Nvidia Jetson Nano

- https://github.com/Qengineering/Jetson-Nano-Ubuntu-20-image
- interesting projects
- cameras
- camera search engine

remove desktop

```
apt remove -y ubuntu-desktop
apt remove -y gnome-*
apt remove -y chromium-browser chromium-codecs-ffmpeg-extra
apt remove -y libreoffice* docker.io thunderbird bluez*
apt remove -y libnvinfer-samples
# remove snap
systemctl stop snapd
apt remove --purge --assume-yes snapd gnome-software-plugin-snap
rm -rf /var/cache/snapd/

apt autoremove -y
```

```
systemctl disable lightdm whoopsie  ModemManager bluetooth.service gpsd
```

tools

jtop

```
sudo -H pip install -U jetson-stats
```

check CUDA

```
cd /usr/local/cuda-10.0/samples/1_Utilities/deviceQuery
sudo make
./deviceQuery

cd /usr/local/cuda-10.0/samples/1_Utilities/bandwidthTest/
sudo make
./bandwidthTest
```

```
Device 0: "NVIDIA Tegra X1"
  CUDA Driver Version / Runtime Version          10.0 / 10.0
  CUDA Capability Major/Minor version number:    5.3
  Total amount of global memory:                 3957 MBytes (4148756480
bytes)
  ( 1) Multiprocessors, (128) CUDA Cores/MP:     128 CUDA Cores
```

```
  GPU Max Clock rate:                            922 MHz (0.92 GHz)
```

python 3.8

```
sudo apt install libssl-dev zlib1g-dev libncurses5-dev libncursesw5-dev
libreadline-dev libsqlite3-dev
sudo apt install libgdbm-dev libdb5.3-dev libbz2-dev libexpat1-dev liblzma-
dev libffi-dev uuid-dev
```

```
wget https://www.python.org/ftp/python/3.8.0/Python-3.8.0b3.tgz
tar zxvf Python-3.8.0b3.tgz
cd Python-3.8.0b3
CFLAGS=-DOPENSSL_NO_SSL2 ./configure
 --prefix=/opt/python3.8 \
 --enable-optimizations \
 --with-lto

make -j3 PROFILE_TASK="-m test.regrtest --pgo test_array test_base64
test_binascii test_binhex test_binop test_c_locale_coercion test_csv
test_json test_hashlib test_unicode test_codecs test_traceback test_decimal
test_math test_compile test_threading test_time test_fstring test_re
test_float test_class test_cmath test_complex test_iter test_struct
test_slice test_set test_dict test_long test_bytes test_memoryview test_io
test_pickle"

sudo make install
```

numpy

Patch to numpy/random/mt19937.c numpy/random/bit_generator.c numpy/random/generator.c

```
from
PyCode_New(a, 0, k, l, s, f, code, c, n, v, fv, cell, fn, name, fline, lnos)

to
PyCode_New(a, k, l, s, f, code, c, n, v, fv, cell, fn, name, fline, lnos)
```

opencv 4.1

- http://docs.donkeycar.com/guide/robot_sbc/setup_jetson_nano/
- https://devtalk.nvidia.com/default/topic/1056594/jetson-nano-opencv-4-1-0/?offset=9

```
apt remove -y libopencv libopencv-dev libopencv-python libopencv-samples
apt install -y python3.7-dev python3.7-venv curl
apt install -y libv4l-dev v4l-utils qv4l2 v4l2ucp
apt install -y build-essential cmake git libgtk2.0-dev pkg-config
```

```
libavcodec-dev libavformat-dev libswscale-dev
apt install -y libgstreamer1.0-dev libgstreamer-plugins-base1.0-dev
```

add swap to compile opencv

```
fallocate -l 6G /mnt/6GB.swap
mkswap /mnt/6GB.swap
swapon /mnt/6GB.swap
swapon -s
```

```
RELEASE=4.1.1
cd ~/
curl -L https://github.com/opencv/opencv/archive/${RELEASE}.zip -o opencv-
${RELEASE}.zip
curl -L https://github.com/opencv/opencv_contrib/archive/${RELEASE}.zip -o
opencv_contrib-${RELEASE}.zip
unzip opencv-${RELEASE}.zip
unzip opencv_contrib-${RELEASE}.zip
cd opencv-${RELEASE}/
mkdir release
cd release/
cmake -D WITH_CUDA=ON -D CUDA_ARCH_BIN="5.3" -D CUDA_ARCH_PTX="" -D
OPENCV_EXTRA_MODULES_PATH=../../opencv_contrib-${RELEASE}/modules -D
WITH_GSTREAMER=ON -D WITH_LIBV4L=ON -D BUILD_opencv_python2=OFF -D
BUILD_opencv_python3=ON -D BUILD_TESTS=OFF -D BUILD_PERF_TESTS=OFF -D
BUILD_EXAMPLES=OFF -D CMAKE_BUILD_TYPE=RELEASE -D
CMAKE_INSTALL_PREFIX=/usr/local -D PYTHON_EXECUTABLE=/usr/bin/python3.7 -D
PYTHON_DEFAULT_EXECUTABLE=/usr/bin/python3.7 -D BUILD_DOCS=OFF -
DENABLE_PRECOMPILED_HEADERS=OFF ..
make -j3
```

install

```
sudo make install/strip
sudo ldconfig
```

da perfezionare...

```
cp /usr/local/lib/python3.7/site-packages/cv2/python-3.7/cv2.cpython-37m-
aarch64-linux-gnu.so lib/lib/python3.7/site-packages/
```

benchmarks

- **darknet** alexey **20FPS** with YOLOV3-tiny and image size 416 <code> cd /lab/dnn/models/plates-tiny darknet detector demo model.data model.test.cfg backup/model*final.weights /lab/dnn/dataset/dataset/marocco/originals/video/01.mkv -dont*show </code>
- yolo34py **12FPS** <code> python process_video.py –model plates-tiny dataset/dataset/marocco/originals/video/01.mkv </code>