**JesonTX2 + torch + opencv +gpu+ yolov5环境搭建**

一、Jetson刷机（参见https://blog.csdn.net/weixin\_72812064/article/details/130627061）

1. 刷jetson os包

先刷系统再刷组件，安装组件会尝试连接nvidian.com和apt仓库，所以需要换源再下载。

1. 刷jetson SDK components

Jetpack已预装CUDA和cudnn。

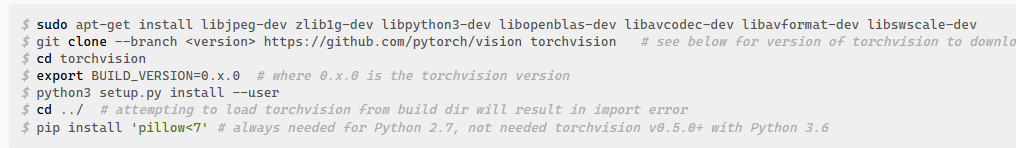
此时可能遇到的问题参见https://github.com/individualCodeSpace/Jetson-

1. 挂载固态硬盘

二、torch编译（参见https://i7y.org/en/pytorch-build-on-jetson-nano/）

X86架构和arm架构的Torch框架并不通用，在虚拟环境里安装的torch并不支持jetson的gpu模块，官网（https://forums.developer.nvidia.com/t/pytorch-for-jetson/72048）有一些jetson linux的预编译版本，但并不齐全，面对其他特定python版本，特定jetpack包版本都需要从源码编译。Jetson TX2 使用4.6.1,yolov5要求python 3.8, torch版本1.8.0。官网只支持到python3.6，需要自行编译。若使用python3.6版本支持的yolov5 5.0版参见<https://blog.csdn.net/m0_62013374/article/details/125736316?spm=1001.2014.3001.5502>。

三、安装torchvision



版本对应表：



测试：

>>>import torch

>>>print(torch.\_version\_)

>>>print(torchvision.\_version\_)

>>>print(torch.cuda.is\_available())

四、openCV编译

Jetpack预装的openCV不支持gpu加速，需要从源码重新编译构建。

1. 更新编译所需依赖

sudo apt-get install -y build-essential cmake git unzip pkg-config

sudo apt-get install -y libjpeg-dev libpng-dev libtiff-dev

sudo apt-get install -y libavcodec-dev libavformat-dev libswscale-dev

sudo apt-get install -y libgtk2.0-dev libcanberra-gtk\*

sudo apt-get install -y python3-dev python3-numpy python3-pip

sudo apt-get install -y libxvidcore-dev libx264-dev libgtk-3-dev

sudo apt-get install -y libtbb2 libtbb-dev libdc1394-22-dev

sudo apt-get install -y gstreamer1.0-tools libv4l-dev v4l-utils

sudo apt-get install -y libgstreamer1.0-dev libgstreamer-plugins-base1.0-dev

sudo apt-get install -y libavresample-dev libvorbis-dev libxine2-dev

sudo apt-get install -y libfaac-dev libmp3lame-dev libtheora-dev

sudo apt-get install -y libopencore-amrnb-dev libopencore-amrwb-dev

sudo apt-get install -y libopenblas-dev libatlas-base-dev libblas-dev

sudo apt-get install -y liblapack-dev libeigen3-dev gfortran

sudo apt-get install -y libhdf5-dev protobuf-compiler

sudo apt-get install -y libprotobuf-dev libgoogle-glog-dev libgflags-dev

1. 下载源码

wget -O opencv.zip https://github.com/opencv/opencv/archive/4.5.1.zip

wget -O opencv\_contrib.zip https://github.com/opencv/opencv\_contrib/archive/4.5.1.zip

unzip opencv.zip

unzip opencv\_contrib.zip

1. Build+Cmake

cd path/to/opencv

mkdir build

cd build

cmake -D CMAKE\_BUILD\_TYPE=RELEASE \

-D CMAKE\_INSTALL\_PREFIX=/usr \

-D OPENCV\_EXTRA\_MODULES\_PATH=~/opencv\_contrib/modules \

-D EIGEN\_INCLUDE\_PATH=/usr/include/eigen3 \

-D WITH\_OPENCL=OFF \

-D WITH\_CUDA=ON \

-D CUDA\_ARCH\_BIN=6.2 \#与自己的板子算力匹配

-D CUDA\_ARCH\_PTX="" \

-D WITH\_CUDNN=ON \

-D WITH\_CUBLAS=ON \

-D ENABLE\_FAST\_MATH=ON \

-D CUDA\_FAST\_MATH=ON \

-D OPENCV\_DNN\_CUDA=ON \

-D ENABLE\_NEON=ON \

-D WITH\_QT=OFF \

-D WITH\_OPENMP=ON \

-D WITH\_OPENGL=ON \

-D BUILD\_TIFF=ON \

-D WITH\_FFMPEG=ON \

-D WITH\_GSTREAMER=ON \

-D WITH\_TBB=ON \

-D BUILD\_TBB=ON \

-D BUILD\_TESTS=OFF \

-D WITH\_EIGEN=ON \

-D WITH\_V4L=ON \

-D WITH\_LIBV4L=ON \

-D OPENCV\_ENABLE\_NONFREE=ON \

-D INSTALL\_C\_EXAMPLES=OFF \

-D INSTALL\_PYTHON\_EXAMPLES=OFF \

-D BUILD\_NEW\_PYTHON\_SUPPORT=ON \

-D BUILD\_opencv\_python3=TRUE \

-D OPENCV\_GENERATE\_PKGCONFIG=ON \

-D BUILD\_EXAMPLES=OFF ..

报错情况处理

1. ippicv下载失败

解决方法：手动下载，修改cmake.txt文件

地址： https://github.com/opencv/opencv\_3rdparty/tree/ippicv/master\_20170822/ippicv

cd /home/nvidia/opencv\_contrib/modules/xfeatures2d/src #文件放这

sudo nano /path/to/opencv/3rdparty/ippicv/ippicv.cmake

"file://home/nvidia/opencv\_contrib/modules/xfeatures2d/src/"

#"https://raw.githubusercontent.com/opencv/opencv\_3rdparty/${IPPICV\_COMMIT}/ippicv/"

1. face\_landmark\_model.dat下载失败

解决方法：手动下载，修改cmake.txt文件

cd /home/nvidia/opencv\_contrib/modules/xfeatures2d/src

curl <https://raw.githubusercontent.com/opencv/opencv_3rdparty/>

8afa57abc8229d611c4937165d20e2a2d9fc5a12/face\_landmark\_model.dat >face\_landmark\_model.dat

Sudo nano /path/to/opencv\_contrib/modules/face/CMakeLists.txt

"file://home/nvidia/opencv\_contrib/modules/xfeatures2d/src/"

1. boostdesc\_bgm.i 下载失败

解决方法：手动下载，修改cmake.txt文件

cd /home/nvidia/opencv\_contrib/modules/xfeatures2d/src

curl <https://raw.githubusercontent.com/opencv/opencv_3rdparty/34e4206aef44d50e6bbcd0ab06354b52e7466d26/boostdesc_lbgm.i> > boostdesc\_lbgm.i

curl <https://raw.githubusercontent.com/opencv/opencv_3rdparty/34e4206aef44d50e6bbcd0ab06354b52e7466d26/boostdesc_binboost_256.i> > boostdesc\_binboost\_256.i  
curl <https://raw.githubusercontent.com/opencv/opencv_3rdparty/34e4206aef44d50e6bbcd0ab06354b52e7466d26/boostdesc_binboost_128.i> > boostdesc\_binboost\_128.i  
curl <https://raw.githubusercontent.com/opencv/opencv_3rdparty/34e4206aef44d50e6bbcd0ab06354b52e7466d26/boostdesc_binboost_064.i> > boostdesc\_binboost\_064.i  
curl <https://raw.githubusercontent.com/opencv/opencv_3rdparty/34e4206aef44d50e6bbcd0ab06354b52e7466d26/boostdesc_bgm_hd.i> > boostdesc\_bgm\_hd.i  
curl <https://raw.githubusercontent.com/opencv/opencv_3rdparty/34e4206aef44d50e6bbcd0ab06354b52e7466d26/boostdesc_bgm_bi.i> > boostdesc\_bgm\_bi.i  
curl <https://raw.githubusercontent.com/opencv/opencv_3rdparty/34e4206aef44d50e6bbcd0ab06354b52e7466d26/boostdesc_bgm.i> > boostdesc\_bgm.i  
curl <https://raw.githubusercontent.com/opencv/opencv_3rdparty/fccf7cd6a4b12079f73bbfb21745f9babcd4eb1d/vgg_generated_120.i> > vgg\_generated\_120.i  
curl <https://raw.githubusercontent.com/opencv/opencv_3rdparty/fccf7cd6a4b12079f73bbfb21745f9babcd4eb1d/vgg_generated_64.i> > vgg\_generated\_64.i  
curl <https://raw.githubusercontent.com/opencv/opencv_3rdparty/fccf7cd6a4b12079f73bbfb21745f9babcd4eb1d/vgg_generated_48.i> > vgg\_generated\_48.i  
curl <https://raw.githubusercontent.com/opencv/opencv_3rdparty/fccf7cd6a4b12079f73bbfb21745f9babcd4eb1d/vgg_generated_80.i> > vgg\_generated\_80.i

cd path/to/opencv\_contrib/modules/xfeatures2d/cmake/

sudo nano download\_boostdesc.cmake

将下载地址替换为本地：

"file://home/nvidia/opencv\_contrib/modules/xfeatures2d/src/"

1. 编译出错需要重新编译

删除原编译文件：

sudo rm -rf /usr/local/include/opencv4

sudo rm -rf /usr/local/lib/libopencv\*

sudo rm -rf /usr/local/bin/opencv\_\*

sudo rm -rf /usr/local/share/opencv4

sudo rm -rf /usr/local/lib/python3\*/site-packages/cv2\*

删除build文件夹，重新新建并Cmake

1. Make -j4
2. Sudo make install
3. 将编译好的cv2添加到虚拟环境包里、

Cd /miniconda/envs/<env name> /lib/python<version>/site-packages

Cp /usr/lib/python<version>/dist-packages/cv2/python-<version>/<cv2.so> .#复制到当前路径

sudo ldconfig #更新动态库连接

1. 测试

Python3

>>> import cv2

>>> print(cv2.getBuildInformation)

1. Make clean

五、关联TensorRT

1.从源码编译安装，源码地址<https://github.com/wang-xinyu/tensorrtx.git>

（C++版本，没有python绑定，所以待会测试在python虚拟环境里提示缺少该包）

2.查看所需模型对应文件夹下的README.txt照做就可以。

出错提醒：找不到pycuda 和 tensorrt  
pip install pycuda

# 从jetpack预装的tensorrt对应python是3.6版本的，设置软连接后，在3.8环境会显示不匹配，所以得用pybind11进行绑定，这样就可以在python.里调用C++编译好的tensorrt了。

六、配置yolov5

不建议直接pip install -r requirement.txt，文件里有检查版本命令，有可能会自动把装好的torch库等更新到最新版本，把依赖项一个个pip安装即可

七、调用板载摄像头

利用opencv的方法打开摄像头即可。

查看设备：ls /dev/video0