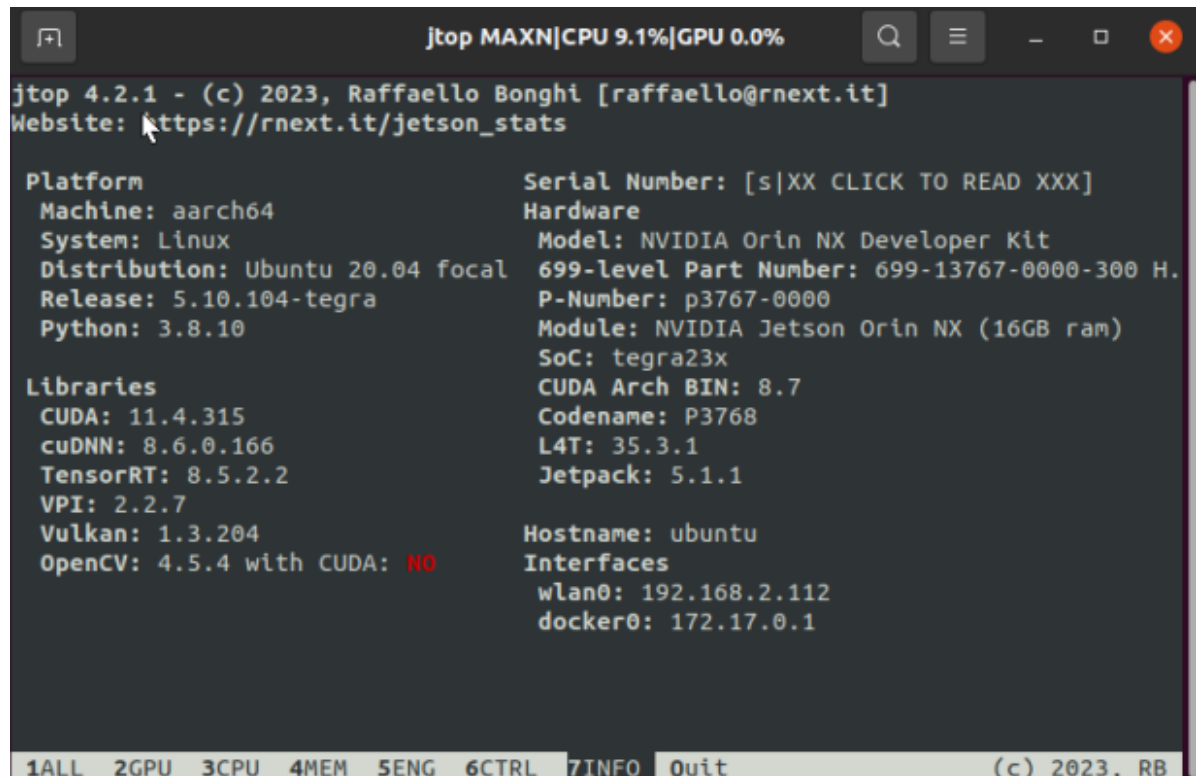


Jetson reference environment construction

1. Instructions before use

This tutorial is suitable for independently building images of Jetson orin nx. Directly using the YAHBOOM version of the image can be ignored for the tutorial.

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



If you don't want to build it completely on your own, you can use the Jetson reference compressed package we provided, pass the compressed package into Jetson orin nx, decompress it, and start looking at the "installation module" directly

3. Start building

3.1 Dependencies required for download

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

3.2 Download relevant source code

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

3.3 Download relevant Python modules

Find torch-1.8.0-cp36-cp36m-Linux from the attachment we built in our environment_
Aarch64.whl Transfer this file to Jetson orin nx

```
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 Modifying Files

Edit Jetson reference/CMakePrebuild.sh. Put/ Download models. sh comment out (with a # comment added 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
libgl2.0-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 the 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, and scientific internet access is required to download it normally

Method 2: You can find the package required for Jetson reference in the attachment we provided for environment setup, transfer the compressed package to Jetson orin nx's Jetson reference/data/network, and then decompress itDecompression command

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

notes:

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

5.Start Compilation

```
cd jetson-inference
mkdir build
cd build
cmake ../
make (或者make -j4)    # (build)
sudo make install      # (build)
```

If an error is reported midway, it indicates that the source code download is incomplete. Please go back to step 3.2 and execute the command git submodule update - init again, or download from a browser using Baidu

6.Verify if the installation was successful

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

./imagenet-console ./images/bird_0.jpg output.jpg
```

```
yahboom@yahboom-desktop: ~/yahboom/jetson-inference/build/aarch64/bin

-- dim #0 3 (CHANNEL)
-- dim #1 224 (SPATIAL)
-- dim #2 224 (SPATIAL)
[TRI] binding -- index 1
-- name 'prob'
-- type FP32
-- in/out OUTPUT
-- # dims 3
-- dim #0 1000 (CHANNEL)
-- dim #1 1 (SPATIAL)
-- dim #2 1 (SPATIAL)
[TRI] binding to input 0 data binding index: 0
[TRI] binding to input 0 data dims (b=1 c=3 h=224 w=224) size=602112
[TRI] binding to output 0 prob binding index: 1
[TRI] binding to output 0 prob dims (b=1 c=1000 h=1 w=1) size=4000
device GPU, networks/bvlc_googlenet.caffemodel initialized.
[TRI] 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 migratorius)

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

[TRI] 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 follows. The recognition results will be displayed at the top of the image.



Other reference tutorials:

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