TX1 Set Up Notes

1. Official TX1 starter guide

http://developer.download.nvidia.com/embedded/L4T/r23_Release_v1.0/NVIDIA_Jetson_TX1_Developer_Kit_User_Guide.pdf

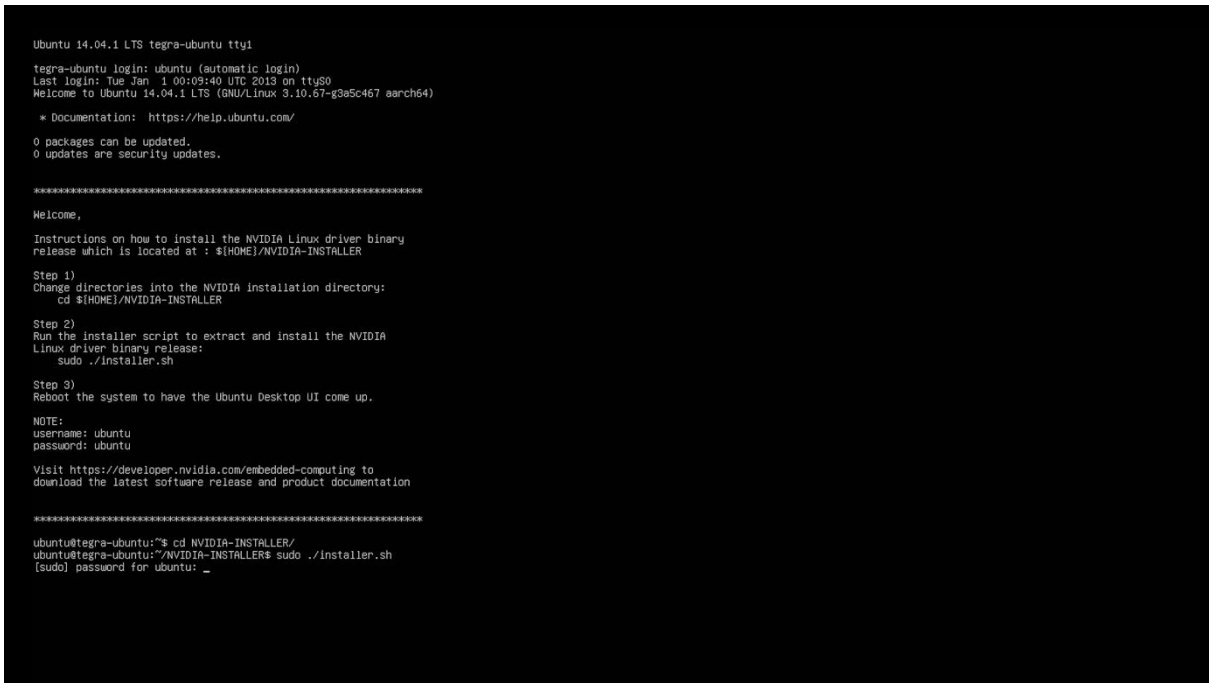
https://www.youtube.com/watch?v=WFUcGGuWhdk

Before powering on the board, plug in the ethernet cable first to enable downloads.

2. Power on

1) Plug in

2) press the rightmost red button



This is just a screenshot from a tutorial. Refer to the actual instructions.

Login: ubuntu

Password: ubuntu (not nvidia)

Reboot: $ sudo reboot

- After reboot, if you are still at the command line. Use $ls to check if **NVIDIA-INSTALL** exists.

- If the graphical desktop is not seen, it might not have been installed. Refer to

http://developer.download.nvidia.com/embedded/L4T/r23_Release_v1.0/l4t_quick_start_guide.txt

You should be able to see the graphical desktop after rebooting.

- **Install web browser:** e.g. firefox *(optional)*

3. Install JetPack Before You Proceed with Neural Network

Here we use **JetPack 3.1**

https://developer.nvidia.com/embedded/jetpack

- The package is available on Nvidia website
- Refer to the JetPack release notes for installation
- Useful video: https://www.youtube.com/watch?v=RJkOGMC8IrY (some outputs could be different due to the difference in various versions)
- After flashing, the user name and password are both **'nvidia'**

1. Recommended: install using an **Ubuntu host PC** with at least **35 GB** free disk space. Alternatively, one could use a VBox VM for installation. (Important: use **Ubuntu 14.04**). Note that VM tends to encounter more problems.

2. If Using VM

- There was not enough space in my Ubuntu partition. Therefore, I used a virtual machine.
- Creating an Ubuntu VM: https://linus.nci.nih.gov/bdge/installUbuntu.html (remember to insert **guest addition** as shown in the post)
- Allocate at least **50 GB** to the VM upon creation and try not to install anything else. Under **Devices/Network/Network** settings, set the network connection to **Bridged Adapter** (default: NAT). This is to allow the host PC to retrieve the IP address of TX1 later. Then start installing JetPack.

- When TX1 is in **force recovery** mode: If you do not see '**Nvidia Corp**' after the running $lsusb command, go to **'Devices/USB'** and select the **"Nvidia ..."** option. Then run $lsusb again. 'Nvidia Corp' should appear by then.
- During the installation, ensure stable network connection and remember to check the storage consumption (go to **system monitor**). It might be a good idea to set the sleeping time of host Ubuntu to "Never" (in **System settings**)

3. Testing

- For JetPack 3.1: download https://developer.nvidia.com/embedded/dlc/l4t-multimedia-api-reference-28-1 as reference for running sample applications. Refer to **Update 1** for specific samples to run.
- E.g. the ocean simulation, the car detection in video, etc.

4. Troubleshooting the installation

- if you encounter any other problems, try google it. If there is no know solution, retry the installation (i.e. reflash the TX1 board) This could be time-consuming and multiple attempts might be required.

Update 1

a. Testing onboard camera:

1) Run $dmesg | grep –i ov5693

If the output contains something like: [OV5693]: probing v4l2 sensor, then the onboard camera is functioning.

If the above output is missing, you might want to reflash the bootloader (replace the **cboot.bin** file). Refer to comment #18 in the following thread for the procedure:

https://devtalk.nvidia.com/default/topic/1019986/jetson-tx1/getting-errors-in-using-onboard-camera-jetpack-3-1-/2

Specific procedure to be performed on the host PC:

- Download the .tar.gz file in the comment and save it in a folder. Cd to that folder and run $tar –xvzf filename.tar.gz to extract the file.
- Rename the extracted file as 'cboot.bin' and replace the original cboot.bin (found at *JetPack/3.1/64_TX1/Linux_for_Tegra_64_tx1/bootloader/t210ref/cboot.bin*) with this new file.
- Put the TX1 into force recovery mode. Then cd to/bootloader/ and run $sudo ./flash.sh -r -k EBT jetson-tx1 mmcblk0p1
- Restart the TX1. Then run $dmesg | grep -i ov5693 again to see if the onboard camera can be detected.

I tried reflashing the bootloader only as stated. However, the onboard camera still could not be detected. Therefore, I reflashed the entire system (still using the new bootloader). After that the onboard camera was detected.

2) For the samples, only **09_camera_jpeg_capture** and **10_camera_recording** can be applied on the onboard camera. For the rest of the samples, a USB camera is needed. We can test the USB camera with one of the remaining examples. E.g. **12_camera_v4l2_cuda**

1. Cd to *~/tegra_multimedia_api/samples/12_camera_v4l2_cuda*
2. Run $ ./camera_v4l2_cuda -d /dev/video1 -s 1280x720 -f YUYV –c (N.B. dev/video0 refers to the onboard camera while dev/video1 refers to the USB camera)

Reference:

https://devtalk.nvidia.com/default/topic/1016726/cannot-run-example-12_camera_v4l2_cuda-/

b. Maximise Performance

run $ sudo ./jetson_clocks.sh

c. Problems with apt-get update

You might encounter a date format warning. This is due to a library in OpenCV4Tegra. However, you may just ignore the warning.

Reference: https://askubuntu.com/questions/194651/why-use-apt-get-upgrade-instead-of-apt-get-dist-upgrade

Update 2: Expand the Storage (Transfer the System onto an SD Card)

1. When first bought, SD cards are normally not in ext4 format. (could be in extFAT). Therefore, we have to format it to ext4 format so that TX1 can read the card. In this case, Eassos PartitionGuru is used for formatting. Refer to http://www.eassos.com/blog/how-to-format-ext4-in-windows-10-8-7-xp/ for more details if the host system is Windows.
2. Follow the instructions at http://www.jetsonhacks.com/2017/01/26/run-jetson-tx1-sd-card/ Note that the directory of the SD card should be 'media/nvidia/SD Root' (if you have named the card 'SD Root') Just drag the icon as shown in the video such that you can always can the correct path of the SD Card.

Update 3: Install OpenCV3 and Test Faster RCNN

1. Replace OpenCV4Tegra with OpenCV3

- **Uninstall OpenCV4Tegra:** http://developer.download.nvidia.com/embedded/L4T/r23_Release_v1.0/OpenCV4Tegra-2.4.12.3-README.txt
- **Install OpenCV3:** http://cyaninfinite.com/tutorials/installing-opencv-in-ubuntu-for-python-3/ (Unfortunately, this link is down by the time this update is written. The method has been tested and it indeed installed OpenCV3 for both python2 and 3)
- **Another link for OpenCV3 installation:** https://askubuntu.com/questions/783956/how-to-install-opencv-3-1-for-python-3-5-on-ubuntu-16-04-lts (I tried this method as but opencv3 was installed only for Python2. Luckily it still worked.)

2. Compile and Test Faster RCNN:

- Follow the instructions here: http://www.tk4479.net/abc869788668/article/details/71802566 (This article is in Chinese, will translate soon)

- Important Changes on the instructions:

Change 1:

之后打开caffe-faster-rcnn将安装好的的caffe目录下的Makefile.config和Makefile.config.example复制到该目录下

然后对 Makefile.config 文件进行修改

```
USE_CUDNN := 1
OPENCV_VERSION := 3
CUSTOM_CXX := g++
WITH_PYTHON_LAYER := 1
INCLUDE_DIRS := $(PYTHON_INCLUDE) /usr/local/include /usr/lib/x86_64-Linux-gnu/hdf5/serial /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
USE_PKG_CONFIG := 1
```

The processors on TX1 are of ARM architecture. Thus, we need to **change "x86_64-linux-gnu" to "aarch64-linux-gnu".**

Change 2:

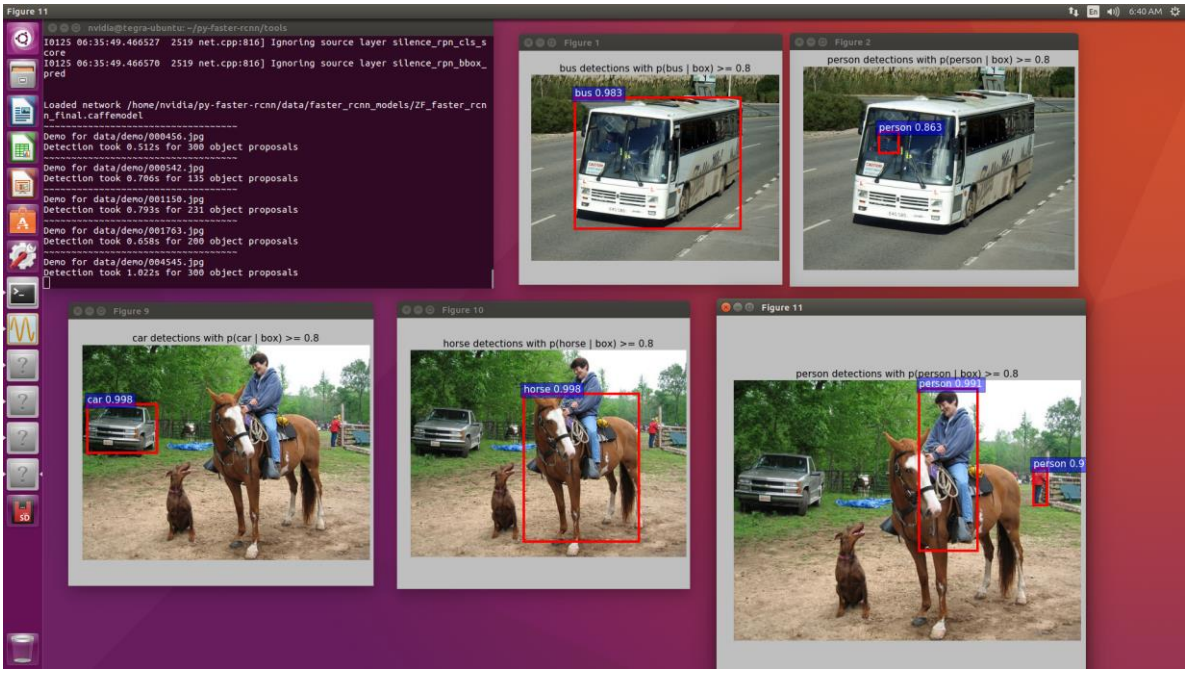
就能通过编译，原因貌似是make只能在当前目录下寻找动态链接库，具体原因还希望有大神进行解答
之后在caffe-fast-rcnn路径下编译

```
sudo make -j8 test
```

The processor on TX1 is a quad-core processor. Therefore, I think sudo make –j4 test is a better option.

- When trying to run demo.py, if the processed failed with a 'killed' but not any error messages, the RAM has to be purged to free up more space. To do so, restart TX1. Then directly run demo.py without creating any other processes beforehand.

Detection Results (Detection on Pictures):



Update 4: Install and run darknet YOLO (new SD card used)

I used a new SD card and copied the system on the internal EMMC to the new SD card (refer to Update2)

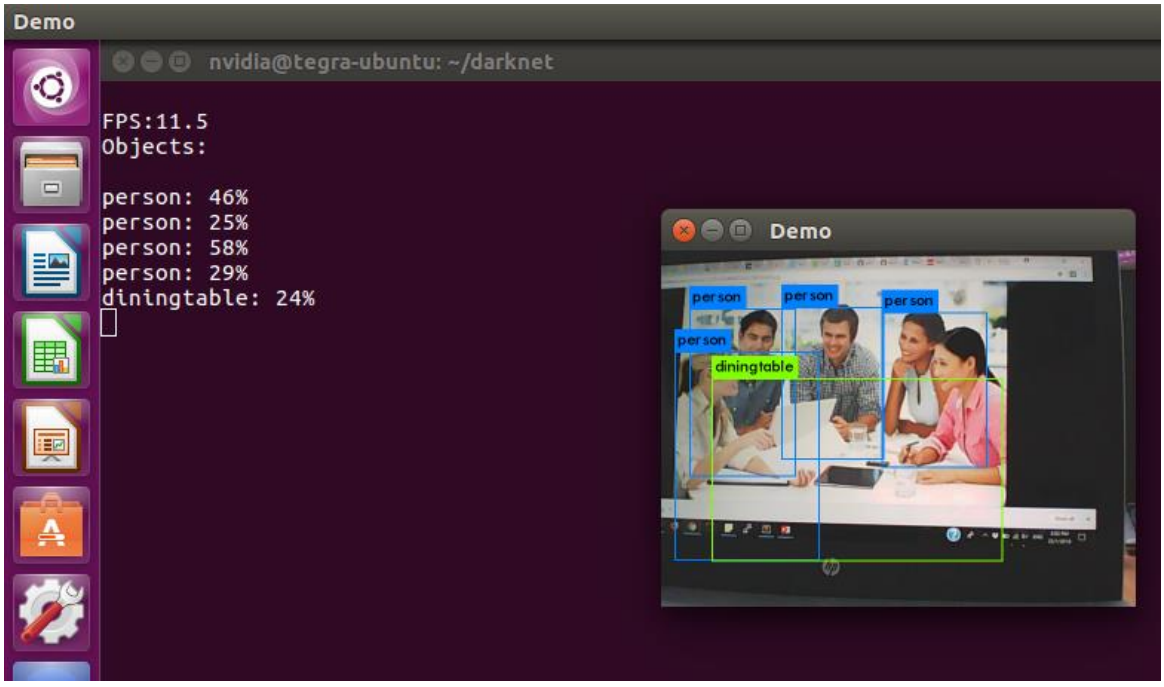
- Refer to https://www.yuthon.com/2016/11/10/YOLO-on-NVIDIA-Jetson-TX1/ for the installation process.
- After running

```
$ sed 's/GPU=0/GPU=1/g' Makefile
$ sed 's/CUDNN=0/CUDNN=1/g' Makefile
$ sed 's/OPENCV=0/OPENCV=1/g' Makefile
```

Double check the Makefile. Ensure that **GPU =1, CUDNN = 1** and **OPENCV = 1**

- A USB webcam is necessary. Connect the webcam to TX1 through a USB hub. Instead of running `$ ./darknet yolo demo cfg/tiny-yolo.cfg tiny-yolo.weights` , run `$ ./darknet yolo demo cfg/tiny-yolo.cfg tiny-yolo.weights -c 1` Then the USB webcam instead of the onboard camera will be used. Various weights can be downloaded from https://pjreddie.com/darknet/yolo/ Note that during execution, the .cfg file name should correspond to the weight file name.

Detection Results (real-time detection):



Other References:

https://devtalk.nvidia.com/default/topic/898129/enabling-camera-on-jetson-tx1-board/

TensorRT Doc:

usr/share/doc/tensorrt/TensotRT User Guide.html (Link found in Jetpack)

Install tensorflow (haven't tested)

http://jany.st/post/2017-05-20-tensorflow-on-nvidia-jetson-tx1.html