

WSL下OpenCV和CUDA的环境配置

安装wsl2

这步很关键！！因为wsl1不支持cuda

在微软商店搜索Windows Subsystem for Linux安装即可

安装ubuntu子系统

在微软商店搜索ubuntu20.04安装即可

安装完后用wsl -l -v来确定一下wsl的版本，不对的话转换一下

安装CUDA Toolkit

注意：1.一定要先安装cuda再安装opencv的cuda版本。不然会缺少cuda模块

2.安装前请在windows系统下使用nvidia-smi命令来确认支持的最高的cuda版本

3.在wsl上安装的是cuda toolkit，不需要也不能安装显卡驱动，只需要在windows本机上安装显卡驱动，如果两端均安装驱动则会报错

在下面的链接中找到对应的cuda toolkit版本

<https://developer.nvidia.com/cuda-toolkit-archive>

选择wsl，并根据下面提示的指令一步步安装即可

例如：11.3版本的安装步骤

```
wget https://developer.download.nvidia.com/compute/cuda/repos/wsl-ubuntu/x86_64/cuda-wsl-ubuntu.pin
sudo mv cuda-wsl-ubuntu.pin /etc/apt/preferences.d/cuda-repository-pin-600
wget https://developer.download.nvidia.com/compute/cuda/11.3.0/local_installers/cuda-repo-wsl-ubuntu-11-3-local_11.3.0-1_amd64.deb
sudo dpkg -i cuda-repo-wsl-ubuntu-11-3-local_11.3.0-1_amd64.deb
sudo apt-key add /var/cuda-repo-wsl-ubuntu-11-3-local/7fa2af80.pub
sudo apt-get update
sudo apt-get -y install cuda
```

在采用如下方式安装完后，还需要手动配置一下环境变量

找到需要设置环境变量用户的家目录

向其家目录下的隐藏文件.bashrc文件末尾添加如下内容

```
export PATH=/usr/local/cuda/bin:$PATH
export LD_LIBRARY_PATH=/usr/local/cuda/lib64:$LD_LIBRARY_PATH
```

保存并使用source命令使其生效

安装openCV（带有cuda的版本）

1.安装编译OpenCV所需的依赖项：

```
sudo apt-get install build-essential cmake git libgtk2.0-dev pkg-config  
libavcodec-dev libavformat-dev libswscale-dev
```

2.获取OpenCV的源代码：

```
git clone https://github.com/opencv/opencv.git
```

3.下载 opencv_contrib 仓库：

```
git clone https://github.com/opencv/opencv_contrib.git
```

3.进入OpenCV源代码目录，并创建一个新的文件夹用于构建：

```
cd opencv  
mkdir build  
cd build
```

4.然后，在运行CMake配置OpenCV构建时，添加 `-D OPENCV_EXTRA_MODULES_PATH=<path_to_opencv_contrib>/modules`。将 `<path_to_opencv_contrib>` 替换为你下载的 opencv_contrib 仓库的实际路径。例如，如果 opencv_contrib 位于你当前用户的家目录中，则路径为 `~/opencv_contrib`。

完整的CMake命令如下：

```
cmake -D CMAKE_BUILD_TYPE=RELEASE -D CMAKE_INSTALL_PREFIX=/usr/local -D  
OPENCV_EXTRA_MODULES_PATH=<path_to_opencv_contrib>/modules -D WITH_CUDA=ON -D  
ENABLE_FAST_MATH=1 -D CUDA_FAST_MATH=1 -D WITH_CUBLAS=1 -D WITH_CUDNN=1 ..
```

6. 编译OpenCV：

```
make -j$(nproc)
```

7. 安装编译完成的OpenCV库：

```
sudo make install
```

8. 配置动态链接库：

```
sudo ldconfig
```

安装eigen3矩阵库

```
sudo apt-get install libeigen3-dev
```

VSCODE配置

首先需要安装c++ 等插件。

Cuda还需要Nsight Visual Studio Code Edition

配置代码提示需要告知vscode cuda和opencv库头文件所在的位置

在.vscode文件夹的c_cpp_properties.json文件中（没有就新建一个），配置以下内容

```
{
  "configurations": [
    {
      "name": "WSL",
      "intellisenseMode": "gcc-x64",
      "compilerPath": "/usr/bin/gcc",
      "cStandard": "c11",
      "cppStandard": "c++11",
      "includePath": [
        "${workspaceFolder}/**",
        "/usr/include/c++/9",
        "/usr/include/x86_64-linux-gnu/c++/9",
        "/usr/include/c++/9/backward",
        "/usr/lib/gcc/x86_64-linux-gnu/9/include",
        "/usr/local/include",
        "/usr/include/x86_64-linux-gnu",
        "/usr/include",
        "/usr/local/include/opencv4",
        "/usr/include/eigen3",
        "/usr/local/cuda/include"
      ],
      "defines": [
        "_DEBUG",
        "UNICODE",
        "_UNICODE"
      ]
    }
  ],
  "version": 4
}
```

其中下面几行分别配置的是安装的位置

```
"/usr/local/include/opencv4",
"/usr/include/eigen3",
"/usr/local/cuda/include"
```

按照自己安装的目录来指定即可，如果是按照上面的过程，则使用这几个配置即可。