

Swin Transformer实战-Windows

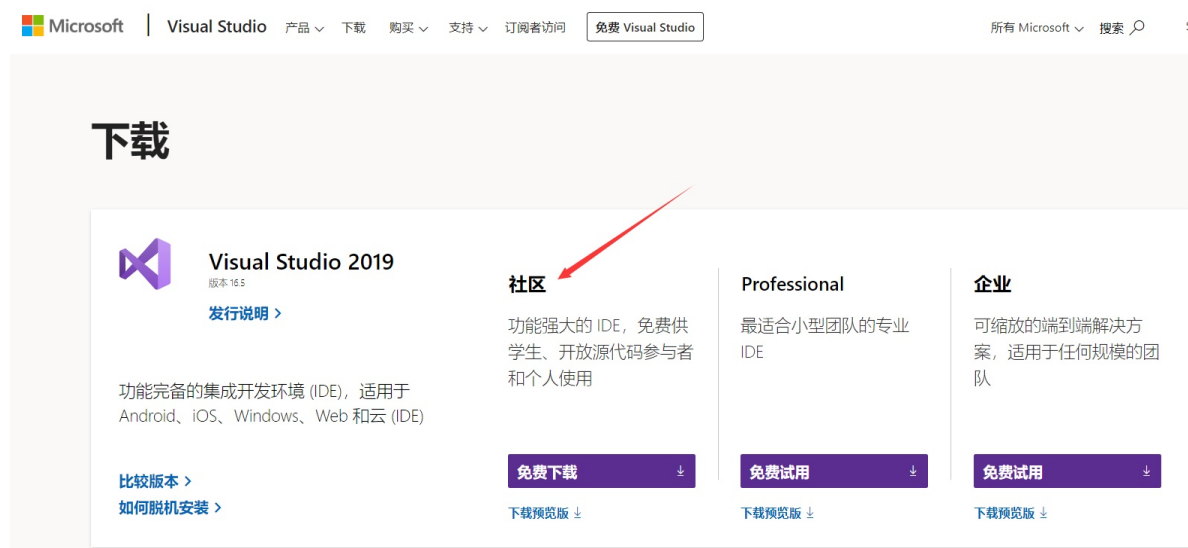
课程演示环境: Windows10, cuda 10.2, cudnn7.6.5

1 软件安装

1) 安装Visual Studio 2019

下载Visual Studio 社区版

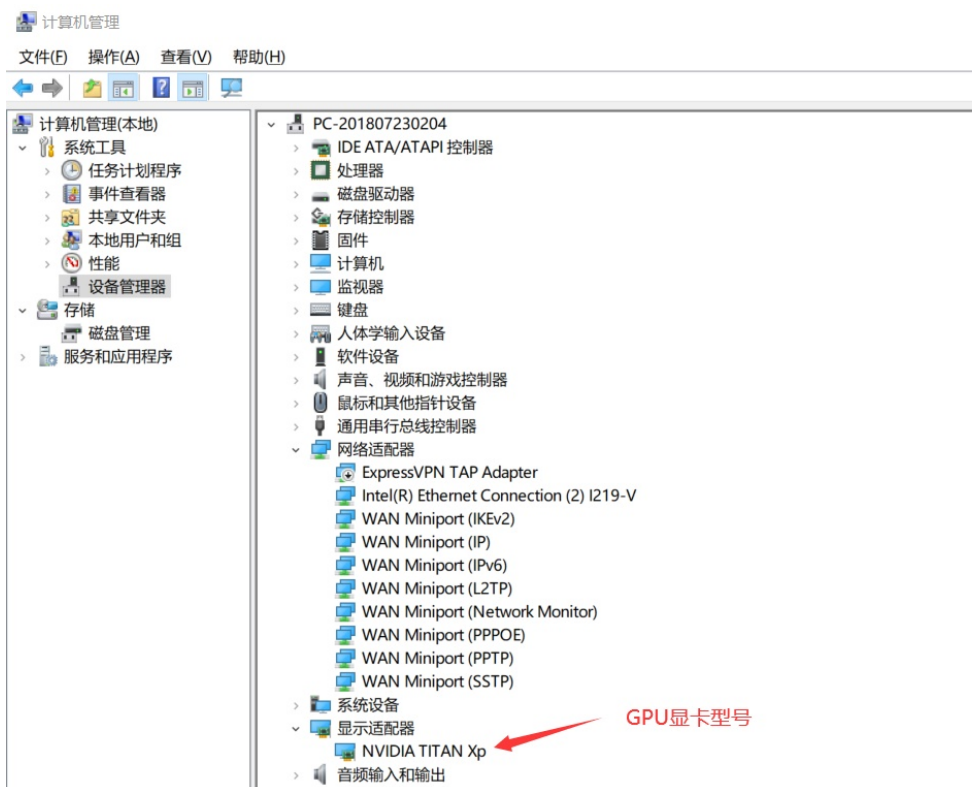
下载链接: <https://visualstudio.microsoft.com/zh-hans/downloads/>



注意: 安装时勾选“Python开发”和“C++桌面开发”

2) 下载和安装nvidia显卡驱动

首先要在设备管理器中查看你的显卡型号, 比如在这里可以看到我的显卡型号为Titan XP。



NVIDIA 驱动下载: <https://www.nvidia.cn/Download/index.aspx?lang=cn>

下载对应你的英伟达显卡驱动。

驱动程序下载

NVIDIA > 驱动程序下载

NVIDIA 驱动程序下载

选项 1: 手动查找适用于我的 NVIDIA 产品的驱动程序。

| | |
|-------|----------------------|
| 产品类型: | TITAN |
| 产品系列: | NVIDIA TITAN Series |
| 产品家族: | NVIDIA TITAN Xp |
| 操作系统: | Windows 10 64-bit |
| 下载类型: | Studio 驱动程序 (SD) |
| 语言: | Chinese (Simplified) |

下载之后就是简单的下一步直到完成。

完成之后, 在cmd中输入执行:

```
nvidia-smi
```

如果有错误:

'nvidia-smi' 不是内部或外部命令, 也不是可运行的程序
或批处理文件。

把C:\Program Files\NVIDIA Corporation\NVSMI添加到环境变量的path中。再重新打开cmd窗口。

如果输出下图所示的显卡信息, 说明你的驱动安装成功。

```
C:\Windows 命令处理程序
Microsoft Windows [版本 10.0.17134.472]
(c) 2018 Microsoft Corporation。保留所有权利。

C:\Windows\System32>nvidia-smi
Thu Apr 30 17:22:50 2020
```

| NVIDIA-SMI 441.22 | | | | Driver Version: 441.22 | | CUDA Version: 10.2 | |
|-------------------|----------|----------|------------------|------------------------|----------|--------------------|---------|
| GPU | Name | TCC/WDDM | Bus-Id | Disp.A | Volatile | Uncorr. | ECC |
| Fan | Temp | Perf | Pwr:Usage/Cap | Memory-Usage | GPU-Util | Compute | M. |
| 0 | TITAN Xp | WDDM | 00000000:65:00:0 | On | | | N/A |
| 23% | 26C | P8 | 11W / 250W | 727MiB / 12288MiB | 1% | | Default |

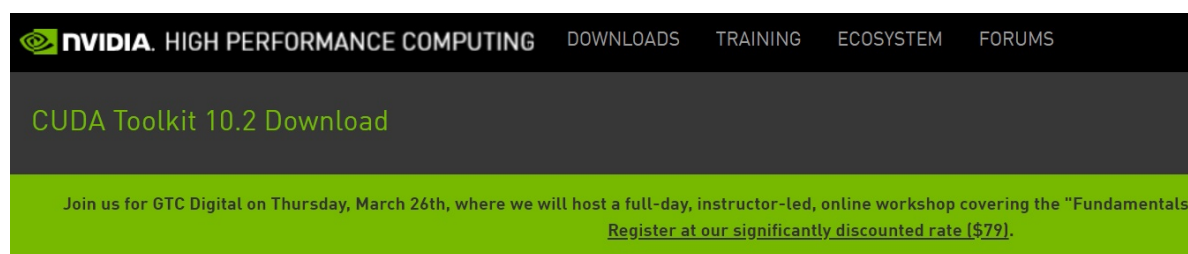
| Processes: | | | | GPU Memory Usage |
|------------|-------|------|--|------------------|
| GPU | PID | Type | Process name | |
| 0 | 1160 | C+G | Insufficient Permissions | N/A |
| 0 | 7316 | C+G | C:\Windows\explorer.exe | N/A |
| 0 | 7412 | C+G | ...t_cw5nlh2txyewy\ShellExperienceHost.exe | N/A |
| 0 | 7624 | C+G | C:\Program Files\Typora\Typora.exe | N/A |
| 0 | 8220 | C+G | ...dows.Cortana_cw5nlh2txyewy\SearchUI.exe | N/A |
| 0 | 9064 | C+G | ...hell.Experiences.TextInput.InputApp.exe | N/A |
| 0 | 11856 | C+G | ...ta\Roaming\360se6\Application\360se.exe | N/A |

注：图中的 CUDA Version是当前Driver版本能支持的最高的CUDA版本

3) 下载CUDA

CUDA用的是10.2版本

cuda下载链接: https://developer.nvidia.com/cuda-downloads?target_os=Windows&target_arch=x86_64&target_version=10&target_type=exe-local



Home > High Performance Computing > CUDA Toolkit > CUDA Toolkit 10.2 Download

Select Target Platform

Click on the green buttons that describe your target platform. Only supported platforms will be shown.

| | | | |
|------------------|---------------|-------------|----------------|
| Operating System | Windows | Linux | Mac OSX |
| Architecture | x86_64 | | |
| Version | 10 | 8.1 | 7 |
| | | Server 2019 | Server 2016 |
| | | | Server 2012 R2 |
| Installer Type | exe (network) | exe (local) | |

下载后得到文件: cuda_10.2.89_441.22_win10.exe

4) 下载cuDNN

cuda下载地址: <https://developer.nvidia.com/cudnn>

需要有账号

cuDNN Download

NVIDIA cuDNN is a GPU-accelerated library of primitives for deep neural networks.

☒ I Agree To the Terms of the [cuDNN Software License Agreement](#)

Note: Please refer to the [Installation Guide](#) for release prerequisites, including supported GPU architectures and compute capabilities, before downloading.

For more information, refer to the cuDNN Developer Guide, Installation Guide and Release Notes on the [Deep Learning SDK Documentation](#) web page.

Download cuDNN v7.6.5 (November 18th, 2019), for CUDA 10.2

Library for Windows, Mac, Linux, Ubuntu and RedHat/Centos(x86_64architecture)

cuDNN Library for Windows 7

cuDNN Library for Windows 10

cuDNN Library for Linux

cuDNN Runtime Library for Ubuntu18.04 [Deb]

cuDNN Developer Library for Ubuntu18.04 [Deb]

cuDNN Code Samples and User Guide for Ubuntu18.04 [Deb]

下载后得到文件: cudnn-10.2-windows10-x64-v7.6.5.32.zip

5) 安装cuda

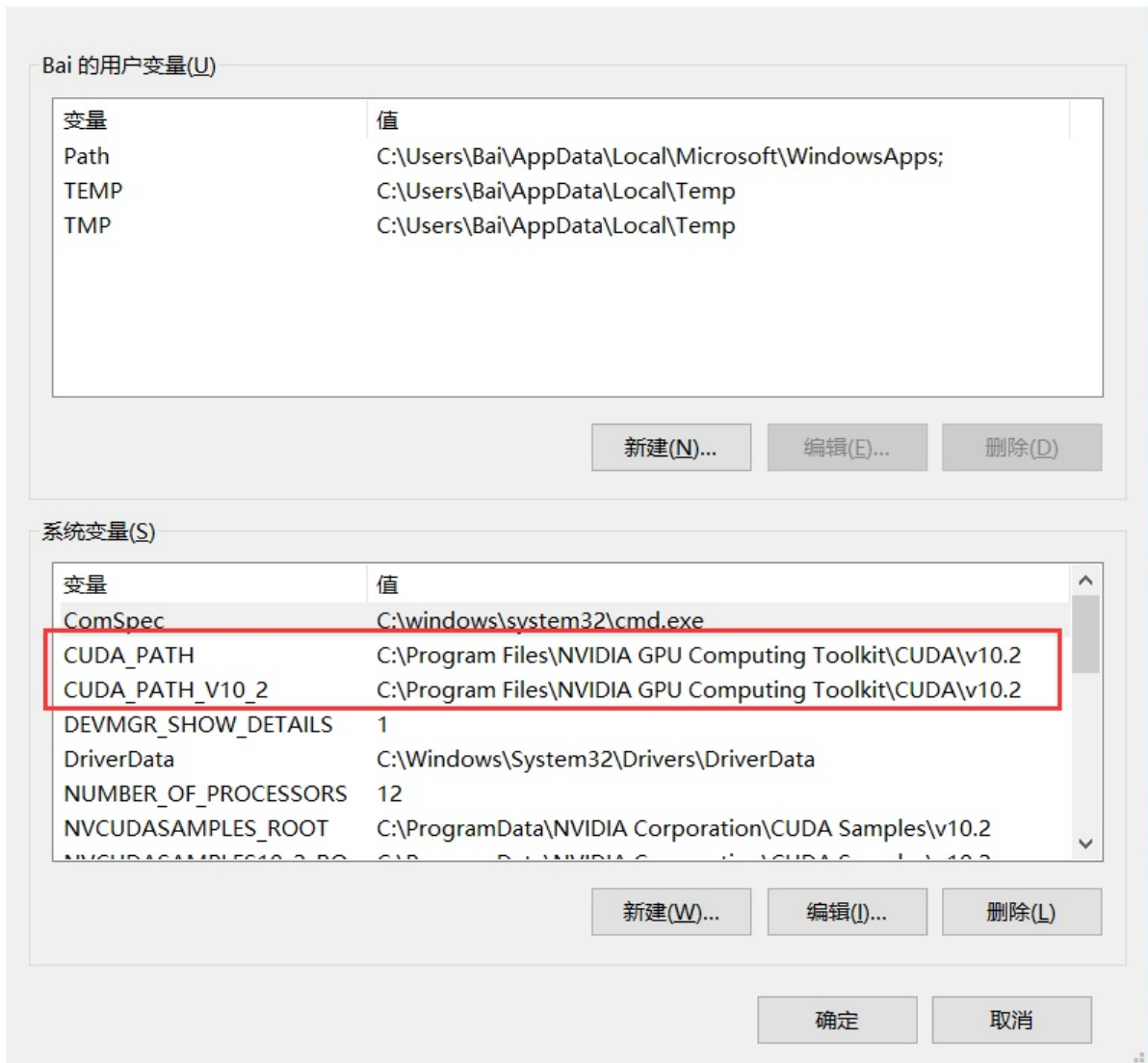
(1) 将cuda运行安装, 建议默认路径





安装时可以勾选Visual Studio Integration

(2) 安装完成后设置环境变量



计算机上点右键，打开属性->高级系统设置->环境变量，可以看到系统中多了CUDA_PATH和CUDA_PATH_V10_2两个环境变量。

接下来，还要在系统中添加以下几个环境变量：

这是默认安装位置的路径：C:\ProgramData\NVIDIA Corporation\CUDA Samples\v10.2

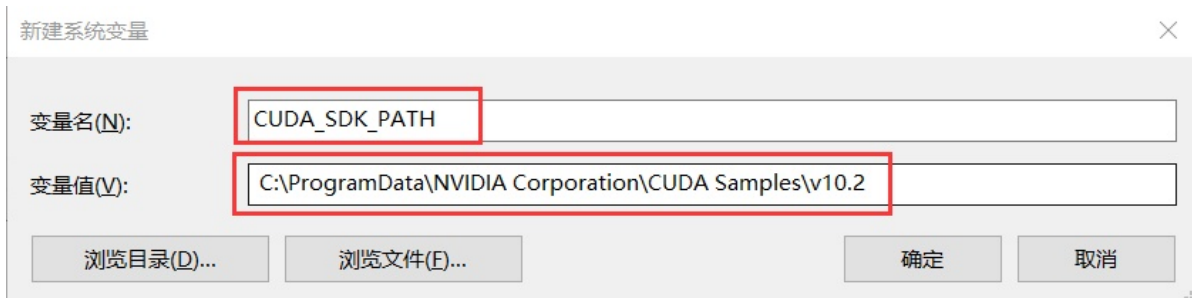
CUDA_SDK_PATH = C:\ProgramData\NVIDIA Corporation\CUDA Samples\v10.2

CUDA_LIB_PATH = %CUDA_PATH%\lib\x64

CUDA_BIN_PATH = %CUDA_PATH%\bin

CUDA_SDK_BIN_PATH = %CUDA_SDK_PATH%\bin\win64

CUDA_SDK_LIB_PATH = %CUDA_SDK_PATH%\common\lib\x64



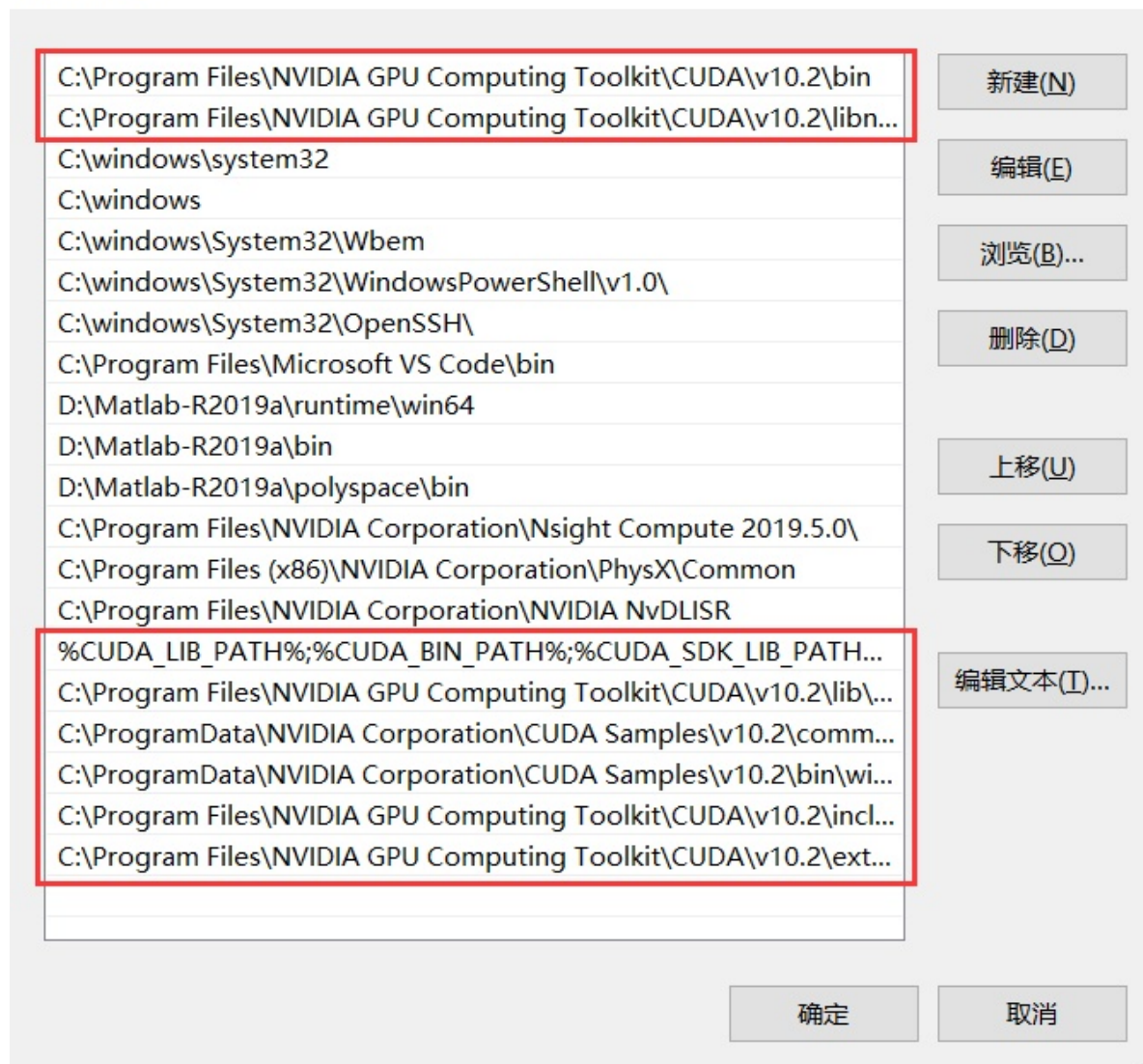
在系统变量 Path 的末尾添加：

%CUDA_LIB_PATH%;%CUDA_BIN_PATH%;%CUDA_SDK_LIB_PATH%;%CUDA_SDK_BIN_PATH%;

再添加如下5条（默认安装路径）：

C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v10.2\lib\x64
 C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v10.2\include
 C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v10.2\extras\CUPTI\lib64
 C:\ProgramData\NVIDIA Corporation\CUDA Samples\v10.2\bin\win64
 C:\ProgramData\NVIDIA Corporation\CUDA Samples\v10.2\common\lib\x64

编辑环境变量



注：与CUDA Samples相关的几个路径也可以不设置

6) 安装cuDNN

复制cudnn文件

对于cudnn直接将其解压缩包，然后需要将bin,include,lib中的文件复制粘贴到cuda的文件夹下

C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v10.2

注意：对整个文件夹bin,include,lib复制粘贴

7) CUDA安装测试

最后测试cuda是否配置成功:

打开CMD执行:

```
nvcc -V
```

即可看到cuda的信息

```
C:\Users\Bai>nvcc -V
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2019 NVIDIA Corporation
Built on Wed_Oct_23_19:32:27_Pacific_Daylight_Time_2019
Cuda compilation tools, release 10.2, V10.2.89
```

8) 安装Anaconda

Anaconda 是一个用于科学计算的 Python 发行版, 支持 Linux, Mac, Windows, 包含了众多流行的科学计算、数据分析的 Python 包。

1. 下载安装包

Anaconda下载Windows版: <https://www.anaconda.com/products/individual>

2. 然后安装anaconda

3. 添加Aanaconda国内镜像配置

清华TUNA提供了 Anaconda 仓库的镜像, 运行以下命令:

```
conda config --add channels
https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/free/
```

```
conda config --add channels
https://mirrors.tuna.tsinghua.edu.cn/anaconda/pkgs/main/
```

```
conda config --set show_channel_urls yes
```

9) 安装pytorch

创建虚拟环境, 环境名字可自己确定, 这里本人使用mypytorch作为环境名:

```
conda create -n mypytorch python=3.8
```

安装成功后激活mypytorch环境:

```
conda activate mypytorch
```

在所创建的mypytorch环境下安装pytorch, 执行命令:

```
conda install pytorch torchvision cudatoolkit=10.2 -c pytorch
```


注意：10.2处应为自己电脑上的cuda版本号

离线安装：

下载网址: <https://mirrors.tuna.tsinghua.edu.cn/anaconda/cloud/pytorch/win-64/>

安装pytorch1.8版本: pytorch-1.8.1-py3.8_cuda102_cudnn7_0.tar.bz2

```
conda install --offline pytorch-1.8.1-py3.8_cuda102_cudnn7_0.tar.bz2
```

2 Swin Transformer项目安装

1) 安装mmdcv

安装Git软件 (<https://git-scm.com/downloads>) , 克隆mmdcv项目到本地 (如d:)

```
git clone -b v1.3.1 https://github.com/open-mmlab/mmdcv.git
```

```
cd mmdcv  
pip install -r requirements.txt
```

然后执行以下步骤：

注意：以下步骤使用Power Shell命令窗口

(1) 找到cl.exe所在位置。设置环境变量，添加 C:\Program Files (x86)\Microsoft Visual Studio\2019\Community\VC\Tools\MSVC\14.28.29910\bin\Hostx64\x86 到 Path 使得cl.exe可以在命令窗口中执行

注意：cl.exe所在的具体路径和自己的电脑有关

具体做法：

- 把上述路径加入Path环境变量，并上移到最顶端
- 执行立即生效命令 set Path=c
- 执行 cl

(2) 查找GPU的计算能力from <https://developer.nvidia.cn/zh-cn/cuda-gpus>

例如: **Suppose you are using GTX 1080, which is of capability 6.1**

在Power Shell命令窗口中执行：

```
$env:Torch_CUDA_ARCH_LIST="6.1"  
$env:MMDCV_WITH_OPS = 1  
$env:MAX_JOBS = 4
```

最后一个命令中的数目based on available number of CPU cores and amount of memory

(3) 改变路径

```
cd mmdcv
```

(4) build

在mypytorch虚拟环境下执行:

```
python setup.py build_ext
```

(5) **install**

```
python setup.py develop
```

(6) **check**

```
pip list
```

2) 安装mmdetection

参考: https://mmdetection.readthedocs.io/zh_CN/v2.19.1/get_started.html

```
git clone -b v2.11.0 https://github.com/open-mmlab/mmdetection.git
cd mmdetection
pip install -r requirements/build.txt
pip install -v -e . # or "python setup.py develop"
```

3) 安装apex

```
git clone https://github.com/NVIDIA/apex
```

```
cd apex
```

```
python setup.py install
```

注意: cuda的版本应和cudatoolkit一致

If you would like to disable apex, modify the type of runner as `EpochBasedRunner` and comment out the following code block in the [configuration files](#)

4) 安装Swin-Transformer-Object-Detection

```
git clone https://github.com/SwinTransformer/Swin-Transformer-Object-Detection.git
```

执行

```
cd Swin-Transformer-Object-Detection
python setup.py develop
```

下载预训练权重文件

下载mask_rcnn_swin_tiny_patch4_window7_1x.pth权重文件，并放置在Swin-Transformer-Ot_Detection文件夹下

百度网盘下载链接:

链接: https://pan.baidu.com/s/14K_5UZLG9sPO_ZEe2-g1iQ
提取码: 2iuv

测试命令:

```
python demo/image_demo.py demo/demo.jpg
configs/swin/mask_rcnn_swin_tiny_patch4_window7_mstrain_480-800_adamw_1x_coco.py
mask_rcnn_swin_tiny_patch4_window7_1x.pth
```

如果出现信息**UserWarning: Matplotlib is currently using agg, which is a non-GUI backend**

可在demo/image_demo.py中加入

```
import matplotlib
matplotlib.use('TkAgg')
```

3 图像标注工具labelme的安装与使用

1) 安装图像标注工具labelme

```
pip install pyqt5
```

```
pip install labelme
```

或

```
pip install labelme -i https://pypi.tuna.tsinghua.edu.cn/simple
```

如果安装过程中提示缺少某个包，可再安装上，如：

```
pip install pyyaml
```

如上述安装方法不能成功，使用下面的命令安装：

```
pip install git+https://github.com/wkentaro/labelme.git
```

2) 使用labelme进行图像标注

执行：

```
labelme
```

标注后生成json文件

课程pothole项目案例的数据集为1280*720的图片，136张用于训练，16张用于测试。

课程中的数据集有5个类别："car", "dashedline", "midlane", "pothole", "rightlane"

数据集图像文件放置在mydataset目录下

4 数据格式转换与组织

1) 图像标注后的数据转换

从百度网盘下载

- mydataset.tar.gz (下载并解压)

把labelme标注的json数据格式转换成COCO数据格式。

```
cd mydataset
```

```
python labelme2cocoAll.py roadscene_train --output roadscene_train.json
```

```
python labelme2cocoAll.py roadscene_val --output roadscene_val.json
```

```
jupyter notebook &
```

在Jupyter Notebook中打开mydataset/COCO_Image_Viewer.ipynb

注意：由labelme标注的数据格式转成COCO数据格式后只包含3个字段信息：images, annotations, categories。而原始COCO数据集包含5个字段信息：info, licenses, images, annotations, categories。

2) 项目数据准备

把转成的COCO数据格式的数据的目录结构准备成COCO目录结构格式。

在Swin-Transformer-Object-Detection根目录下面，创建目录结构如下：

```
├── data
│   ├── coco
│       ├── annotations
│           ├── instances_train2017.json
│           └── instances_val2017.json
│       ├── train2017
│       └── val2017
```

其中：

train2017放置roadscene_train中的图片

val2017放置roadscene_val中的图片

roadscene_train.json改名为instances_train2017.json

roadscene_val.json改名为instances_val2017.json

5 修改配置文件

1) 修改changemaskrcnn.py中num_class并执行

产生新的权重文件

2) 修改 configs_base_models\mask_rcnn_swin_fpn.py 中num_classes, 共两处

3) 修改 configs_base_default_runtime.py 中interval, load_from

4) 修改 configs\swin\mask_rcnn_swin_tiny_patch4_window7_mstrain_480-800_adamw_1x_coco.py 中的 max_epochs, lr_config

5) 修改 configs_base_datasets\coco_instance.py 中的

```
samples_per_gpu=2,
```

```
workers_per_gpu=0,
```

6) 修改mmdet\datasets\coco.py中的CLASSES

6 训练自己的数据集

从百度网盘下载文件

testfiles.zip (下载到Swin-Transformer-Object-Detection目录下并解压)

链接: https://pan.baidu.com/s/14K_5UZLG9sPO_ZEe2-g1iQ

提取码: 2iuv

1) 训练命令

在Swin-Transformer-Object-Detection路径下执行:

```
python tools/train.py
configs/swin/mask_rcnn_swin_tiny_patch4_window7_mstrain_480-800_adamw_1x_coco.py
```

如出现以下错误

AssertionError: Incompatible version of pycocotools is installed. Run pip uninstall pycocotools first. Then run pip install mmpycocotools to install open-mmlab forked pycocotools.

执行

```
pip install --ignore-installed mmpycocotools
```

并且找到C:\Users\Bai Yong\AppData\Roaming\Python\Python38\site-packages中的pycocotools目录并改名

(具体路径和自己的电脑安装有关)

2) 训练结果的查看

查看Swin-Transformer-Object-Detection/work_dirs目录下的文件

7 测试训练出的网络模型

1) 测试图片

```
python demo/image_demo.py testfiles/img_val001.jpg
configs/swin/mask_rcnn_swin_tiny_patch4_window7_mstrain_480-800_adamw_1x_coco.py
work_dirs/mask_rcnn_swin_tiny_patch4_window7_mstrain_480-800_adamw_1x_coco/latest.pth
```

2) 测试视频

```
python demo/video_demo.py testfiles/drive.mp4
configs/swin/mask_rcnn_swin_tiny_patch4_window7_mstrain_480-800_adamw_1x_coco.py
work_dirs/mask_rcnn_swin_tiny_patch4_window7_mstrain_480-800_adamw_1x_coco/latest.pth --show
```

3) 性能统计

```
python tools/test.py configs/swin/mask_rcnn_swin_tiny_patch4_window7_mstrain_480-800_adamw_1x_coco.py work_dirs/mask_rcnn_swin_tiny_patch4_window7_mstrain_480-800_adamw_1x_coco/latest.pth --eval bbox segm
```

4) 日志分析

```
python tools/analysis_tools/analyze_logs.py plot_curve
work_dirs/mask_rcnn_swin_tiny_patch4_window7_mstrain_480-800_adamw_1x_coco/20211226_093405.log.json
```

(注意要指定自己的log.json文件)