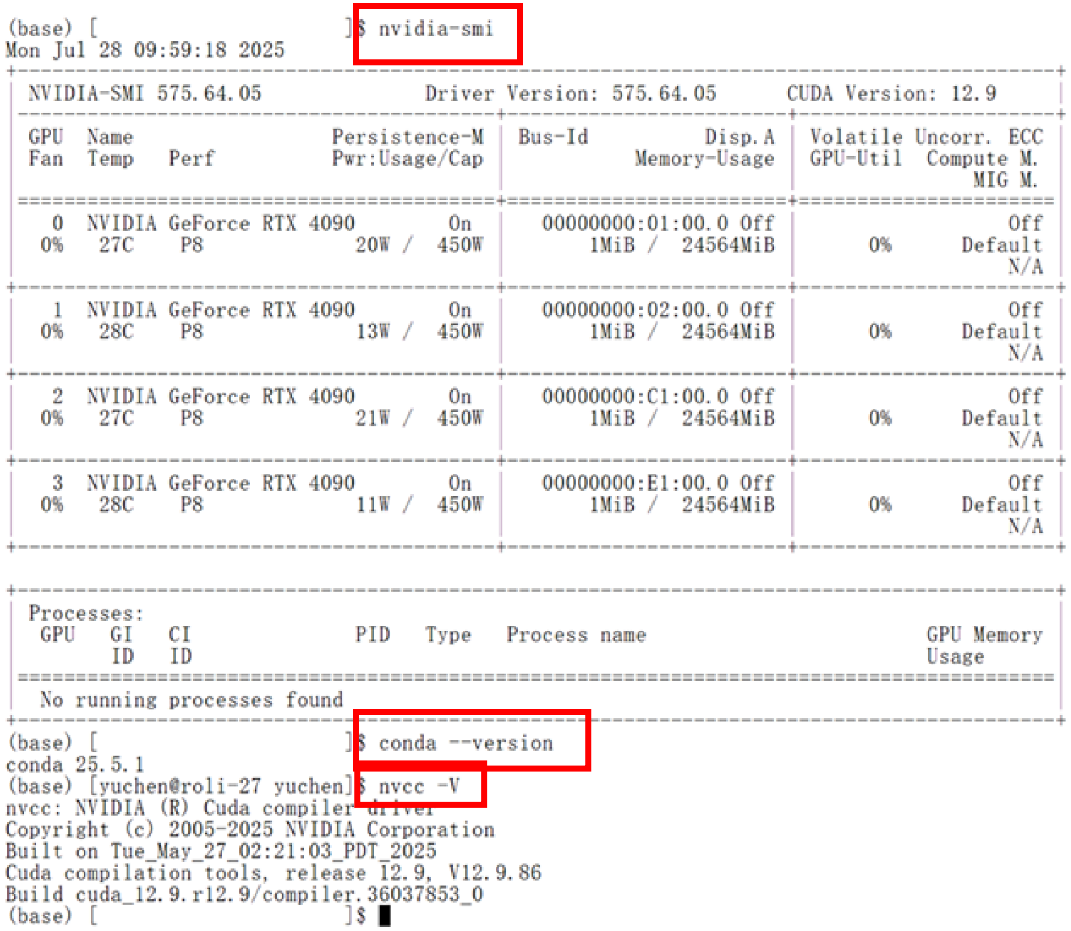
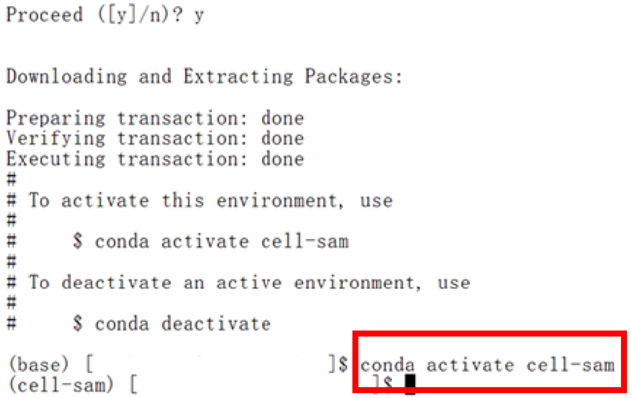
More detailed instructions can be found in [YCWu-nk/Cell_SAM: Large-scale segmentation model facilitates intraoperative histopathology by third harmonic generation microscopy](), you can directly copy them

1. We suggest confirming that the basic environment is installed correctly first:



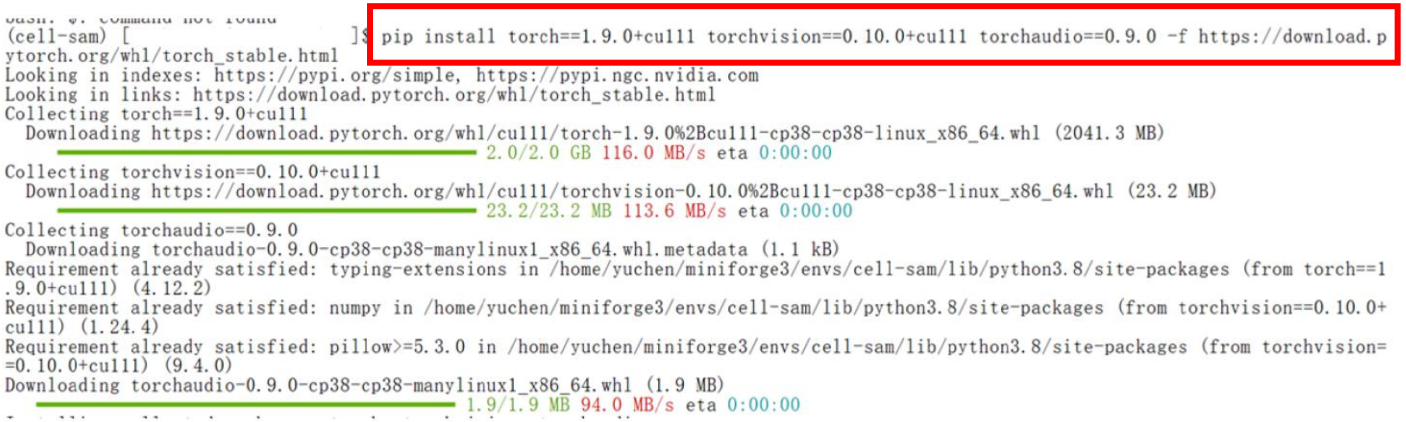
```
(base) [                        ]$ nvidia-smi
Mon Jul 28 09:59:18 2025
+-----------------------------------------------------------------------------------------+
| NVIDIA-SMI 575.64.05              Driver Version: 575.64.05      CUDA Version: 12.9     |
|-----------------------------------------+------------------------+----------------------+
| GPU  Name                 Persistence-M | Bus-Id          Disp.A | Volatile Uncorr. ECC |
| Fan  Temp   Perf          Pwr:Usage/Cap |           Memory-Usage | GPU-Util  Compute M. |
|                                         |                        |               MIG M. |
|=========================================+========================+======================|
|   0  NVIDIA GeForce RTX 4090        On  |   00000000:01:00.0 Off |                  Off |
|  0%   27C    P8             20W /  450W |       1MiB /  24564MiB |      0%      Default |
|                                         |                        |                  N/A |
+-----------------------------------------+------------------------+----------------------+
|   1  NVIDIA GeForce RTX 4090        On  |   00000000:02:00.0 Off |                  Off |
|  0%   28C    P8             13W /  450W |       1MiB /  24564MiB |      0%      Default |
|                                         |                        |                  N/A |
+-----------------------------------------+------------------------+----------------------+
|   2  NVIDIA GeForce RTX 4090        On  |   00000000:C1:00.0 Off |                  Off |
|  0%   27C    P8             21W /  450W |       1MiB /  24564MiB |      0%      Default |
|                                         |                        |                  N/A |
+-----------------------------------------+------------------------+----------------------+
|   3  NVIDIA GeForce RTX 4090        On  |   00000000:E1:00.0 Off |                  Off |
|  0%   28C    P8             11W /  450W |       1MiB /  24564MiB |      0%      Default |
|                                         |                        |                  N/A |
+-----------------------------------------+------------------------+----------------------+

+-----------------------------------------------------------------------------------------+
| Processes:                                                                              |
|  GPU   GI   CI              PID   Type   Process name                        GPU Memory |
|        ID   ID                                                               Usage      |
|=========================================================================================|
|  No running processes found                                                             |
+-----------------------------------------------------------------------------------------+
(base) [                        ]$ conda --version
conda 25.5.1
(base) [yuchen@roli-27 yuchen]$ nvcc -V
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2025 NVIDIA Corporation
Built on Tue_May_27_02:21:03_PDT_2025
Cuda compilation tools, release 12.9, V12.9.86
Build cuda_12.9.r12.9/compiler.36037853_0
(base) [                        ]$ █
```

2. Creating a conda environment, and install ***the GPU version*** of PyTorch in your preferred way:

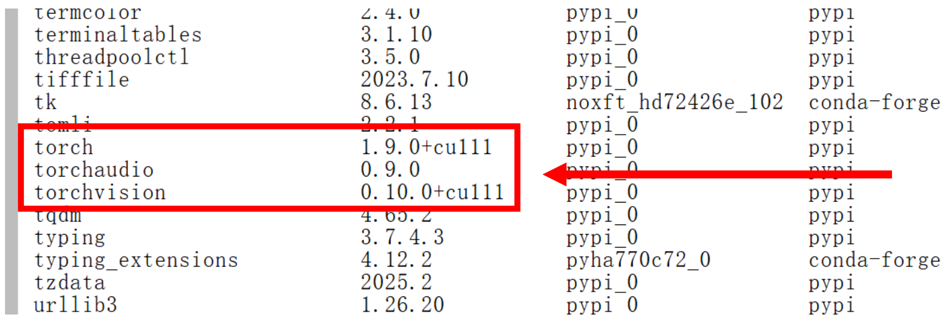


```
Proceed ([y]/n)? y


Downloading and Extracting Packages:

Preparing transaction: done
Verifying transaction: done
Executing transaction: done
#
# To activate this environment, use
#
#     $ conda activate cell-sam
#
# To deactivate an active environment, use
#
#     $ conda deactivate

(base) [                        ]$ conda activate cell-sam
(cell-sam) [                     ]$ █
```

Install the GPU version of PyTorch using any method you prefer (the CPU version is not compatible with other installation packages).



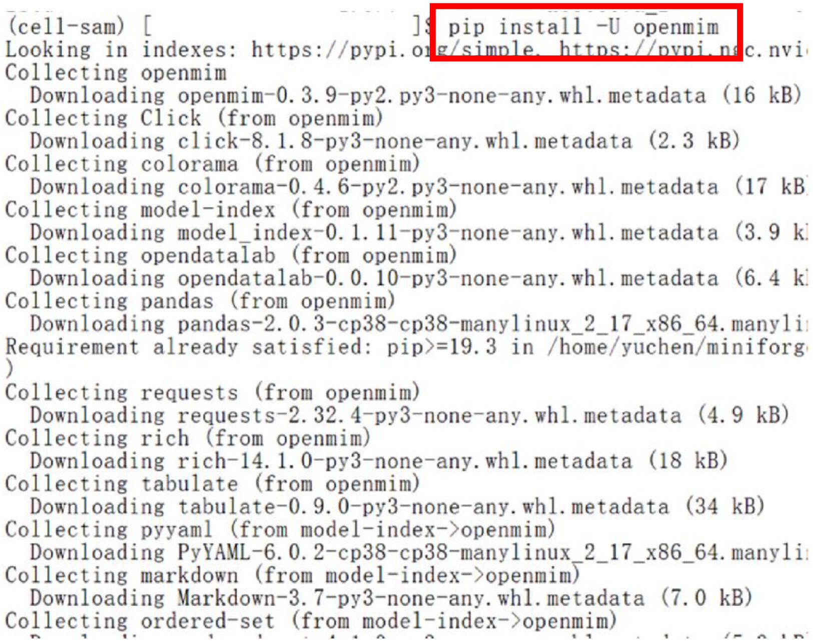
```
bash: $: command not found
(cell-sam) [                        ]$ pip install torch==1.9.0+cu111 torchvision==0.10.0+cu111 torchaudio==0.9.0 -f https://download.p
ytorch.org/whl/torch_stable.html
Looking in indexes: https://pypi.org/simple, https://pypi.ngc.nvidia.com
Looking in links: https://download.pytorch.org/whl/torch_stable.html
Collecting torch==1.9.0+cu111
  Downloading https://download.pytorch.org/whl/cu111/torch-1.9.0%2Bcu111-cp38-cp38-linux_x86_64.whl (2041.3 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.0/2.0 GB 116.0 MB/s eta 0:00:00
Collecting torchvision==0.10.0+cu111
  Downloading https://download.pytorch.org/whl/cu111/torchvision-0.10.0%2Bcu111-cp38-cp38-linux_x86_64.whl (23.2 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 23.2/23.2 MB 113.6 MB/s eta 0:00:00
Collecting torchaudio==0.9.0
  Downloading torchaudio-0.9.0-cp38-cp38-manylinux1_x86_64.whl.metadata (1.1 kB)
Requirement already satisfied: typing-extensions in /home/yuchen/miniforge3/envs/cell-sam/lib/python3.8/site-packages (from torch==1
.9.0+cu111) (4.12.2)
Requirement already satisfied: numpy in /home/yuchen/miniforge3/envs/cell-sam/lib/python3.8/site-packages (from torchvision==0.10.0+
cu111) (1.24.4)
Requirement already satisfied: pillow>=5.3.0 in /home/yuchen/miniforge3/envs/cell-sam/lib/python3.8/site-packages (from torchvision=
=0.10.0+cu111) (9.4.0)
Downloading torchaudio-0.9.0-cp38-cp38-manylinux1_x86_64.whl (1.9 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 1.9/1.9 MB 94.0 MB/s eta 0:00:00
```



```
termcolor                  2.4.0              pypi_0               pypi
terminaltables             3.1.10             pypi_0               pypi
threadpoolctl              3.5.0              pypi_0               pypi
tifffile                   2023.7.10          pypi_0               pypi
tk                         8.6.13             noxft_hd72426e_102   conda-forge
tomli                      2.2.1              pypi_0               pypi
torch                      1.9.0+cu111        pypi_0               pypi
torchaudio                 0.9.0              pypi_0               pypi
torchvision                0.10.0+cu111       pypi_0               pypi
tqdm                       4.65.2             pypi_0               pypi
typing                     3.7.4.3            pypi_0               pypi
typing_extensions          4.12.2             pyha770c72_0         conda-forge
tzdata                     2025.2             pypi_0               pypi
urllib3                    1.26.20            pypi_0               pypi
```

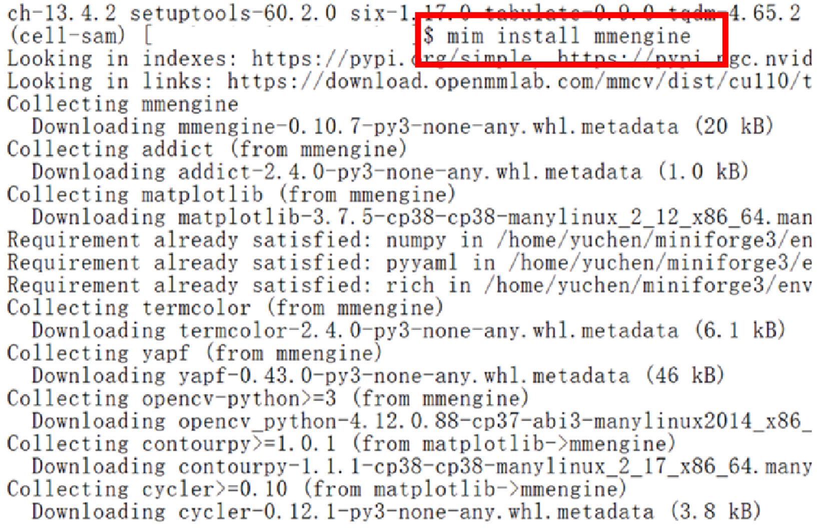
Please confirm again through ‘conda list’
‘+cu’ is right
‘cpu’ is wrong

3. Install openmim, it's similar to pip:



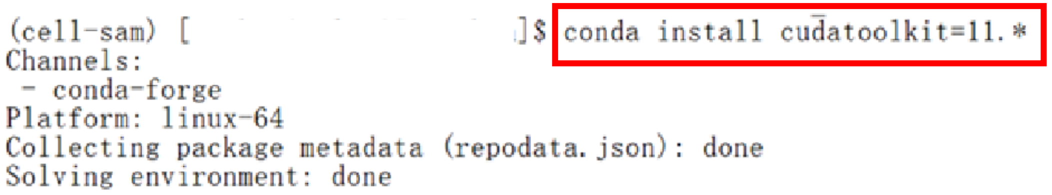
```
(cell-sam) [                              ]$ pip install -U openmim
Looking in indexes: https://pypi.org/simple, https://pypi.ngc.nvi
Collecting openmim
  Downloading openmim-0.3.9-py2.py3-none-any.whl.metadata (16 kB)
Collecting Click (from openmim)
  Downloading click-8.1.8-py3-none-any.whl.metadata (2.3 kB)
Collecting colorama (from openmim)
  Downloading colorama-0.4.6-py2.py3-none-any.whl.metadata (17 kB
Collecting model-index (from openmim)
  Downloading model_index-0.1.11-py3-none-any.whl.metadata (3.9 k
Collecting opendatalab (from openmim)
  Downloading opendatalab-0.0.10-py3-none-any.whl.metadata (6.4 k
Collecting pandas (from openmim)
  Downloading pandas-2.0.3-cp38-cp38-manylinux_2_17_x86_64.manyli
Requirement already satisfied: pip>=19.3 in /home/yuchen/miniforg
)
Collecting requests (from openmim)
  Downloading requests-2.32.4-py3-none-any.whl.metadata (4.9 kB)
Collecting rich (from openmim)
  Downloading rich-14.1.0-py3-none-any.whl.metadata (18 kB)
Collecting tabulate (from openmim)
  Downloading tabulate-0.9.0-py3-none-any.whl.metadata (34 kB)
Collecting pyyaml (from model-index->openmim)
  Downloading PyYAML-6.0.2-cp38-cp38-manylinux_2_17_x86_64.manyli
Collecting markdown (from model-index->openmim)
  Downloading Markdown-3.7-py3-none-any.whl.metadata (7.0 kB)
Collecting ordered-set (from model-index->openmim)
```

4. Install mmengine , using mim:



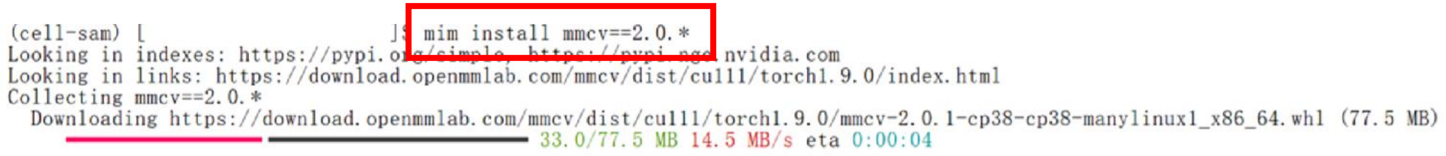
```
ch-13.4.2 setuptools-60.2.0 six-1.17.0 tabulate-0.9.0 tqdm-4.65.2
(cell-sam) [                              ]$ mim install mmengine
Looking in indexes: https://pypi.org/simple, https://pypi.ngc.nvid
Looking in links: https://download.openmmlab.com/mmcv/dist/cu110/t
Collecting mmengine
  Downloading mmengine-0.10.7-py3-none-any.whl.metadata (20 kB)
Collecting addict (from mmengine)
  Downloading addict-2.4.0-py3-none-any.whl.metadata (1.0 kB)
Collecting matplotlib (from mmengine)
  Downloading matplotlib-3.7.5-cp38-cp38-manylinux_2_12_x86_64.man
Requirement already satisfied: numpy in /home/yuchen/miniforge3/en
Requirement already satisfied: pyyaml in /home/yuchen/miniforge3/e
Requirement already satisfied: rich in /home/yuchen/miniforge3/env
Collecting termcolor (from mmengine)
  Downloading termcolor-2.4.0-py3-none-any.whl.metadata (6.1 kB)
Collecting yapf (from mmengine)
  Downloading yapf-0.43.0-py3-none-any.whl.metadata (46 kB)
Collecting opencv-python>=3 (from mmengine)
  Downloading opencv_python-4.12.0.88-cp37-abi3-manylinux2014_x86_
Collecting contourpy>=1.0.1 (from matplotlib->mmengine)
  Downloading contourpy-1.1.1-cp38-cp38-manylinux_2_17_x86_64.many
Collecting cycler>=0.10 (from matplotlib->mmengine)
  Downloading cycler-0.12.1-py3-none-any.whl.metadata (3.8 kB)
```

5. Install CUDAToolkit by hand , if needed (If it is uncertain whether the server has pre configured CUDA, proceed with this step):



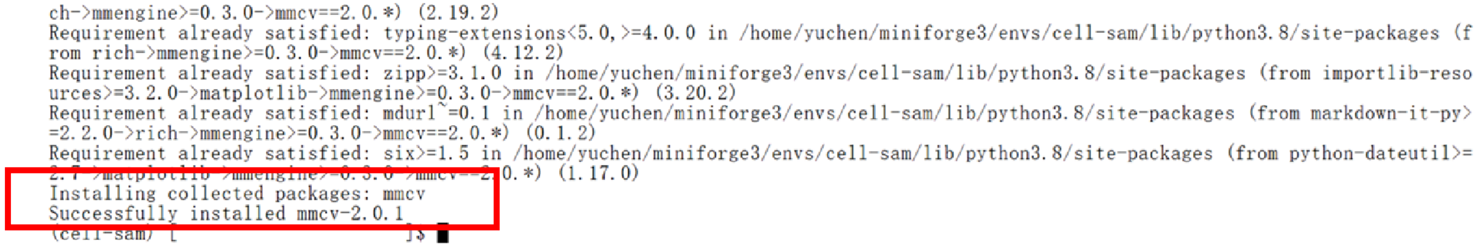
```
(cell-sam) [                    ]$ conda install cudatoolkit=11.*
Channels:
 - conda-forge
Platform: linux-64
Collecting package metadata (repodata.json): done
Solving environment: done
```

6. The most important step: configure MMCV:



```
(cell-sam) [                              ]$ mim install mmcv==2.0.*
Looking in indexes: https://pypi.org/simple, https://pypi.ngc.nvidia.com
Looking in links: https://download.openmmlab.com/mmcv/dist/cu111/torch1.9.0/index.html
Collecting mmcv==2.0.*
  Downloading https://download.openmmlab.com/mmcv/dist/cu111/torch1.9.0/mmcv-2.0.1-cp38-cp38-manylinux1_x86_64.whl (77.5 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 33.0/77.5 MB 14.5 MB/s eta 0:00:04
```

Wait for a few minutes. If there are no errors, the installation could be successful

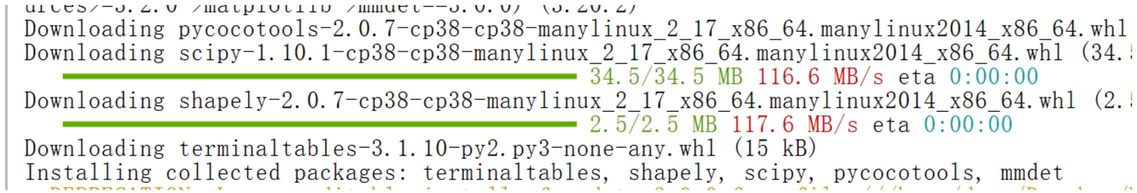


```
ch->mmengine>=0.3.0->mmcv==2.0.*) (2.19.2)
Requirement already satisfied: typing-extensions<5.0,>=4.0.0 in /home/yuchen/miniforge3/envs/cell-sam/lib/python3.8/site-packages (f
rom rich->mmengine>=0.3.0->mmcv==2.0.*) (4.12.2)
Requirement already satisfied: zipp>=3.1.0 in /home/yuchen/miniforge3/envs/cell-sam/lib/python3.8/site-packages (from importlib-reso
urces>=3.2.0->matplotlib->mmengine>=0.3.0->mmcv==2.0.*) (3.20.2)
Requirement already satisfied: mdurl~=0.1 in /home/yuchen/miniforge3/envs/cell-sam/lib/python3.8/site-packages (from markdown-it-py>
=2.2.0->rich->mmengine>=0.3.0->mmcv==2.0.*) (0.1.2)
Requirement already satisfied: six>=1.5 in /home/yuchen/miniforge3/envs/cell-sam/lib/python3.8/site-packages (from python-dateutil>=
2.7->matplotlib->mmengine>=0.3.0->mmcv==2.0.*) (1.17.0)
Installing collected packages: mmcv
Successfully installed mmcv-2.0.1
(cell-sam) [                              ]$ █
```

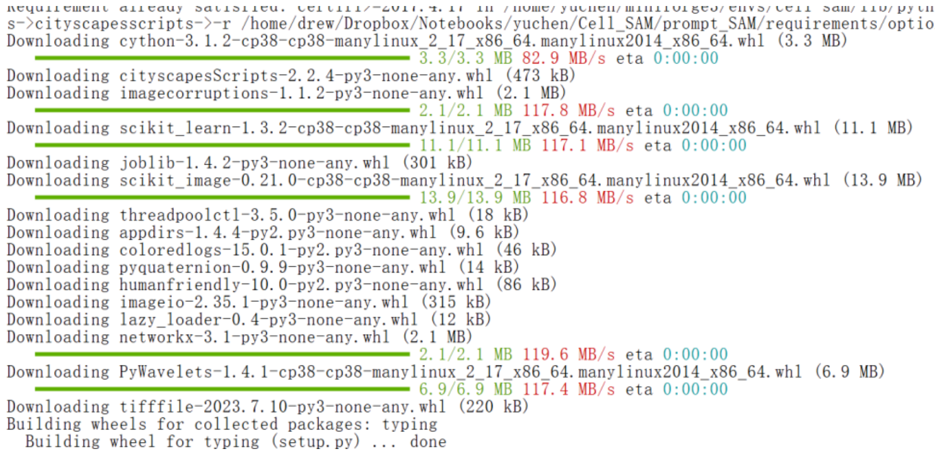
If there is an error, please check the above steps, such as whether the CUDAToolkit is installed correctly, and try different versions of PyTorch (GPU version as a prerequisite), which can be fixed after a simple one or two version changes. Or find more solutions from Issues · open-mmlab/mmdetection (We have deployed it on more than 6 servers, including different CPUs and graphics cards (1080Ti, 3060, V100, 3080, 3090Ti, 4090), and different operating systems (Windows, Ubuntu, Arch Linux). Please report any bugs on your server to us.).

6. Install other used packages and SAM :

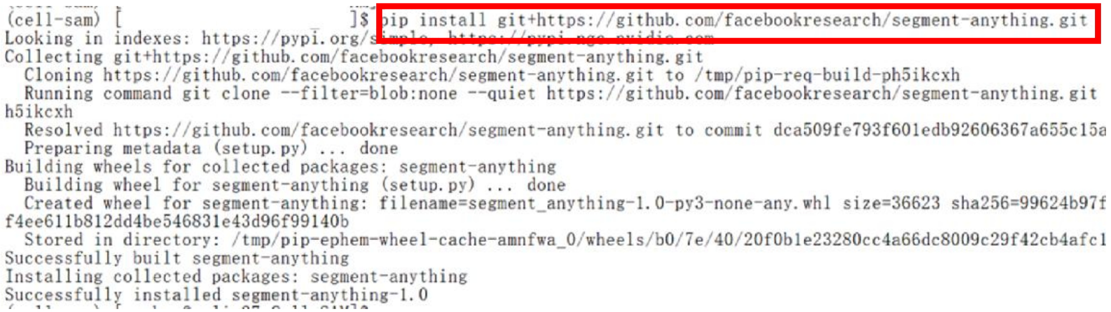
cd prompt_SAM; pip install -v -e .; pip install -r requirements.txt



```
Downloading pycocotools-2.0.7-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl
Downloading scipy-1.10.1-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (34.
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 34.5/34.5 MB 116.6 MB/s eta 0:00:00
Downloading shapely-2.0.7-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (2.
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.5/2.5 MB 117.6 MB/s eta 0:00:00
Downloading terminaltables-3.1.10-py2.py3-none-any.whl (15 kB)
Installing collected packages: terminaltables, shapely, scipy, pycocotools, mmdet
```



```
Requirement already satisfied: certifi>=2017.4.17 in /home/yuchen/miniforge3/envs/cell_sam/lib/python3.8
s->cityscapesscripts->-r /home/drew/Dropbox/Notebooks/yuchen/Cell_SAM/prompt_SAM/requirements/optional.t
Downloading cython-3.1.2-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (3.3 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 3.3/3.3 MB 82.9 MB/s eta 0:00:00
Downloading cityscapesScripts-2.2.4-py3-none-any.whl (473 kB)
Downloading imagecorruptions-1.1.2-py3-none-any.whl (2.1 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.1/2.1 MB 117.8 MB/s eta 0:00:00
Downloading scikit_learn-1.3.2-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (11.1 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 11.1/11.1 MB 117.1 MB/s eta 0:00:00
Downloading joblib-1.4.2-py3-none-any.whl (301 kB)
Downloading scikit_image-0.21.0-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (13.9 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 13.9/13.9 MB 116.8 MB/s eta 0:00:00
Downloading threadpoolctl-3.5.0-py3-none-any.whl (18 kB)
Downloading appdirs-1.4.4-py2.py3-none-any.whl (9.6 kB)
Downloading coloredlogs-15.0.1-py2.py3-none-any.whl (46 kB)
Downloading pyquaternion-0.9.9-py3-none-any.whl (14 kB)
Downloading humanfriendly-10.0-py2.py3-none-any.whl (86 kB)
Downloading imageio-2.35.1-py3-none-any.whl (315 kB)
Downloading lazy_loader-0.4-py3-none-any.whl (12 kB)
Downloading networkx-3.1-py3-none-any.whl (2.1 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 2.1/2.1 MB 119.6 MB/s eta 0:00:00
Downloading PyWavelets-1.4.1-cp38-cp38-manylinux_2_17_x86_64.manylinux2014_x86_64.whl (6.9 MB)
   ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 6.9/6.9 MB 117.4 MB/s eta 0:00:00
Downloading tifffile-2023.7.10-py3-none-any.whl (220 kB)
Building wheels for collected packages: typing
  Building wheel for typing (setup.py) ... done
```



```
(cell-sam) [                    ]$ pip install git+https://github.com/facebookresearch/segment-anything.git
Looking in indexes: https://pypi.org/simple, https://pypi.ngc.nvidia.com
Collecting git+https://github.com/facebookresearch/segment-anything.git
  Cloning https://github.com/facebookresearch/segment-anything.git to /tmp/pip-req-build-ph5ikcxh
  Running command git clone --filter=blob:none --quiet https://github.com/facebookresearch/segment-anything.git
h5ikcxh
  Resolved https://github.com/facebookresearch/segment-anything.git to commit dca509fe793f601edb92606367a655c15a
  Preparing metadata (setup.py) ... done
Building wheels for collected packages: segment-anything
  Building wheel for segment-anything (setup.py) ... done
  Created wheel for segment-anything: filename=segment_anything-1.0-py3-none-any.whl size=36623 sha256=99624b97f
f4ee611b812dd4be546831e43d96f99140b
  Stored in directory: /tmp/pip-ephem-wheel-cache-amnfwa_0/wheels/b0/7e/40/20f0b1e23280cc4a66dc8009c29f42cb4afc1
Successfully built segment-anything
Installing collected packages: segment-anything
Successfully installed segment-anything-1.0
```

Normally, SAM will automatically download the weight file called, but you can also manually download it, if SAM cannot access its website. We use 'sam_vit_b_01ec64.pth'

7. Test demo:

Download the dataset and weight files as instructed by GitHub, and place them in the required location. (prompt SAM)

datasam — Put the data here

outputs — Wait for the results here

det4sam_spark_8xb32_r50-300e.py — Config file of the detection model

detector_sam_demo.py

detector_sam2_demo.py

Python files combining SAM/SAM2 and detection model

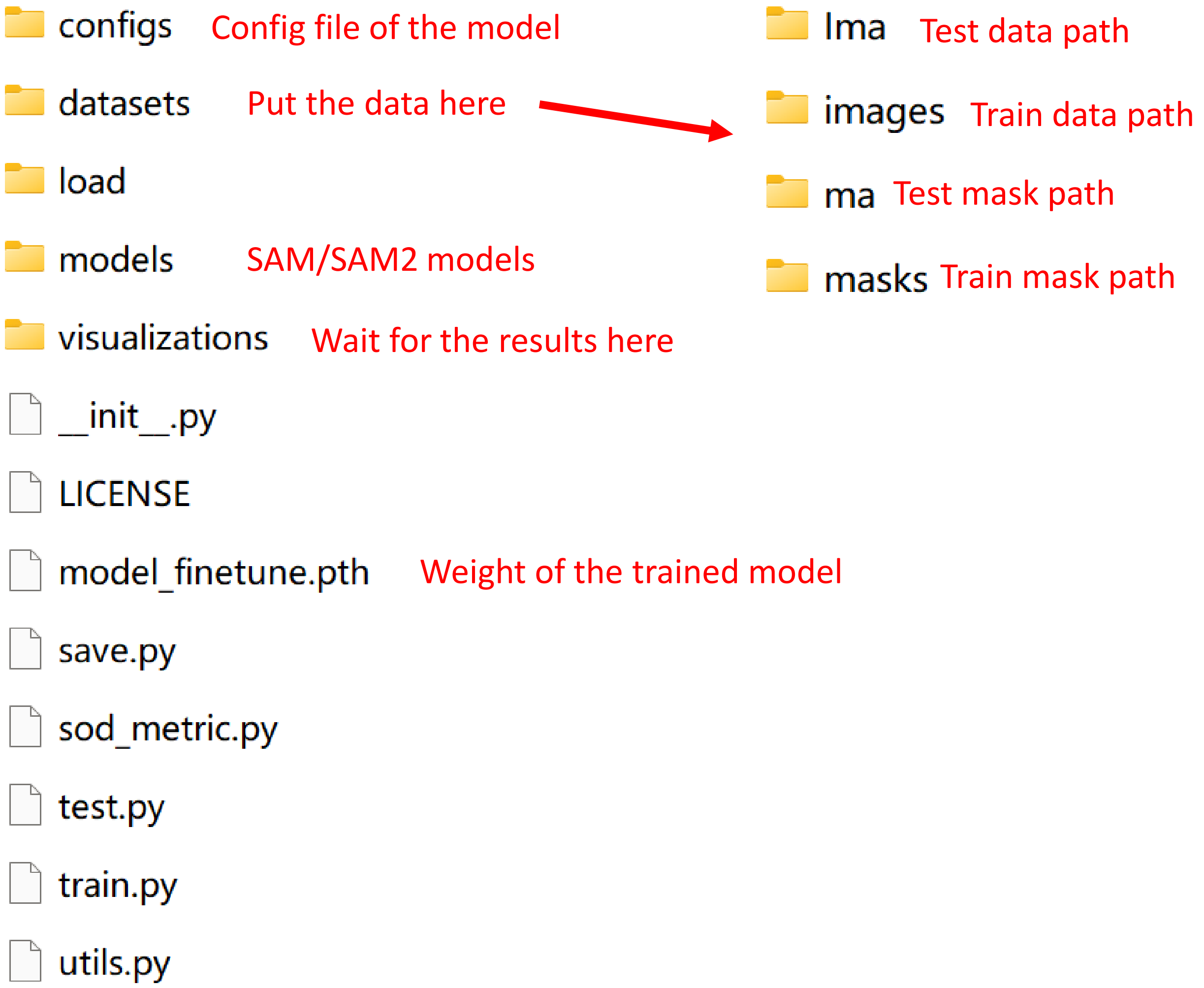
epoch_300.pth — Weight of the detection model

sam_vit_b_01ec64.pth — Weight of SAM

utils.py

7. Test demo:

Download the dataset and weight files as instructed by GitHub, and place them in the required location. (finetune SAM)



configs Config file of the model

datasets Put the data here

load

models SAM/SAM2 models

visualizations Wait for the results here

__init__.py

LICENSE

model_finetune.pth Weight of the trained model

save.py

sod_metric.py

test.py

train.py

utils.py

Ima Test data path

images Train data path

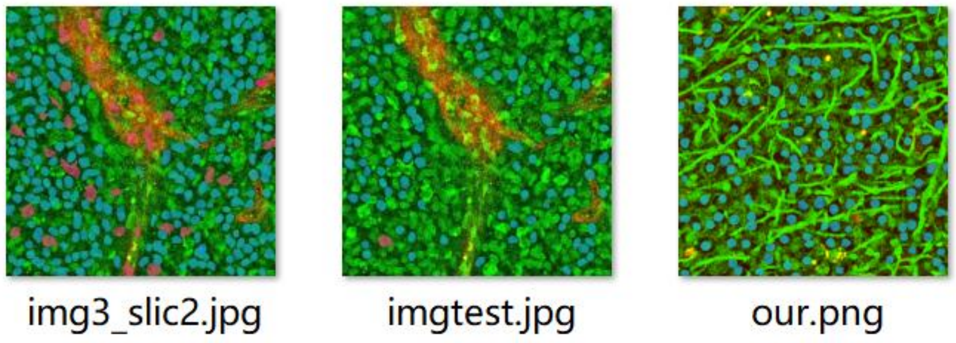
ma Test mask path

masks Train mask path

8. Get results:

Enter instructions, wait for the program to run, and obtain visible results.

Prompt-SAM like these, in outputs

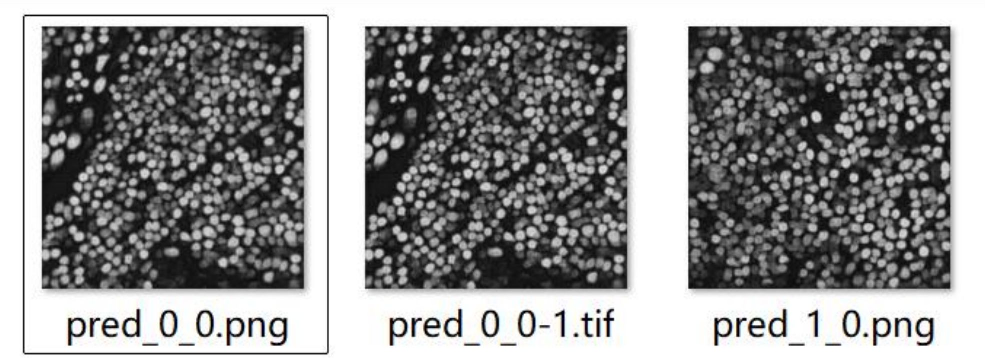


img3_slic2.jpg

imgtest.jpg

our.png

Finetined-SAM like these , in visualizations



pred_0_0.png

pred_0_0-1.tif

pred_1_0.png