一.开机安装

cd NVIDIA-INSTALLER/

sudo ./installer.sh

reboot

密码：nvidia

右上角时间

Time&Date settings

设置为shanghai

环境安装包（百度网盘）：

链接：https://pan.baidu.com/s/1AmGQPZyGIk0cNsQhyU6\_Ng

提取码：m63q

二.pip安装

Ifconfig：查询地址（192.168.3.36）

拷贝文件

sudo python get-pip.py

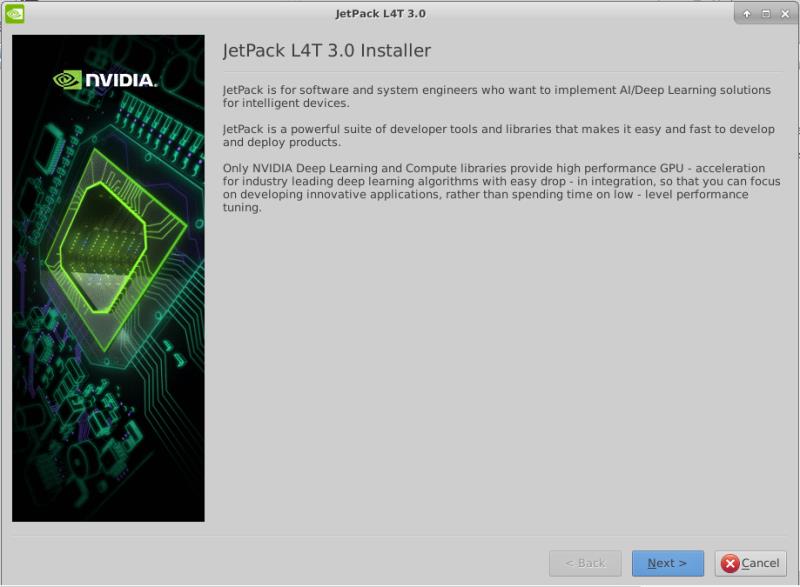
pip --version

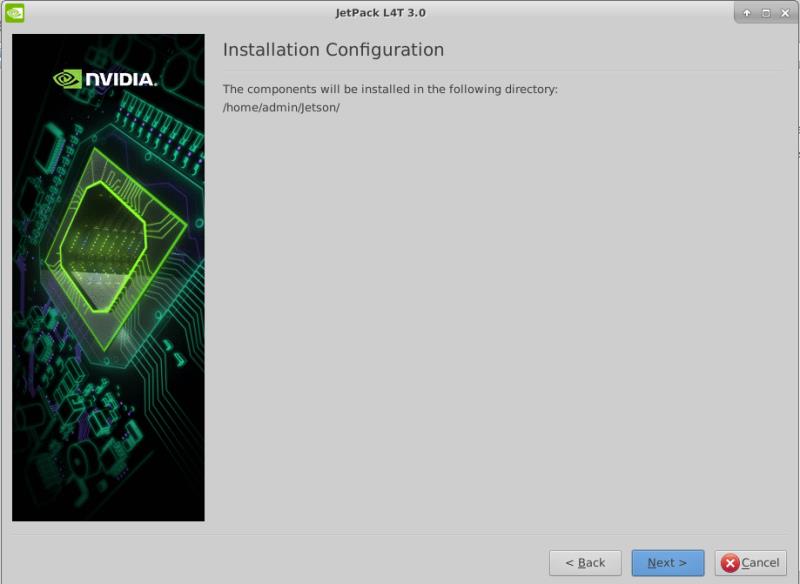
三.jetson tx2刷机(安装CUDA+opencv+cudnn)

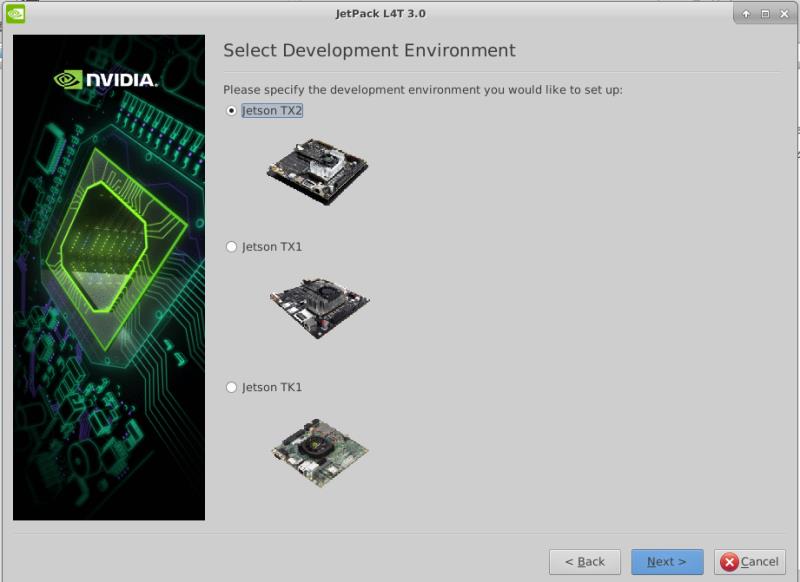
一台host

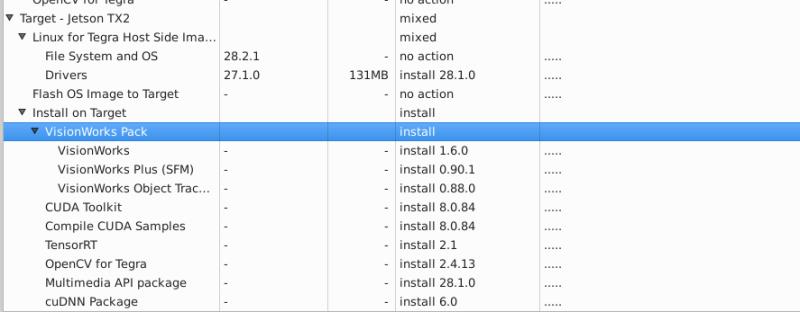
sudo chmod +x JetPack-L4T-3.0-linux-x64.run

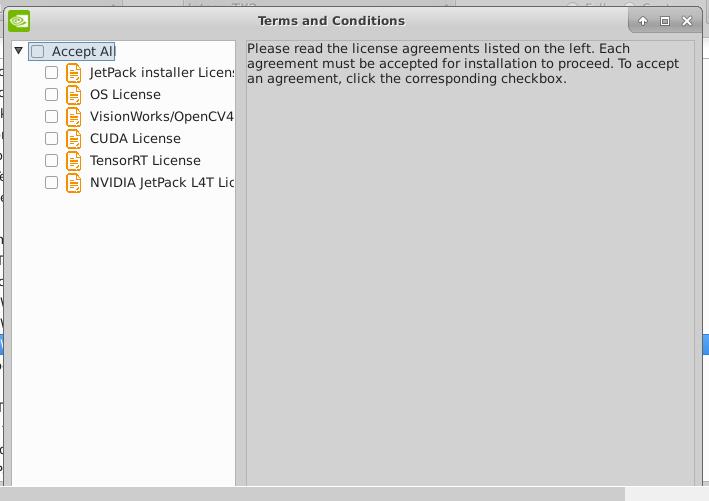
./ JetPack-L4T-3.0-linux-x64.run

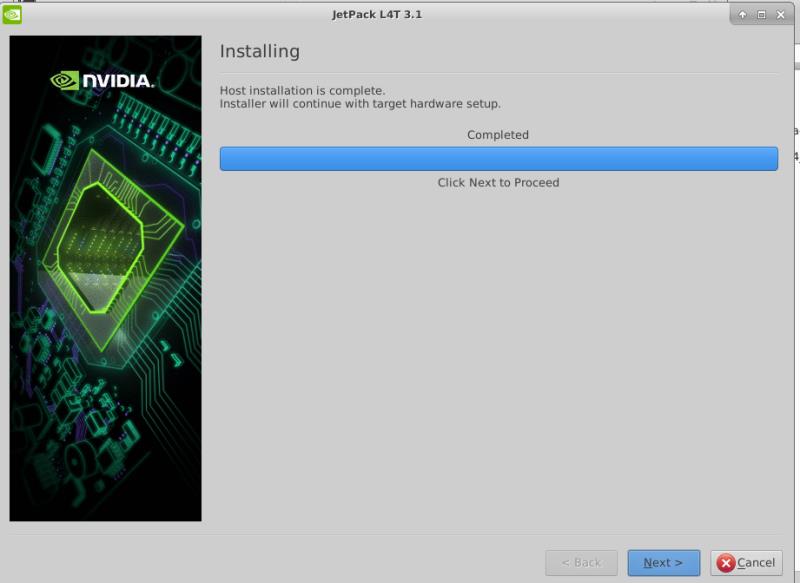




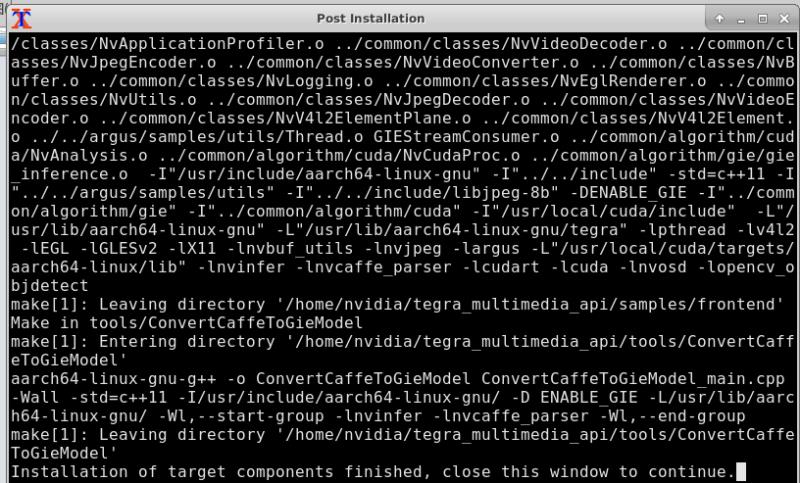












cd cuda-l4t

sudo dpkg –i \*.deb

报错：

error: dpkg status database is locked by another process

解决：sudo rm /var/lib/dpkg/lock

sudo dpkg --configure -a

cat /usr/include/cudnn.h | grep CUDNN\_MAJOR –A 2

sudo cp /usr/include/cudnn.h /usr/local/cuda/include/

sudo apt-get upgrade

sudo apt-get --fix-broken install

四.tensorflow安装

安装依赖：

sudo add-apt-repository ppa:webupd8team/java

sudo apt-get update

sudo apt-get install oracle-java8-installer -y

sudo apt-get install zip unzip autoconf automake libtool curl zlib1g-dev maven -y

sudo apt install python3-numpy python3-dev python3-pip python3-wheel

sudo pip install tensorflow-1.2.0-cp27-cp27m-linux\_aarch.whl

五.mxnet安装

1.安装依赖：

sudo apt-get update

sudo apt-get -y install git build-essential libatlas-base-dev libopencv-dev graphviz python-pip

sudo pip install pip --upgrade

sudo pip install setuptools numpy --upgrade

sudo pip install graphviz jupyter

2.下载mxnet

git clone https://github.com/dmlc/mxnet.git --recursive

cd mxnet

3.修改配置

* 修改配置增加CUDA支持，无CUDA可不改：
* 进入mxnet里面的make文件夹，更改config.mk

cp make/config.mk .

echo "USE\_CUDA=1" >> config.mk

echo "USE\_CUDA\_PATH=/usr/local/cuda" >> config.mk

echo "USE\_CUDNN=1" >> config.mk

* 修改mshadow/make/mshadow.mk

vim mshadow/make/mshadow.mk

* 修改内容：
* 先找到122行，增加如下

MSHADOW\_CFLAGS += -DMSHADOW\_USE\_PASCAL=1

* 回到mxnet文件夹里面：

make -j $(nproc)

**测试：**

* 执行命令：

python example/image-classification/train\_mnist.py

六.opencv其他版本安装

opencv3.4\_install.sh和opencv-3.4.0.zip

chmod +x opencv3.4\_instll.sh

两个半小时后

pkg-config --modversion opencv

七.caffe安装

安装依赖：

sudo apt-get install libprotobuf-dev libleveldb-dev libsnappy-dev libopencv-dev libhdf5-serial-dev protobuf-compiler

sudo apt-get install --no-install-recommends libboost-all-dev

sudo apt-get install libopenblas-dev liblapack-dev libatlas-base-dev

sudo apt-get install libgflags-dev libgoogle-glog-dev liblmdb-dev

sudo apt-get install python-numpy python-scipy python-matplotlib

sudo apt-get install python

sudo apt-get install python-dev

sudo apt-get install ipython

sudo apt-get install ipython-notebook

sudo apt-get install python-sklearn

sudo apt-get install python-skimage

sudo apt-get install python-protobuf

makefile.config配置：

A:cudnn

1. 将
2. #USE\_CUDNN := 1
3. 修改成：
4. USE\_CUDNN := 1

B:opencv

1. 将
2. #OPENCV\_VERSION := 3
3. 修改为：
4. OPENCV\_VERSION := 3

C:layer

1. 将
2. #WITH\_PYTHON\_LAYER := 1
3. 修改为
4. WITH\_PYTHON\_LAYER := 1

D:hdf5

INCLUDE\_DIRS := $(PYTHON\_INCLUDE) /usr/local/include

LIBRARY\_DIRS := $(PYTHON\_LIB) /usr/local/lib /usr/lib

修改为：

INCLUDE\_DIRS := $(PYTHON\_INCLUDE) /usr/local/include /usr/include/hdf5/serial

LIBRARY\_DIRS := $(PYTHON\_LIB) /usr/local/lib /usr/lib /usr/lib/x86\_64-linux-gnu /usr/lib/x86\_64-linux-gnu/hdf5/serial

Makefile配置：

将：

LIBRARIES += glog gflags protobuf boost\_system boost\_filesystem m hdf5\_hl hdf5

改为：

LIBRARIES += glog gflags protobuf boost\_system boost\_filesystem m hdf5\_serial\_hl hdf5\_serial

make all –j20

sudo make runtest –j20

sudo make pycaffe –j20

报错：

1. error==cudaSuccess(8 vs. 0) invalid device function

原因：显示此问题为显卡计算能力不匹配造成，在caffe目录下的Makefile.config文件当中做修改，原始的文件第35行CUDA\_ARCH项。

改正：删掉sm\_20,sm\_21两行（不删其实也没关系，会疯狂给你报warnning）

1. make完之后，make runtest时报错 Check failed: out\_bbox.xmax() == 50. (50 vs. 50)

原因：还是计算能力的问题，jetson TX2计算能力为6.2

改正：CUDA\_ARCH第一行变为：CUDA\_ARCH := -gencode arch=compute\_62,cde=sm\_62 \

1. make: protoc: Command not found

解决方案：

sudo apt-get install proto –c-compiler protobuf-compiler

1. fatal error:numpy/arrayobject.h

解决方案：

在Makefile.config找到PYTHON\_INCLUDE，发现有点不同：

PYTHON\_INCLUDE := /usr/include/python2.7 \  
        /usr/lib/python2.7/dist-packages/numpy/core/include

要加一个local，变成：

PYTHON\_INCLUDE := /usr/include/python2.7 \  
        /usr/local/lib/python2.7/dist-packages/numpy/core/include  
再make pycaffe就ok了

Depends: libopencv-ocl-dev (= 2.4.9.1+dfsg-1.5ubuntu1) but it is not going to be installed

Depends: libopencv2.4-java (= 2.4.9.1+dfsg-1.5ubuntu1) but it is not going to be installed

Depends: libopencv2.4-jni (= 2.4.9.1+dfsg-1.5ubuntu1) but it is not going to be installed

解决方案：

sudo apt install libopencv-ocl-dev libopencv2.4-java libopencv2.4-jni

八．附加

jetson tx2 程序自启动：

方法一（推荐）：

(1)进入HOME目录，HOME是当前用户目录的环境变量

(2)使用命令列出当前目录下的所有文件和目录，可以看到一个.config目录，这是一个隐藏目录

$ ls -al

(3)进入.config目录，列出.config下的目录和文件，找到autostart目录

$ ls -l

若没有，则创建

$ mkdir autostart

(5)进入该目录

$ cd autostart

(6)创建一个desktop文件，这里文件名可以自己定义，vim是一个编辑器，可以使用其他的编辑器，如nano，gedit等

$ vim \*.desktop

(7)在desktop文件中写入内容,例子如下：

[Desktop Entry]

Type=Application

Exec=/home/chaowei/test

其中Exec后面放的就是需要自启动的可执行程序的路径，这里最好选择绝对路径

方法二：

打开终端，在终端中执行gnome-session-properties，会弹出一个“Startup Applications Preferences”的菜单，点击添加Add

Name：输入命令名称

Command：输入启动指令

Comment：输入指令描述

在Command里可以输入运行的脚本，如

gnome-terminal -x /home/jobs/mecAGV/autoLoad.sh

如果想要后台运行则换成

bash /home/jobs/mecAGV/autoLoad.sh

也可以直接输入终端指令，如

ffmpeg -i "rtsp://192.168.1.50:554/user=admin&password=&channel=1&stream=1.sdp?" -f flv -r 25 -s 640x480 -an "rtmp://192.168.1.10:1935/mylive/node1"

完成将rtsp转rtmp推流到服务器功能