6.TensorRT on-board camera real-time image recognition tutorial

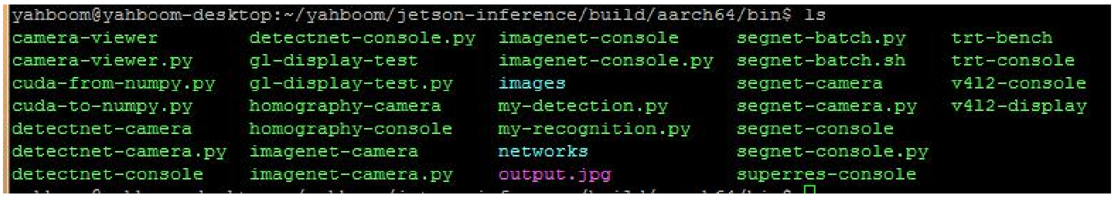
1. Preparation

Before we start this step, we need to make sure that we have completed all the steps in Tutorials 4 and 5 and can test simple examples.

2.Enter the executable directory

If you are in the **jetson-inference** directory, please execute the following directory.

cd   build/aarch64/bin/



```
yahboom@yahboom-desktop:~/yahboom/jetson-inference/build/aarch64/bin$ ls
camera-viewer        detectnet-console.py  imagenet-console     segnet-batch.py    trt-bench
camera-viewer.py     gl-display-test       imagenet-console.py  segnet-batch.sh    trt-console
cuda-from-numpy.py   gl-display-test.py    images               segnet-camera      v4l2-console
cuda-to-numpy.py     homography-camera     my-detection.py      segnet-camera.py   v4l2-display
detectnet-camera     homography-console    my-recognition.py    segnet-console
detectnet-camera.py  imagenet-camera       networks             segnet-console.py
detectnet-console    imagenet-camera.py    output.jpg           superres-console
```

3.Execute image recognition command

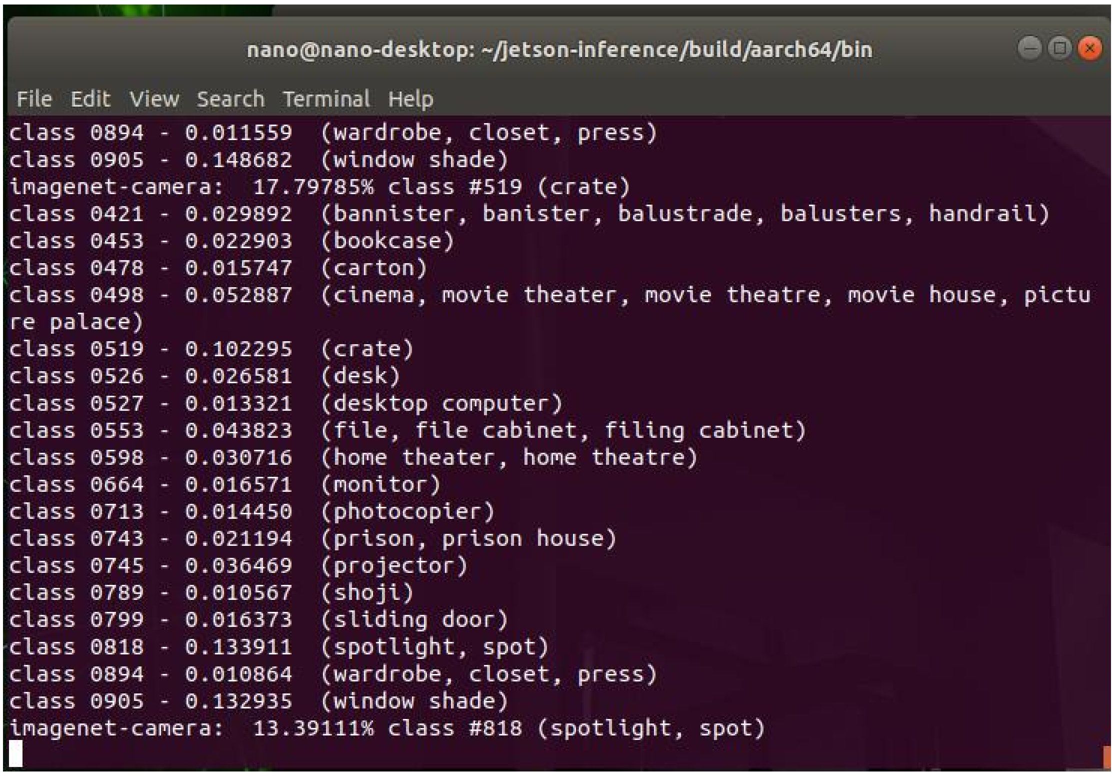
At this point we are better able to execute by the remote desktop, otherwise you may not see the camera interface, or connect by VNC remote desktop.

Enter the bin directory:

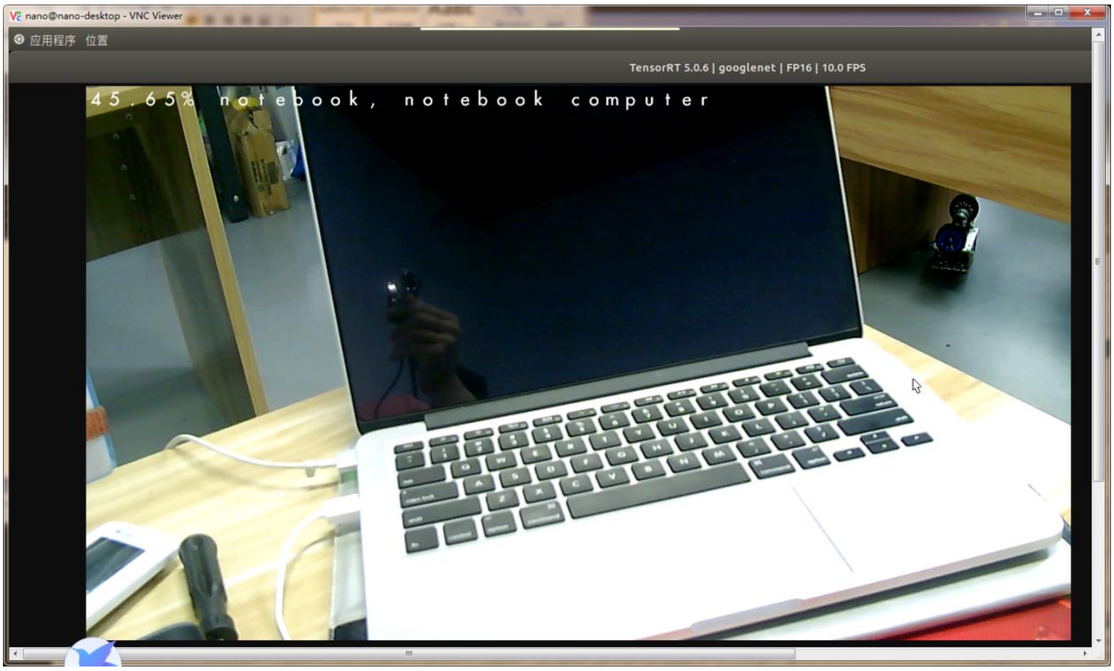
The live image recognition demo is located in **/aarch64/bin** and call **imagenet-camera**. It runs on the live camera stream and loads googlenet or alexnet using TensorRT based on user parameters.

```
$ ./imagenet-camera googlenet                    # Run with googlenet
$ ./imagenet-camera alexnet                      # Run with alexnet
```

Frames per second (FPS), the classification object name from the video and the confidence of the classification object are printed to the OpenGL window title bar. By default, the application can recognize up to 1000 different types of objects, name mappings for 1000 types of objects, which can be found under repo: data/networks/ilsvrc12_synset_words.txt


nano@nano-desktop: ~/jetson-inference/build/aarch64/bin
File Edit View Search Terminal Help

```
class 0894 - 0.011559  (wardrobe, closet, press)
class 0905 - 0.148682  (window shade)
imagenet-camera:  17.79785% class #519 (crate)
class 0421 - 0.029892  (bannister, banister, balustrade, balusters, handrail)
class 0453 - 0.022903  (bookcase)
class 0478 - 0.015747  (carton)
class 0498 - 0.052887  (cinema, movie theater, movie theatre, movie house, pictu
re palace)
class 0519 - 0.102295  (crate)
class 0526 - 0.026581  (desk)
class 0527 - 0.013321  (desktop computer)
class 0553 - 0.043823  (file, file cabinet, filing cabinet)
class 0598 - 0.030716  (home theater, home theatre)
class 0664 - 0.016571  (monitor)
class 0713 - 0.014450  (photocopier)
class 0743 - 0.021194  (prison, prison house)
class 0745 - 0.036469  (projector)
class 0789 - 0.010567  (shoji)
class 0799 - 0.016373  (sliding door)
class 0818 - 0.133911  (spotlight, spot)
class 0894 - 0.010864  (wardrobe, closet, press)
class 0905 - 0.132935  (window shade)
imagenet-camera:  13.39111% class #818 (spotlight, spot)
```



```
TensorRT 5.0.6 | googlenet | FP16 | 10.0 FPS
45.65% notebook, notebook computer
```

When an object is recognized, the English name of the object is displayed on the interface, and the percentage is the matching percentage.

！ Note:

After the camera mount is installed, if the video picture is upside down, you can use the following methods to set and modify:

```
$ cd ~/jetson-inference/utils/camera/
$ gedit gstCamera.cpp
```

~~~
- Change the folloing code:
```bash
#if NV_TENSORRT_MAJOR > 1 && NV_TENSORRT_MAJOR < 5 // if JetPack 3.1-3.3
(different flip-method)
const int flipMethod = 0; // Xavier (w/TRT5) camera is mounted inverted
#else
const int flipMethod = 2;
#endif

We need to change the above code to the code shown below:
#if NV_TENSORRT_MAJOR > 1 && NV_TENSORRT_MAJOR < 5 // if JetPack 3.1-3.3
(different flip-method)
const int flipMethod = 0; // Xavier (w/TRT5) camera is mounted inverted
#else
const int flipMethod = 0;  // 2 change 0
#endif
~~~

Build the Code:
```bash
$ cd ~/jetson-inference/build/
$ make
$sudo make install
```