**Jetpack**

**Jetpack version 4.3**

**cuda 10.0.326-1**

**cudnn 7.6.3.28-1**

**tensorrt 6.0.1.10**

Jetpack version 4.4

cuda 10.2.89

cudnn 8.0.0.180

tensorrt 7.1.3.0

Logbook

<https://www.baeldung.com/linux/limit-resource-consumption>

<https://www.nvidia.com/gtc/keynote/>

sudo systemctl set-default graphical.target

Check for update by command “sudo apt-get update”

Dpkg (Operating System) is interrupted, manually run “sudo dpkg –configure -a’

Error occur “nvidia-l4t-bootloader”

Go to “/usr/sbin/l4t\_payload\_updater\_t210”

Change user permission to 777 by “sudo chmod 777 l4t\*”

Search function “def \_skip\_check\_old\_ver(self):”

Disable version check by bypassing “return true”

Upgrade the system by “sudo apt-get upgrade”

Double check upgrade by “sudo apt update” && “sudo apt upgrade”

Make sure python3 is installed by “python3 –version”

If not installed redo the update and upgrade / manually install

Make sure pip3 is installed by “pip3 –version”

“sudo apt-get install python3-pip”

“sudo pip3 install -U pip testresources setuptools==49.6.0”

Install the jetbot setup file by executing the setup.py in jetbot folder

“sudo python3 setup.py install”

It might causes error “No module named ‘setuptools’”

To fix this, type ”sudo apt-get install python3-setuptools” to install the module

And then redo the setup.py step

No module named ensurepip

“sudo apt-get install python3.6-venv”

Install traitlets, “pip3 install traitlets”

Install packaging, “pip3 install packaging” (Required for ipwidgets)

Install ipywidgets “pip3 install ipywidgets”

Install pytorch, “wget https://nvidia.box.com/shared/static/p57jwntv436lfrd78inwl7iml6p13fzh.whl -O torch-1.8.0-cp36-cp36m-linux\_aarch64.whl”

“sudo apt-get install python3-pip libopenblas-base libopenmpi-dev libomp-dev”

“pip3 install Cython”

“pip3 install numpy torch-1.8.0-cp36-cp36m-linux\_aarch64.whl”

pip3 install adafruit-circuitpython-motorkit

pi3 install Adafruit\_motorHAT

<https://www.youtube.com/watch?v=wY3rezjj9es>

<https://developer.nvidia.com/embedded/diy-ai-race>

<https://developer.nvidia.com/mpi-solutions-gpus>

<https://github.com/garyexplains/examples/blob/master/how_to_build_nvidia_jetson_gpu_cluster.md>

<https://www.open-mpi.org/faq/?category=rsh>

<https://stackoverflow.com/questions/7589012/combining-two-images-with-opencv>

<https://answers.opencv.org/question/46288/stitching-2-images-together-in-cuda-like-hconcat/>

<https://developer.ridgerun.com/wiki/index.php/Image_Stitching_for_NVIDIA_Jetson/Getting_Started/Evaluating_the_Stitcher>

<https://note.nkmk.me/en/python-opencv-hconcat-vconcat-np-tile/>

HOSTFILE FIRST IP IS THE ROOT

Jetbot single camera: 10.130.26.37

Jetbot 0: 10.129.153.108

Jetbot 1: 10.128.120.46

Jetbot 2: 10.131.72.27

Jetbot 3: 10.130.219.77

Jetson Nano 1: 10.128.34.54

Jetson Nano 2: 10.131.92.21

$ ssh-keygen

$ ssh-copy-id -i ~/.ssh/id\_rsa.pub remoteIP

10.129.153.108

10.128.78.59

10.131.72.27

10.130.181.251

10.128.34.54

10.131.92.21

Mvapich

Download and unpack

wget <http://mvapich.cse.ohio-state.edu/download/mvapich/mv2/mvapich2-2.3.7.tar.gz>

gzip -dc mvapich2-2.3.7.tar.gz | tar -x

cd mvapich2-2.3.7

Error found: Fortran 77 Compiler is not installed

./configure --with-device=ch3:sock --disable-fortran --enable-g=dbg --enable-debuginfo

make -j4

sudo make install

ssh-keygen

sudo apt-get install sshpass

Copy ID to ALL Remote Host and localhost

sshpass -p ‘jetbot’ ssh-copy-id -i ~/.ssh/id\_rsa.pub remoteIP

Establish fingerprint of ALL remoteIP and localhost

ssh-keyscan -H remoteIP >> ~/.ssh/known\_hosts

Need to SSH first time to see if the passwordless setting is setup

Make hosts and ADD all nodes IP into the file

vim hosts

sudo vim /etc/hosts

change other ip (not localhost)

sudo vim /etc/hosts.allow

add sshd: ALL

Compile file [Make sure the exe name is the same across all nodes]

mpicc -g -o mpihello ./hellow.c

run the code

mpirun\_rsh -hostfile hosts -n 3 ./mpihello

HOSTNAME MUST BE DIFFERENT AND UNIQUE IN ALL MACHINES!

EXECUTABLE NAME MUST BE THE SAME!

sudo vim /etc/hostname

sudo vim /etc/hosts

sudo reboot

<https://docs.nvidia.com/jetson/jetpack/introduction/index.html>

HOSTNAME MUST BE DIFFERENT AND UNIQUE IN ALL MACHINES!

Joystick Right

Up: B0

Right: B1

Down: B2

Left: B3

Click: B11

Joystick Left

Axis 0 (Right +1; Left -1)

Axis 1 (Down +1; Up -1)

Click: B10

Y: B0

B: B1

A: B2

X: B3

Axis 0 (Right +1; Left -1)

Axis 1 (Down +1; Up -1)

R1: B5

R2: B7

L1: B4

L2: B6

Start: B9

Select: B8

Home: B12

2 light means intermediate

Joystick Right

Axis 2: (Right +1; Left -1)

Axis 5: (Down +1; Up -1)

Click: B11

Joystick Left

Axis 0 (Right +1; Left -1)

Axis 1 (Down +1; Up -1)

Click: B10

Y: B0

B: B1

A: B2

X: B3

Axis 9:

Anti-Clockwise

+1

0.71

0.43

0.14

-0.14

-0.43

-0.71

R1: B5

R2: B7

L1: B4

L2: B6

Start: B9

Select: B8

Home: B12

Xbox mode

Joystick Right:

Axis 2: (Right +1; Left -1)

Axis 3: (Down +1; Up -1)

Click: B11

Joystick Left:

Axis 0: (Right +1; Left -1)

Axis 1: (Down +1; Up -1)

Click: B10

U: B12

D: B13

L: B14

R: B15

Click: ALL

A: B0

B: B1

X: B2

Y: B3

R1: B5

R2: B7

L1: B4

L2: B6

alias sshjb0=’ssh jetbot@10.129.153.108’

alias sshjb1=’ssh jetbot@10.128.221.63’

alias sshjb2=’ssh jetbot@10.131.72.27’

alias sshjb3=’ssh jetbot@10.130.181.251’

alias sshjn1=’ssh jetbot@10.128.34.54’

alias sshjn2=’ssh jetbot@10.131.92.21’

Auto password

sudo -i

visudo

username ALL=(ALL) NOPASSWD:ALL

Restart docker daemon

Resize partition

10.128.17.96

remove red tint

download <https://www.dropbox.com/s/u80hr1o8n9hqeaj/camera_overrides.isp?dl=0>

scp to jetbot

sudo cp camera\_overrides.isp /var/nvidia/nvcam/settings/

sudo chmod 664 /var/nvidia/nvcam/settings/camera\_overrides.isp

sudo chown root:root /var/nvidia/nvcam/settings/camera\_overrides.ispredredred

collect images

sudo python3 image-capture --flip-method=rotate-180

**classification**

* train/
  + blocked/
  + free/
* test/
  + blocked/
  + free/
* val/
  + blocked/
  + free/

$ cd jetson-inference/python/training/classification

$ python3 train.py --model-dir=models/<YOUR-MODEL> data/<YOUR-DATASET> --epochs 10

default number of epochs is 35

default batch-size is 8

$ python3 onnx\_export.py --model-dir=models/<YOUR-MODEL>

Check if onnx file is correct (optional)

cd ~/jetson-inference/python/training/classification/

sudo apt-get install protobuf-compiler libprotoc-dev

pip3 install onnx

python3 onnx\_validate.py models/<YOUR-MODEL>/resnet18.onnx

May need to use OPENBLAS\_CORETYPE=ARMV8 if it says illegal instruction

test model

$ cd ~/jetson-inference/build/aarch64/bin/

$ ./imagenet --model=/home/$USER/jetson-inference/python/training/classification/models/<YOUR-MODEL>/resnet18.onnx --labels=/path/to/labels.txt --input\_blob=input\_0 --output\_blob=output\_0 csi://0

[TensorRT] WARNING: Using an engine plan file across different models of devices is not recommended and is likely to affect performance or even cause errors.

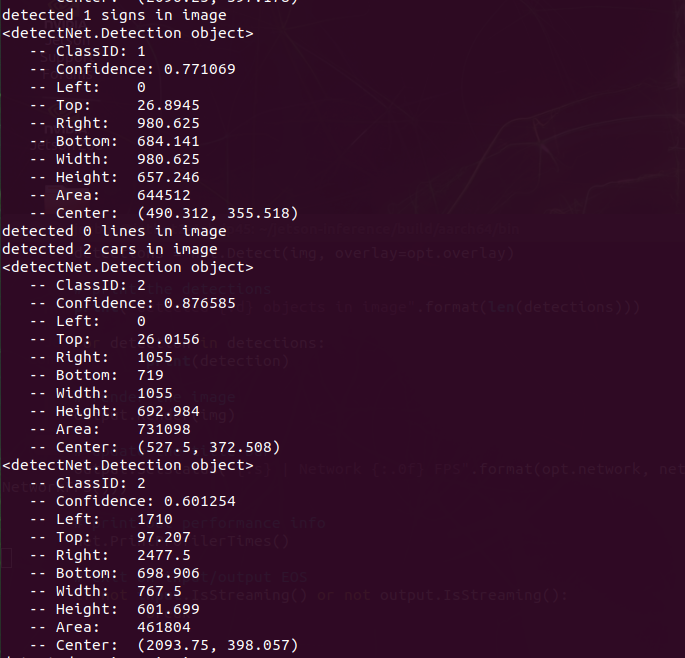
[TensorRT] ERROR: Parameter check failed at: engine.cpp::setBindingDimensions::1044, condition: profileMaxDims.d[i] >= dimensions.d[i]

installed nvidia-l4t-bootloader package post-installation script subprocess returned error exit status 1

Errors were encountered while processing:

nvidia-l4t-bootloader

<https://rawgit.com/dusty-nv/jetson-inference/python/docs/html/python/jetson.inference.html#detectNet>



sign and intersection information

sign 100000-150000 area

collision\_false = [‘SLL’, … ]

collision\_true = [‘SLS’, … ]

collision\_false = ['SLL', ‘SSS’, ‘SSL’, ‘RLR’, ‘RLL’, ‘RSS’, ‘RSL’, ‘RRS’, ‘RRR’, ‘RRL’, ‘LLL’, ‘LRL’]

collision\_true = [‘SLS’, ‘SLR', ‘SSR', ‘SRS', ‘SRR', ‘SRL', ‘RLS', ‘RSR', ‘LLS', ‘LLR', ‘LSS', ‘LSR', ‘LSL', ‘LRS', ‘LRR']

def load(self, arr):

self.date = arr[0]

self.time = arr[1]

self.status = arr[2]

self.id = arr[3]

self.next\_dir = arr[4]

self.path = arr[5]

self.lspd = arr[6]

self.rspd = arr[7]

def get\_date():

return self.date

def get\_time():

return self.time

def get\_stats():

return self.status

def get\_id():

return self.id

def get\_next\_dir():

return self.next\_dir

def get\_path():

return self.path

def get\_lspd():

return self.lspd

def get\_rspd():

return self.rspd

