

7、TensorRT USB 摄像头实时图像识别教程

准备工作：

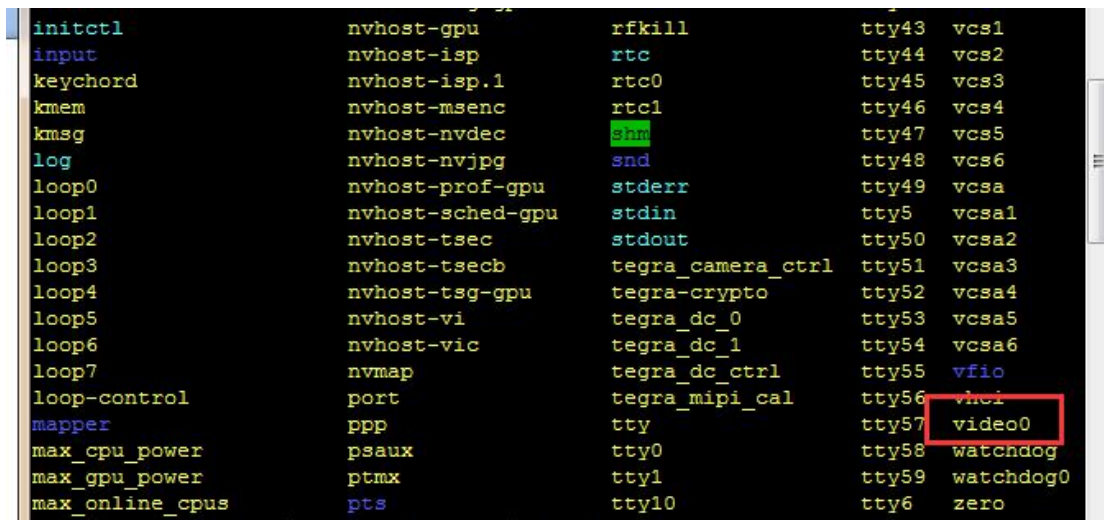
开始之前，确定已经完成教程 4、5 的所有步骤，能测试简单的例子。最好拔掉板载摄像头，插上 USB 摄像头再上电。

1、检测摄像头设备

假如你在 jetson-inference 目录下。

如下图，执行：

ls /dev



```

initctl          nvhost-gpu      rfkill           tty43            vcs1
input            nvhost-isp      rtc              tty44            vcs2
keychord         nvhost-isp.1    rtc0             tty45            vcs3
kmem             nvhost-msenc    rtc1             tty46            vcs4
kmsg             nvhost-nvdec    shm              tty47            vcs5
log              nvhost-nvjpg    snd              tty48            vcs6
loop0            nvhost-prof-gpu stderr           tty49            vcsa
loop1            nvhost-sched-gpu stdin            tty5             vcsa1
loop2            nvhost-tsec     stdout           tty50            vcsa2
loop3            nvhost-tsecb    tegra_camera_ctrl tty51            vcsa3
loop4            nvhost-tsg-gpu tegra-crypto     tty52            vcsa4
loop5            nvhost-vi       tegra_dc_0       tty53            vcsa5
loop6            nvhost-vic      tegra_dc_1       tty54            vcsa6
loop7            nvmap           tegra_dc_ctrl    tty55            vfio
loop-control     port            tegra_mipi_cal   tty56            vchi
mapper           ppp             tty              tty57            video0
max_cpu_power    psaux           tty0             tty58            watchdog
max_gpu_power    ptmx           tty1             tty59            watchdog0
max_online_cpus  pts            tty10            tty6             zero
  
```

确定是否有 video0 这个设备，有可能多个摄像头时，注意后面的编号不同。

2、参数介绍

与前面的 imagenet-console 示例类似，相机应用程序构建在该/aarch64/bin 目录中。它们在带有 OpenGL 渲染的实时摄像机流上运行，并接受 4 个可选的命令行参数：

- --network 标志设置分类模型（默认为 GoogleNet）
- 请参阅下载其他分类模型以获取可用的网络。
- --camera 标志设置要使用的摄像头设备

通过指定传感器索引（0 或 1 等）来使用 MIPI CSI 摄像机

V4L2 USB 摄像机通过指定其/dev/video 节点（/dev/video0，/dev/video1 等）使用。

- 默认为使用 MIPI CSI 传感器 0（--camera=0）
- --width 和--height 标志设置相机分辨率（默认为 1280x720）
- 分辨率应设置为相机支持的格式。
- 使用以下命令查询可用格式：

```
$ sudo apt-get install v4l-utils
```

```
$ v4l2-ctl --list-formats-ext
```

您可以根据需要组合使用这些标志，并且还有其他命令行参数可用于加载自定义模型。启动带有--help 标志的应用程序以获取更多信息，或参阅 Examples 自述文件。

以下是启动程序的一些典型方案:

C++

```
$ ./imagenet-camera # 使用 GoogLeNet, 默认 MIPI CSI 相机 (1280x720)
$ ./imagenet-camera --network=alexnet # 使用 alexnet, 默认 MIPI CSI 相机 (1280x720)
$ ./imagenet-camera --camera=/dev/video1 # 使用 GoogLeNet, V4L2 摄像机/ dev / video1 (1280x720)
$ ./imagenet-camera --width=640 --height=480 # 使用 GoogLeNet, 默认为 MIPI CSI 摄像机 (640x480)
```

Python

```
$ ./imagenet-camera.py # 使用 GoogLeNet, 默认 MIPI CSI 相机 (1280x720)
$ ./imagenet-camera.py --network=alexnet # 使用 RESNET-18, 默认 MIPI CSI 相机 (1280x720)
$ ./detectnet-camera.py --camera=/dev/video1 # 使用 PedNet, V4L2 摄像机/dev/video1 (1280x720)
$ ./imagenet-camera.py --width=640 --height=480 # 使用 GoogLeNet, 默认为 MIPI CSI 相机 (640x480)
```

3、执行图像识别命令

这个时候最好能通过桌面执行, 否则可能看不到摄像头的界面, 或者通过 VNC 远程桌面连接。

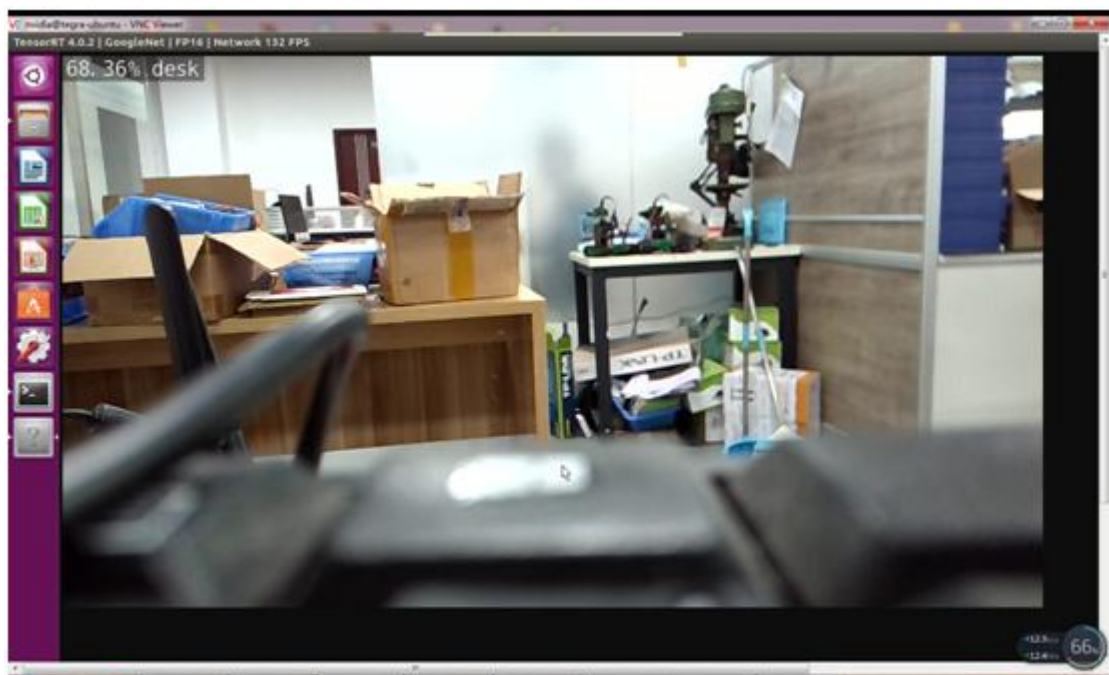
进入到 bin 目录下:

实时图像识别演示位于 / aarch64 / bin 中并被调用 imagenet-camera。它在实时摄像机流上运行, 并根据用户参数, 使用 TensorRT 加载 googlenet 或 alexnet。

```
$ ./imagenet-camera --network=googlenet --camera=/dev/video1 # 使用 googlenet 的 USB 摄像头运行
```

```
$ ./imagenet-camera --network=alexnet --camera=/dev/video1 # 使用 alexnet 运行
```

每秒帧数 (FPS), 来自视频的分类对象名称和分类对象的置信度被打印到 OpenGL 窗口标题栏。默认情况下, 应用程序可以识别多达 1000 种不同类型的对象, 因为 Googlenet 和 Alexnet 是在包含 1000 类对象的 ILSVRC12 ImageNet 数据库上进行培训的。1000 种类型对象的名称映射, 可以在 repo 下找到 data/networks/ilsvrc12_synset_words.txt



这样识别到物体就会在上面显示物体英文名称，百分数就是匹配百分比。