

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/

```
nano@nano-desktop:~/jetson-inference$ cd build/aarch64/bin/
nano@nano-desktop:~/jetson-inference/build/aarch64/bin$ ls
                    drone_0255.png
airplane_0.jpg
                                         homography-console red_apple_0.jpg
                    drone_0427.png
                                         imagenet-camera
banana_0.jpg
                                                               segnet-batch.sh
bird_0.jpg
                    drone_0428.png
                                         imagenet-console
                                                               segnet-camera
black_bear.jpg
                    drone_0435.png
                                         networks
                                                               segnet-console
                    drone_0436.png
bottle_0.jpg
                                                               superres-console
                                         orange_0.jpg
                                         orange_1.jpg
output_0.jpg
brown_bear.jpg
                    fontmapA.png
                                                               trt-bench
cat_0.jpg
                    fontmapB.png
                                                               trt-console
                   gl-display-test
                                         peds-001.jpg
detectnet-camera
                                                               v4l2-console
detectnet-console granny_smith_0.jpg peds-00; jpg
                                                               v4l2-display
dog_0.jpg
                    granny_smith_1.jpg peds-003.jpg
dog_1.jpg
                    gst-camera
                                          peds-004.jpg
dog_2.jpg homography-camera polar_bear.jpg
nano@nano-desktop:~/jetson-inference/build/aarch64/bin$
```

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



```
File Edit Tabs Help
         fc6 + relu6 (caskFullyConnectedFP32)    Set Tactic Name: maxwell_sgemm_128x64_relu_nn_v1
TRT
         fc6 + relu6 (caskFullyConnectedFP32) Set Tactic Name: maxwell sgemm 64x64 relu nn v1
fc6 + relu6 (caskFullyConnectedFP32) Set Tacti System throttled due to Over-curre
TRT]
                                                           System throttled due to Over-current.
         fc6 + relu6 (caskFullyConnectedFP32) Set Tacti
TRT]
                          Timing Runner: fc6 + relu6 (Caskrullyconnected)
TRT
         fc6 + relu6 (caskFullyConnectedFP32) Set Tactic Name: maxwell_sgemm_128x128_relu_nn_v1
TRT]
TRT]
        Tactic: 8883888914904656451 skipped. Persistent device scratch requested: 151011840, ava
able:
       103736320
         fc6 + relu6 (caskFullyConnectedFP32) Set Tactic Name: maxwell_sgemm 128x64 relu nn v1
TRT]
         Tactic: 5453137127347942357 skipped. Persistent device scratch requested: 151011840, avai
TRT]
able: 103736320
TRT]
         fc6 + relu6 (caskFullyConnectedFP32) Set Tactic Name: maxwell sgemm 64x64 relu nn v1
         Tactic: 5373503982740029499 skipped. Persistent device scratch requested: 151011840, avai
TRT1
able: 103736320
         fc6 + relu6 (caskFullyConnectedFP32) Set Tactic Name: maxwell_sgemm_32x128 relu_nn_v1
TRT]
         Tactic: 4133936625481774016 skipped. Persistent device scratch requested: 151011840, avai
able: 103736320
         fc6 + relu6 (caskFullyConnectedFP32) Set Tactic Name: maxwell_sgemm_128x32_relu_nn_v1
TRT
         Tactic: 1933552664043962183 skipped. Persistent device scratch requested: I5101I840, avai
TRT]
able:
      103736320
TRT1
         Fastest Tactic: -3360065831133338131 Time: 3.40282e+38
                         Timing Runner: fc6 + relu6 (CudaFullyConnected)
TRT1
```



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
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
```