Jetson-inference environment construction

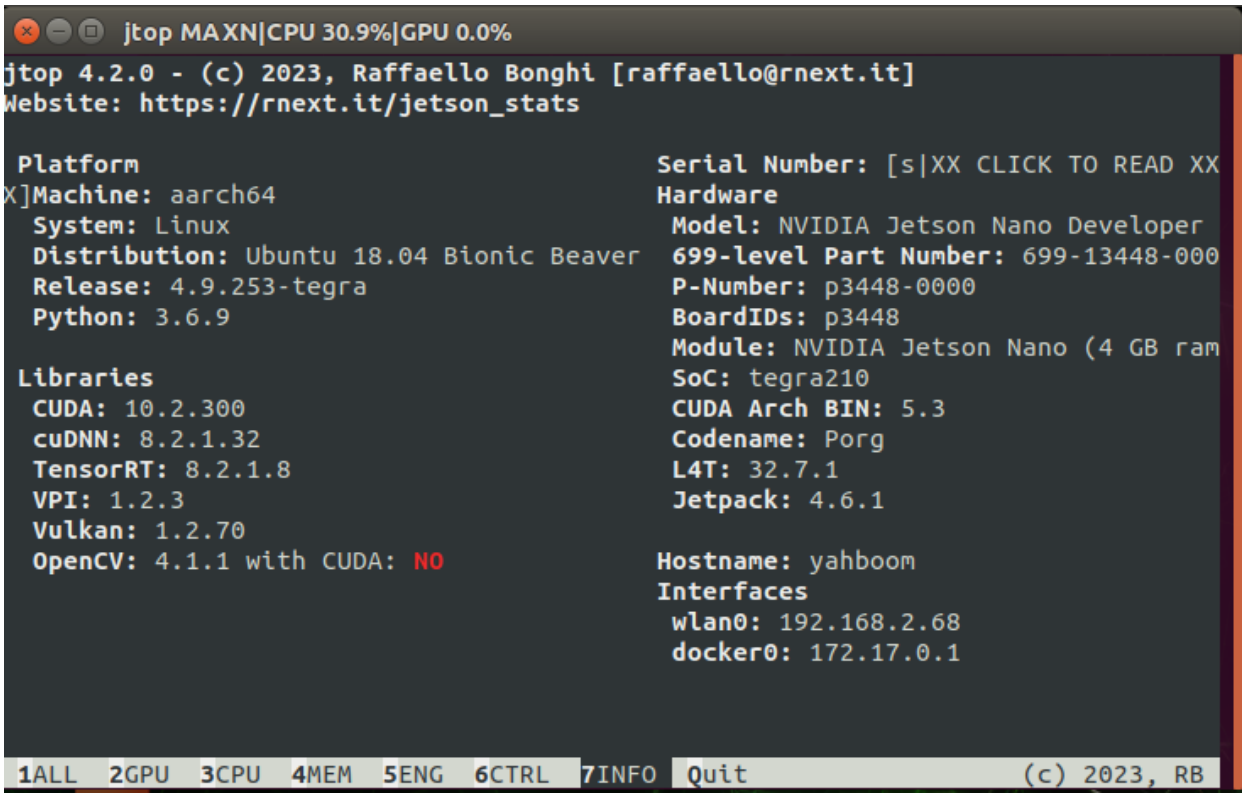
The Raspberry Pi motherboard series does not currently support this tutorial.

一、Install jetson-inference environment

1.Instructions before use

This tutorial is suitable for building a jetson nano image independently. If you use the YAHBOOM version of the image directly, you can ignore this tutorial.

2.The environment version configuration of this tutorial is as shown in the figure：



If you don't want to build it completely by yourself, you can use the jetson-inference compressed package we provide, pass the compressed package into jetson nano, unzip it, and start directly from "Installing Modules"

3. Start building

3.1 Download required dependencies

```
sudo apt-get update
sudo apt-get install git cmake
```

3.2 Download related source code

```
git clone https://github.com/dusty-nv/jetson-inference
cd jetson-inference
git submodule update --init
```

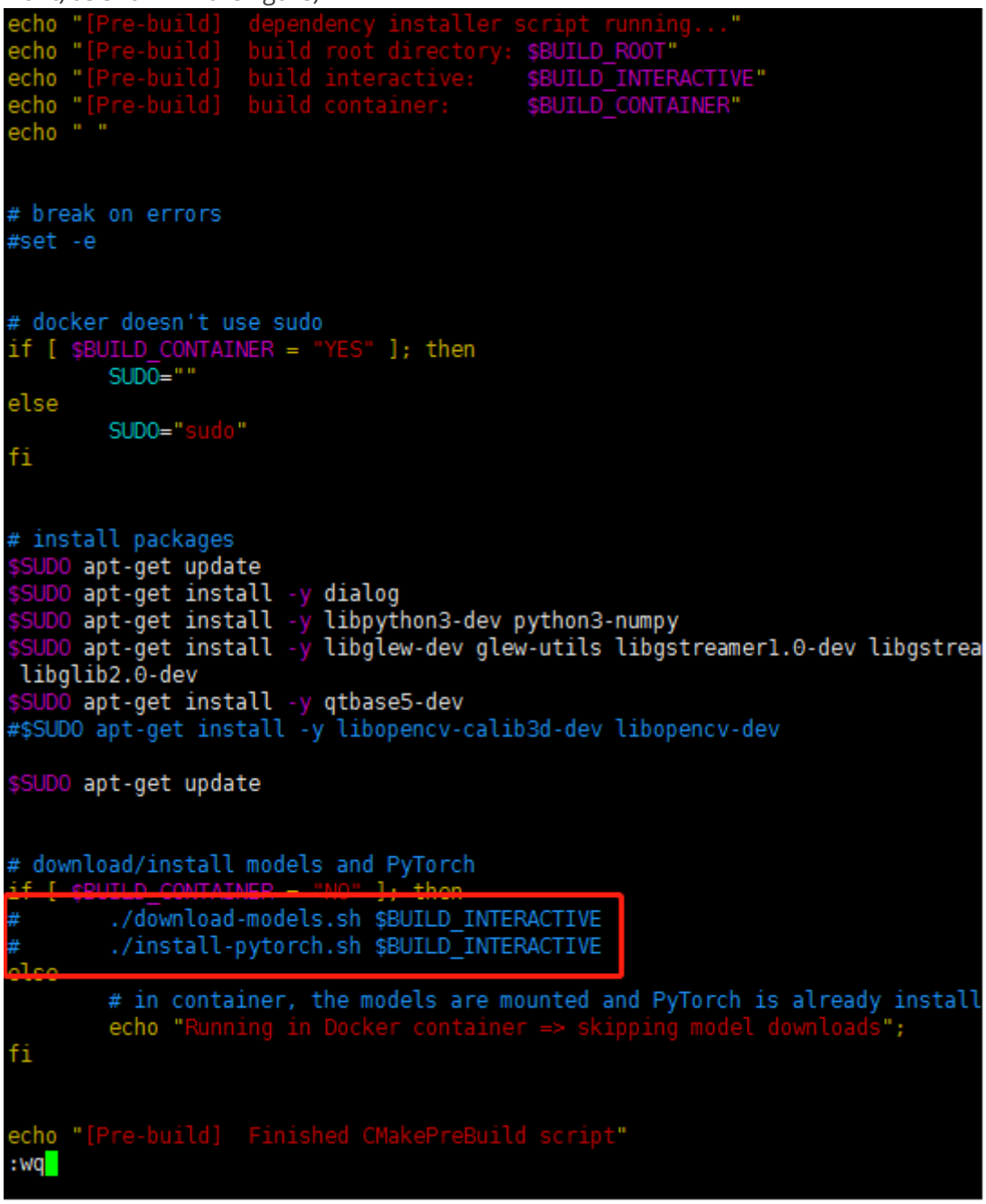
3.3 Download the relevant python module

Find the file torch-1.8.0-cp36-cp36m-linux_aarch64.whl from the attachment built by our environment and transfer it to jetson nano

```
sudo apt-get install libpython3-dev python3-numpy
sudo apt-get install python3-scipy
sudo apt-get install python3-pandas
sudo apt-get install python3-matplotlib
sudo apt-get install python3-sklearn
pip3 install torch-1.8.0-cp36-cp36m-linux_aarch64.whl
```

3.4 Make changes to files

Edit jetson-inference/CMakePrebuild.sh. Comment out ./download-models.sh (add a # comment in front) as shown in the figure)



```
echo "[Pre-build]  dependency installer script running..."
echo "[Pre-build]  build root directory: $BUILD_ROOT"
echo "[Pre-build]  build interactive:    $BUILD_INTERACTIVE"
echo "[Pre-build]  build container:      $BUILD_CONTAINER"
echo " "


# break on errors
#set -e


# docker doesn't use sudo
if [ $BUILD_CONTAINER = "YES" ]; then
        SUDO=""
else
        SUDO="sudo"
fi


# install packages
$SUDO apt-get update
$SUDO apt-get install -y dialog
$SUDO apt-get install -y libpython3-dev python3-numpy
$SUDO apt-get install -y libglew-dev glew-utils libgstreamer1.0-dev libgstrea
 libglib2.0-dev
$SUDO apt-get install -y qtbase5-dev
#$SUDO apt-get install -y libopencv-calib3d-dev libopencv-dev

$SUDO apt-get update


# download/install models and PyTorch
if [ $BUILD_CONTAINER = "NO" ]; then
#       ./download-models.sh $BUILD_INTERACTIVE
#       ./install-pytorch.sh $BUILD_INTERACTIVE
else
        # in container, the models are mounted and PyTorch is already install
        echo "Running in Docker container => skipping model downloads";
fi


echo "[Pre-build]  Finished CMakePreBuild script"
:wq
```

4.Install model

Method 1: You can perform the following steps

```
cd jetson-inference/tools
./download-models.sh
```

After making a selection, the model will be automatically downloaded to the file path of data/network.

Method 2: You can find the packages required by jetson-inference in the attachments we provide for environment construction, transfer the compressed packages inside to jetson-inference/data/network of jetso nano, and then decompress them.
Unzip command

```
for tar in *.tar.gz; do tar xvf $tar; done
```

Note:

1. For decompressing multiple .gz files, use this command:
   for gz in *.gz; do gunzip $gz; done
2. For decompressing multiple .tar.gz files, use the following command
   for tar in *.tar.gz; do tar xvf $tar; done

5. Start compiling

```
cd jetson-inference
mkdir build
cd build
cmake ../
make (or make -j4)    #（In the build directory）
sudo make install      #（In the build directory）
```

If an error is reported during the process, it means that the source code download is incomplete. please go back to step 3.2 and execute the command git submodule update --init.

6.Verify whether the installation is successful

```
cd jetson-inference/build/aarch64/bin

./imagenet-console ./images/bird_0.jpg output.jpg
# After the execution waits for a long time, the following appears (the first time
it takes a long time, the subsequent execution will be very fast)
```

```
yahboom@yahboom-desktop: ~/yahboom/jetson-inference/build/aarch64/bin
                -- dim #0  3 (CHANNEL)
                -- dim #1  224 (SPATIAL)
                -- dim #2  224 (SPATIAL)
[TRT]   binding -- index   1
                -- name    'prob'
                -- type    FP32
                -- in/out  OUTPUT
                -- # dims  3
                -- dim #0  1000 (CHANNEL)
                -- dim #1  1 (SPATIAL)
                -- dim #2  1 (SPATIAL)
[TRT]   binding to input 0 data  binding index:  0
[TRT]   binding to input 0 data  dims (b=1 c=3 h=224 w=224) size=602112
[TRT]   binding to output 0 prob  binding index:  1
[TRT]   binding to output 0 prob  dims (b=1 c=1000 h=1 w=1) size=4000
device GPU, networks/bvlc_googlenet.caffemodel initialized.
[TRT]   networks/bvlc_googlenet.caffemodel loaded
imageNet -- loaded 1000 class info entries
networks/bvlc_googlenet.caffemodel initialized.
[image] loaded './images/bird_0.jpg'  (368 x 500, 3 channels)
class 0015 - 0.998702  (robin, American robin, Turdus migratorius)
imagenet-console:  './images/bird_0.jpg' -> 99.87018% class #15 (robin, American robin, Turdus migrator
ius)

[TRT]   ------------------------------------------------
[TRT]   Timing Report networks/bvlc_googlenet.caffemodel
[TRT]   ------------------------------------------------
[TRT]   Pre-Process   CPU   0.08995ms  CUDA   0.64693ms
[TRT]   Network       CPU  72.14478ms  CUDA  71.47083ms
[TRT]   Post-Process  CPU   0.97890ms  CUDA   1.06088ms
[TRT]   Total         CPU  73.21364ms  CUDA  73.17864ms
[TRT]   ------------------------------------------------

[TRT]   note -- when processing a single image, run 'sudo jetson_clocks' before
                to disable DVFS for more accurate profiling/timing measurements

imagenet-console:  attempting to save output image to 'output.jpg'
imagenet-console:  completed saving 'output.jpg'
imagenet-console:  shutting down...
imagenet-console:  shutdown complete
```

Find the corresponding directory and view output.jpg as shown below. The recognition result will be displayed at the top of the picture.

二、Install Mediapipe environment

1.Preparing Files

Transfer the two files bazel and mediapipe-0.8-cp36-cp36m-linux_aarch64.whl in the attachment of the environment setup to the jetson nano

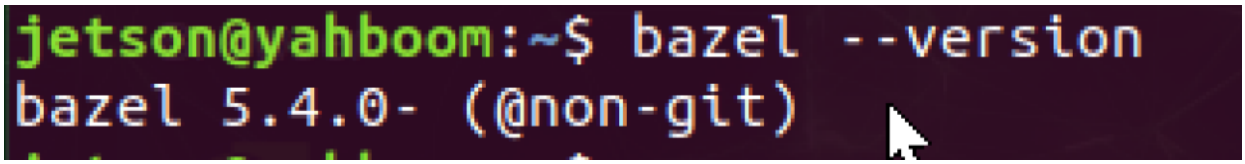
2.Install bazel

Open a terminal and run the following command

```
sudo chmod +x bazel
mv bazel /usr/local/bin
```

Check whether the installation of bazel is complete. If the version number can be printed, the installation is complete.

```
bazel --version
```



3.Install mediapipe

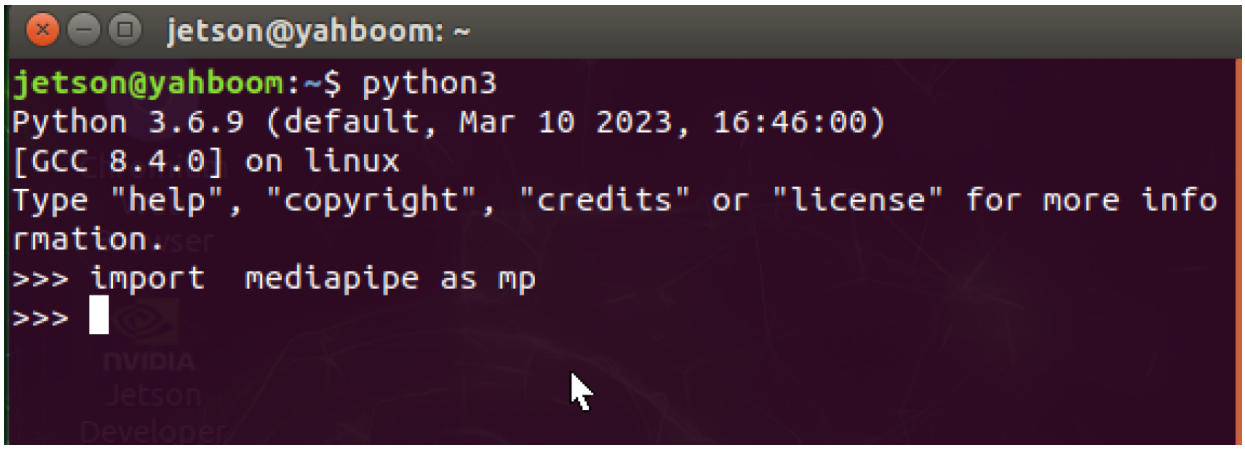
Open a terminal and run the following command

```
pip3 install opencv-contrib-python==3.4.18.65
pip3 install mediapipe-0.8-cp36-cp36m-linux_aarch64.whl
pip3 uninstall opencv-contrib-python
```

Verify successful installation

```
python3

import mediapipe as mp
```



appendix

Other reference tutorial URLs：

1.https://blog.csdn.net/aal779/article/details/122055432

2.https://github.com/dusty-nv/jetson-inference/blob/master/docs/building-repo-2.md

3.https://blog.csdn.net/weixin_43659725/article/details/120211312