

因为我们的编译选项CUDA_ARCH_BIN的值为 8.6, 8.9, 这个选项支持显卡列表如下:

8.9	NVIDIA L4 NVIDIA L40 NVIDIA L40S	NVIDIA RTX 6000 Ada NVIDIA RTX 5000 Ada NVIDIA RTX 4500 Ada NVIDIA RTX 4000 Ada NVIDIA RTX 4000 SFF Ada NVIDIA RTX 2000 Ada GeForce RTX 4090 GeForce RTX 4080 GeForce RTX 4070 Ti GeForce RTX 4070 GeForce RTX 4060 Ti GeForce RTX 4060 GeForce RTX 4050	
8.7			Jetson AGX Orin Jetson Orin NX Jetson Orin Nano
8.6	NVIDIA A40 NVIDIA A10 NVIDIA A16 NVIDIA A2	NVIDIA RTX A6000 NVIDIA RTX A5000 NVIDIA RTX A4000 NVIDIA RTX A3000 NVIDIA RTX A2000 GeForce RTX 3090 Ti GeForce RTX 3090 GeForce RTX 3080 Ti GeForce RTX 3080 GeForce RTX 3070 Ti GeForce RTX 3070 GeForce RTX 3060 Ti GeForce RTX 3060 GeForce RTX 3050 Ti GeForce RTX 3050	

此表更新于2025年11月10日。

1.2 文件列表说明

文件名/文件夹名	说明
opencv.zip	OpenCV 4.10源码, 此源码自定义了本地文件链接插件
opencv-contrib.zip	OpenCV 4.10 contrib 源码, 此源码自定义了本地文件链接插件
readmeMD	readme的markdown 文档
thirdparty.zip	下载第三方文件存到的压缩包
WinWSL.bat	进入WSL的bat脚本, 方便编译等操作
installOpencvGPU.sh	自动编译shell脚本

1.3 操作方法

1.3.1 自动化脚本编译

双击 [WinWSL.bat](#) 脚本, 打开WSL终端, 然后输入 `./installOpencvGPU.sh` 运行一键编译安装脚本

1.3.2 手动输入命令编译

```

1  ##### 解压三个压缩包 #####
2  #需要注意解压后, 对应压缩包解压后的不能包含同名的子文件夹, 比如:
3  opencv/opencv/ #解压后不能出现这样的目录结构
4
5  ##### 一般OpenCV 编译 #####
6  # 创建个build目录, yu opencv, opencv_crontib ,thirdparty 目同一级
7  mkdir build && cd build
8
9  #配置 cmake 参数
10 cmake -D CMAKE_BUILD_TYPE=RELEASE -D CMAKE_INSTALL_PREFIX=/usr/local -D OPENCV_GENERATE_PKGCONFIG=ON -D OPENCV_EXTRA
11
12 # 编译安装
13 make && make install

```

注意上面的命令执行完后, opencv 会安装到/usr/local里面去, 而对应编译的python 库安装在/usr/local/lib/python3.10/dist-packages

```

/usr/local/include/opencv4/opencv2/videostab/log.hpp
/usr/local/include/opencv4/opencv2/videostab/motion_core.hpp
/usr/local/include/opencv4/opencv2/videostab/motion_stabilizing.hpp
/usr/local/include/opencv4/opencv2/videostab/optical_flow.hpp
/usr/local/include/opencv4/opencv2/videostab/outlier_rejection.hpp
/usr/local/include/opencv4/opencv2/videostab/ring_buffer.hpp
/usr/local/include/opencv4/opencv2/videostab/stabilizer.hpp
/usr/local/include/opencv4/opencv2/videostab/wobble_suppression.hpp
/usr/local/share/java/opencv4/libopencv_java4100.so
/usr/local/share/java/opencv4/opencv-4100.jar
/usr/local/lib/python3.10/dist-packages/cv2/__init__.py
/usr/local/lib/python3.10/dist-packages/cv2/load_config_py2.py
/usr/local/lib/python3.10/dist-packages/cv2/load_config_py3.py
/usr/local/lib/python3.10/dist-packages/cv2/config.py
/usr/local/lib/python3.10/dist-packages/cv2/misc/__init__.py
/usr/local/lib/python3.10/dist-packages/cv2/misc/version.py
/usr/local/lib/python3.10/dist-packages/cv2/mat_wrapper/__init__.py
/usr/local/lib/python3.10/dist-packages/cv2/utils/__init__.py
/usr/local/lib/python3.10/dist-packages/cv2
/usr/local/lib/python3.10/dist-packages/cv2/typing
/usr/local/lib/python3.10/dist-packages/cv2/python-3.10/cv2.cpython-310-x86_64-linux-gnu.so
/usr/local/lib/python3.10/dist-packages/cv2/config-3.10.py
/usr/local/share/opencv4/haarcascades/haarcascade_eye.xml
/usr/local/share/opencv4/haarcascades/haarcascade_eye_tree_eyeglasses.xml
/usr/local/share/opencv4/haarcascades/haarcascade_frontalcatface.xml

```

卸载原来的opencv-python 库

```
1 | pip install opencv-python opencv-contrib-python
```

2. 源码编译后功能验证

2.1 验证C++版本cuda 版本功能

opencv的基本功能验证，这类就不提供了，主要是要验证OpenCV cuda功能是否可用，C++ cuda 验证需要编译一个程序，略过于麻烦，这里只是展示下opencv C++ 调用的库配置。

```

1 | # 确保 pkg-config --modversion opencv 能起作用
2 | cd /usr/local/lib/pkgconfig/
3 |
4 | cp opencv4.pc opencv.pc
5 |
6 | pkg-config --modversion opencv

```

python版本的OpenCV底层也是调用了C++的库，因此只要python版本opencv cuda可用，C++版本的就可用

2.2 验证Python版本cuda功能

源码编译的opencv python验证，需要重新运行WinWSL.bat脚本另开终端，或者执行命令"source ~/.bashrc"刷新下环境变量。

```

1 | # 运行WinWSL.bat脚本，打开终端，输入"python3" 打开python解释器，注意命令前面的符号">>", 不是命令的一部分。
2 | python3
3 | >>> import cv2
4 | >>> print(cv2.__version__)
5 | >>> import cv2.cuda
6 | >>> print(cv2.cuda.get_cuda_enabled_device_count())

```

```

root@King-Win:/mnt/e/work/RflySimGit/8.RflySimVision/0.ApiExps/0.Preparation/7.OpencvC++/opencv/Ubuntu22.04# python3
Python 3.10.12 (main, Aug 15 2025, 14:32:43) [GCC 11.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import cv2
>>> print(cv2.__version__)
4.10.0
>>> import cv2.cuda
>>> print(cv2.cuda.get_cuda_enabled_device_count())
1
>>>

```

2.2.1 查看python版的cuda支持哪些模块

cuda 不仅仅只是把先关数据拷贝的GPU里面去计算，还有一些原生态的模块接口也只是支持，我们可以通过打印当前opencv cuda 支持的所有模块和以及变量

```
1 python3
2 >>import cv2
3 >>print("cv2.cuda的子模块:", [x for x in dir(cv2.cuda) if not x.startswith('_')])
```

```
>>> print("cv2.cuda的子模块:", [x for x in dir(cv2.cuda) if not x.startswith('_')])
cv2.cuda的子模块: ['ALPHA_ATOP', 'ALPHA_ATOP_PREMUL', 'ALPHA_IN', 'ALPHA_IN_PREMUL', 'ALPHA_OUT', 'ALPHA_OUT_PREMUL', 'ALPHA_OVER', 'ALPHA_OVER_A_XOR', 'ALPHA_XOR_PREMUL', 'BackgroundSubtractorMOG', 'BackgroundSubtractorMOG2', 'BroxOpticalFlow', 'BroxOpticalFlow_create', 'BufferPool', 'Bayer_BG2GRAY_MHT', 'COLOR_BAYER_BG2RGB_MHT', 'COLOR_BAYER_GB2BGR_MHT', 'COLOR_BAYER_GB2GRAY_MHT', 'COLOR_BAYER_GB2RGB_MHT', 'COLOR_BAYER_GR2BayerBG2RGB_MHT', 'COLOR_BayerGR2BGR_MHT', 'COLOR_BayerGR2GRAY_MHT', 'COLOR_BayerGR2RGB_MHT', 'COLOR_BayerRG2BGR_MHT', 'COLOR_BayerRG2GRAY_MHT', 'COLOR_BayerRG2RGB_MHT', 'CascadeClassifier_create', 'Convolution', 'CornersCriteria', 'CornersDetector', 'DEVICE_INFO_COMPUTE_MODE_DEFAULT', 'DEVICE_INFO_COMPUTE_EVIC', 'GpuMatND', 'GpuMat_defaultAllocator', 'GpuMat_setDefaultAllocator', 'HOG', 'HOG_create', 'HOST_MEM_PAGE_LOCKED', 'HOST_MEM_SHARED', 'HOST_HARD', 'HostMem_WRITE_COMBINED', 'HoughCirclesDetector', 'HoughLinesDetector', 'HoughSegmentDetector', 'LookupTable', 'NATIVE_DOUBLE', 'NVIDIA_NV_OF_PERF_LEVEL_MAX', 'NVIDIA_OPTICAL_FLOW_1_0_NV_OF_PERF_LEVEL_MEDIUM', 'NVIDIA_OPTICAL_FLOW_1_0_NV_OF_PERF_LEVEL_SLOW', 'NVIDIA_OPTICAL_FLOW_HINT_VECTOR_GRID_SIZE_1', 'NVIDIA_OPTICAL_FLOW_2_0_NV_OF_HINT_VECTOR_GRID_SIZE_2', 'NVIDIA_OPTICAL_FLOW_2_0_NV_OF_HINT_VECTOR_GRID_SIZE_4', 'NVCAL_FLOW_2_0_NV_OF_HINT_VECTOR_GRID_SIZE_MAX', 'NVIDIA_OPTICAL_FLOW_2_0_NV_OF_HINT_VECTOR_GRID_SIZE_UNDEFINED', 'NVIDIA_OPTICAL_FLOW_2_0_NV_OF_VECTOR_GRID_SIZE_2', 'NVIDIA_OPTICAL_FLOW_2_0_NV_OF_OUTPUT_VECTOR_GRID_SIZE_4', 'NVIDIA_OPTICAL_FLOW_2_0_NV_OF_OUTPUT_VECTOR_GRID_SIZE_MAX', 'NVIDIA_OPTICAL_FLOW_2_0_NV_OF_PERF_LEVEL_FAST', 'NVIDIA_OPTICAL_FLOW_2_0_NV_OF_PERF_LEVEL_MAX', 'NVIDIA_OPTICAL_FLOW_2_0_NV_OF_PERF_LEVEL_MEDIUM', 'NvidiaHWOpticalFlow', 'NvidiaOpticalFlow_1_0', 'NvidiaOpticalFlow_1_0_NV_OF_PERF_LEVEL_FAST', 'NvidiaOpticalFlow_1_0_NV_OF_PERF_LEVEL_MEDIUM', 'NvidiaOpticalFlow_1_0_NV_OF_PERF_LEVEL_SLOW', 'NvidiaOpticalFlow_1_0_NV_OF_PERF_LEVEL_UNDEFINED', 'NvidiaOpticalFlow_1_0_NV_OF_HINT_VECTOR_GRID_SIZE_1', 'NvidiaOpticalFlow_2_0_NV_OF_HINT_VECTOR_GRID_SIZE_2', 'NvidiaOpticalFlow_2_0_NV_OF_HINT_VECTOR_GRID_SIZE_4', 'NvidiaOpticalFlow_2_0_NV_OF_HINT_VECTOR_GRID_SIZE_MAX', 'NvidiaOpticalFlow_2_0_NV_OF_OUTPUT_VECTOR_GRID_SIZE_2', 'NvidiaOpticalFlow_2_0_NV_OF_OUTPUT_VECTOR_GRID_SIZE_4', 'NvidiaOpticalFlow_2_0_NV_OF_OUTPUT_VECTOR_GRID_SIZE_MAX', 'NvidiaOpticalFlow_2_0_NV_OF_PERF_LEVEL_FAST', 'NvidiaOpticalFlow_2_0_NV_OF_PERF_LEVEL_MAX', 'NvidiaOpticalFlow_2_0_NV_OF_PERF_LEVEL_MEDIUM', 'NvidiaOpticalFlow_2_0_NV_OF_PERF_LEVEL_UNDEFINED', 'NvidiaOpticalFlowDual_TV11', 'OpticalFlowDual_TV11_create', 'SECOND_ORDER_MOMENTS', 'SURIAN_ROW', 'SURF_CUDA_LAPLACIAN_ROW', 'SURF_CUDA_OCTAVE_ROW', 'SURF_CUDA_ROWS_COUNT', 'SURF_CUDA_SIZE_ROW', 'SURF_CUDA_X_ROW', 'SURF_CUDA_Y_ROW', 'SparsePyrLKOpticalFlow_create', 'StereoBM', 'StereoBeliefPropagation', 'StereoBeliefPropagation_estimateRecommendedParams', 'StereoConstantSGM', 'Stream', 'Stream_Null', 'THIRD_ORDER_MOMENTS', 'TargetArchs', 'TargetArchs_has', 'TargetArchs_hasBin', 'TargetArchs_hasEqualOrGreaterPtx', 'TargetArchs_hasEqualOrLessPtx', 'TargetArchs_hasPtx', 'TemplateMatching', 'WARP_SHUFFLE_FUNCTIONS', 'abs', 'absSum', 'absdiff', 'add', 'itwise_not', 'bitwise_or', 'bitwise_xor', 'blendLinear', 'buildWarpAffineMaps', 'buildWarpPerspectiveMaps', 'calcAbsSum', 'calcHist', 'calcNorme', 'connectedComponents', 'connectedComponentsWithAlgorithm', 'convertSpatialMoments', 'copyMakeBorder', 'countNonZero', 'createBackgroundSubt', 'createBoxMaxFilter', 'createBoxMinFilter', 'createCLAHE', 'createCannyEdgeDetector', 'createColumnSumFilter', 'createContinuous', 'createConvolveFilter', 'createGaussianFilter', 'createGeneralizedHoughBallard', 'createGeneralizedHoughGuil', 'createGoodFeaturesToTrackDetector', 'createGpctor', 'createHoughLinesDetector', 'createHoughSegmentDetector', 'createLaplacianFilter', 'createLinearFilter', 'createLookupTable', 'createMed', 'createRowSumFilter', 'createScharrFilter', 'createSeparableLinearFilter', 'createSobelFilter', 'createStereoBM', 'createStereoBeliefPropagation', 'createTemplateMatching', 'cvtColor', 'demosaiing', 'dft', 'divide', 'drawColorDisp', 'ensureSizeIsEnough', 'equalizeHist', 'evenLevels', 'exp', 'fastNlMeanMaxLoc', 'flip', 'gammaCorrection', 'gemm', 'getCudaEnabledDeviceCount', 'getDevice', 'histEven', 'histRange', 'inRange', 'integral', 'log', 'meanShiftProc', 'meanShiftSegmentation', 'meanStdDev', 'merge', 'min', 'minMax', 'minMaxLoc', 'moments', 'mulAndScaleSpectrums', 'mulSpectrum', 'phase', 'polarToCart', 'pow', 'printCudaDeviceInfo', 'printShortCudaDeviceInfo', 'pyrDown', 'pyrUp', 'rectStdDev', 'reduce', 'registerPageLo', 'rshift', 'setBufferPoolConfig', 'setBufferPoolUsage', 'setDevice', 'spatialMoments', 'split', 'sqr', 'sqrIntegral', 'sqrSum', 'sqrt', 'warpAffine', 'warpPerspective', 'wrapStream']
>>>
```

3.其他可能出现的问题

3.1 如输入import cv2 报错

```
root@King-Win:/mnt/e/work/RflySimGit/8.RflySimVision/0.ApiExps/0.Preparation/7.OpencvC++/opencv/Ubuntu22.04/build# python3
Python 3.10.12 (main, Aug 15 2025, 14:32:43) [GCC 11.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import cv2
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File "/usr/local/lib/python3.10/dist-packages/cv2/__init__.py", line 181, in <module>
    bootstrap()
  File "/usr/local/lib/python3.10/dist-packages/cv2/__init__.py", line 175, in bootstrap
    if __load_extra_py_code_for_module("cv2", submodule, DEBUG):
  File "/usr/local/lib/python3.10/dist-packages/cv2/__init__.py", line 28, in __load_extra_py_code_for_module
    py_module = importlib.import_module(module_name)
  File "/usr/lib/python3.10/importlib/__init__.py", line 126, in import_module
    return _bootstrap.gcd_import(name[level:], package, level)
  File "/usr/local/lib/python3.10/dist-packages/cv2/gapi/__init__.py", line 290, in <module>
    cv.gapi.wip.GStreamerPipeline = cv.gapi_wip_gst_GStreamerPipeline
AttributeError: partially initialized module 'cv2' has no attribute 'gapi_wip_gst_GStreamerPipeline' (most likely due to a circular import)
>>>
KeyboardInterrupt
>>>
```

这里需要卸载 原来的opencv

```
1 | pip uninstall opencv-python opencv-contrib-python
```