Nvidia Titan V - Real time object detection

Notebook:	1.Getting Started		
Created:	4/9/2018 11:01 AM	**Updated:**	20/2/2019 10:11 AM
Author:	pengpeng		
Tags:	CV, DL, GPU, TF		

This article is briefing about my experience on setting real time object detection using Nvidia Titan V.
Here highlighted the working combination of software version and step.

Hardware required:

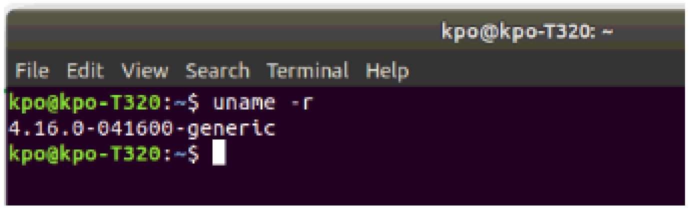
1. Workstation - Dell PowerEdge T320
2. Nvidia Graphic Card - Titan V

Software required :

1. OS - Ubuntu 18.04 Bionic Beaver
2. Linux kernel header - 4.16.0-041600-generic
3. Nvidia Driver - 396.44
4. CUDA release version - 9.2
5. Cudnn - v7.2.1
6. TensorFlow 1.8.0
7. OpenCV 3.4.2

Step 1 - Setup Linux kernel :
Install ubuntu 18.04 on the workstation. By default, ubuntu 18.04, we get "4.15.0–23-generic".
Upgrade kernel to 4.16 version as command :

```
# Download linux kernel files :
        $ mkdir -p ~/kernel-4_16 && cd ~/kernel-4_16
        $ wget http://kernel.ubuntu.com/~kernel-ppa/mainline/v4.16/linux-headers-4.16.0-
        041600_4.16.0-041600.201804012230_all.deb
        $ wget http://kernel.ubuntu.com/~kernel-ppa/mainline/v4.16/linux-headers-4.16.0-
        041600-generic_4.16.0-041600.201804012230_amd64.deb
        $ wget http://kernel.ubuntu.com/~kernel-ppa/mainline/v4.16/linux-image-4.16.0-
        041600-generic_4.16.0-041600.201804012230_amd64.deb
        files
# Install linux-headers :
    $ sudo dpkg -i *.deb
# After installation is complete, reboot the system and Verify :
    $ uname -sr
# Output :
```



```
kpo@kpo-T320:~$ uname -r
4.16.0-041600-generic
kpo@kpo-T320:~$
```

Step 2 - Install Nvidia - Cuda toolkit

<u># Remove previous cuda installation(if you installed cuda before):</u>

```
    $ sudo apt-get purge nvidia*
    $ sudo apt-get autoremove
    $ sudo apt-get auto clean
    $ sudo rm -rf /usr/local/cuda*
```

<u># Install cuda:</u>

```
    $ sudo apt-key adv --fetch-keys
http://developer.download.nvidia.com/compute/cuda/repos/ubuntu1710/x86_64/7fa2af80.pub
    $ echo "deb
https://developer.download.nvidia.com/compute/cuda/repos/ubuntu1710/x86_64 /" | sudo tee
/etc/apt/sources.list.d/cuda.list
    $ sudo apt-get update
    $ sudo apt-get -o Dpkg::Options::="--force-overwrite" install cuda-9-2 cuda-drivers
 cuda=9.0.176-1
```

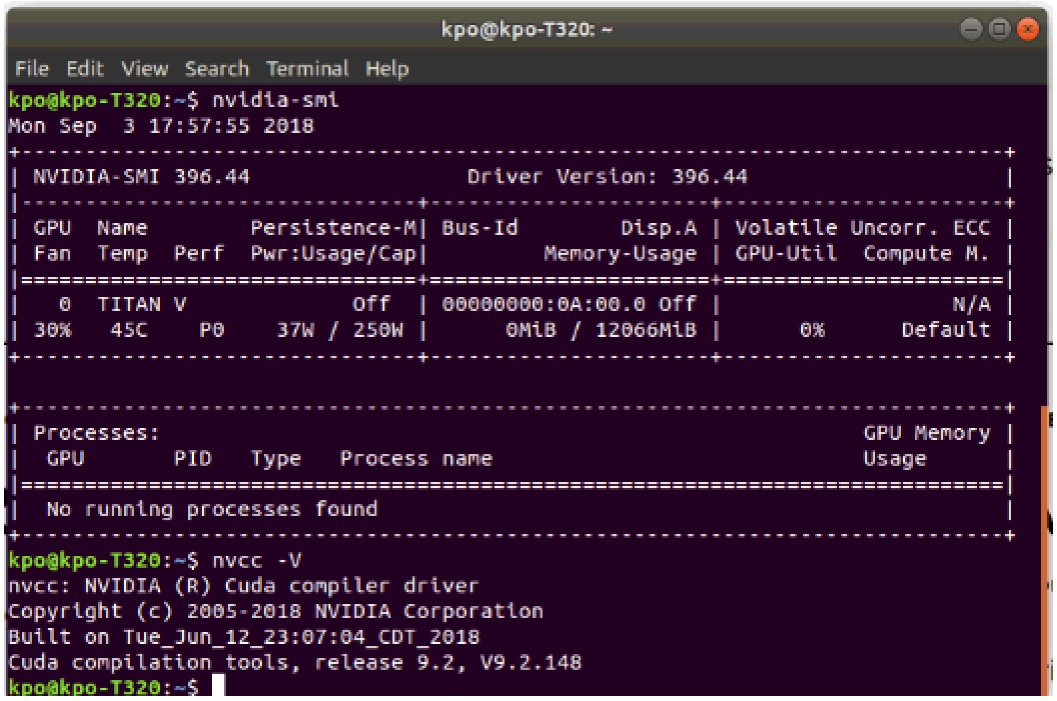
<u># After installation is complete, reboot the system to reload driver.</u>
<u># Setup linux environment for CUDA compilation :</u>

```
    $ esudo apt-get install libcupti-dev
/usr/local/cuda-9.2/bin
echo 'export PATH=/usr/local/cuda-9.2/bin:$PATH' >> ~/.bashrc


    $ echo 'export LD_LIBRARY_PATH=/usr/local/cuda-
9.2/lib64${LD_LIBRARY_PATH:+:${LD_LIBRARY_PATH}}' >> ~/.bashrc
    $ source ~/.bashrc
    $ sudo ldconfig
```

<u># Verify Nvidia-driver and Cuda version:</u>

```
    $ nvidia-smi
    $ nvcc -V
```

<u># Output :</u>


```
kpo@kpo-T320:~$ nvidia-smi
Mon Sep  3 17:57:55 2018
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 396.44                 Driver Version: 396.44                    |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  TITAN V             Off  | 00000000:0A:00.0 Off |                  N/A |
| 30%   45C    P0    37W / 250W |      0MiB / 12066MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID   Type   Process name                             Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
kpo@kpo-T320:~$ nvcc -V
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2018 NVIDIA Corporation
Built on Tue_Jun_12_23:07:04_CDT_2018
Cuda compilation tools, release 9.2, V9.2.148
kpo@kpo-T320:~$
```

# Install GPU-accelerated library cuDNN v7.2.1 :

#Download cudnn-9.2-linux-x64-v7.2.1.38.tgz from https://developer.nvidia.com/cudnn

```
    $ cd ~/Downloads #asssume file is saved in folder "~/Downloads"
    $ tar -xf cudnn-9.2-linux-x64-v7.2.1.38.tgz
    $ sudo cp -R cuda/include/* /usr/local/cuda-9.2/include
    $ sudo cp -R cuda/lib64/* /usr/local/cuda-9.2/lib64
```

*#Install Dependencies (**libcupti**):*

```
    $ sudo apt-get install libcupti-dev
    $ echo 'export LD_LIBRARY_PATH=/usr/local/cuda/extras/CUPTI/lib64:$LD_LIBRARY_PATH' >>
~/.bashrc
```

Step 3 : Instal opencv 3.4.2

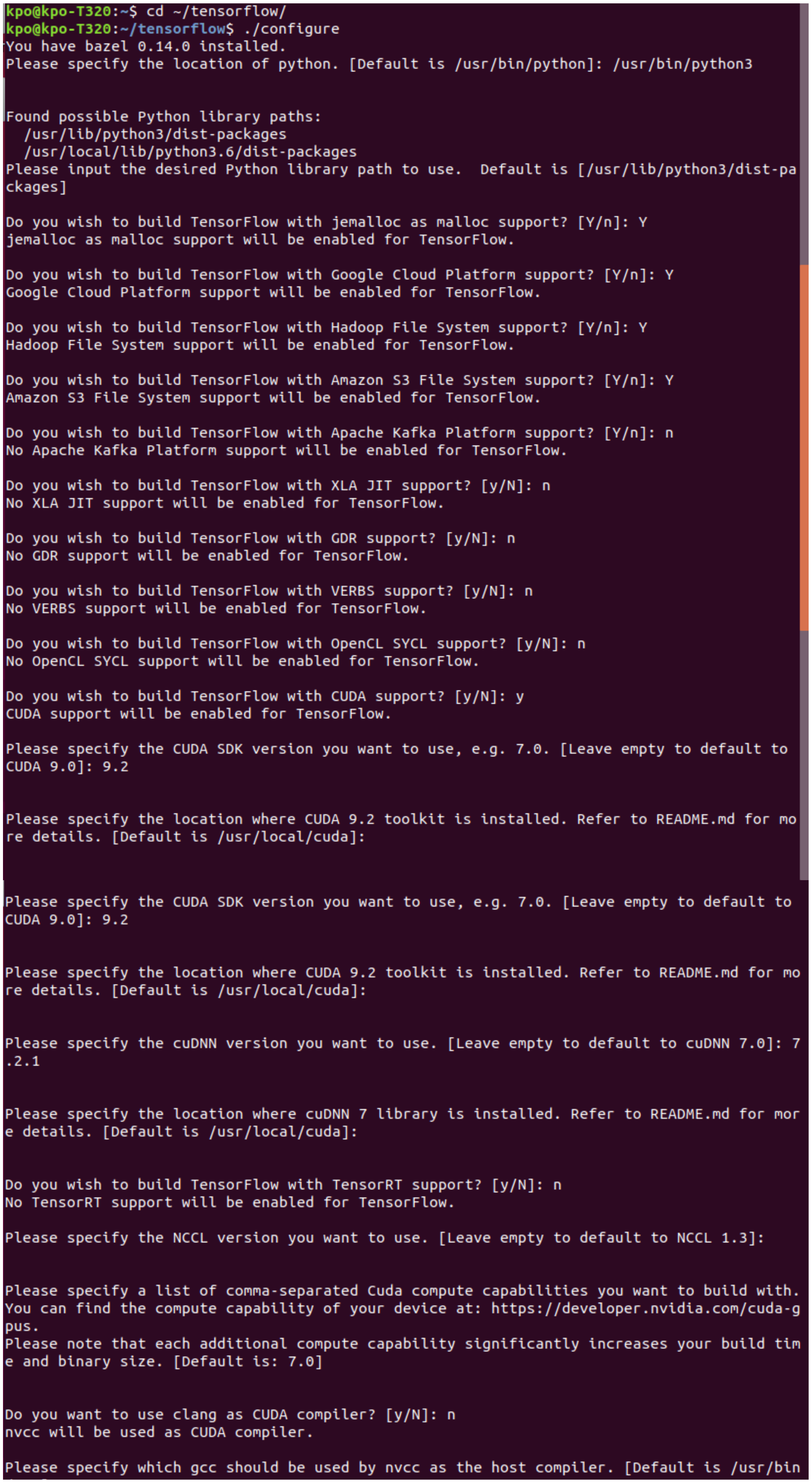
# Run the script below to install opencv 3.4.2

install-opencv.sh
1/4/2019 5:23 PM, 2.4 KB

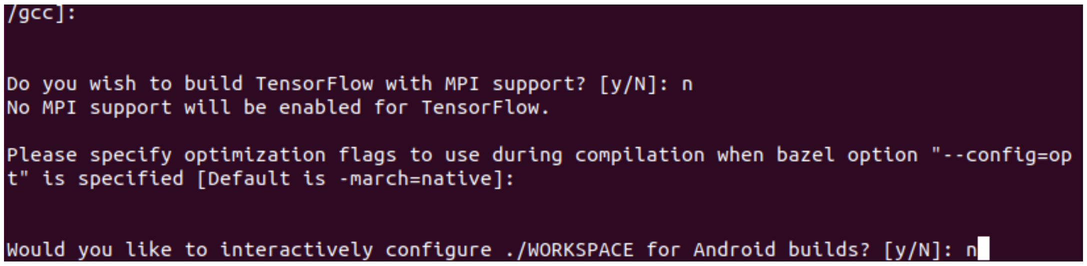
Step 4 : Install TensorFlow v1.8 by source :

<u># Install dependencies :</u>

```
$ sudo apt-get install python-numpy python-dev python-pip python-wheel
$ sudo apt-get install python3-numpy python3-dev python3-pip python3-wheel
# Install Bazel - to build tensorflow source file:
$ cd ~/
$ wget https://github.com/bazelbuild/bazel/releases/download/0.14.0/bazel-0.14.0-installer-linux-x86_64.sh
$ chmod +x bazel-0.14.0-installer-linux-x86_64.sh
$ ./bazel-0.14.0-installer-linux-x86_64.sh --user
$ echo 'export PATH="$PATH:$HOME/bin"' >> ~/.bashrc
# Reload environment variables
$ source ~/.bashrc
$ sudo ldconfig
# Download tensorflow 1.8.0 and configure
$ cd ~/
$ git clone https://github.com/tensorflow/tensorflow.git
$ cd tensorflow
$ git pull
$ git checkout r1.8
$ ./configure
# tensorflow configuration as :
```



```
kpo@kpo-T320:~$ cd ~/tensorflow/
kpo@kpo-T320:~/tensorflow$ ./configure
You have bazel 0.14.0 installed.
Please specify the location of python. [Default is /usr/bin/python]: /usr/bin/python3


Found possible Python library paths:
  /usr/lib/python3/dist-packages
  /usr/local/lib/python3.6/dist-packages
Please input the desired Python library path to use.  Default is [/usr/lib/python3/dist-pa
ckages]

Do you wish to build TensorFlow with jemalloc as malloc support? [Y/n]: Y
jemalloc as malloc support will be enabled for TensorFlow.

Do you wish to build TensorFlow with Google Cloud Platform support? [Y/n]: Y
Google Cloud Platform support will be enabled for TensorFlow.

Do you wish to build TensorFlow with Hadoop File System support? [Y/n]: Y
Hadoop File System support will be enabled for TensorFlow.

Do you wish to build TensorFlow with Amazon S3 File System support? [Y/n]: Y
Amazon S3 File System support will be enabled for TensorFlow.

Do you wish to build TensorFlow with Apache Kafka Platform support? [Y/n]: n
No Apache Kafka Platform support will be enabled for TensorFlow.

Do you wish to build TensorFlow with XLA JIT support? [y/N]: n
No XLA JIT support will be enabled for TensorFlow.

Do you wish to build TensorFlow with GDR support? [y/N]: n
No GDR support will be enabled for TensorFlow.

Do you wish to build TensorFlow with VERBS support? [y/N]: n
No VERBS support will be enabled for TensorFlow.

Do you wish to build TensorFlow with OpenCL SYCL support? [y/N]: n
No OpenCL SYCL support will be enabled for TensorFlow.

Do you wish to build TensorFlow with CUDA support? [y/N]: y
CUDA support will be enabled for TensorFlow.

Please specify the CUDA SDK version you want to use, e.g. 7.0. [Leave empty to default to
CUDA 9.0]: 9.2


Please specify the location where CUDA 9.2 toolkit is installed. Refer to README.md for mo
re details. [Default is /usr/local/cuda]:


Please specify the CUDA SDK version you want to use, e.g. 7.0. [Leave empty to default to
CUDA 9.0]: 9.2


Please specify the location where CUDA 9.2 toolkit is installed. Refer to README.md for mo
re details. [Default is /usr/local/cuda]:


Please specify the cuDNN version you want to use. [Leave empty to default to cuDNN 7.0]: 7
.2.1


Please specify the location where cuDNN 7 library is installed. Refer to README.md for mor
e details. [Default is /usr/local/cuda]:


Do you wish to build TensorFlow with TensorRT support? [y/N]: n
No TensorRT support will be enabled for TensorFlow.

Please specify the NCCL version you want to use. [Leave empty to default to NCCL 1.3]:


Please specify a list of comma-separated Cuda compute capabilities you want to build with.
You can find the compute capability of your device at: https://developer.nvidia.com/cuda-g
pus.
Please note that each additional compute capability significantly increases your build tim
e and binary size. [Default is: 7.0]


Do you want to use clang as CUDA compiler? [y/N]: n
nvcc will be used as CUDA compiler.

Please specify which gcc should be used by nvcc as the host compiler. [Default is /usr/bin
```



```
/gcc]:

Do you wish to build TensorFlow with MPI support? [y/N]: n
No MPI support will be enabled for TensorFlow.

Please specify optimization flags to use during compilation when bazel option "--config=op
t" is specified [Default is -march=native]:

Would you like to interactively configure ./WORKSPACE for Android builds? [y/N]: n
```

# Configuration finished

# Build tensorflow using bazel **This process takes a fairly long time.*

```
$ bazel build --config=opt //tensorflow/tools/pip_package:build_pip_package
```

# To build whl file issue following command:

```
$ bazel-bin/tensorflow/tools/pip_package/build_pip_package tensorflow_pkg
```

# Install tensorflow with pip:

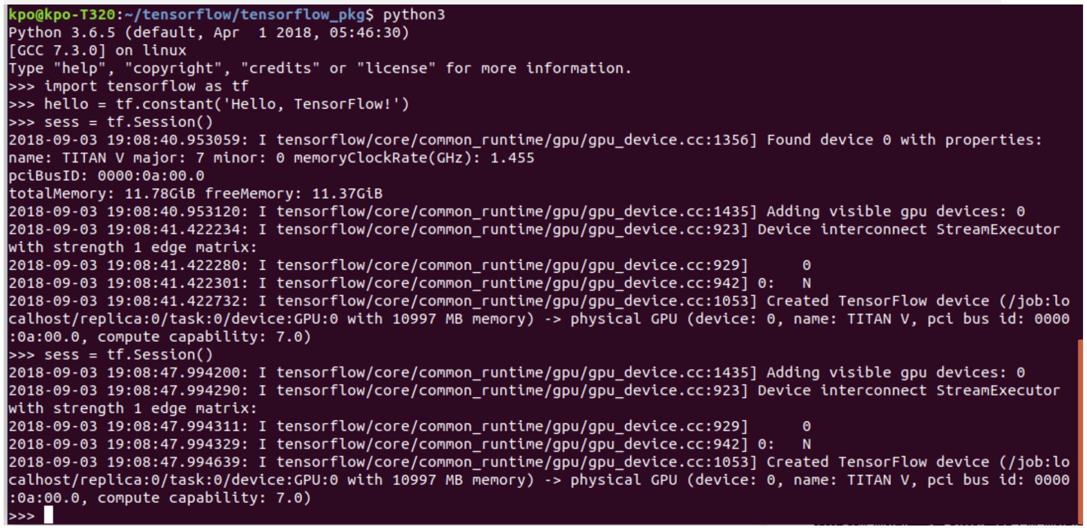
```
$ cd tensorflow_pkg
$ pip3 install tensorflow*.whl
```

# Verify tensorflow installation

- Run in terminal

# output as :

```
python3
import tensorflow as tf
hello = tf.constant('Hello, TensorFlow!')
sess = tf.Session()
print(sess.run(hello))
```



```
kpo@kpo-T320:~/tensorflow/tensorflow_pkg$ python3
Python 3.6.5 (default, Apr  1 2018, 05:46:30)
[GCC 7.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import tensorflow as tf
>>> hello = tf.constant('Hello, TensorFlow!')
>>> sess = tf.Session()
2018-09-03 19:08:40.953059: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1356] Found device 0 with properties:
name: TITAN V major: 7 minor: 0 memoryClockRate(GHz): 1.455
pciBusID: 0000:0a:00.0
totalMemory: 11.78GiB freeMemory: 11.37GiB
2018-09-03 19:08:40.953120: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1435] Adding visible gpu devices: 0
2018-09-03 19:08:41.422234: I tensorflow/core/common_runtime/gpu/gpu_device.cc:923] Device interconnect StreamExecutor
with strength 1 edge matrix:
2018-09-03 19:08:41.422280: I tensorflow/core/common_runtime/gpu/gpu_device.cc:929]      0
2018-09-03 19:08:41.422301: I tensorflow/core/common_runtime/gpu/gpu_device.cc:942] 0:   N
2018-09-03 19:08:41.422732: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1053] Created TensorFlow device (/job:lo
calhost/replica:0/task:0/device:GPU:0 with 10997 MB memory) -> physical GPU (device: 0, name: TITAN V, pci bus id: 0000
:0a:00.0, compute capability: 7.0)
>>> sess = tf.Session()
2018-09-03 19:08:47.994200: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1435] Adding visible gpu devices: 0
2018-09-03 19:08:47.994290: I tensorflow/core/common_runtime/gpu/gpu_device.cc:923] Device interconnect StreamExecutor
with strength 1 edge matrix:
2018-09-03 19:08:47.994311: I tensorflow/core/common_runtime/gpu/gpu_device.cc:929]      0
2018-09-03 19:08:47.994329: I tensorflow/core/common_runtime/gpu/gpu_device.cc:942] 0:   N
2018-09-03 19:08:47.994639: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1053] Created TensorFlow device (/job:lo
calhost/replica:0/task:0/device:GPU:0 with 10997 MB memory) -> physical GPU (device: 0, name: TITAN V, pci bus id: 0000
:0a:00.0, compute capability: 7.0)
>>> 
```

Step 5 : Real time Object Detection - tensorflow - CoCo Model

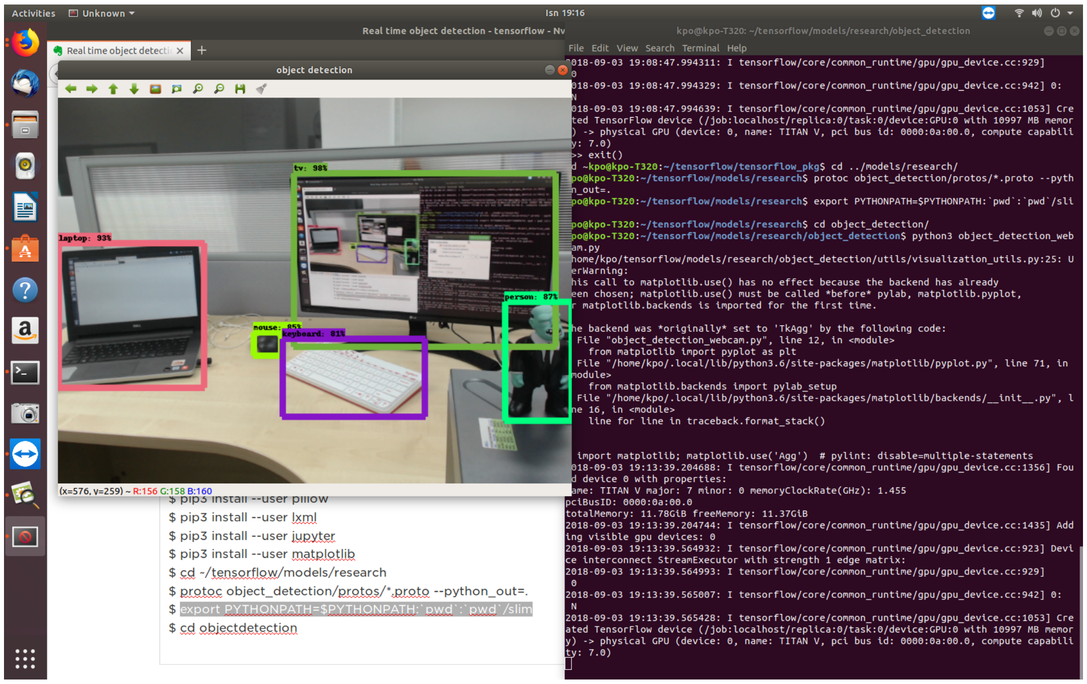
```
$ cd ~/tensorflow
$ git clone https://github.com/tensorflow/models.git
$ sudo apt-get install protobuf-compiler python-pil python-lxml python-tk
$ pip3 install --user Cython
$ pip3 install --user contextlib2
$ pip3 install --user pillow
$ pip3 install --user lxml
$ pip3 install --user jupyter
$ pip3 install --user matplotlib
$ cd ~/tensorflow/models/research
$ protoc object_detection/protos/*.proto --python_out=.
$ export PYTHONPATH=$PYTHONPATH:`pwd`:`pwd`/slim

$ cd object_detection
```

# Place 'object_detection_webcam.py' in ~/tensorflow/models/research/objectdetection directory

object_detection_webcam.py
1/4/2019 5:23 PM, 4.2 KB

```
$ python3 object_detection_webcam.py
```

# output as :



```
$ pip3 install --user pillow
$ pip3 install --user lxml
$ pip3 install --user jupyter
$ pip3 install --user matplotlib
$ cd ~/tensorflow/models/research
$ protoc object_detection/protos/*.proto --python_out=.
$ export PYTHONPATH=$PYTHONPATH:`pwd`:`pwd`/slim
$ cd objectdetection
```

kpo@kpo-T320: ~/tensorflow/models/research/object_detection

File Edit View Search Terminal Help

```
018-09-03 19:08:47.994311: I tensorflow/core/common_runtime/gpu/gpu_device.cc:929]
0
018-09-03 19:08:47.994329: I tensorflow/core/common_runtime/gpu/gpu_device.cc:942] 0:
N
018-09-03 19:08:47.994639: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1053] Cre
ted TensorFlow device (/job:localhost/replica:0/task:0/device:GPU:0 with 10997 MB memor
) -> physical GPU (device: 0, name: TITAN V, pci bus id: 0000:0a:00.0, compute capabili
y: 7.0)
>> exit()
d ~kpo@kpo-T320:~/tensorflow/tensorflow_pkg$ cd ../models/research/
po@kpo-T320:~/tensorflow/models/research$ protoc object_detection/protos/*.proto --pyth
n_out=.
po@kpo-T320:~/tensorflow/models/research$ export PYTHONPATH=$PYTHONPATH:`pwd`:`pwd`/sli

po@kpo-T320:~/tensorflow/models/research$ cd object_detection/
po@kpo-T320:~/tensorflow/models/research/object_detection$ python3 object_detection_web
am.py
home/kpo/tensorflow/models/research/object_detection/utils/visualization_utils.py:25: U
erWarning:
his call to matplotlib.use() has no effect because the backend has already
een chosen; matplotlib.use() must be called *before* pylab, matplotlib.pyplot,
r matplotlib.backends is imported for the first time.

he backend was *originally* set to 'TkAgg' by the following code:
 File "object_detection_webcam.py", line 12, in <module>
   from matplotlib import pyplot as plt
 File "/home/kpo/.local/lib/python3.6/site-packages/matplotlib/pyplot.py", line 71, in
module>
   from matplotlib.backends import pylab_setup
 File "/home/kpo/.local/lib/python3.6/site-packages/matplotlib/backends/__init__.py", l
ne 16, in <module>
   line for line in traceback.format_stack()


 import matplotlib; matplotlib.use('Agg')  # pylint: disable=multiple-statements
018-09-03 19:13:39.204688: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1356] Fou
d device 0 with properties:
ame: TITAN V major: 7 minor: 0 memoryClockRate(GHz): 1.455
pciBusID: 0000:0a:00.0
totalMemory: 11.78GiB freeMemory: 11.37GiB
2018-09-03 19:13:39.204744: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1435] Add
ing visible gpu devices: 0
2018-09-03 19:13:39.564932: I tensorflow/core/common_runtime/gpu/gpu_device.cc:923] Devi
ce interconnect StreamExecutor with strength 1 edge matrix:
2018-09-03 19:13:39.564993: I tensorflow/core/common_runtime/gpu/gpu_device.cc:929]
 0
2018-09-03 19:13:39.565007: I tensorflow/core/common_runtime/gpu/gpu_device.cc:942] 0:
 N
2018-09-03 19:13:39.565428: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1053] Cre
ated TensorFlow device (/job:localhost/replica:0/task:0/device:GPU:0 with 10997 MB memor
y) -> physical GPU (device: 0, name: TITAN V, pci bus id: 0000:0a:00.0, compute capabili
ty: 7.0)
```