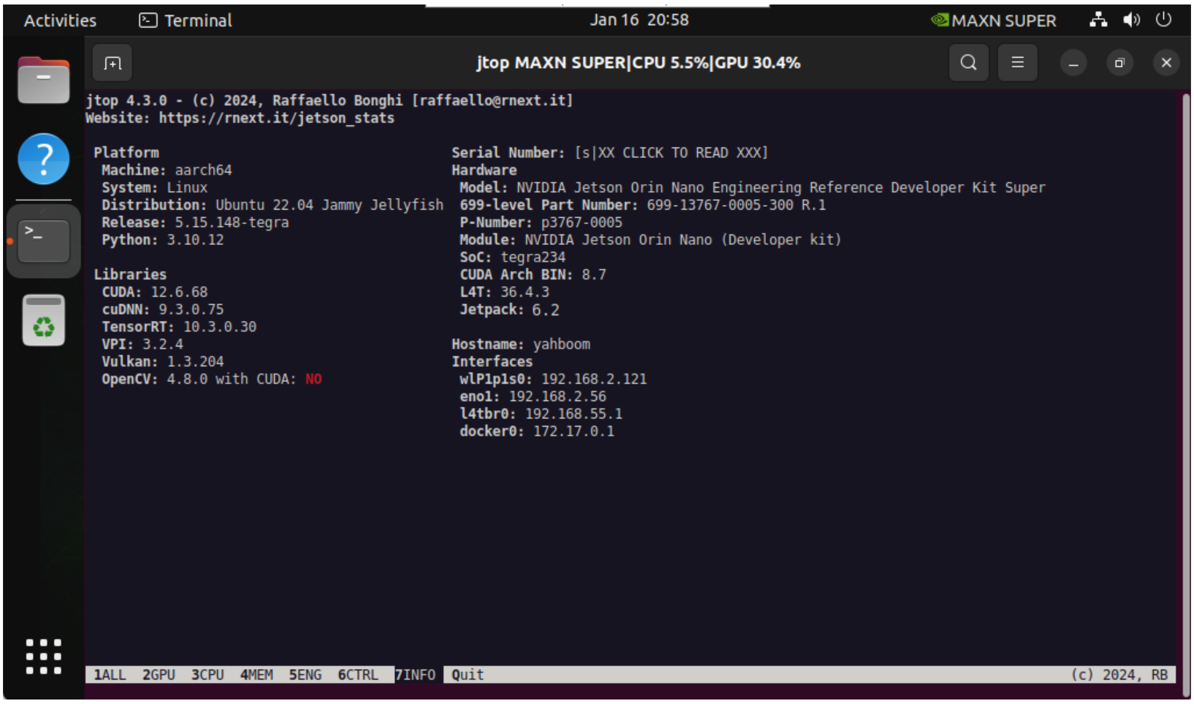
YOLO environment construction

The factory image comes with a pre-installed environment; this tutorial is for users who want to build their own environment!

1. System information



```
jtop 4.3.0 - (c) 2024, Raffaello Bonghi [raffaello@rnext.it]
Website: https://rnext.it/jetson_stats

 Platform                                     Serial Number: [s|XX CLICK TO READ XXX]
  Machine: aarch64                            Hardware
  System: Linux                                Model: NVIDIA Jetson Orin Nano Engineering Reference Developer Kit Super
  Distribution: Ubuntu 22.04 Jammy Jellyfish   699-level Part Number: 699-13767-0005-300 R.1
  Release: 5.15.148-tegra                      P-Number: p3767-0005
  Python: 3.10.12                              Module: NVIDIA Jetson Orin Nano (Developer kit)
                                               SoC: tegra234
 Libraries                                     CUDA Arch BIN: 8.7
  CUDA: 12.6.68                                L4T: 36.4.3
  cuDNN: 9.3.0.75                              Jetpack: 6.2
  TensorRT: 10.3.0.30
  VPI: 3.2.4                                  Hostname: yahboom
  Vulkan: 1.3.204                             Interfaces
  OpenCV: 4.8.0 with CUDA: NO                  wlP1p1s0: 192.168.2.121
                                               eno1: 192.168.2.56
                                               l4tbr0: 192.168.55.1
                                               docker0: 172.17.0.1

 1ALL  2GPU  3CPU  4MEM  5ENG  6CTRL 7INFO Quit                                                   (c) 2024, RB
```

2. Preliminary preparation

```
sudo apt update
sudo apt install python3-pip -y
sudo pip install -U pip
```

3. Install Ultralytics

```
sudo pip3 install ultralytics[export]
sudo reboot
```

4. Configure GPU acceleration

> **torch**

```
sudo pip3 install https://github.com/ultralytics/assets/releases/download/v0.0.0/torch-2.5.0a0+872d972e41.nv24.08-cp310-cp310-linux_aarch64.whl
```

> **torchvision**

```
sudo pip3 install https://github.com/ultralytics/assets/releases/download/v0.0.0/torchvision-0.20.0a0+afc54f7-cp310-cp310-linux_aarch64.whl
```

> **cuSPARSELt**

```
wget https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2204/arm64/cuda-keyring_1.1-1_all.deb
sudo dpkg -i cuda-keyring_1.1-1_all.deb
sudo apt-get update
sudo apt-get -y install libcusparselt0 libcusparselt-dev
```

> **onnxruntime-gpu**

```
sudo pip3 install https://github.com/ultralytics/assets/releases/download/v0.0.0/onnxruntime_gpu-1.20.0-cp310-cp310-linux_aarch64.whl
```

Note: Using onnxruntime-gpu requires installing a specific version of numpy. If it is not 1.23.5, you can run the following command to install the specified version

```
sudo pip3 install numpy==1.23.5
```

5. Verify the installation

> Verifying Ultralytics

```
python3 -c "import ultralytics; print(ultralytics.__version__)"
```

> Verifying Torch

```
python3 -c "import torch; print(torch.__version__); print(torch.cuda.is_available())"
```

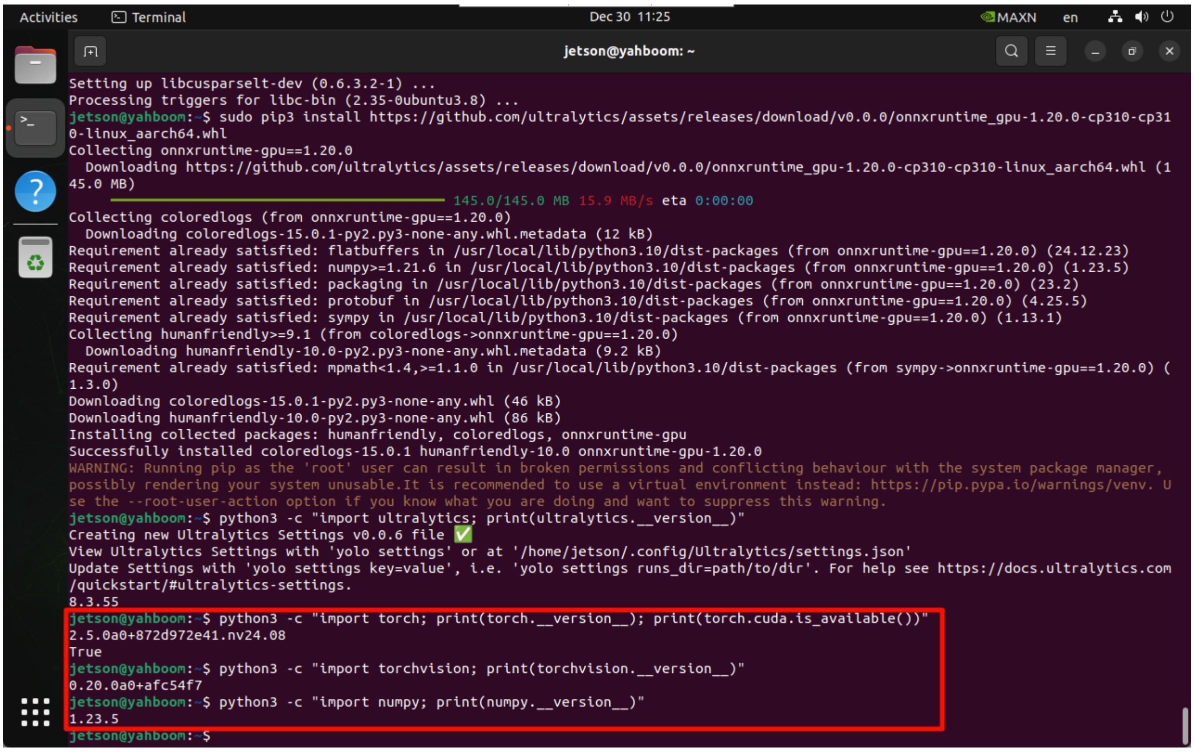
> Verifying Torchvision

```
python3 -c "import torchvision; print(torchvision.__version__)"
```

> Verify Numpy

```
python3 -c "import numpy; print(numpy.__version__)"
```

Note: The ultralytics version will be updated later, please refer to the version information queried in the system



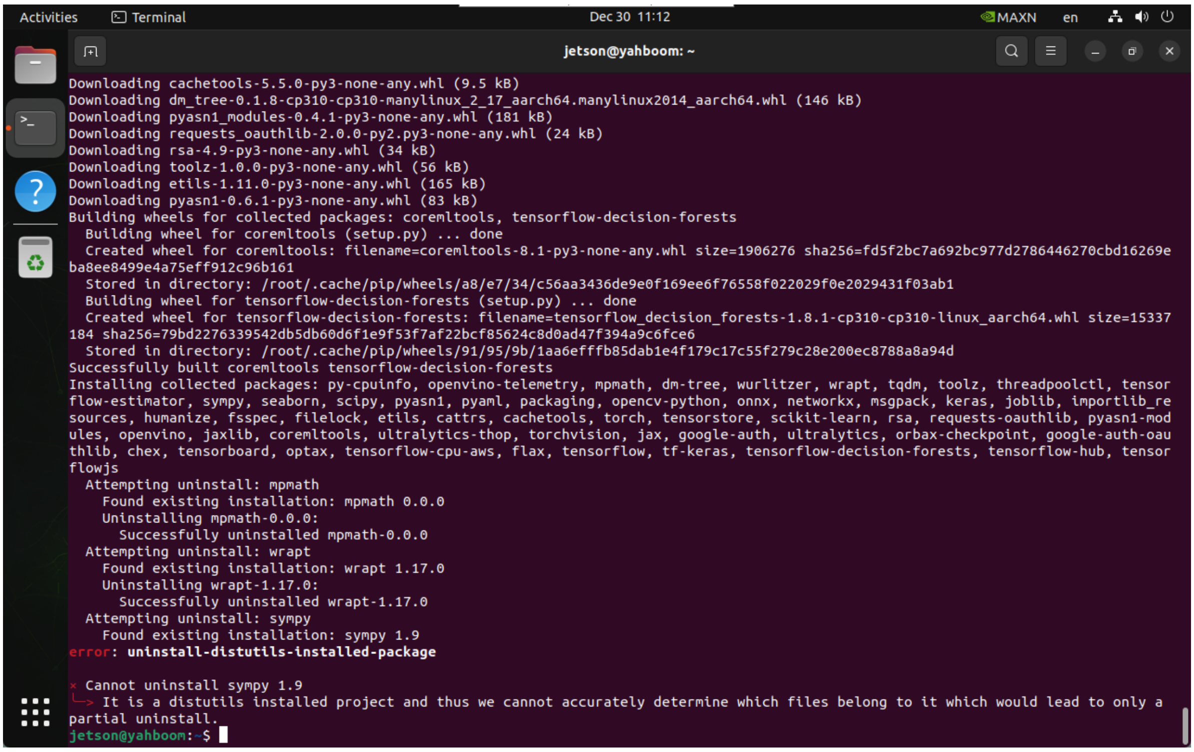
```
Setting up libcusparselt-dev (0.6.3.2-1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.8) ...
jetson@yahboom:~$ sudo pip3 install https://github.com/ultralytics/assets/releases/download/v0.0.0/onnxruntime_gpu-1.20.0-cp310-cp31
0-linux_aarch64.whl
Collecting onnxruntime-gpu==1.20.0
  Downloading https://github.com/ultralytics/assets/releases/download/v0.0.0/onnxruntime_gpu-1.20.0-cp310-cp310-linux_aarch64.whl (1
45.0 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 145.0/145.0 MB 15.9 MB/s eta 0:00:00
Collecting coloredlogs (from onnxruntime-gpu==1.20.0)
  Downloading coloredlogs-15.0.1-py2.py3-none-any.whl.metadata (12 kB)
Requirement already satisfied: flatbuffers in /usr/local/lib/python3.10/dist-packages (from onnxruntime-gpu==1.20.0) (24.12.23)
Requirement already satisfied: numpy>=1.21.6 in /usr/local/lib/python3.10/dist-packages (from onnxruntime-gpu==1.20.0) (1.23.5)
Requirement already satisfied: packaging in /usr/local/lib/python3.10/dist-packages (from onnxruntime-gpu==1.20.0) (23.2)
Requirement already satisfied: protobuf in /usr/local/lib/python3.10/dist-packages (from onnxruntime-gpu==1.20.0) (4.25.5)
Requirement already satisfied: sympy in /usr/local/lib/python3.10/dist-packages (from onnxruntime-gpu==1.20.0) (1.13.1)
Collecting humanfriendly>=9.1 (from coloredlogs->onnxruntime-gpu==1.20.0)
  Downloading humanfriendly-10.0-py2.py3-none-any.whl.metadata (9.2 kB)
Requirement already satisfied: mpmath<1.4,>=1.1.0 in /usr/local/lib/python3.10/dist-packages (from sympy->onnxruntime-gpu==1.20.0) (
1.3.0)
Downloading coloredlogs-15.0.1-py2.py3-none-any.whl (46 kB)
Downloading humanfriendly-10.0-py2.py3-none-any.whl (86 kB)
Installing collected packages: humanfriendly, coloredlogs, onnxruntime-gpu
Successfully installed coloredlogs-15.0.1 humanfriendly-10.0 onnxruntime-gpu-1.20.0
WARNING: Running pip as the 'root' user can result in broken permissions and conflicting behaviour with the system package manager,
possibly rendering your system unusable.It is recommended to use a virtual environment instead: https://pip.pypa.io/warnings/venv. U
se the --root-user-action option if you know what you are doing and want to suppress this warning.
jetson@yahboom:~$ python3 -c "import ultralytics; print(ultralytics.__version__)"
Creating new Ultralytics Settings v0.0.6 file ✅
View Ultralytics Settings with 'yolo settings' or at '/home/jetson/.config/Ultralytics/settings.json'
Update Settings with 'yolo settings key=value', i.e. 'yolo settings runs_dir=path/to/dir'. For help see https://docs.ultralytics.com
/quickstart/#ultralytics-settings.
8.3.55
jetson@yahboom:~$ python3 -c "import torch; print(torch.__version__); print(torch.cuda.is_available())"
2.5.0a0+872d972e41.nv24.08
True
jetson@yahboom:~$ python3 -c "import torchvision; print(torchvision.__version__)"
0.20.0a0+afc54f7
jetson@yahboom:~$ python3 -c "import numpy; print(numpy.__version__)"
1.23.5
jetson@yahboom:~$
```

Common Errors

Cannot uninstall sympy

Error phenomenon

Unable to uninstall sympy



```
Downloading cachetools-5.5.0-py3-none-any.whl (9.5 kB)
Downloading dm_tree-0.1.8-cp310-cp310-manylinux_2_17_aarch64.manylinux2014_aarch64.whl (146 kB)
Downloading pyasn1_modules-0.4.1-py3-none-any.whl (181 kB)
Downloading requests_oauthlib-2.0.0-py2.py3-none-any.whl (24 kB)
Downloading rsa-4.9-py3-none-any.whl (34 kB)
Downloading toolz-1.0.0-py3-none-any.whl (56 kB)
Downloading etils-1.11.0-py3-none-any.whl (165 kB)
Downloading pyasn1-0.6.1-py3-none-any.whl (83 kB)
Building wheels for collected packages: coremltools, tensorflow-decision-forests
  Building wheel for coremltools (setup.py) ... done
  Created wheel for coremltools: filename=coremltools-8.1-py3-none-any.whl size=1906276 sha256=fd5f2bc7a692bc977d2786446270cbd16269e
ba8ee8499e4a75eff912c96b161
  Stored in directory: /root/.cache/pip/wheels/a8/e7/34/c56aa3436de9e0f169ee6f76558f022029f0e2029431f03ab1
  Building wheel for tensorflow-decision-forests (setup.py) ... done
  Created wheel for tensorflow-decision-forests: filename=tensorflow_decision_forests-1.8.1-cp310-cp310-linux_aarch64.whl size=15337
184 sha256=79bd2276339542db5db60d6f1e9f53f7af22bcf85624c8d0ad47f394a9c6fce6
  Stored in directory: /root/.cache/pip/wheels/91/95/9b/1aa6efffb85dab1e4f179c17c55f279c28e200ec8788a8a94d
Successfully built coremltools tensorflow-decision-forests
Installing collected packages: py-cpuinfo, openvino-telemetry, mpmath, dm-tree, wurlitzer, wrapt, tqdm, toolz, threadpoolctl, tensor
flow-estimator, sympy, seaborn, scipy, pyasn1, pyaml, packaging, opencv-python, onnx, networkx, msgpack, keras, joblib, importlib_re
sources, humanize, fsspec, filelock, etils, cattrs, cachetools, torch, tensorstore, scikit-learn, rsa, requests-oauthlib, pyasn1-mod
ules, openvino, jaxlib, coremltools, ultralytics-thop, torchvision, jax, google-auth, ultralytics, orbax-checkpoint, google-auth-oau
thlib, chex, tensorboard, optax, tensorflow-cpu-aws, flax, tensorflow, tf-keras, tensorflow-decision-forests, tensorflow-hub, tensor
flowjs
  Attempting uninstall: mpmath
    Found existing installation: mpmath 0.0.0
    Uninstalling mpmath-0.0.0:
      Successfully uninstalled mpmath-0.0.0
  Attempting uninstall: wrapt
    Found existing installation: wrapt 1.17.0
    Uninstalling wrapt-1.17.0:
      Successfully uninstalled wrapt-1.17.0
  Attempting uninstall: sympy
    Found existing installation: sympy 1.9
error: uninstall-distutils-installed-package

× Cannot uninstall sympy 1.9
╰─> It is a distutils installed project and thus we cannot accurately determine which files belong to it which would lead to only a
partial uninstall.
jetson@yahboom:~$
```

Solution

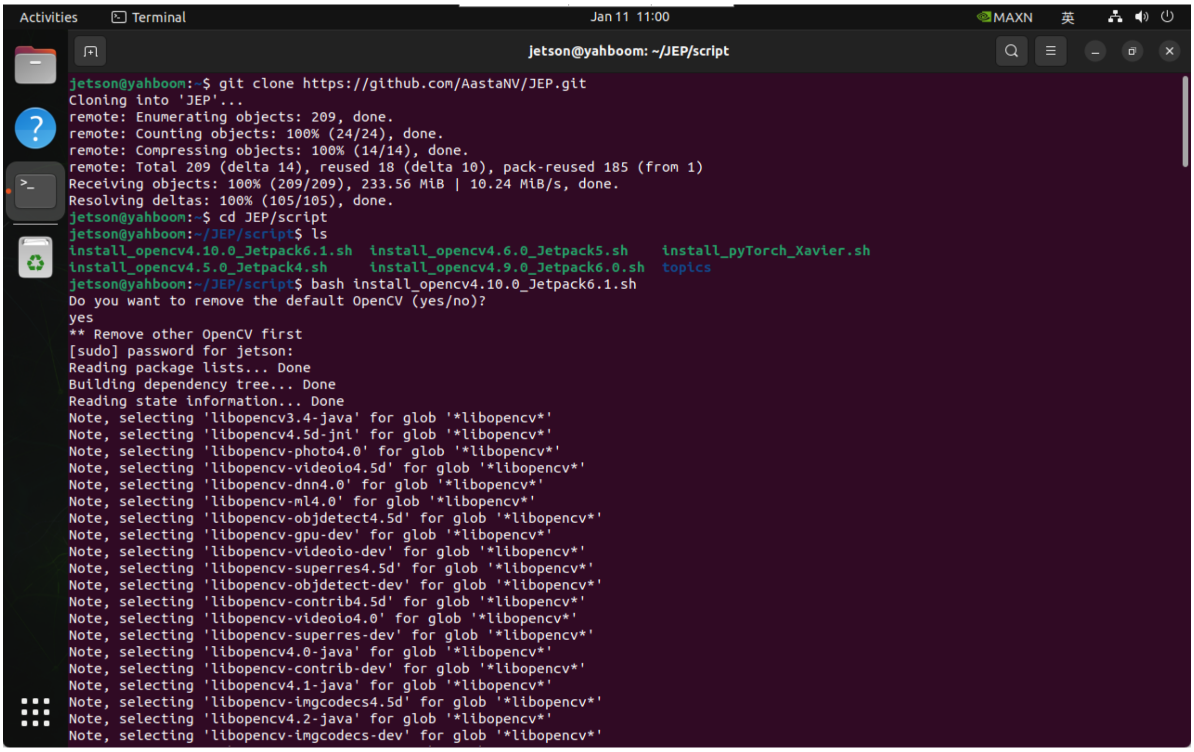
Uninstall python3-sympy: Reinstall PyTorch after uninstallation

```
sudo apt remove python3-sympy -y
```

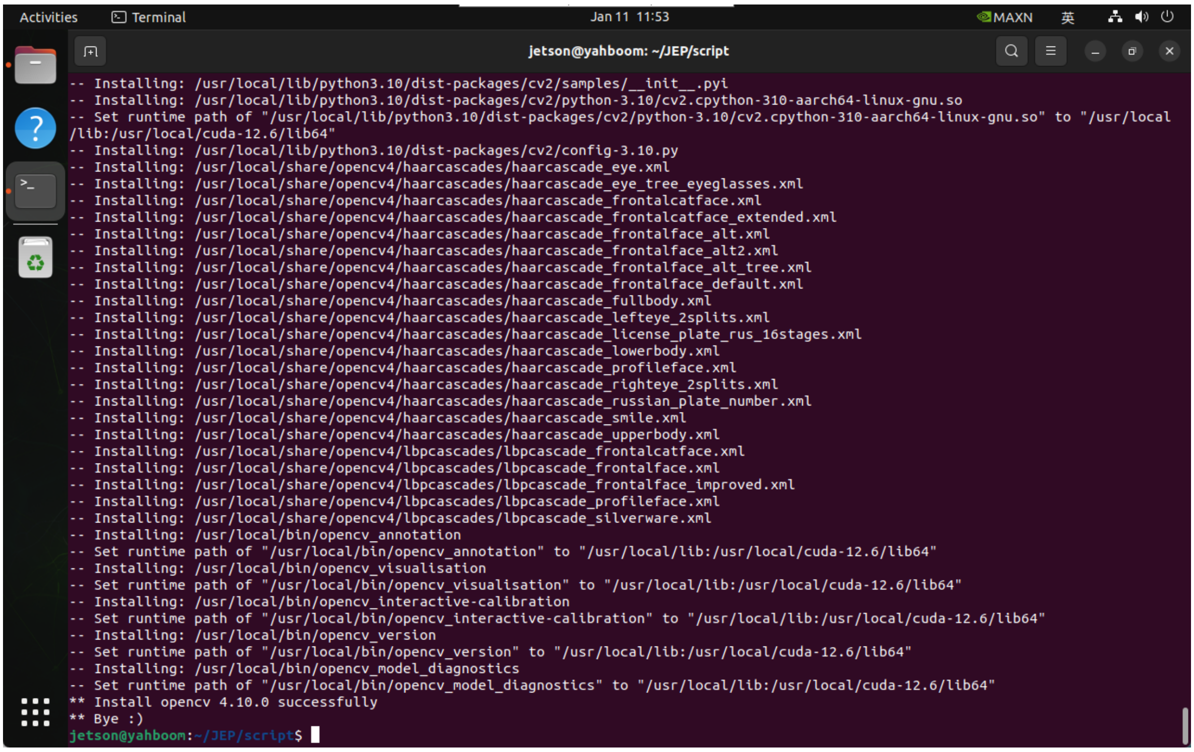
CSI camera cannot be called

Compile OpenCV from source code and enable GStreamer support: basically the entire process is automatically installed. It is recommended to uninstall the old version and install the new version (the script automatically enables CUDA and GStreamer functions)

```
git clone https://github.com/AastaNV/JEP.git
cd JEP/script
bash install_opencv4.10.0_Jetpack6.1.sh
```



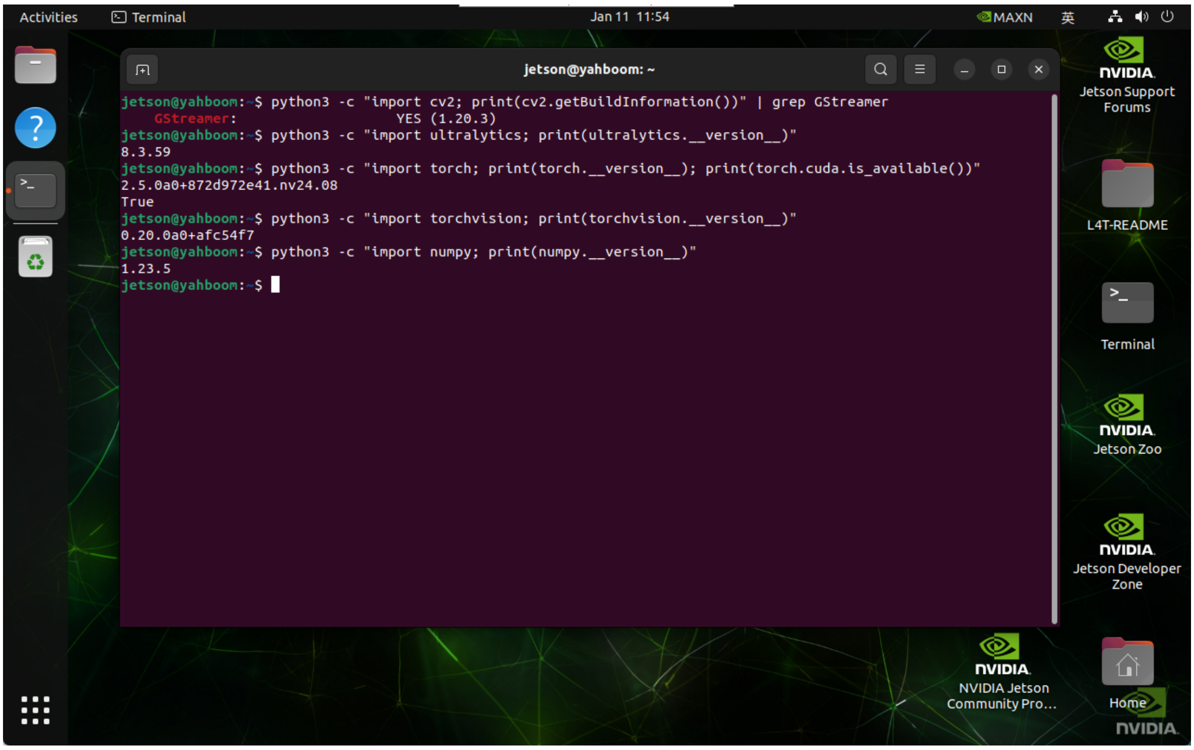
```
jetson@yahboom:~$ git clone https://github.com/AastaNV/JEP.git
Cloning into 'JEP'...
remote: Enumerating objects: 209, done.
remote: Counting objects: 100% (24/24), done.
remote: Compressing objects: 100% (14/14), done.
remote: Total 209 (delta 14), reused 18 (delta 10), pack-reused 185 (from 1)
Receiving objects: 100% (209/209), 233.56 MiB | 10.24 MiB/s, done.
Resolving deltas: 100% (105/105), done.
jetson@yahboom:~$ cd JEP/script
jetson@yahboom:~/JEP/script$ ls
install_opencv4.10.0_Jetpack6.1.sh  install_opencv4.6.0_Jetpack5.sh    install_pyTorch_Xavier.sh
install_opencv4.5.0_Jetpack4.sh     install_opencv4.9.0_Jetpack6.0.sh  topics
jetson@yahboom:~/JEP/script$ bash install_opencv4.10.0_Jetpack6.1.sh
Do you want to remove the default OpenCV (yes/no)?
yes
** Remove other OpenCV first
[sudo] password for jetson:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Note, selecting 'libopencv3.4-java' for glob '*libopencv*'
Note, selecting 'libopencv4.5d-jni' for glob '*libopencv*'
Note, selecting 'libopencv-photo4.0' for glob '*libopencv*'
Note, selecting 'libopencv-videoio4.5d' for glob '*libopencv*'
Note, selecting 'libopencv-dnn4.0' for glob '*libopencv*'
Note, selecting 'libopencv-ml4.0' for glob '*libopencv*'
Note, selecting 'libopencv-objdetect4.5d' for glob '*libopencv*'
Note, selecting 'libopencv-gpu-dev' for glob '*libopencv*'
Note, selecting 'libopencv-videoio-dev' for glob '*libopencv*'
Note, selecting 'libopencv-superres4.5d' for glob '*libopencv*'
Note, selecting 'libopencv-objdetect-dev' for glob '*libopencv*'
Note, selecting 'libopencv-contrib4.5d' for glob '*libopencv*'
Note, selecting 'libopencv-videoio4.0' for glob '*libopencv*'
Note, selecting 'libopencv-superres-dev' for glob '*libopencv*'
Note, selecting 'libopencv4.0-java' for glob '*libopencv*'
Note, selecting 'libopencv-contrib-dev' for glob '*libopencv*'
Note, selecting 'libopencv4.1-java' for glob '*libopencv*'
Note, selecting 'libopencv-imgcodecs4.5d' for glob '*libopencv*'
Note, selecting 'libopencv4.2-java' for glob '*libopencv*'
Note, selecting 'libopencv-imgcodecs-dev' for glob '*libopencv*'
```



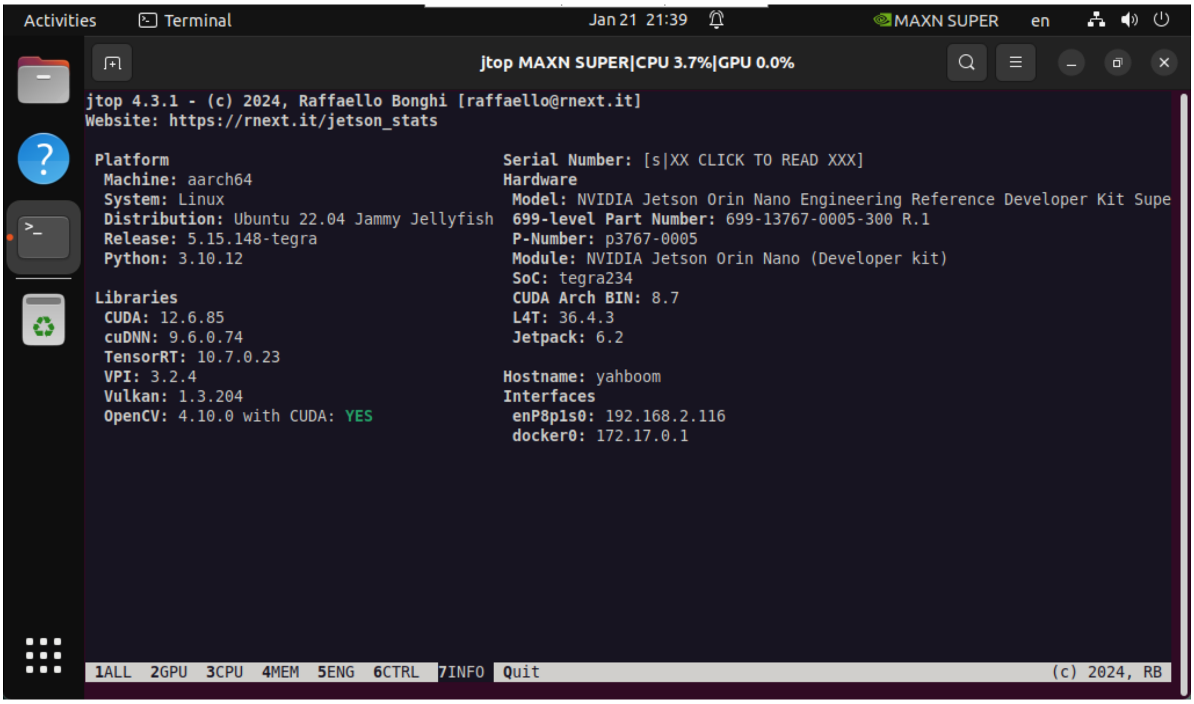
```
-- Installing: /usr/local/lib/python3.10/dist-packages/cv2/samples/__init__.pyi
-- Installing: /usr/local/lib/python3.10/dist-packages/cv2/python-3.10/cv2.cpython-310-aarch64-linux-gnu.so
-- Set runtime path of "/usr/local/lib/python3.10/dist-packages/cv2/python-3.10/cv2.cpython-310-aarch64-linux-gnu.so" to "/usr/local
/lib:/usr/local/cuda-12.6/lib64"
-- Installing: /usr/local/lib/python3.10/dist-packages/cv2/config-3.10.py
-- Installing: /usr/local/share/opencv4/haarcascades/haarcascade_eye.xml
-- Installing: /usr/local/share/opencv4/haarcascades/haarcascade_eye_tree_eyeglasses.xml
-- Installing: /usr/local/share/opencv4/haarcascades/haarcascade_frontalcatface.xml
-- Installing: /usr/local/share/opencv4/haarcascades/haarcascade_frontalcatface_extended.xml
-- Installing: /usr/local/share/opencv4/haarcascades/haarcascade_frontalface_alt.xml
-- Installing: /usr/local/share/opencv4/haarcascades/haarcascade_frontalface_alt2.xml
-- Installing: /usr/local/share/opencv4/haarcascades/haarcascade_frontalface_alt_tree.xml
-- Installing: /usr/local/share/opencv4/haarcascades/haarcascade_frontalface_default.xml
-- Installing: /usr/local/share/opencv4/haarcascades/haarcascade_fullbody.xml
-- Installing: /usr/local/share/opencv4/haarcascades/haarcascade_lefteye_2splits.xml
-- Installing: /usr/local/share/opencv4/haarcascades/haarcascade_license_plate_rus_16stages.xml
-- Installing: /usr/local/share/opencv4/haarcascades/haarcascade_lowerbody.xml
-- Installing: /usr/local/share/opencv4/haarcascades/haarcascade_profileface.xml
-- Installing: /usr/local/share/opencv4/haarcascades/haarcascade_righteye_2splits.xml
-- Installing: /usr/local/share/opencv4/haarcascades/haarcascade_russian_plate_number.xml
-- Installing: /usr/local/share/opencv4/haarcascades/haarcascade_smile.xml
-- Installing: /usr/local/share/opencv4/haarcascades/haarcascade_upperbody.xml
-- Installing: /usr/local/share/opencv4/lbpcascades/lbpcascade_frontalcatface.xml
-- Installing: /usr/local/share/opencv4/lbpcascades/lbpcascade_frontalface.xml
-- Installing: /usr/local/share/opencv4/lbpcascades/lbpcascade_frontalface_improved.xml
-- Installing: /usr/local/share/opencv4/lbpcascades/lbpcascade_profileface.xml
-- Installing: /usr/local/share/opencv4/lbpcascades/lbpcascade_silverware.xml
-- Installing: /usr/local/bin/opencv_annotation
-- Set runtime path of "/usr/local/bin/opencv_annotation" to "/usr/local/lib:/usr/local/cuda-12.6/lib64"
-- Installing: /usr/local/bin/opencv_visualisation
-- Set runtime path of "/usr/local/bin/opencv_visualisation" to "/usr/local/lib:/usr/local/cuda-12.6/lib64"
-- Installing: /usr/local/bin/opencv_interactive-calibration
-- Set runtime path of "/usr/local/bin/opencv_interactive-calibration" to "/usr/local/lib:/usr/local/cuda-12.6/lib64"
-- Installing: /usr/local/bin/opencv_version
-- Set runtime path of "/usr/local/bin/opencv_version" to "/usr/local/lib:/usr/local/cuda-12.6/lib64"
-- Installing: /usr/local/bin/opencv_model_diagnostics
-- Set runtime path of "/usr/local/bin/opencv_model_diagnostics" to "/usr/local/lib:/usr/local/cuda-12.6/lib64"
** Install opencv 4.10.0 successfully
** Bye :)
jetson@yahboom:~/JEP/script$
```

Verify the environment

```
python3 -c "import cv2; print(cv2.getBuildInformation())" | grep GStreamer
python3 -c "import ultralytics; print(ultralytics.__version__)"
python3 -c "import torch; print(torch.__version__);
print(torch.cuda.is_available())"
python3 -c "import torchvision; print(torchvision.__version__)"
python3 -c "import numpy; print(numpy.__version__)"
jtop
```



```
jetson@yahboom:~$ python3 -c "import cv2; print(cv2.getBuildInformation())" | grep GStreamer
    GStreamer:                   YES (1.20.3)
jetson@yahboom:~$ python3 -c "import ultralytics; print(ultralytics.__version__)"
8.3.59
jetson@yahboom:~$ python3 -c "import torch; print(torch.__version__); print(torch.cuda.is_available())"
2.5.0a0+872d972e41.nv24.08
True
jetson@yahboom:~$ python3 -c "import torchvision; print(torchvision.__version__)"
0.20.0a0+afc54f7
jetson@yahboom:~$ python3 -c "import numpy; print(numpy.__version__)"
1.23.5
jetson@yahboom:~$
```



```
jtop 4.3.1 - (c) 2024, Raffaello Bonghi [raffaello@rnext.it]
Website: https://rnext.it/jetson_stats

 Platform                                    Serial Number: [s|XX CLICK TO READ XXX]
  Machine: aarch64                           Hardware
  System: Linux                               Model: NVIDIA Jetson Orin Nano Engineering Reference Developer Kit Supe
  Distribution: Ubuntu 22.04 Jammy Jellyfish  699-level Part Number: 699-13767-0005-300 R.1
  Release: 5.15.148-tegra                     P-Number: p3767-0005
  Python: 3.10.12                             Module: NVIDIA Jetson Orin Nano (Developer kit)
                                              SoC: tegra234
 Libraries                                    CUDA Arch BIN: 8.7
  CUDA: 12.6.85                               L4T: 36.4.3
  cuDNN: 9.6.0.74                             Jetpack: 6.2
  TensorRT: 10.7.0.23
  VPI: 3.2.4                                 Hostname: yahboom
  Vulkan: 1.3.204                            Interfaces
  OpenCV: 4.10.0 with CUDA: YES               enP8p1s0: 192.168.2.116
                                              docker0: 172.17.0.1

 1ALL  2GPU  3CPU  4MEM  5ENG  6CTRL  7INFO  Quit                                                        (c) 2024, RB
```

References

https://docs.ultralytics.com/guides/nvidia-jetson/

https://github.com/AastaNV/JEP