

通用物体检测



主讲人 张士峰

中国科学院自动化研究所
模式识别国家重点实验室博士





目录

-  物体检测环境配置
-  通用物体检测概述
-  基于锚框的检测算法
-  无需锚框的检测算法
-  物体检测算法的对比总结
-  实用检测算法的研究思路



目录



物体检测环境配置



通用物体检测概述



基于锚框的检测算法



无需锚框的检测算法



物体检测算法的对比总结

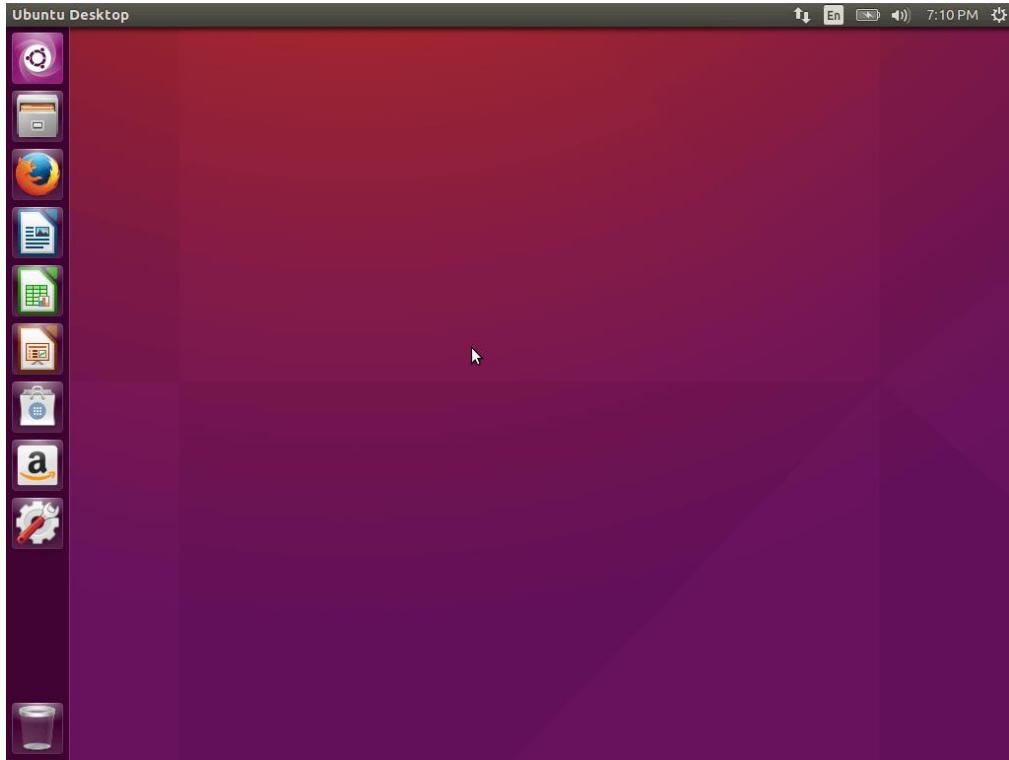


实用检测算法的研究思路



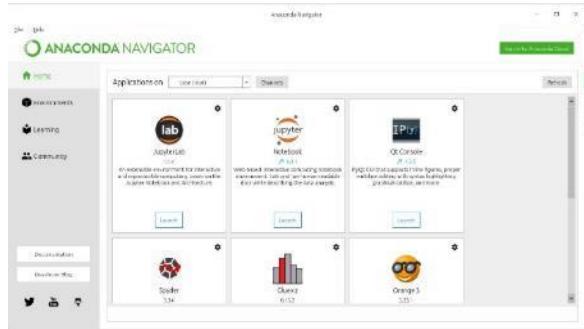
物体检测环境配置

➤ Ubuntu系统 (16.04) + 英伟达显卡 (TITAN X)





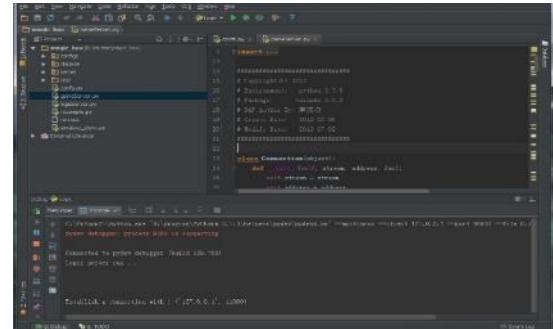
物体检测环境配置



环境管理软件Anaconda



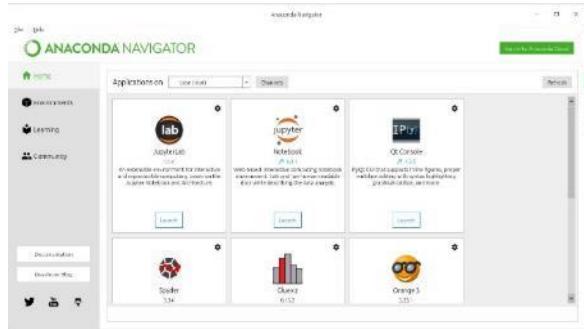
物体检测平台Detectron2



代码调试软件PyCharm



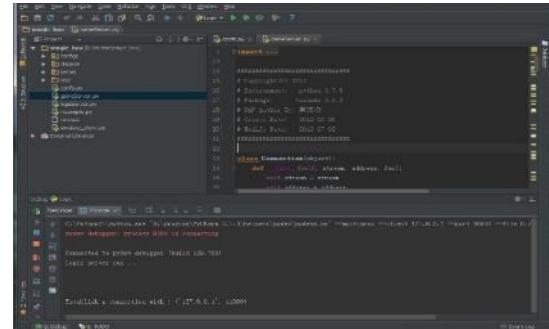
物体检测环境配置



环境管理软件Anaconda



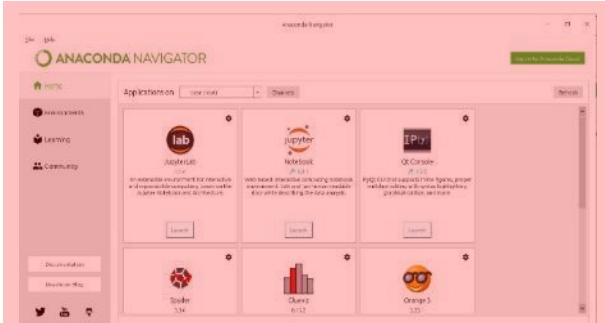
物体检测平台Detectron2



代码调试软件PyCharm



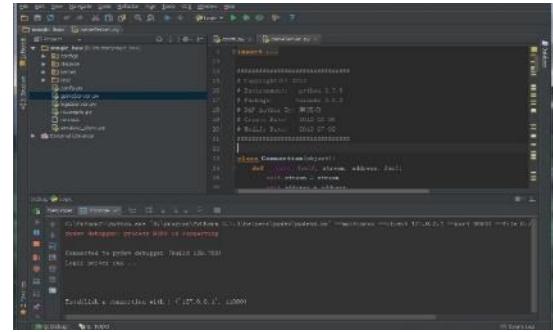
物体检测环境配置



环境管理软件Anaconda



物体检测平台Detectron2

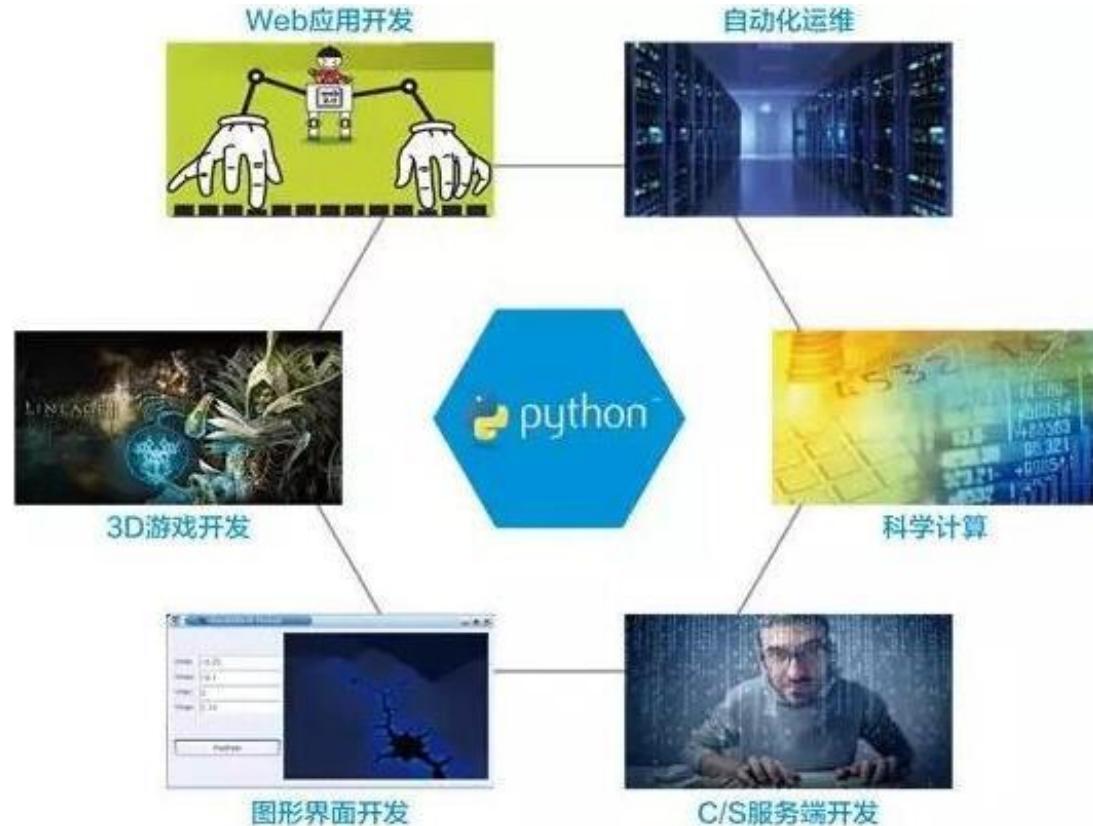


代码调试软件PyCharm



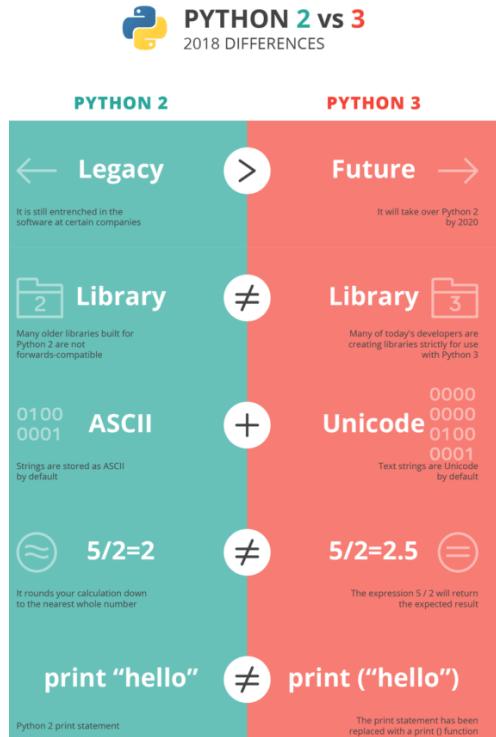
编程语言Python简介

- Python编程语言
- 深度学习最常用语言
- 各行各业都在用Python
- 各式各样的包
- PyTorch
- TensorFlow
- MXNet





环境管理软件Anaconda



Python有不同的版本

如何让一个系统有不同的Python环境?

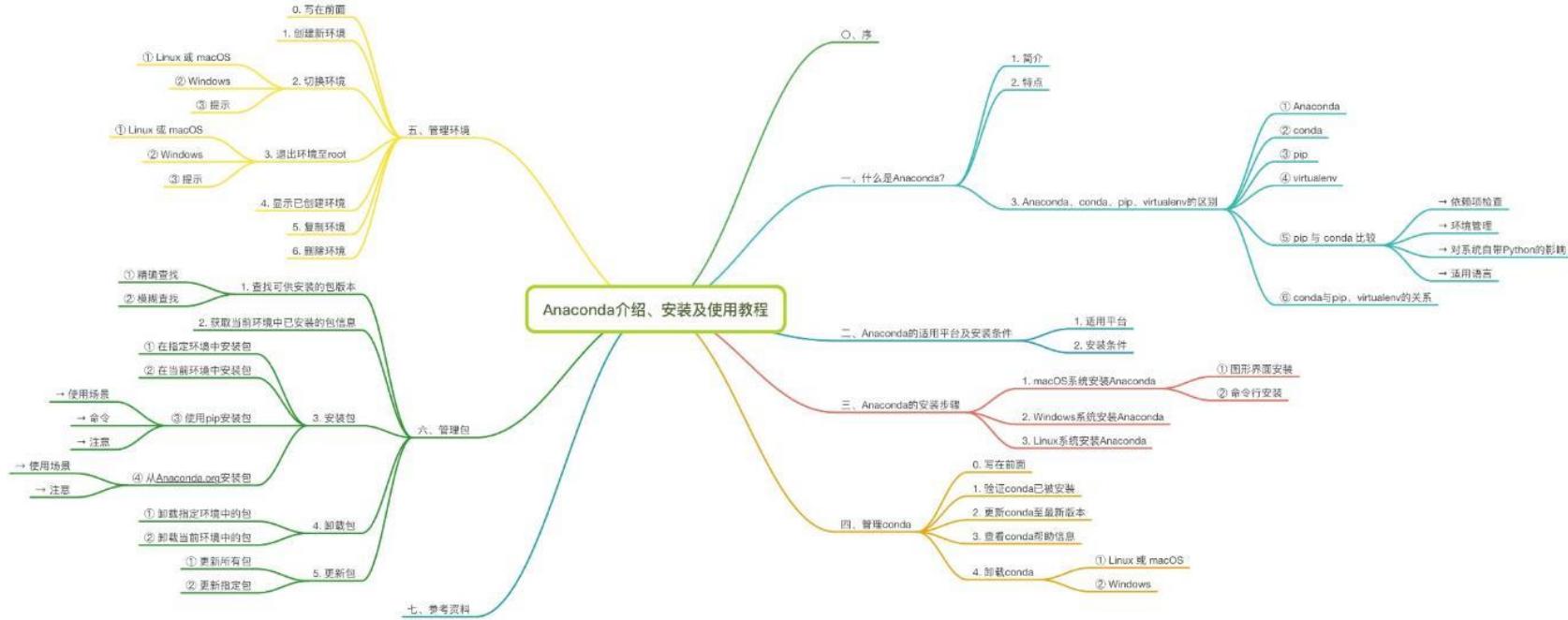
PyTorch Build	Stable (1.5)	Preview (Nightly)
Your OS	Linux	Mac Windows
Package	Conda	Pip LibTorch Source
Language	Python	C++ / Java
CUDA	9.2	10.1 10.2 None
Run this Command:	<pre>conda install pytorch torchvision cudatoolkit=10.2 -c pytorch</pre>	

PyTorch有不同的版本



环境管理软件Anaconda

- Anaconda可以便捷获取包且能够对包进行管理，同时对环境可以统一管理的发行版本
- Anaconda包含了conda、Python在内的超过180个科学包及其依赖项
- 推荐教程：Anaconda介绍、安装及使用教程 (<https://zhuanlan.zhihu.com/p/32925500>, 下图来自该教程)





Anaconda的安装

- 前往官方下载页面下载。有两个版本可供选择：Python 3.x 和 Python 2.x

The screenshot shows a web browser window with the title "Individual Edition | Anaconda" and the URL "anaconda.com/products/individual". The main content is titled "Anaconda Installers". It lists download options for three operating systems:

- Windows**:
 - Python 3.7:
 - 64-Bit Graphical Installer (466 MB)
 - 32-Bit Graphical Installer (423 MB)
 - Python 2.7:
 - 64-Bit Graphical Installer (413 MB)
 - 32-Bit Graphical Installer (356 MB)
- MacOS**:
 - Python 3.7:
 - 64-Bit Graphical Installer (442 MB)
 - 64-Bit Command Line Installer (430 MB)
 - Python 2.7:
 - 64-Bit Graphical Installer (637 MB)
 - 64-Bit Command Line Installer (409 MB)
- Linux**:
 - Python 3.7:
 - 64-Bit (x86) Installer (522 MB)
 - 64-Bit (Power8 and Power9) Installer (276 MB)
 - Python 2.7:
 - 64-Bit (x86) Installer (477 MB)
 - 64-Bit (Power8 and Power9) Installer (295 MB)



Anaconda的安装

- 启动终端找到安装文件
- 输入命令`md5sum filename` 检查文件

- 输入命令`bash filename` 安装文件
- 敲回车键查看许可证协议
- 敲Ctrl+C退出查看许可证协议
- 输入`yes`同意许可证协议
- 敲回车选择默认路径，否则输入指定路径
- 输入`yes`添加Anaconda路径，完成安装

```
root@cbsr219:~/Downloads# ls  
Anaconda3-2020.02-Linux-x86_64.sh
```

```
root@cbsr219:~/Downloads# md5sum Anaconda3-2020.02-Linux-x86_64.sh  
17600d1f12b2b047b62763221f29f2bc Anaconda3-2020.02-Linux-x86_64.sh
```

```
root@cbsr219:~/Downloads# bash Anaconda3-2020.02-Linux-x86_64.sh  
Welcome to Anaconda3 2020.02  
  
In order to continue the installation process, please review the license  
agreement.  
Please, press ENTER to continue  
>>>  
Notice of Third Party Software Licenses  
=====  
  
Do you accept the license terms? [yes|no]  
[no] >>> yes  
Anaconda3 will now be installed into this location:  
/home/shifeng/anaconda3  
  
- Press ENTER to confirm the location  
- Press CTRL-C to abort the installation  
- Or specify a different location below  
  
[/home/shifeng/anaconda3] >>> /home/shifeng/Software/anaconda3  
Do you wish the installer to initialize Anaconda3  
by running conda init? [yes|no]  
[no] >>> yes
```



Anaconda的安装

- 启动终端找到安装文件
- 输入命令`md5sum filename` 检查文件

- 输入命令`bash filename` 安装文件
- 敲回车键查看许可证协议
- 敲Ctrl+C退出查看许可证协议
- 输入`yes`同意许可证协议
- 敲回车选择默认路径，否则输入指定路径
- 输入`yes`添加Anaconda路径，完成安装

```
root@cbsr219:~/Downloads# ls  
Anaconda3-2020.02-Linux-x86_64.sh
```

```
root@cbsr219:~/Downloads# md5sum Anaconda3-2020.02-Linux-x86_64.sh  
17600d1f12b2b047b62763221f29f2bc Anaconda3-2020.02-Linux-x86_64.sh
```

```
root@cbsr219:~/Downloads# bash Anaconda3-2020.02-Linux-x86_64.sh  
Welcome to Anaconda3 2020.02  
  
In order to continue the installation process, please review the license  
agreement.  
Please, press ENTER to continue  
>>>  
Notice of Third Party Software Licenses  
=====
```

```
Do you accept the license terms? [yes|no]  
[no] >>> yes
```

```
Anaconda3 will now be installed into this location:  
/home/shifeng/anaconda3
```

```
- Press ENTER to confirm the location  
- Press CTRL-C to abort the installation  
- Or specify a different location below
```

```
[/home/shifeng/anaconda3] >>> /home/shifeng/Software/anaconda3
```

```
Do you wish the installer to initialize Anaconda3  
by running conda init? [yes|no]  
[no] >>> yes
```



Anaconda的安装

- 关闭终端，再打开终端以启动Anaconda
- 终端左边出现base，说明安装成功
- base即Anaconda的默认环境
- 输入命令`conda list`，查看已安装的包
- Anaconda的默认环境安装了几十个包
- 名称、版本、build、channel
- conda：管理不同的包和环境
- python：启动Python交互界面

```
(base) root@cbsr219:~# conda list
# packages in environment at /home/shifeng/Software/anaconda3:
#
# Name           Version        Build  Channel
_ipyw_jlab_nb_ext_conf  0.1.0          py3_0  defaults
_libgcc_mutex          0.1            main   defaults
alabaster             0.7.12         py37_0  defaults
anaconda              2020.02        py37_0  defaults
anaconda-client        1.7.2          py37_0  defaults
anaconda-navigator    1.9.12         py37_0  defaults
anaconda-project      0.8.4          py_0    defaults
argh                  0.26.2         py37_0  defaults
asn1crypto             1.3.0          py37_0  defaults
astroid                2.3.3          py37_0  defaults
astropy                4.0            py37h7b6447c_0 defaults
atomicwrites           1.3.0          py37_1  defaults
attrs                 19.3.0         py_0    defaults
autopep8               1.4.4          py_0    defaults
colorama               0.4.3          py_0    defaults
conda                 4.8.2          py37_0  defaults
conda-build             3.18.11        py37_0  defaults
conda-env               2.6.0          py_1    defaults
conda-package-handling 1.6.0          py37h7b6447c_0 defaults
conda-verify             3.4.2          py_1    defaults
pytest                 5.3.5          py37_0  defaults
pytest-arraydiff        0.3            py37h39e3cac_0 defaults
pytest-astropy           0.8.0          py_0    defaults
pytest-astropy-header   0.1.2          py_0    defaults
pytest-doctestplus      0.5.0          py_0    defaults
pytest-openfiles         0.4.0          py_0    defaults
pytest-remotedata       0.3.2          py37_0  defaults
python                 3.7.6          h0371630_2 defaults
python-dateutil          2.8.1          py_0    defaults
python-jsonrpc-server   0.3.4          py_0    defaults
python-language-server  0.31.7         py37_0  defaults
python-libarchive-c     2.8            py37_13 defaults
```



Anaconda的使用教程

- 查看现存的环境: conda env list

```
root@cbsr219:~#
File Edit View Search Terminal Help
(base) root@cbsr219:~# conda env list
# conda environments:
#
base          * /home/shifeng/Software/anaconda3

(base) root@cbsr219:~#
```



Anaconda的使用教程

- 创建新的环境: conda create --name <env_name> <package_names>

```
x  root@cbsr219:~ 
File Edit View Search Terminal Help
(base) root@cbsr219:~# conda create --name test python=3.6
Collecting package metadata (current_repodata.json): done
Solving environment: done
## Package Plan ##

environment location: /home/shifeng/Software/anaconda3/envs/test

added / updated specs:
- python=3.6

The following packages will be downloaded:

```

package	build	size	url
_libgcc_mutex-0.1	main	3 KB	https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
ca-certificates-2020.1.1	0	125 KB	https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
certifi-2020.4.5.1	py36_0	155 KB	https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
ld_impl_linux-64-2.33.1	h53a641e_7	568 KB	https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
libedit-3.1.20181209	hc058e9b_0	163 KB	https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
libffi-3.3	he6710b0_1	50 KB	https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
libgcc-ng-9.1.0	hdf63c60_0	5.1 MB	https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
libstdcxx-ng-9.1.0	hdf63c60_0	3.1 MB	https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
ncurses-6.2	he6710b0_1	817 KB	https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
openssl-1.1.1g	h7b6447c_0	2.5 MB	https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
pip-20.0.2	py36_3	1.7 MB	https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
python-3.6.10	h7579374_2	29.7 MB	https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main

```
(base) root@cbsr219:~# conda env list
# conda environments:
#
base          * /home/shifeng/Software/anaconda3
test          /home/shifeng/Software/anaconda3/envs/test
```



Anaconda的使用教程

- 切换到指定环境: conda activate <env_name>

```
x - root@cbsr219: ~
File Edit View Search Terminal Help
(base) root@cbsr219:~# conda env list
# conda environments:
#
base          * /home/shifeng/Software/anaconda3
test

(base) root@cbsr219:~# conda activate test
(test) root@cbsr219:~# conda list
# packages in environment at /home/shifeng/Software/anaconda3/envs/test:
#
# Name      Version   Build  Channel
_libgcc_mutex    0.1        main
ca-certificates  2020.1.1       0    https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
certifi         2020.4.5.1      py36_0  https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
ld_impl_linux-64 2.33.1     h53a641e_7
libedit         3.1.20181209  hc05e9b_0
libffi           3.3        he6710b0_1
libgcc-ng        9.1.0      hdf63c60_0
libstdcxx-ng     9.1.0      hdf63c60_0
ncurses          6.2        he6710b0_1
openssl         1.1.1g     h7b6447c_0
pip              20.0.2      py36_3
python           3.6.10     h7579374_2
readline         8.0        h7b6447c_0
setuptools       46.4.0      py36_0
sqlite           3.31.1     h62c20be_1
tk               8.6.8      hbc83047_0
wheel            0.34.2      py36_0
xz               5.2.5      h7b6447c_0
zlib             1.2.11     h7b6447c_3
(test) root@cbsr219:~#
```



Anaconda的使用教程

- 退出当前环境: conda deactivate

```
root@cbsr219:~#
File Edit View Search Terminal Help
(base) root@cbsr219:~# conda env list
# conda environments:
#
base          * /home/shifeng/Software/anaconda3
test          /home/shifeng/Software/anaconda3/envs/test

(base) root@cbsr219:~# conda activate test
(test) root@cbsr219:~# conda deactivate
(base) root@cbsr219:~# conda deactivate
root@cbsr219:~#
```

A screenshot of a terminal window titled "root@cbsr219:~". The window shows a command-line interface with several lines of text. The first few lines show the output of the "conda env list" command, listing two environments: "base" and "test". The "base" environment is marked with an asterisk (*) and its path is "/home/shifeng/Software/anaconda3". The "test" environment is listed below it with its path. Following this, three commands are entered sequentially: "conda activate test", "conda deactivate", and another "conda deactivate". The last "conda deactivate" command is highlighted with a red rectangular box. The terminal window has a dark background and light-colored text.



Anaconda的使用教程

- 复制环境: conda create --name <new_env_name> --clone <copied_env_name>

```
root@cbsr219: ~
File Edit View Search Terminal Help
(base) root@cbsr219:~# conda env list
# conda environments:
#
base          * /home/shifeng/Software/anaconda3
test          /home/shifeng/Software/anaconda3/envs/test

(base) root@cbsr219:~# conda create --name copy --clone test
Source:      /home/shifeng/Software/anaconda3/envs/test
Destination: /home/shifeng/Software/anaconda3/envs/copy
Packages: 19
Files: 0
Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#     $ conda activate copy
#
# To deactivate an active environment, use
#
#     $ conda deactivate

(base) root@cbsr219:~# conda env list
# conda environments:
#
base          * /home/shifeng/Software/anaconda3
copy          /home/shifeng/Software/anaconda3/envs/copy
test          /home/shifeng/Software/anaconda3/envs/test
```



Anaconda的使用教程

- 删除环境: conda remove --name <env_name> --all

```
root@cbsr219:~  
File Edit View Search Terminal Help  
(base) root@cbsr219:~# conda env list  
# conda environments:  
#  
base          * /home/shifeng/Software/anaconda3  
test           /home/shifeng/Software/anaconda3/envs/test  
  
(base) root@cbsr219:~# conda remove --name test --all  
Remove all packages in environment /home/shifeng/Software/anaconda3/envs/test:  
  
## Package Plan ##  
  
environment location: /home/shifeng/Software/anaconda3/envs/test  
  
The following packages will be REMOVED:  
  
_libgcc_mutex-0.1-main  
ca-certificates-2020.1.1.0  
certifi-2020.4.5.1-py36_0  
ld_impl_linux-64-2.33.1-h53a641e_7  
libedit-3.1.20181209-hc058e9b_0  
libffi-3.3-he6710b0_1  
libgcc-ng-9.1.0-hdf63c60_0  
libstdcxx-ng-9.1.0-hdf63c60_0  
ncurses-6.2-he6710b0_1  
openssl-1.1.1g-h7b6447c_0  
pip-20.0.2-py36_3  
python-3.6.10-h7579374_2  
readline-8.0-hb6447c_0  
setuptools-46.4.0-py36_0  
sqlite-3.31.1-h62c20be_1  
tk-8.6.8-hbc83047_0  
wheel-0.34.2-py36_0  
xz-5.2.5-h7b6447c_0  
zlib-1.2.11-h7b6447c_3  
  
Proceed ([y]/n)? y  
Preparing transaction: done  
Verifying transaction: done  
Executing transaction: done  
(base) root@cbsr219:~# conda env list  
# conda environments:  
#  
base          * /home/shifeng/Software/anaconda3
```



Anaconda的使用教程

■ 创建detectron2环境

Requirements

- Linux or macOS with Python ≥ 3.6
- PyTorch ≥ 1.4
- [torchvision](#) that matches the PyTorch installation. You can install them together at [pytorch.org](#) to make sure of this.
- OpenCV, optional, needed by demo and visualization
- pycocotools: `pip install cython; pip install -U
'git+https://github.com/cocodataset/cocoapi.git#subdirectory=PythonAPI'`

<https://github.com/facebookresearch/detectron2/blob/master/INSTALL.md>



Anaconda的使用教程

■ 创建detectron2环境

Requirements

- Linux or macOS with Python ≥ 3.6
- PyTorch ≥ 1.4
- `torchvision` that matches the PyTorch installation. You can install them together at pytorch.org to make sure of this.
- OpenCV, optional, needed by demo and visualization
- pycocotools: `pip install cython; pip install -U 'git+https://github.com/cocodataset/cocoapi.git#subdirectory=PythonAPI'`

```
(base) root@cbsr219:~# conda create --name detectron2 python=3.6
Collecting package metadata (current_repodata.json): done
Solving environment: done
```

```
(base) root@cbsr219:~# conda env list
# conda environments:
#
base                  * /home/shifeng/Software/anaconda3
detectron2            /home/shifeng/Software/anaconda3/envs/detectron2

(base) root@cbsr219:~# conda activate detectron2
(detectron2) root@cbsr219:~# █
```



Anaconda的使用教程

■ 创建detectron2环境

Requirements

- Linux or macOS with Python ≥ 3.6
- PyTorch ≥ 1.4
- [torchvision](#) that matches the PyTorch installation. You can install them together at pytorch.org to make sure of this.
- OpenCV, optional, needed by demo and visualization
- pycocotools: `pip install cython; pip install -U 'git+https://github.com/cocodataset/cocoapi.git#subdirectory=PythonAPI'`

PyTorch Build	Stable (1.5)	Preview (Nightly)		
Your OS	Linux	Mac	Windows	
Package	Conda	Pip	LibTorch	Source
Language	Python			
CUDA	9.2	10.1	10.2	None
Run this Command:	<code>conda install pytorch torchvision cudatoolkit=10.1 -c pytorch</code>			

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



Anaconda的使用教程

■ 创建detectron2环境

Requirements

- Linux or macOS with Python ≥ 3.6
- PyTorch ≥ 1.4
- [torchvision](#) that matches the PyTorch installation. You can install them together at pytorch.org to make sure of this.
- OpenCV, optional, needed by demo and visualization
- pycocotools: `pip install cython; pip install -U 'git+https://github.com/cocodataset/cocoapi.git#subdirectory=PythonAPI'`

```
(detectron2) root@cbsr219:~# conda install pytorch torchvision cudatoolkit=10.1 -c pytorch
Collecting package metadata (current_repodata.json): done
Solving environment: done
The following packages will be downloaded:

  package          | build
cudatoolkit-10.1.243 | h6bb024c_0    347.4 MB  https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
freetype-2.9.1        | h8a8880c_1    550 KB   https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
intel-openmp-2020.1   | 217           780 KB   https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
jpeg-9b               | h024ee3a_2    214 KB   https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
libgfortran-ng-7.3.0  | hdf63c60_0    1006 KB  https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
mkl-2020.1            | 217           129.0 MB  https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
mkl-service-2.3.0     | py36he904bf0_0 219 KB   https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
mkl_fft-1.0.15         | py36ha843d7b_0 155 KB   https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
mkl_random-1.1.1       | py36h0573a6f_0 327 KB   https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
ninja-1.9.0             | py36hfd86e86_0 1.2 MB   https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
numpy-1.18.1            | py36h4f9e942_0  5 KB    https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
numpy-base-1.18.1       | py36hdes5b4d6_1 4.2 MB   https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
olefile-0.46              | py36_0          48 KB   https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
pillow-7.1.2             | py36hb39fc3d_0  604 KB  https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
pytorch-1.5.0             | pv3.6_cuda10.1.243_cudnn7.6.3_0 399.5 MB  pytorch
six-1.14.0                | py36_0          27 KB   https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
torchvision-0.6.0          | py36_cu101      11.8 MB  pytorch
```



Anaconda的使用教程

■ 创建detectron2环境

Requirements

- Linux or macOS with Python ≥ 3.6
- PyTorch ≥ 1.4
- [torchvision](#) that matches the PyTorch installation. You can install them together at [pytorch.org](#) to make sure of this.
- OpenCV, optional, needed by demo and visualization
- pycocotools: pip install cython; pip install -U
`'git+https://github.com/cocodataset/cocoapi.git#subdirectory=PythonAPI'`

```
(detectron2) root@cbsr219:~# python
Python 3.6.10 |Anaconda, Inc.| (default, May  8 2020, 02:54:21)
[GCC 7.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import torch
>>> import torchvision
>>> torch.__version__
'1.5.0'
>>> torchvision.__version__
'0.6.0a0+82fd1c8'
>>> exit()
```



Anaconda的使用教程

■ 创建detectron2环境

Requirements

- Linux or macOS with Python ≥ 3.6
- PyTorch ≥ 1.4
- [torchvision](#) that matches the PyTorch installation. You can install them together at [pytorch.org](#) to make sure of this.
- OpenCV, optional, needed by demo and visualization
- pycocotools: pip install cython; pip install -U
`'git+https://github.com/cocodataset/cocoapi.git#subdirectory=PythonAPI'`

```
(detectron2) root@cbsr219:~# conda install -c https://conda.anaconda.org/menpo opencv3
Collecting package metadata (current_repodata.json): done
Solving environment: done
The following packages will be downloaded:

  package          | build
  opencv3-3.1.0    | py36_0      37.4 MB  menpo
                                         Total: 37.4 MB

(detectron2) root@cbsr219:~# python
Python 3.6.10 |Anaconda, Inc.| (default, May  8 2020, 02:54:21)
[GCC 7.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import cv2
>>> 
```



Anaconda的使用教程

■ 创建detectron2环境

Requirements

- Linux or macOS with Python ≥ 3.6
- PyTorch ≥ 1.4
- [torchvision](#) that matches the PyTorch installation. You can install them together at pytorch.org to make sure of this.
- OpenCV, optional, needed by demo and visualization
- pycocotools: `pip install cython; pip install -U 'git+https://github.com/cocodataset/cocoapi.git#subdirectory=PythonAPI'`

```
(detectron2) root@cbsr219:~# pip install cython; pip install -U 'git+https://github.com/cocodataset/cocoapi.git#subdirectory=PythonAPI'
WARNING: The directory '/home/shifeng/.cache/pip' is not owned or is not writable by the current user. The cache has been disabled. Check the permissions and owner of that directory. If executing pip with sudo, you may want sudo's -H flag.
Looking in indexes: https://pypi.tuna.tsinghua.edu.cn/simple
Collecting cython
  Downloading https://pypi.tuna.tsinghua.edu.cn/packages/bb/37/13249e34he215c30b4dbfc0b90c5ab719a01b2522820e0fcf3035ce130f/cython-0.29.19-cp36-cp36-manylinux1_x86_64.whl (2.0 MB)
Installing collected packages: cython
Successfully installed cython-0.29.19
Collecting PythonAPI
  Downloading https://pypi.tuna.tsinghua.edu.cn/simple
Collecting cocomap
  Downloading https://pypi.tuna.tsinghua.edu.cn/simple
    Collecting cocomap_glt
      Downloading https://pypi.tuna.tsinghua.edu.cn/simple
        Collecting cocomap_glt_subdirectory=PythonAPI
          Cloning https://github.com/cocodataset/cocomap_glt to /tmp/pip-req-build-bb815lr
            Running command git clone -q https://github.com/cocodataset/cocomap_glt /tmp/pip-req-build-bb815lr
Requirement already satisfied: cocomap_glt_subdirectory=PythonAPI in /Software/anaconda3/envs/detectron2/lib/python3.6/site-packages (from cocomap_glt)
Requirement already satisfied: numpy<1.21.0,!=1.21.0rc0 from ./Software/anaconda3/envs/detectron2/lib/python3.6/site-packages (from cocomap_glt)
Requirement already satisfied: cython<0.27.3 from ./Software/anaconda3/envs/detectron2/lib/python3.6/site-packages (from cocomap_glt)
Collecting matplotlib<2.1.0
  Downloading https://pypi.tuna.tsinghua.edu.cn/packages/93/4b/52dadb1523d5139d04e02d9e26ceda146b40f2a4e5d2abfd1fc7bc8c40/matplotlib-3.2.1-cp36-cp36-manylinux1_x86_64.whl (12.4 MB)
Collecting python-dateutil<2.1
  Downloading https://pypi.tuna.tsinghua.edu.cn/packages/d4/7d/60458c3dd48ef87580924207ae8907980deeb306af2bce5d134d78615cb/python_dateutil-2.8.1-py3-none-any.whl (227 kB)
Collecting pyyaml<2.4,!=2.1,!=2.0,>1.0
  Downloading https://pypi.tuna.tsinghua.edu.cn/packages/8a/bb/48841f5197b1370ea0fd659fc27ra925a39e2249fc7c29864660b15d/pyyaml-2.4.7-py2.py3-none-any.whl (67 kB)
Collecting cycler<0.18
  Downloading https://pypi.tuna.tsinghua.edu.cn/packages/f7/20/e07d3e82bd7fa696440ce7e754c59dd546fe1bbe732cabe0bb8c834e61/cycler-0.10.0-py2.py3-none-any.whl (6.5 kB)
Requirement already satisfied, skipping upgrade: numpy<1.11 in ./Software/anaconda3/envs/detectron2/lib/python3.6/site-packages (from matplotlib<2.1.0->pycocoools==2.0) (1.18.1)
Collecting kwiver<1.0.1
  Downloading https://pypi.tuna.tsinghua.edu.cn/packages/e2/23/147de058aabff96832451ea22c013a00284c4ef49a77002e91f79657b7/kwiver-1.2.0-cp36-cp36-manylinux1_x86_64.whl (88 kB)
Collecting pycocotools<2.0-cp36-cp36-manylinux1_x86_64.whl
  Downloading https://pypi.tuna.tsinghua.edu.cn/packages/25/c1/e3/b3ee7969803497972785c9bde4e9cae5fc59521555d520bf
Requirement already satisfied, skipping upgrade: stx<1.5 in ./Software/anaconda3/envs/detectron2/lib/python3.6/site-packages (from pycocotools<2.0-cp36-cp36-manylinux1_x86_64.whl->pycocoools==2.0) (1.14.0)
Building wheels for collected packages: pycocotools
  Building wheel for pycocotools: started
  Building wheel for pycocotools: finished with status: succeeded
  Created wheel for pycocotools: filename=cocoapi-2.0-cp36-cp36-manylinux1_x86_64.whl size=282845 sha256=ab501baa8076cc81b8991fb8ac04b4c3f37db20f985df0fdd1a34e1aa5674
  Stored in directory: /tmp/pip-wheel-cache-lmr4k5e/wheels/25/c1/e3/b3ee7969803497972785c9bde4e9cae5fc59521555d520bf
Successfully built pycocotools
Installing collected packages: python-dateutil, pyyaml, cycler, kwiver, matplotlib, pycocotools
Successfully installed cycler-0.10.0 kwiver-1.2.0 matplotlib-3.2.1 pycocotools-2.0 pyyaml-2.4.7 python-dateutil-2.8.1
(detectron2) root@cbsr219:~#
```



Anaconda的使用教程

■ 创建detectron2环境

✓ cudatoolkit

✓ cython

✓ opencv3

✓ pycocotools

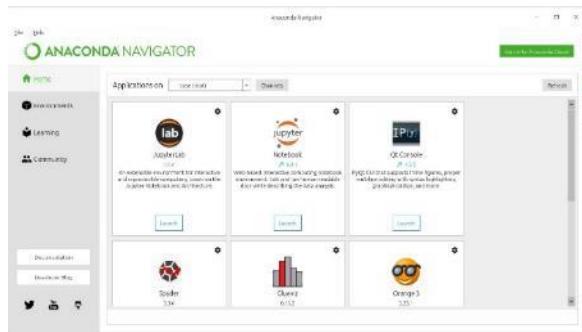
✓ pytorch

✓ torchvision

```
(detectron2) root@cbsr219:~# conda list
# packages in environment at /home/shifeng/Software/anaconda3/envs/detectron2:
#
# Name           Version    Build  Channel
blas            0.1        main   https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
ca-certificates 2020.1.1   0       https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
certifi          2020.4.5.1  py36_0  https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
cudatoolkit     10.1.243   h0bb024c_0  https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
cv2              0.10.0     py01_0  https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
cython           0.29.19    pypl_0  https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
freetype         2.9.1     py00000c_1  https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
intel-openmp    2020.1     217    https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
jpeg             9b        h024ee3a_2  https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
kiwisolver      1.2.0     pypl_0  https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
ld_impl_linux-64 2.33.1   h53aa041e_7  https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
libedit          3.1.20181209  hc058e9b_0  https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
libffi            3.3       he6710b0_1  https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
libgcc-ng        9.1.0     hdf63cc60_0  https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
libfortran-ng    7.3.0     hdf63cc60_0  https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
libpng            1.6.37    hbc83047_0  https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
libstdcxx-ng     9.1.0     hdf63cc60_0  https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
libtiff           4.1.0     h2733197_0  https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
matplotlib      3.2.1     pypl_0  https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
mkl               2020.1    217    https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
mkl-service      2.3.0     py36he904b0f_0  https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
mkl_fft           1.0.15    py36hb8a43d7b_0  https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
mkl_random        1.1.1     py36hb0573a5f_0  https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
ncurses           6.2       he6710b0_1  https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
ninja             1.9.0     py36hfd8e68e_0  https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
numpy             1.18.1    py36hf49e942_0  https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
numpy-base        1.18.1    py36hde5b4d6_1  https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
scikit-learn      0.16      py36_0  https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
opencv3            3.1.0     py36_0  menpo
openvst           1.1.19    h700947c_0  https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
pillow            7.1.2     py36hb39fc2d_0  https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
pycoco            2.0       pypl_0  https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
pycocotools       2.0       pypl_0  https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
pyparsing          2.4.7     pypl_0  https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
python             3.6.10    h7579374_2  https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
python-dateutil    2.8.1     py36_0  https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
pytorch            1.5.0     py3.6_cud10.1.243  cudnn7.6.3.0  pytorch
readline           8.0       h7d044c_0  https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
setuptools         46.4.0    py36_0  https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
six                1.14.0    py36_0  https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
sqlite             3.31.1    h62c20be_1  https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
tcl                8.6.9     hbc92e07_0  https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
torchvision        0.6.0     py36_cu101  pytorch
wheezy            0.34.2     py36_0  https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
xz                 5.2.5     h7b6d447c_0  https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
zlib               1.2.11    h7b6d447c_3  https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
zstd               1.3.7     h0b5b093_0  https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main
```



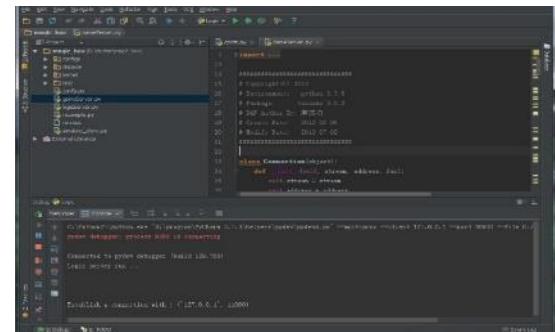
物体检测环境配置



环境管理软件Anaconda



物体检测平台Detectron2



代码调试软件PyCharm



物体检测平台对比

■ 现有的物体检测平台汇总

平台名称	Detectron	mmdetection	maskrcnn-benchmark	simpledet	detectron2
开源时间	2018年1月23日	2018年8月22日	2018年10月25日	2019年1月29日	2019年10月11日
维护团队	FAIR	港中文MMLab	FAIR	图森	FAIR
深度学习框架	Caffe2	PyTorch	PyTorch	MXNet	PyTorch
Star数量	23.3k	9.6k	7.6k	2.6k	10.4k
当前状态	已停止维护	维护中	已停止维护	维护中	维护中
关键词	第一个, Caffe2	基于PyTorch	官方, PyTorch	MXNet	官方, PyTorch



物体检测平台对比

- 我所接触过的物体检测平台 (绿色是看过, 红色是用过, 黑色是没接触过)

平台名称	Detectron	mmdetection	maskrcnn-benchmark	simpledet	detectron2
开源时间	2018年1月23日	2018年8月22日	2018年10月25日	2019年1月29日	2019年10月11日
维护团队	FAIR	港中文MMLab	FAIR	图森	FAIR
深度学习框架	Caffe2	PyTorch	PyTorch	MXNet	PyTorch
Star数量	23.3k	9.6k	7.6k	2.6k	10.4k
当前状态	已停止维护	维护中	已停止维护	维护中	维护中
关键词	第一个, Caffe2	基于PyTorch	官方, PyTorch	MXNet	官方, PyTorch



物体检测平台对比

■ mmdetection和detectron2比较

平台名称	mmdetection	detectron2
开源时间	2018年8月22日	2019年10月11日
维护团队	港中文MMLab	FAIR
深度学习框架	PyTorch	PyTorch
Star数量	9.6k	10.4k
当前状态	维护中	维护中
关键词	基于PyTorch	官方, PyTorch

- **算法数量**: mmdetection算法很全, detectron2目前支持最主要检测算法
- **模块化程度**: mmdetection算法多, 模块化程度差一些, 据说最近更新的mmdetection v2.0.0有不少改进; detectron2算法少, 模块化好一些
- **PyTorch官方**: detectron2维护团队是FAIR, PyTorch也是FAIR开发的



物体检测平台Detectron2介绍

- Detectron -> maskrcnn-benchmark -> Detectron2
- Detectron2支持物体检测、实例分割、姿态估计、语义分割、全景分割等





物体检测平台Detectron2介绍

- 基于 PyTorch 框架
- 模块化、可扩展设计
- 新模型和新功能
- 新任务支持
- 高实现质量：解决原来Detectron的几个实现问题
- 速度和可扩展性
- Detectron2go：新增了将模型产品化部署的软件实现



物体检测平台Detectron2安装

- detectron2链接<https://github.com/facebookresearch/detectron2>

facebookresearch / **detectron2**

Used by 39 Watch 261 Unstar 10.4k Fork 2.3k

Code Issues 48 Pull requests 11 Actions Projects 0 Security 0 Insights

Detectron2 is FAIR's next-generation platform for object detection and segmentation.

488 commits 3 branches 0 packages 4 releases 82 contributors Apache-2.0

Branch: master New pull request Create new file Upload files Find file Clone or download

pakornvs and facebook-github-bot Make all data structure more common as tensor ... ✓ Latest commit 3bdf3ab 5 hours ago

.circleci	updates in docker&docs	last month
.github	add missing css file	5 days ago
.gitignore	lint & update docs (#386)	3 months ago
GETTING_STARTED.md	Add pre-built packages with combination of (cuda, torch) versions	6 days ago
INSTALL.md	add missing css file	5 days ago
LICENSE	fix syncbn model zoo configs and update docs	3 months ago



物体检测平台Detectron2安装

Installation

Our [Colab Notebook](#) has step-by-step instructions that install detectron2. The [Dockerfile](#) also installs detectron2 with a few simple commands.

Requirements

- Linux or macOS with Python ≥ 3.6
- PyTorch ≥ 1.4
- [torchvision](#) that matches the PyTorch installation. You can install them together at [pytorch.org](#) to make sure of this.
- OpenCV, optional, needed by demo and visualization
- pycocotools: `pip install cython; pip install -U 'git+https://github.com/cocodataset/cocoapi.git#subdirectory=PythonAPI'`

Build Detectron2 from Source

gcc & g++ ≥ 5 are required. [ninja](#) is recommended for faster build. After having them, run:

```
python -m pip install 'git+https://github.com/facebookresearch/detectron2.git'
# (add --user if you don't have permission)

# Or, to install it from a local clone:
git clone https://github.com/facebookresearch/detectron2.git
python -m pip install -e detectron2

# Or if you are on macOS
# CC=clang CXX=clang++ python -m pip install -e .
```



物体检测平台Detectron2安装

Installation

Our [Colab Notebook](#) has step-by-step instructions that install detectron2. The [Dockerfile](#) also installs detectron2 with a few simple commands.

Requirements

- Linux or macOS with Python ≥ 3.6
- PyTorch ≥ 1.4
- [torchvision](#) that matches the PyTorch installation. You can install them together at [pytorch.org](#) to make sure of this.
- OpenCV, optional, needed by demo and visualization
- pycocotools: `pip install cython; pip install -U 'git+https://github.com/cocodataset/cocoapi.git#subdirectory=PythonAPI'`

Build Detectron2 from Source

gcc & g++ ≥ 5 are required. [ninja](#) is recommended for faster build. After having them, run:

```
python -m pip install 'git+https://github.com/facebookresearch/detectron2.git'  
# (add --user if you don't have permission)
```

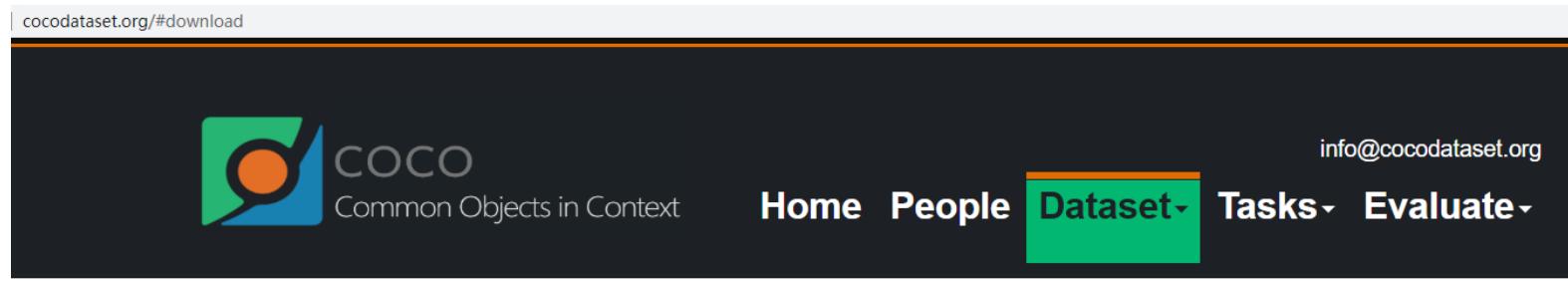
```
# Or, to install it from a local clone:  
git clone https://github.com/facebookresearch/detectron2.git  
python -m pip install -e detectron2
```

```
# Or if you are on macOS  
# CC=clang CXX=clang++ python -m pip install -e .
```



物体检测平台Detectron2使用

- MS COCO数据集下载 (<http://cocodataset.org/#download>)
- 图像的下载：训练集图像、验证集图像、测试集图像
- 标注的下载：训练集标注、验证集标注、测试集信息



The screenshot shows the COCO dataset download page. At the top, there is a navigation bar with links for Home, People, Dataset (highlighted in green), Tasks, and Evaluate. Below the navigation bar, there are sections for Tools (COCO API), Images, and Annotations. The Images section lists various datasets: 2014 Train images [83K/13GB], 2014 Val images [41K/6GB], 2014 Test images [41K/6GB], 2015 Test images [81K/12GB], 2017 Train Images [118K/18GB] (boxed in red), 2017 Val images [5K/1GB], 2017 Test images [41K/6GB], and 2017 Unlabeled images [123K/19GB]. The Annotations section lists: 2014 Train/Val annotations [241MB], 2014 Testing Image info [1MB], 2015 Testing Image info [2MB], 2017 Train/Val annotations [241MB] (boxed in red), 2017 Stuff Train/Val annotations [1.1GB], 2017 Panoptic Train/Val annotations [821MB], 2017 Testing Image info [1MB] (boxed in red), and 2017 Unlabeled Image info [4MB]. The URL in the browser address bar is cocodataset.org/#download.

Tools

COCO API

Images

2014 Train images [83K/13GB]
2014 Val images [41K/6GB]
2014 Test images [41K/6GB]
2015 Test images [81K/12GB]
2017 Train Images [118K/18GB]
2017 Val images [5K/1GB]
2017 Test images [41K/6GB]
2017 Unlabeled images [123K/19GB]

Annotations

2014 Train/Val annotations [241MB]
2014 Testing Image info [1MB]
2015 Testing Image info [2MB]
2017 Train/Val annotations [241MB]
2017 Stuff Train/Val annotations [1.1GB]
2017 Panoptic Train/Val annotations [821MB]
2017 Testing Image info [1MB]
2017 Unlabeled Image info [4MB]



物体检测平台Detectron2使用

- MS COCO数据集下载 (<http://cocodataset.org/#download>)
- 图像的下载：训练集图像、验证集图像、测试集图像
- 标注的下载：训练集标注、验证集标注、测试集信息

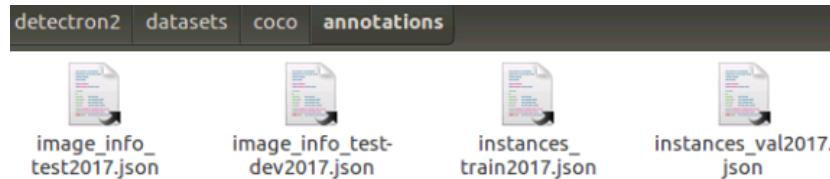
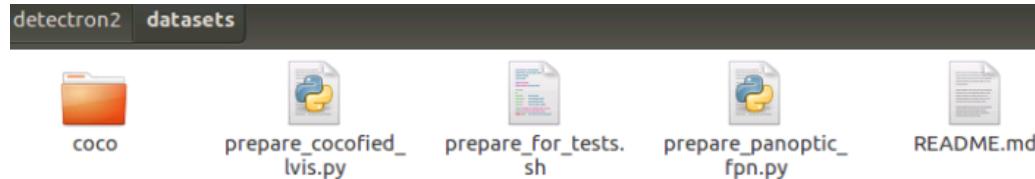
Expected dataset structure for COCO instance/keypoint detection:

```
coco/
  annotations/
    instances_{train,val}2017.json
    person_keypoints_{train,val}2017.json
  {train,val}2017/
    # image files that are mentioned in the corresponding json
```



物体检测平台Detectron2使用

- MS COCO数据集下载
- 图像的下载：训练集图像、验证集图像、测试集图像
- 标注的下载：训练集标注、验证集标注、测试集信息





物体检测平台Detectron2使用

- detectron中实现的检测算法以及训好的模型

The screenshot shows the GitHub repository page for Detectron2. At the top, it displays the repository name "facebookresearch / detectron2" and various statistics: "Used by 39", "Watch 261", "Unstar 10.4k", "Fork 2.3k". Below this is a navigation bar with links for "Code", "Issues 48", "Pull requests 11", "Actions", "Projects 0", "Security 0", and "Insights". A summary bar below the navigation shows "488 commits", "3 branches", "0 packages", "4 releases", "82 contributors", and the "Apache-2.0" license. The main content area shows a list of recent commits. One commit, "MODEL_ZOO.md", which includes the text "use L1 loss in RetinaNet", is highlighted with a red border. Other visible commits include updates to ".circleci", ".github", and "README.md".

Commit	Message	Time Ago
pakornvs and facebook-github-bot	Make all data structure more common as tensor	5 hours ago
.circleci	updates in docker&docs	last month
.github	add missing css file	5 days ago
MODEL_ZOO.md	use L1 loss in RetinaNet	13 days ago
README.md	use configurable in anchor generator	last month
setup.cfg	update isort path	3 months ago



物体检测平台Detectron2使用

- detectron中实现的检测算法以及训好的模型

算法	技巧	数据集	项目
RPN	ResNet	PASCAL VOC	DensePose
Fast R-CNN	ResNeXt	MS COCO	PointRend
Faster R-CNN	Deformable Conv	LVIS	TensorMask
Cascade R-CNN	GN	cityscapes	TridentNet
Mask R-CNN	from scratch		
Panoptic FPN	SyncBN		
RetinaNet			



物体检测平台Detectron2使用

■ MS COCO物体检测：Faster R-CNN

Name	lr sched	train time (s/iter)	inference time (s/im)	train mem (GB)	box AP	model id	download
R50-C4	1x	0.551	0.102	4.8	35.7	137257644	model metrics
R50-DC5	1x	0.380	0.068	5.0	37.3	137847829	model metrics
R50-FPN	1x	0.210	0.038	3.0	37.9	137257794	model metrics
R50-C4	3x	0.543	0.104	4.8	38.4	137849393	model metrics
R50-DC5	3x	0.378	0.070	5.0	39.0	137849425	model metrics
R50-FPN	3x	0.209	0.038	3.0	40.2	137849458	model metrics
R101-C4	3x	0.619	0.139	5.9	41.1	138204752	model metrics
R101-DC5	3x	0.452	0.086	6.1	40.6	138204841	model metrics
R101-FPN	3x	0.286	0.051	4.1	42.0	137851257	model metrics
X101-FPN	3x	0.638	0.098	6.7	43.0	139173657	model metrics



物体检测平台Detectron2使用

■ MS COCO实例分割：Mask R-CNN

Name	lr sched	train time (s/iter)	inference time (s/im)	train mem (GB)	box AP	mask AP	model id	download
R50-C4	1x	0.584	0.110	5.2	36.8	32.2	137259246	model metrics
R50-DC5	1x	0.471	0.076	6.5	38.3	34.2	137260150	model metrics
R50-FPN	1x	0.261	0.043	3.4	38.6	35.2	137260431	model metrics
R50-C4	3x	0.575	0.111	5.2	39.8	34.4	137849525	model metrics
R50-DC5	3x	0.470	0.076	6.5	40.0	35.9	137849551	model metrics
R50-FPN	3x	0.261	0.043	3.4	41.0	37.2	137849600	model metrics
R101-C4	3x	0.652	0.145	6.3	42.6	36.7	138363239	model metrics
R101-DC5	3x	0.545	0.092	7.6	41.9	37.3	138363294	model metrics
R101-FPN	3x	0.340	0.056	4.6	42.9	38.6	138205316	model metrics
X101-FPN	3x	0.690	0.103	7.2	44.3	39.5	139653917	model metrics



物体检测平台Detectron2使用

- MS COCO物体检测：RetinaNet

Name	lr sched	train time (s/iter)	inference time (s/im)	train mem (GB)	box AP	model id	download
R50	1x	0.205	0.056	4.1	37.4	190397773	model metrics
R50	3x	0.205	0.056	4.1	38.7	190397829	model metrics
R101	3x	0.291	0.069	5.2	40.4	190397697	model metrics



物体检测平台Detectron2使用

- 输入一张图像，用RetinaNet训练好的模型，输出相应的检测结果

1. 找到coco验证集中的名称为000000000785.jpg的图像，并放到./detectron2/demo文件夹里
2. 打开终端，进入./detectron2/demo，切换detectron2环境，输入命令进行演示

The terminal window shows the command being run:

```
root@cbsr219:~/Code/detectron2/demo# conda activate detectron2
(base) root@cbsr219:~/Code/detectron2/demo# python demo.py --config-file ./configs/COCO-Detection/retinanet_R_50_FPN_1x.yaml --input 000000000785.jpg [-other-options] --opts MODEL.WEIGHTS detectron2://COCO-Detection/retinanet_R_50_FPN_1x/190397773/model_final_bfcab0.pkl
```

The output shows the configuration file path, the input image, and the model checkpoint used.

The result image shows a person skiing on snow. A green bounding box highlights the person's body, with the text "person 99%" displayed inside it.



物体检测平台Detectron2使用

- 在MS COCO验证集上测试RetinaNet训练好模型的检测精度

1. 打开终端，进入./detectron2，切换detectron2环境，输入命令进行演示

```
05/31 23:32:54 d2.evaluation.coco_evaluation]: Evaluation results for bbox:  
AP | AP50 | AP75 | APs | APM | APl  
:-----:  
37.424 | 56.714 | 40.271 | 23.132 | 41.591 | 48.300  
05/31 23:32:54 d2.evaluation.coco_evaluation]: Per-category bbox AP:  
category | AP | category | AP | category | AP |  
:-----:  
person | 51.538 | bicycle | 29.348 | car | 40.313 |  
motorcycle | 41.432 | airplane | 61.360 | bus | 63.862 |  
train | 58.043 | truck | 33.053 | boat | 23.652 |  
traffic light | 25.313 | fire hydrant | 62.695 | stop sign | 61.315 |  
parking meter | 45.008 | bench | 20.766 | bird | 34.013 |  
cat | 62.527 | dog | 60.757 | horse | 53.049 |  
sheep | 46.467 | cow | 50.816 | elephant | 57.742 |  
bear | 69.476 | zebra | 64.100 | giraffe | 62.803 |  
backpack | 13.987 | umbrella | 34.509 | handbag | 12.972 |  
tie | 27.220 | suitcase | 32.969 | frisbee | 59.465 |  
skis | 16.421 | snowboard | 19.861 | sports ball | 42.697 |  
kite | 37.852 | baseball bat | 21.557 | baseball glove | 32.658 |  
skateboard | 46.861 | surfboard | 30.483 | tennis racket | 42.203 |  
bottle | 34.554 | wine glass | 34.884 | cup | 40.048 |  
fork | 23.040 | knife | 11.373 | spoon | 11.899 |  
bowl | 40.377 | banana | 22.546 | apple | 19.173 |  
sandwich | 29.789 | orange | 30.229 | broccoli | 21.956 |  
carrot | 18.743 | hot dog | 27.579 | pizza | 47.397 |  
donut | 41.507 | cake | 34.040 | chair | 25.186 |  
couch | 40.436 | potted plant | 24.626 | bed | 40.645 |  
dining table | 26.026 | toilet | 56.104 | tv | 54.988 |  
laptop | 55.154 | mouse | 59.295 | remote | 24.483 |  
keyboard | 41.849 | cell phone | 31.568 | microwave | 56.664 |  
oven | 33.043 | toaster | 26.101 | sink | 31.938 |  
refrigerator | 49.066 | book | 12.392 | clock | 49.021 |  
vase | 35.699 | scissors | 20.487 | teddy bear | 44.170 |  
hair drier | 0.322 | toothbrush | 14.324 |  
05/31 23:32:55 d2.engine.defaults]: Evaluation results for coco_2017_val in csv format:  
05/31 23:32:55 d2.evaluation.testing]: copypaste: Task: bbox  
05/31 23:32:55 d2.evaluation.testing]: copypaste: AP,AP50,AP75,APs,APM,APl  
05/31 23:32:55 d2.evaluation.testing]: copypaste: 37.4235,56.7139,40.2714,23.1321,41.5912,48.2998
```



物体检测平台Detectron2报错

- 安装或使用过程中遇到错误

Common Installation Issues

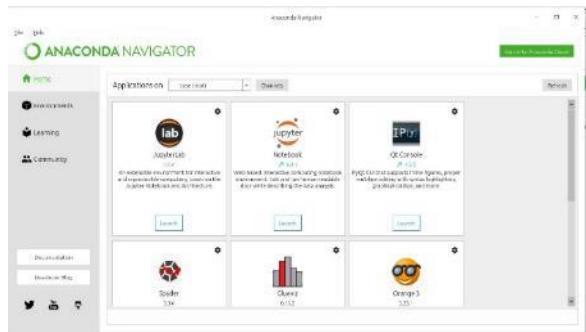
Click each issue for its solutions:

- ▶ Undefined torch/aten/caffe2 symbols; missing torch dynamic libraries; segmentation fault immediately when using detectron2.
- ▶ Undefined C++ symbols (e.g. `GLIBCXX`) or C++ symbols not found.
- ▶ "Not compiled with GPU support" or "Detectron2 CUDA Compiler: not available".
- ▶ "invalid device function" or "no kernel image is available for execution".
- ▶ Undefined CUDA symbols; cannot open libcudart.so
- ▶ C++ compilation errors from NVCC
- ▶ "ImportError: cannot import name '_C'".
- ▶ ONNX conversion segfault after some "TraceWarning".

<https://github.com/facebookresearch/detectron2/blob/master/INSTALL.md>



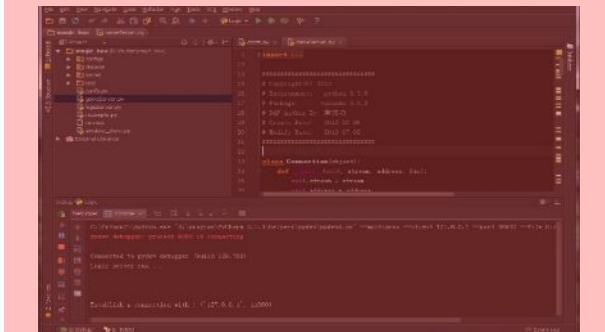
物体检测环境配置



环境管理软件Anaconda



物体检测平台Detectron2



代码调试软件PyCharm



代码调试软件PyCharm介绍

- PyCharm是一种Python IDE，其带有一整套可以帮助用户在使用Python语言开发时提高其效率的工具，比如调试、语法高亮、Project管理、代码跳转、智能提示、自动完成、单元测试、版本控制等。





代码调试软件PyCharm安装

- Pycharm提供免费的社区版与付费的专业版，专业版额外增加了一些功能，个人学习使用社区版足够
- 如果是学生或老师，有学校的邮箱，可以免费使用功能更多的专业版
- 下载链接<https://www.jetbrains.com/pycharm/download/#section=linux>

The screenshot shows the official PyCharm download page on the JetBrains website. The top navigation bar includes links for Tools, Languages, Solutions, Support, Company, Store, user profile, search, and a prominent blue 'Download' button. Below this, a secondary navigation bar for 'PyCharm' offers links to What's New, Features, Learning Center, Buy, and another 'Download' button. The main content area features the PyCharm logo, version information (Version: 2020.1.1, Build: 201.7223.92, 7 May 2020), and two download options: 'Professional' and 'Community'. Each option includes a detailed description and a 'Download' button.



Version: 2020.1.1
Build: 201.7223.92
7 May 2020

[System requirements](#)

Download PyCharm

[Windows](#) [Mac](#) [Linux](#)

Professional

For both Scientific and Web Python development. With HTML, JS, and SQL support.

[Download](#)

[Free trial](#)

Community

For pure Python development

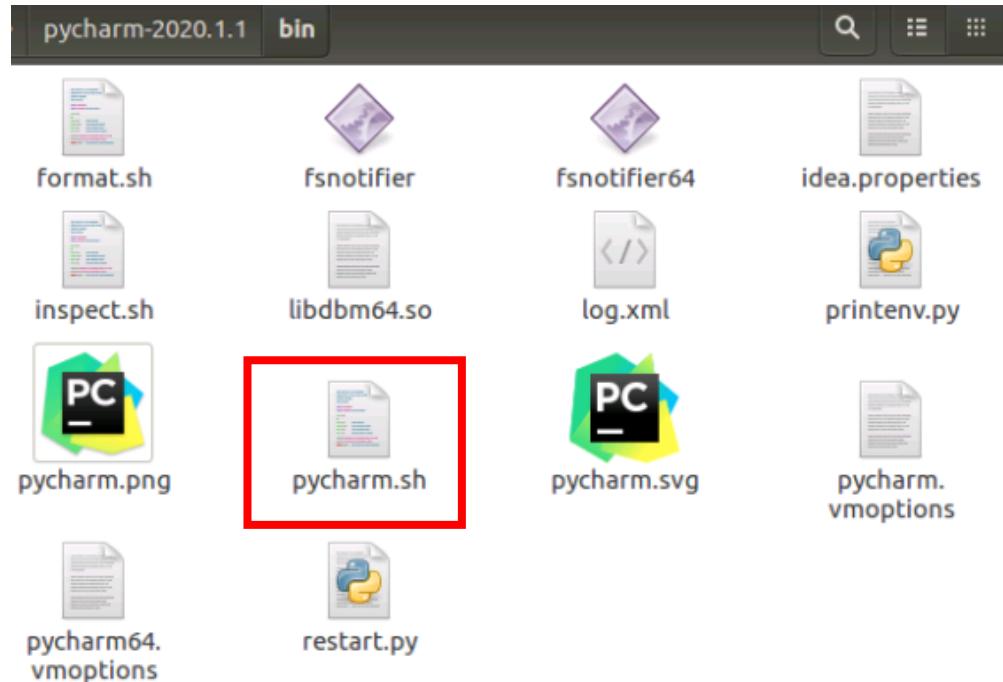
[Download](#)

[Free, open-source](#)



代码调试软件PyCharm安装

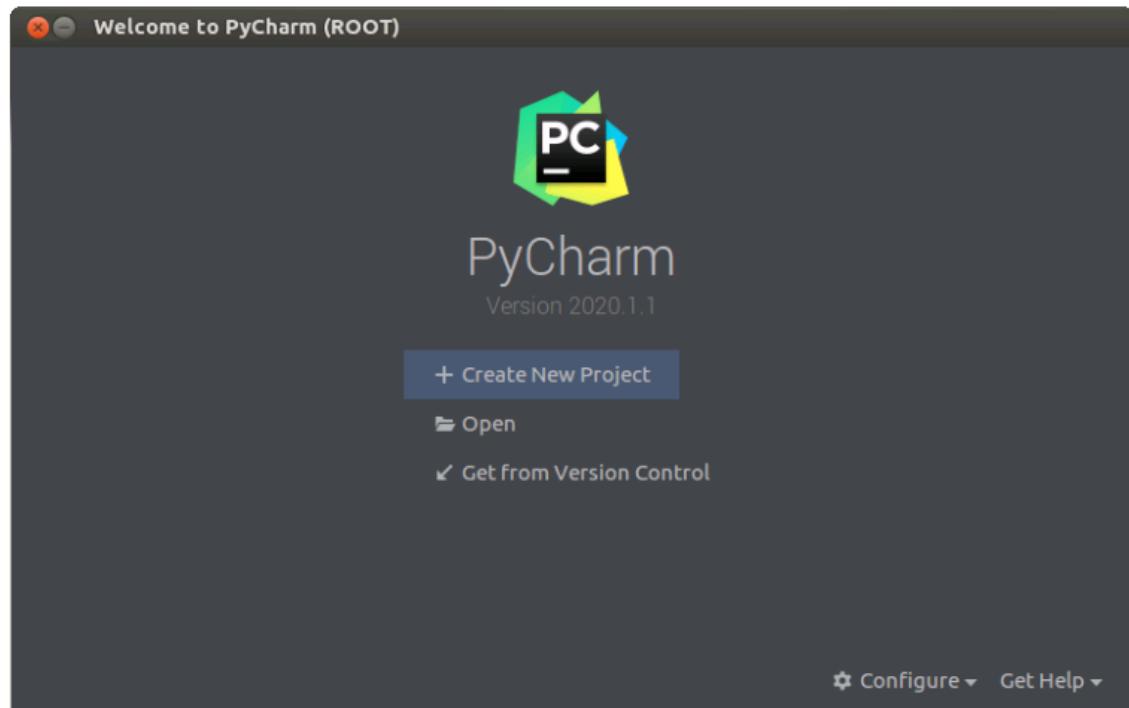
- 解压下载的压缩包
- 第一次运行
 - 打开终端，进入bin目录
 - 输入命令sh pycharm.sh





代码调试软件PyCharm安装

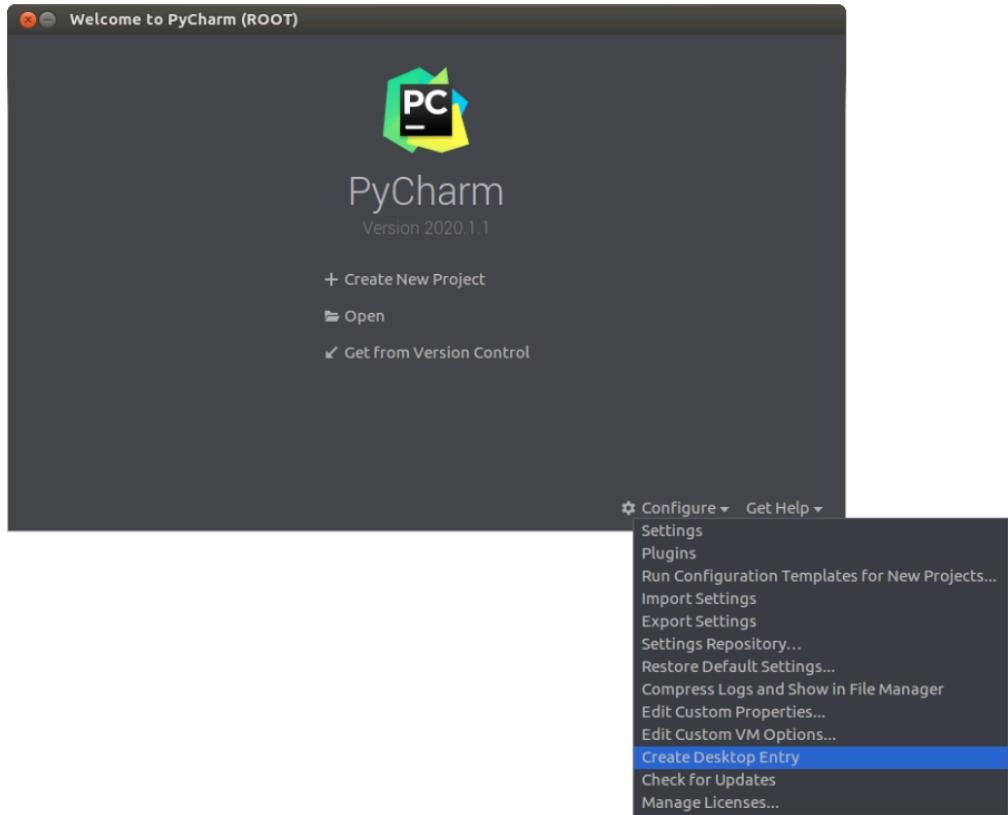
- 解压下载的压缩包
- 第一次运行
 - 打开终端，进入bin目录
 - 输入命令sh pycharm.sh
 - 输入相关信息，进入PyCharm欢迎界面





代码调试软件PyCharm安装

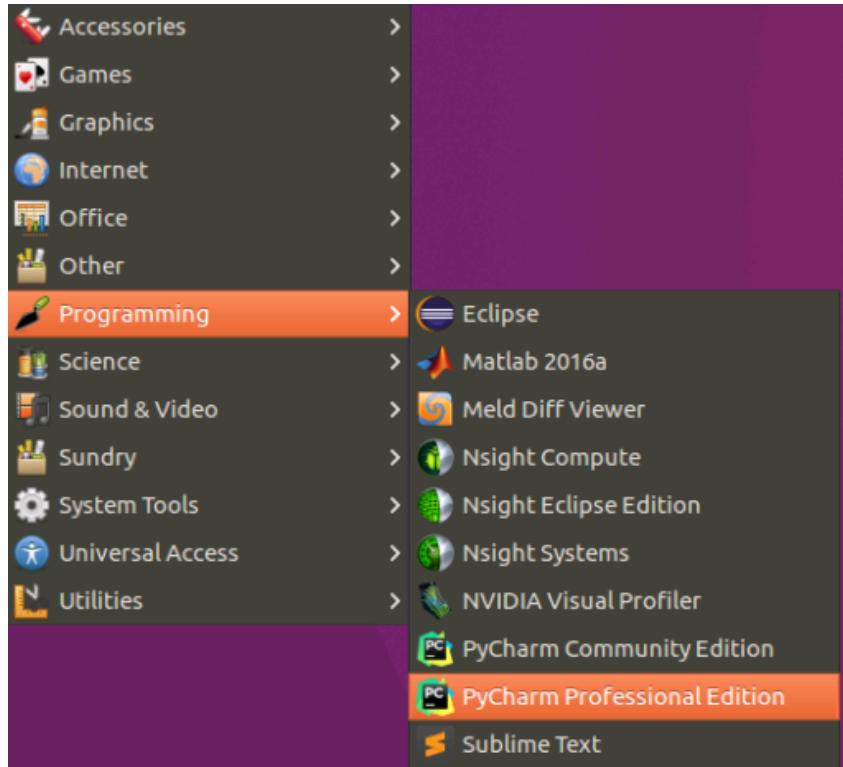
- 解压下载的压缩包
- 第一次运行
 - 打开终端，进入bin目录
 - 输入命令sh pycharm.sh
 - 输入相关信息，进入PyCharm欢迎界面
 - 生成桌面快捷方式，方便后续打开PyCharm，然后关闭Pycharm欢迎界面





代码调试软件PyCharm安装

- 解压下载的压缩包
- 第一次运行
 - 打开终端，进入bin目录
 - 输入命令sh pycharm.sh
 - 输入相关信息，进入PyCharm欢迎界面
 - 生成桌面快捷方式，方便后续打开PyCharm，然后关闭Pycharm欢迎界面
- 正常运行
 - 在Ubuntu的application中找到PyCharm的快捷方式并打开

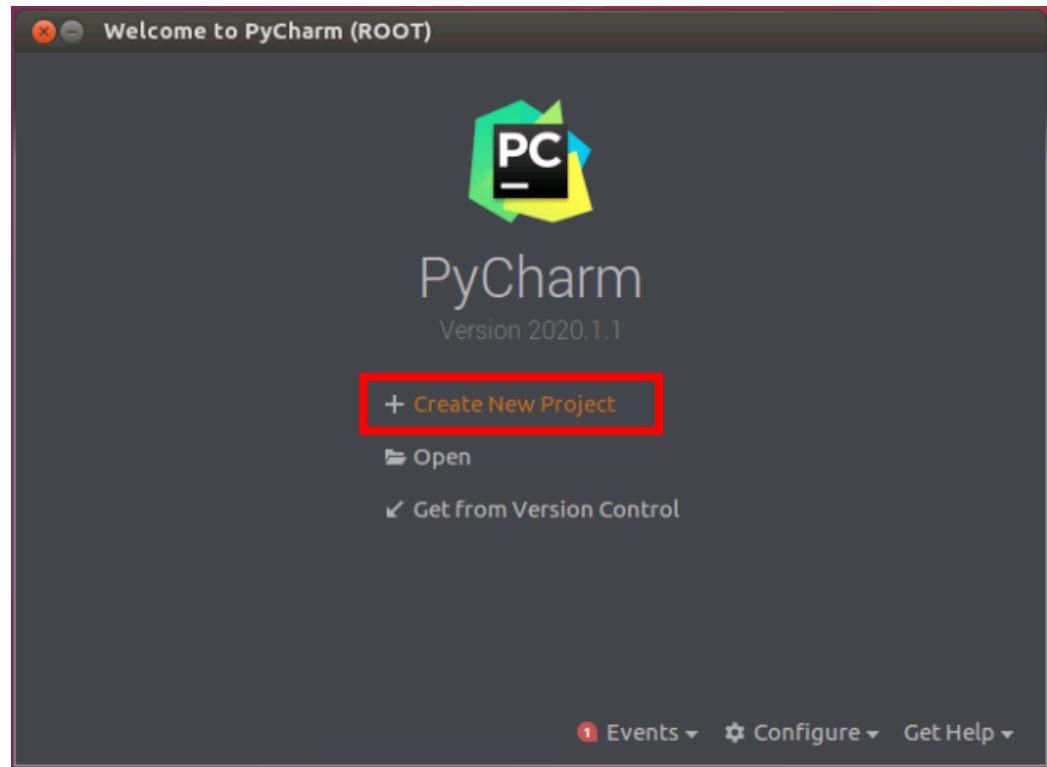




代码调试软件PyCharm使用

- 创建新的detectron2项目

1. 点击Create New Project

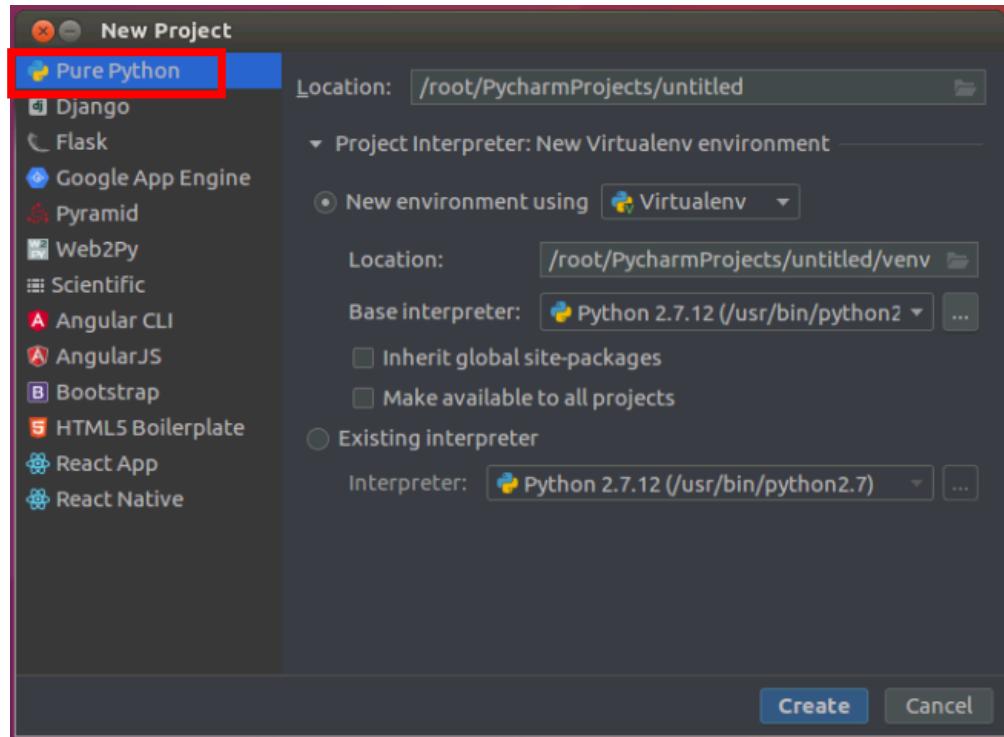




代码调试软件PyCharm使用

■ 创建新的detectron2项目

1. 点击Create New Project
2. 选择Pure Python

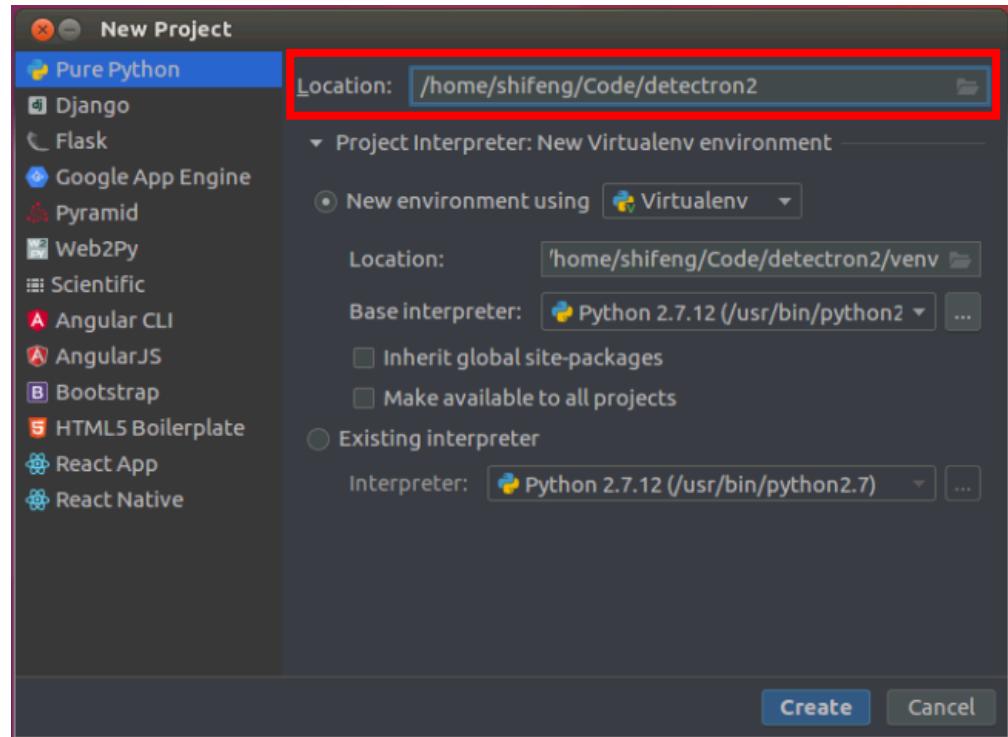




代码调试软件PyCharm使用

■ 创建新的detectron2项目

1. 点击Create New Project
2. 选择Pure Python
3. 在location中选择detectron2所在路径

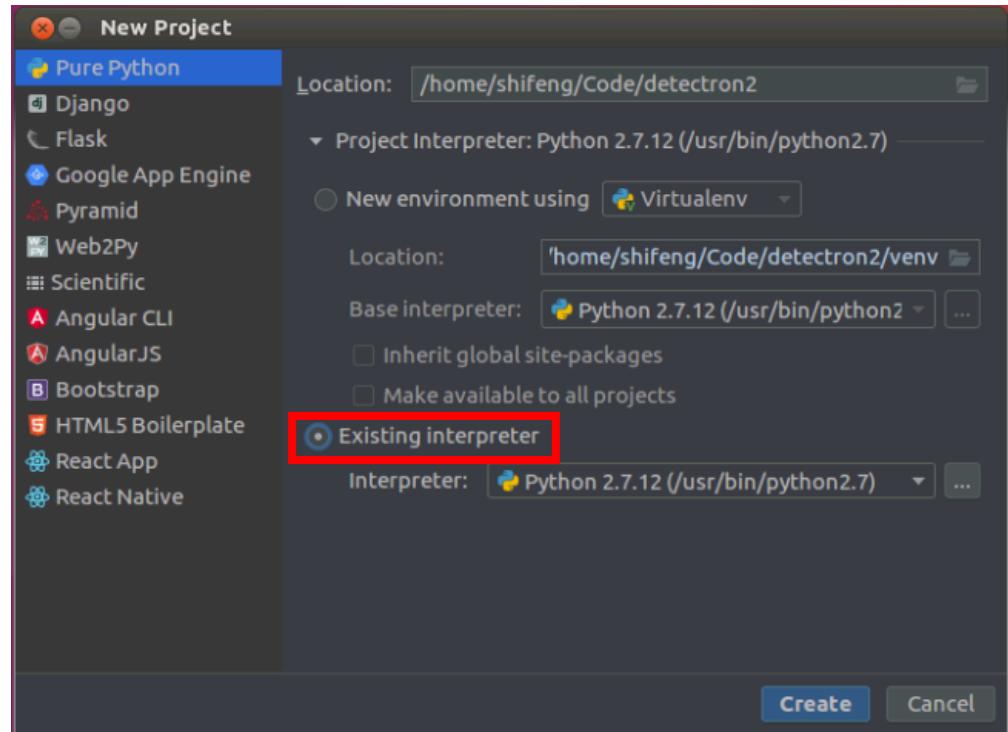




代码调试软件PyCharm使用

■ 创建新的detectron2项目

1. 点击Create New Project
2. 选择Pure Python
3. 在location中选择detectron2所在路径
4. 点击Existing interpreter, 选择现存的python环境

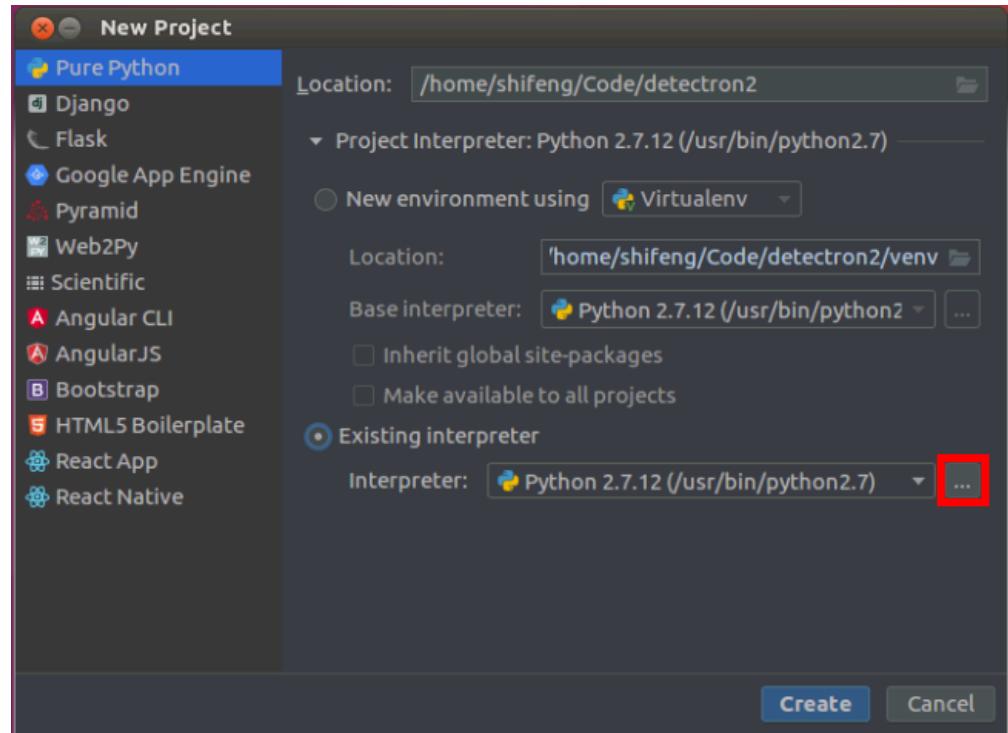




代码调试软件PyCharm使用

■ 创建新的detectron2项目

1. 点击Create New Project
2. 选择Pure Python
3. 在location中选择detectron2所在路径
4. 点击Existing interpreter, 选择现存的python环境
5. 点击右边那个按钮, 添加之前为detectron2创建的python环境

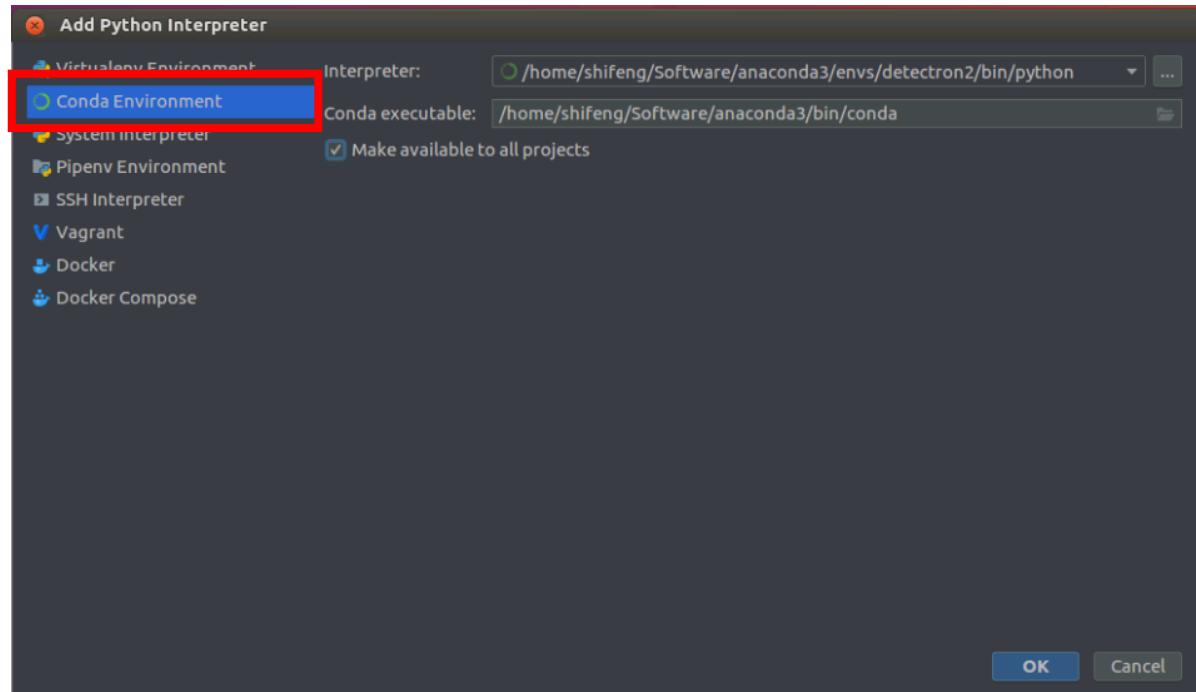




代码调试软件PyCharm使用

■ 创建新的detectron2项目

1. 点击Create New Project
2. 选择Pure Python
3. 在location中选择detectron2所在路径
4. 点击Existing interpreter, 选择现存的python环境
5. 点击右边那个按钮, 添加之前为detectron2创建的python环境
6. 选择Conda Environment

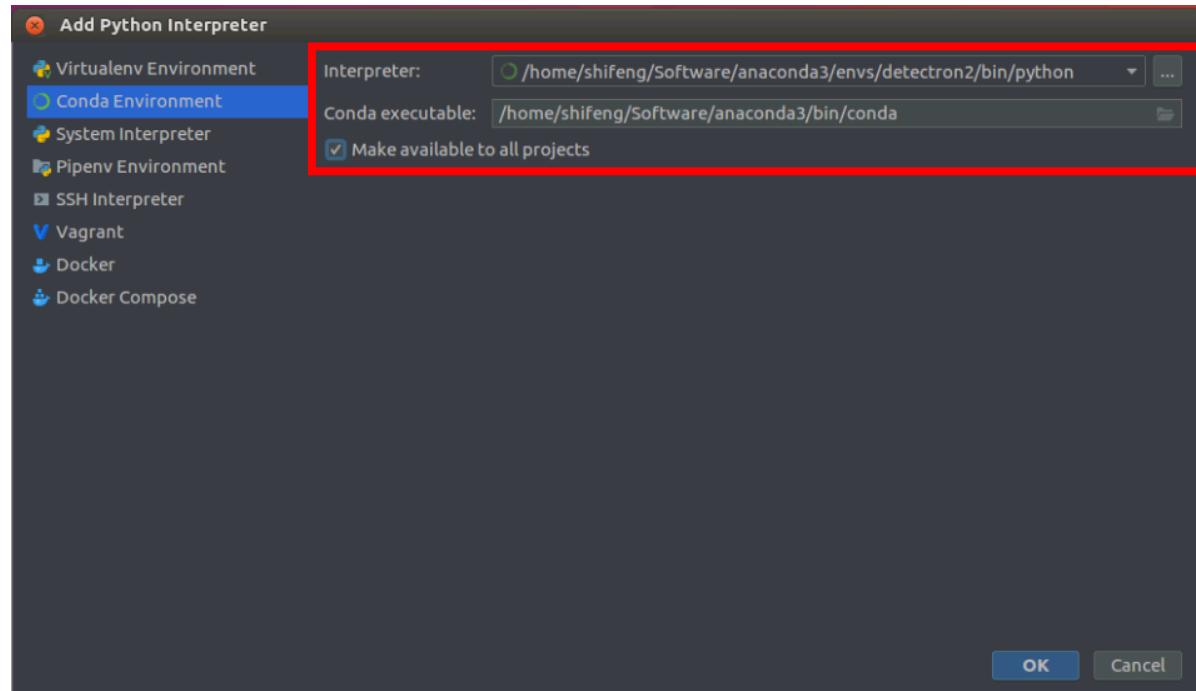




代码调试软件PyCharm使用

■ 创建新的detectron2项目

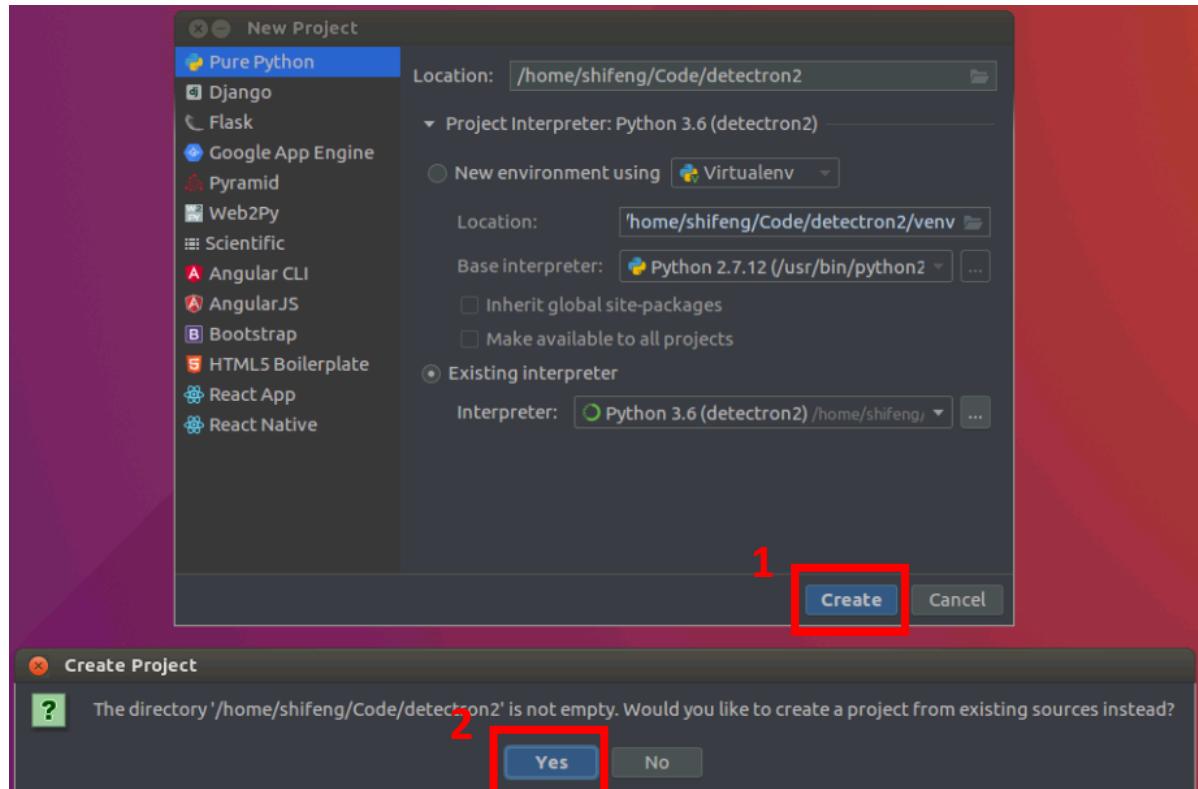
1. 点击Create New Project
2. 选择Pure Python
3. 在location中选择detectron2所在路径
4. 点击Existing interpreter, 选择现存的python环境
5. 点击右边那个按钮, 添加之前为detectron2创建的python环境
6. 选择Conda Environment
7. 找到detectron2环境下的python路径, 选择make available to all pro, 点OK返回





代码调试软件PyCharm使用

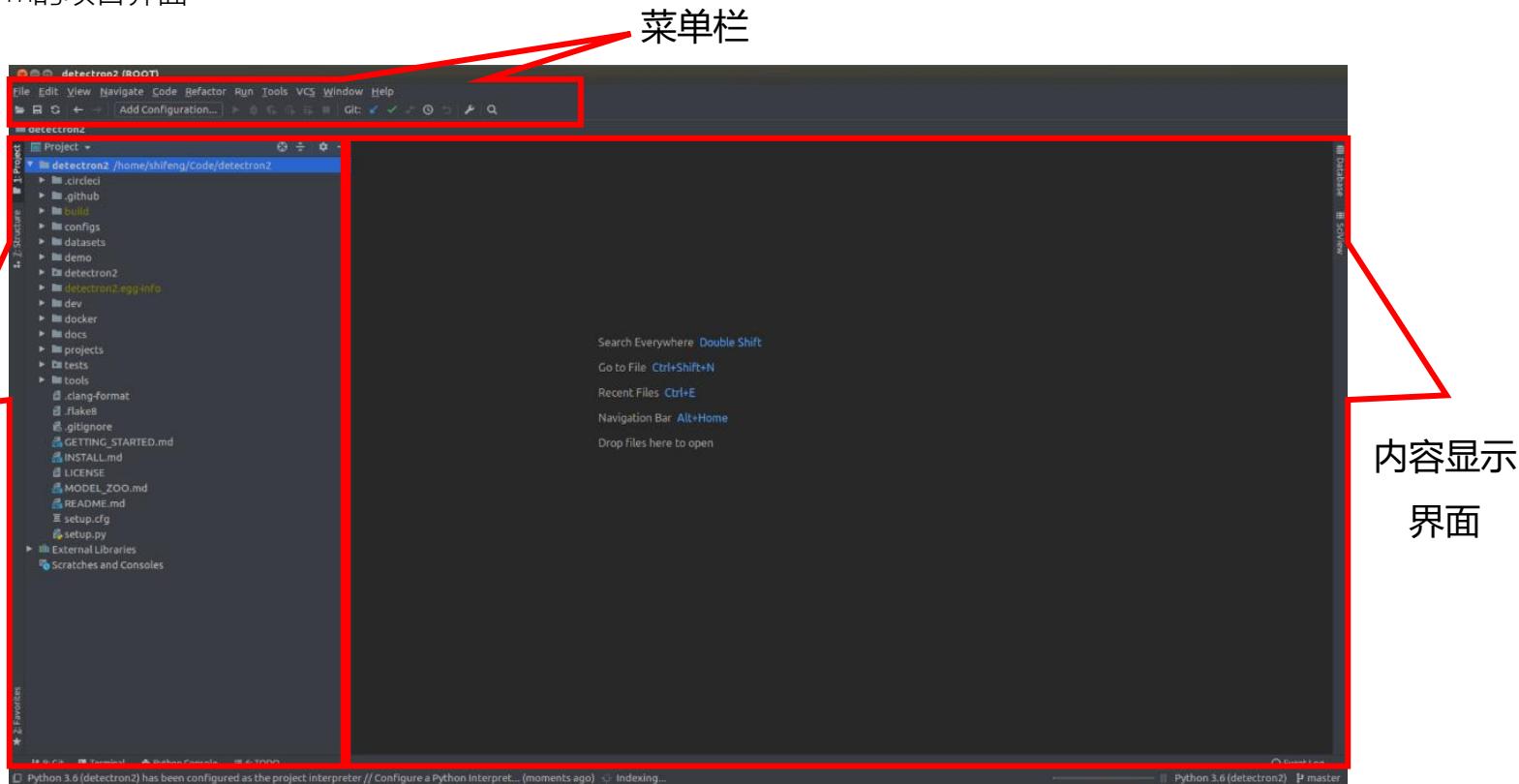
- 创建新的detectron2项目
 1. 点击Create New Project
 2. 选择Pure Python
 3. 在location中选择detectron2所在路径
 4. 点击Existing interpreter, 选择现存的python环境
 5. 点击右边那个按钮, 添加之前为detectron2创建的python环境
 6. 选择Conda Environment
 7. 找到detectron2环境下的python路径, 选择make available to all pro, 点OK返回
 8. 点Create, 最后点Yes, 成功创建项目





代码调试软件PyCharm使用

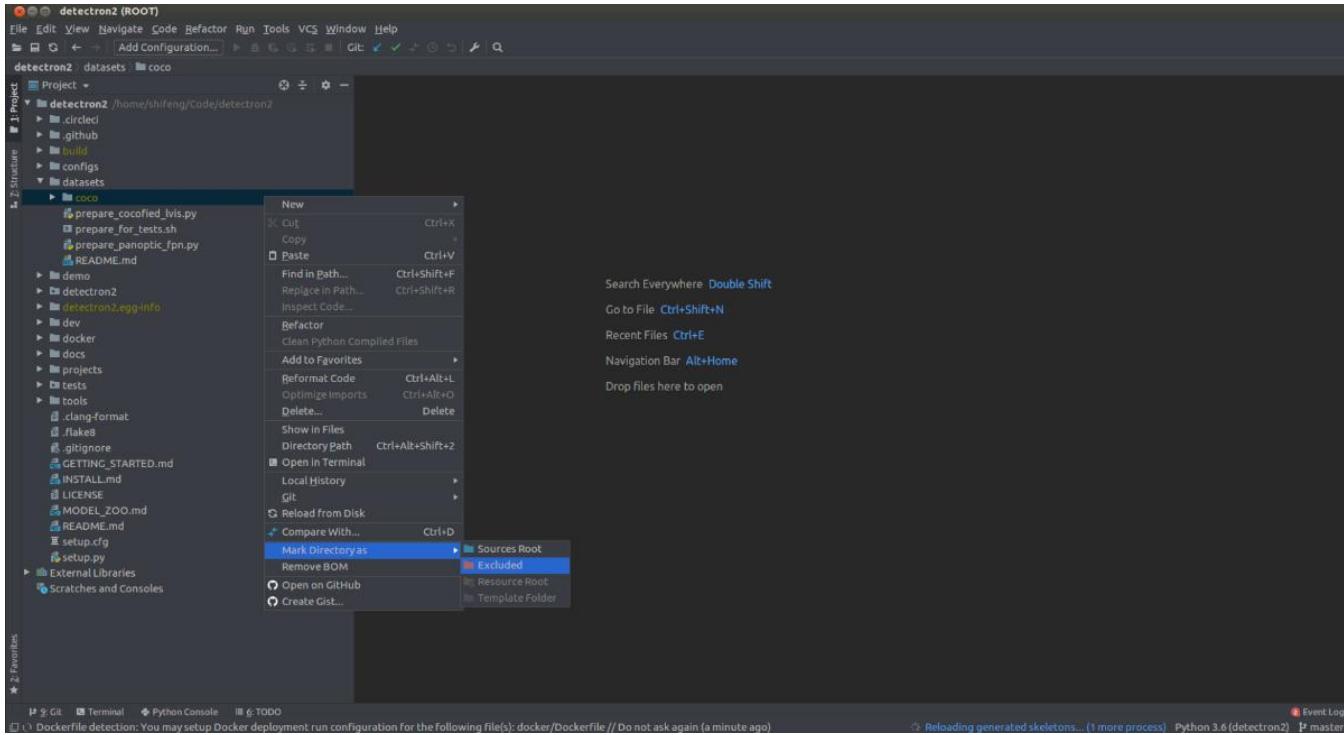
■ PyCharm的项目界面





代码调试软件PyCharm使用

- 对小文件比较多的数据集文件夹进行隐藏，防止PyCharm一直Indexing





代码调试软件PyCharm使用

■ 调试测试代码

1. 右键tools下面的train_net.py文件，选择
create train_net

The screenshot shows the PyCharm IDE interface with the following details:

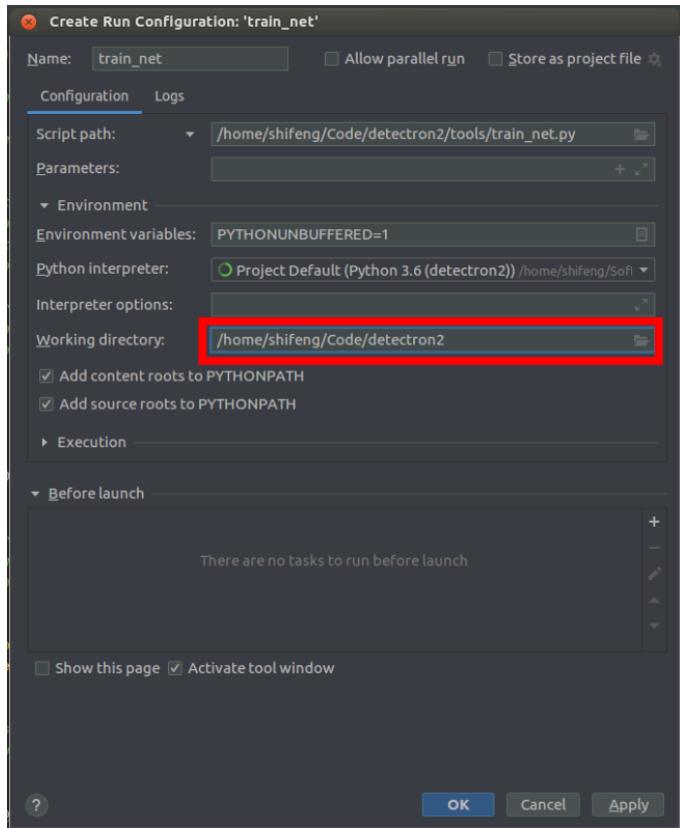
- Project Tree:** Shows the project structure under 'detectron2/tools'. Key files include 'analyze_model.py', 'benchmark.py', 'convert-torchvision-to-d2.py', 'plain_train_net.py', 'README.md', and 'train_net.py'.
- Code Editor:** Displays the content of 'train_net.py'. The code is a Python script for training or evaluating models, with comments explaining its purpose and usage.
- Context Menu:** An open context menu is positioned over the code editor, specifically over the line 'def trainNet(cfg, dataset_name, output_folder=None):'. The menu items include:
 - New
 - Cut (Ctrl+X)
 - Copy
 - Paste (Ctrl+V)
 - Find Usages (Alt+F7)
 - Inspect Code...
 - Refactor
 - Clean Python Compiled Files
 - Add to Favorites
 - Reformat Code (Ctrl+Alt+L)
 - Optimize Imports (Ctrl+Alt+O)
 - Delete...
 - Mark as Plain Text
 - Run 'train_net' (Ctrl+Shift+F10)
 - Debug 'train_net'
 - Run 'train_net' with Coverage
 - Profile 'train_net'
 - Concurrency Diagram for 'train_net'
 - Create 'train_net...' (highlighted)
 - Show In Files
 - File Path (Ctrl+Alt+Shift+F2)
 - Open in Terminal
 - Local History
 - Git
 - Reload from Disk
 - Compare With...
 - Mark Directory as
 - Diagrams
 - Open in GitHub
- Bottom Navigation:** Includes tabs for Git, Terminal, Python Console, and TODO, along with a 'Create run configuration from context' button.



代码调试软件PyCharm使用

■ 调试测试代码

1. 右键tools下面的train_net.py文件，选择
create train_net
2. 把Working directory改到detectron2根目
录下

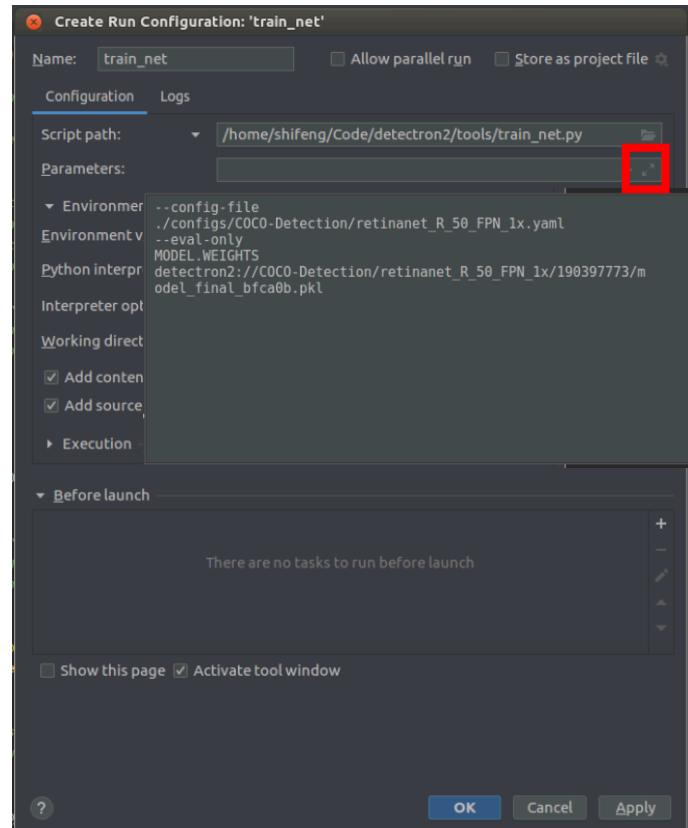




代码调试软件PyCharm使用

■ 调试测试代码

1. 右键tools下面的train_net.py文件，选择
create train_net
2. 把Working directory改到detectron2根目
录下
3. 点击红色框中的按钮，添加运行时所需要
的参数

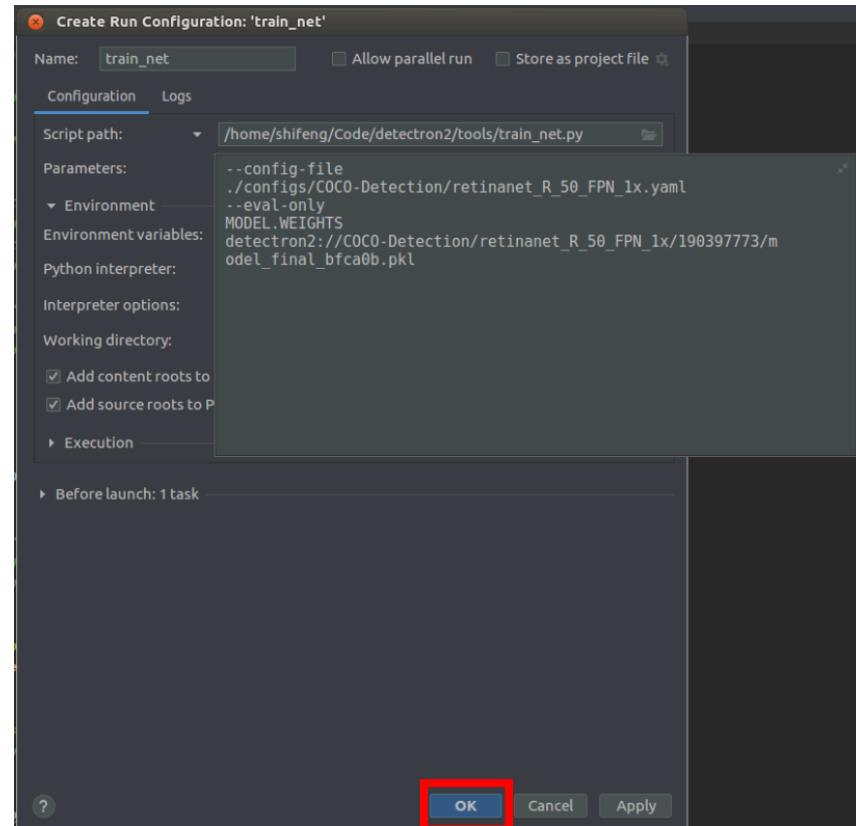




代码调试软件PyCharm使用

■ 调试测试代码

1. 右键tools下面的train_net.py文件，选择
create train_net
2. 把Working directory改到detectron2根目
录下
3. 点击红色框中的按钮，添加运行时所需要
的参数
4. 点击OK回到先前的界面





代码调试软件PyCharm使用

■ 调试测试代码

1. 右键tools下面的train_net.py文件，选择
create train_net
2. 把Working directory改到detectron2根目
录下
3. 点击红色框中的按钮，添加运行时所需要
的参数
4. 点击OK回到先前的界面
5. 点击三角形按钮运行测试代码

```
#!/usr/bin/env python
# Copyright (c) Facebook, Inc. and its affiliates. All Rights Reserved.
"""
Detection Training Script.

This script reads a given config file and runs the training or evaluation.
It is an entry point that is made to train standard models in detectron2.

In order to let one script support training of many models,
this script contains logic that are specific to these built-in models and therefore
may not be suitable for your own project.
For example, your research project perhaps only needs a single "evaluator".

Therefore, we recommend you to use detectron2 as an library and take
this file as an example of how to use the library.
You may want to write your own script with your datasets and other customizations.
"""

import ...

class Trainer(DefaultTrainer):
    """
    We use the "DefaultTrainer" which contains pre-defined default logic for
    standard training workflow. They may not work for you, especially if you
    are working on a new research project. In that case you can write your
    own training loop. You can use "tools/plain_train_net.py" as an example.
    """

    @classmethod
    def build_evaluator(cls, cfg, dataset_name, output_folder=None):
        """
        Create evaluator(s) for a given dataset.
        This uses the special metadata "evaluator_type" associated with each builtin dataset.
        For your own dataset, you can simply create an evaluator manually in your
        script and do not have to worry about the _haskey if-else logic here.
        """
        if output_folder is None:
            output_folder = os.path.join(cfg.OUTPUT_DIR, "Inference")
        evaluation_type = ...
```

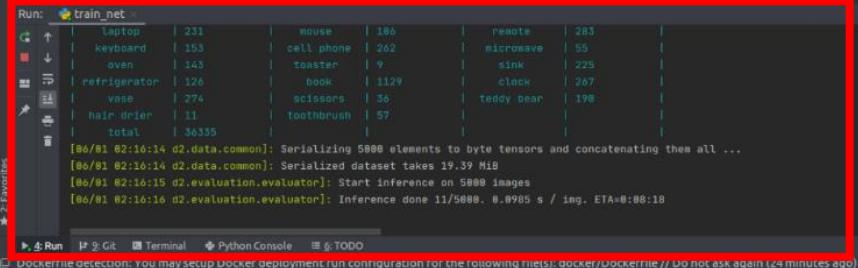
Dockerfile detection: You may setup Docker deployment run configuration for the following file(s): docker/Dockerfile // Do not ask again (22 minutes ago)



代码调试软件PyCharm使用

■ 调试测试代码

1. 右键tools下面的train_net.py文件，选择
create train_net
2. 把Working directory改到detectron2根目
录下
3. 点击红色框中的按钮，添加运行时所需要
的参数
4. 点击OK回到先前的界面
5. 点击三角形按钮运行测试代码
6. 测试代码开始运行，直至结束



The screenshot shows the PyCharm interface with the 'detectron2' project open. The 'train_net.py' file is selected in the 'Project' view. In the bottom-left corner, the 'Run' tool window is open, showing a table of parameters:

	laptop	mouse	remote
Keyboard	231	186	283
oven	153	262	55
refrigerator	143	9	225
vase	126	1129	267
hair drier	274	56	198
total	11	57	
	56355		

Below the table, the terminal output shows the execution of the script:

```
[06/01 02:16:14 d2.data.common]: Serializing 5000 elements to byte tensors and concatenating them all ...
[06/01 02:16:14 d2.data.common]: Serialized dataset takes 19.39 MB
[06/01 02:16:15 d2.evaluation.evaluator]: Start inference on 5000 images
[06/01 02:16:16 d2.evaluation.evaluator]: Inference done 11/5000. 8.8985 s / img. ETA=0:00:18
```

The 'Run' tool window has a red box drawn around its 'Parameters' tab.



代码调试软件PyCharm使用

■ 调试测试代码

1. 右键tools下面的train_net.py文件，选择create train_net
 2. 把Working directory改到detectron2根目录下
 3. 点击红色框中的按钮，添加运行时所需要的参数
 4. 点击OK回到先前的界面
 5. 点击三角形按钮运行测试代码
 6. 测试代码开始运行，直至结束
 7. 在train_net.py的161行点击生成红色的断点，再点击虫子形状的按钮开始调试测试代码

The screenshot shows the PyCharm IDE interface with the following details:

- Title Bar:** "detectron2 - train_net.py (ROOT)"
- Toolbar:** Includes icons for Close Window, Navigate, Code Refactor, Run, Tools, VCS, Window, Help.
- Project Tree (Left):** Shows the project structure under "detectron2". The "tools" directory is selected, highlighted with a red box. Inside "tools", there are files like ".clang-format", ".flake8", ".gitignore", "GETTING_STARTED.md", "INSTALL.md", "LICENSE", "MODEL_ZOO.md", "README.md", "setup.cfg", and "setup.py".
- Code Editor (Right):** Displays the content of "train_net.py". A red box highlights the first line of code: `if __name__ == "__main__":`. Another red box highlights the "Run" button in the toolbar.
- Bottom Status Bar:** Shows "Dockerfile detection: You may setup Docker deployment run configuration for the following file(s): docker/Dockerfile // Do not ask again (27 minutes ago)".



代码调试软件PyCharm使用

■ 调试测试代码

1. 右键tools下面的train_net.py文件，选择
create train_net
2. 把Working directory改到detectron2根目
录下
3. 点击红色框中的按钮，添加运行时所需要
的参数
4. 点击OK回到先前的界面
5. 点击三角形按钮运行测试代码
6. 测试代码开始运行，直至结束
7. 在train_net.py的161行点击生成红色的断
点，再点击虫子形状的按钮开始调试测试
代码
8. 进入调试状态，可以单步调试所有的代码

The screenshot shows the PyCharm IDE interface. The top part displays the code for `train_net.py` with a red circle highlighting the breakpoint at line 161. The bottom part shows the Debug tool window, which is highlighted with a red border. The 'Threads' tab is selected, showing the 'MainThread' thread. The 'Variables' tab is also visible.

```
154     trainer.register_hooks(
155         [hooks.EvalHook(0, lambda: trainer.test_with_TTA(cfg, trainer.model))]
156     )
157     return trainer.train()
158
159
160 if __name__ == "__main__":
161     args = default_argument_parser().parse_args()
162     print("Command Line Args:", args)
163     launch(
164         main,
165         args.num_gpus,
166         num_machines=args.num_machines,
167         machine_rank=args.machine_rank,
168         dist_url=args.dist_url,
169         args=args,
170     )
171
if __name__ == "__main__":

```