00003
Testing 1 0014
Testing 2 0030
Testing 3 ADE_val_00000114
Testing 4 OST_009
Error CUDA out of memory. Tried to allocate 1.09 GiB (GPU 0; 4.00 GiB total capacity; 870.69 MiB a
MiB free; 1.58 GiB reserved in total by PyTorch) If reserved memory is >> allocated memory try se
to avoid fragmentation. See documentation for Memory Management and PYTORCH_CUDA_ALLOC_CONF
If you encounter CUDA out of memory, try to set --tile with a smaller number.
Testing 5 tree_alpha_16bit
    Input is a 16-bit image
Testing 6 video
[ WARN:0@234.232] global D:\a\opencv-python\opencv-python\opencv\modules\imgcodecs\src\loadsave.cp
imread_('inputs\video'): can't open/read file: check file path/integrity
Traceback (most recent call last):
  File "E:\Github_local_test\Code\Real-ESRGAN-master\inference_realesrgan.py", line 128, in <modul
    main()
  File "E:\Github_local_test\Code\Real-ESRGAN-master\inference_realesrgan.py", line 103, in main
    if len(img.shape) == 3 and img.shape[2] == 4:
AttributeError: 'NoneType' object has no attribute 'shape'
```

在目录下会生成results文件夹

realesrgan	2022/4/24 14:56	文件夹	
results	2022/4/24 15:07	文件夹	
scripts	2022/2/23 4:44	文件夹	
tests	2022/2/23 4:44	文件夹	
.gitignore	2022/2/23 4:44	Git Ignore 源文件	2
.pre-commit-config	2022/2/23 4:44	Yaml 源文件	2

结果展示：将inputs和results文件夹图片对比如下





[参考](https://github.com/xinntao/Real-ESRGAN) Real-ESRGAN项目：<https://github.com/xinntao/Real-ESRGAN>