Yolo5+Tensorrt acceleration+DeepStream (turn on camera)

1.Precautions before use

If you are using the YAHBOOM version of the image, there is no need to build the DeepStream environment. If you have built your own image, you need to build the environment for DeepStream. You can refer to the DeepStream building tutorial we provide, or you can also build your own Baidu

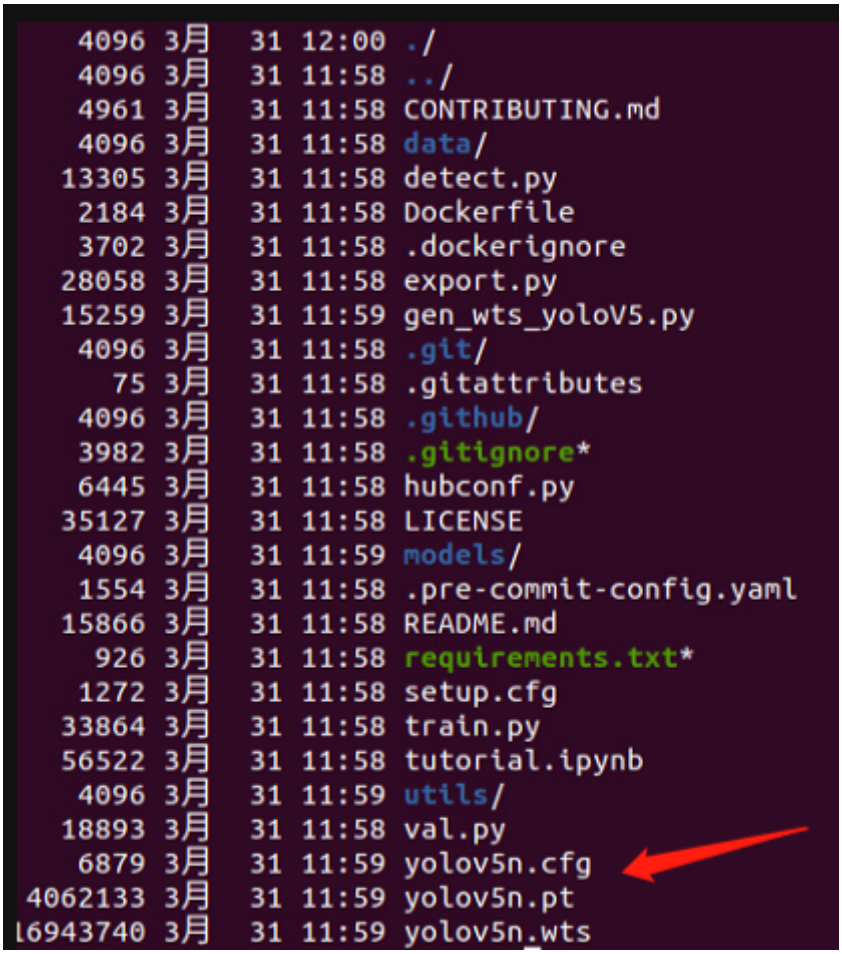
2.Instructions for use

2.1 Model Conversion

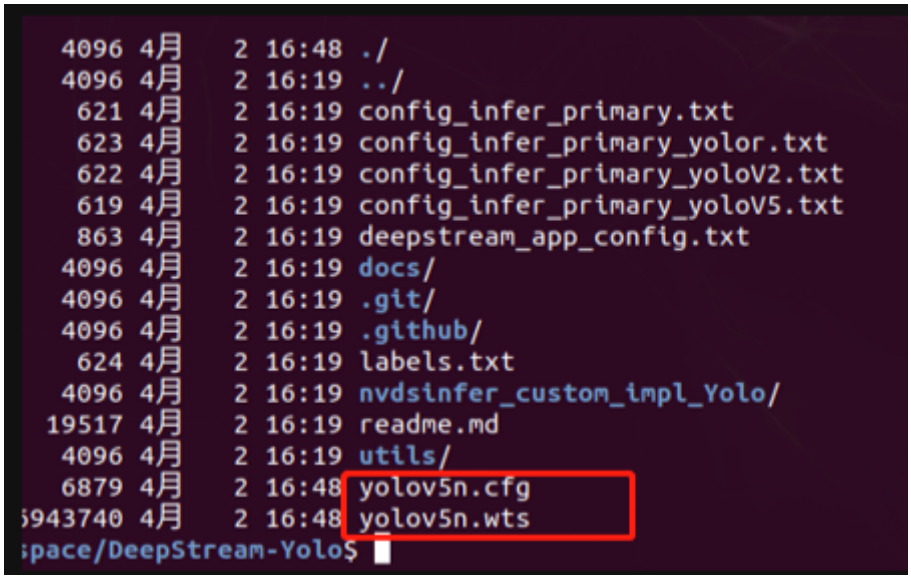
```
git clone https://github.com/marcoslucianops/DeepStream-Yolo.git  #
cd DeepStream-Yolo/utils
cp gen_wts_yoloV5.py ../../yolov5
cd ../../yolov5
python3 gen_wts_yoloV5.py -w  ./yolov5s.pt
```

2.2 Deployment Model

1. After successfully running the previous step, two files will appear in the directory of yolov5, yolov5n.cfg and yolov5n.wts



2. Place yolov5n. cfg and yolov5n. wts in the DeepStream Yolo folder of Jetson nano



2.3 Modifying the Deepstream configuration file (this step can be omitted for images in the YAHBOOM version)

1. Modify Deepstream_ app_ Config.txt fileThe modified content is as follows:Comment on 70 lines, add a line after:config-file=config_ infer_ primary_ yoloV5.txtAs shown in the figure:

```
[primary-gie]
enable=1
gpu-id=0
gie-unique-id=1
nvbuf-memory-type=0
#config-file=config_infer_primary.txt
config-file=config_infer_primary_yoloV5.txt

```

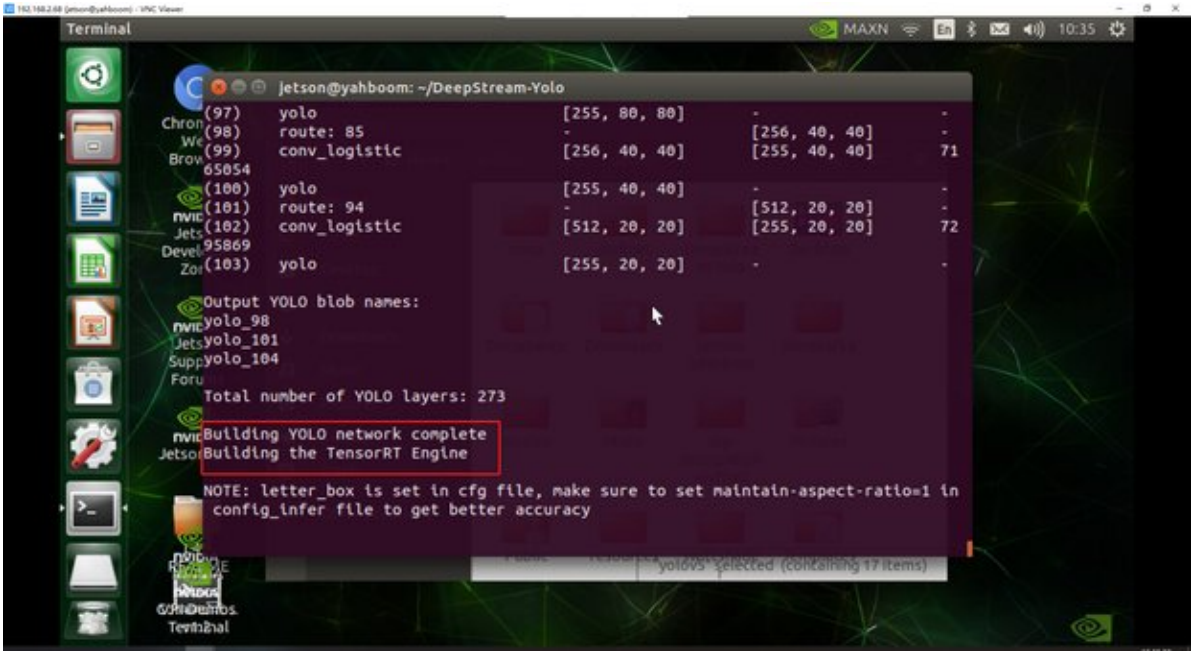
2. Modify the second configuration file config_ infer_ primary_ yoloV5.txt

```
[property]
# 省略 ...
**model-engine-file=model_b2_gpu0_fp16.engine** # 修改 fp32->fp16
batch-size=2 # batch-size 改为2，速度会快一些
# 省略...
**network-mode=2 #** 修改为 2, 强制使用fp16推理
```

Note: FPS is related to parameters such as input image size, batch size, interval, etc., and needs to be optimized according to practical applications. Here, we directly change the batch size of the input to 2, which will significantly improve the inference speed of the model

3.Compile and Run

```
cd nvdsinfer_custom_impl_Yolo/
CUDA_VER=10.2 make -j2 #根据你的CUDA版本修改10.2数字部分
cd ..
deepstream-app -c deepstream_app_config.txt
```



After waiting for a while, you can see that the CSI camera screen has opened

pay attention to

1. If you are using your own built image, you need todeepstream_ app_ Modify config.txtModify as shown in the figure:

```
#[source0]
#enable=1
#type=3
#uri=file:///opt/nvidia/deepstream/deepstream/samples/streams/sample_1080p_h264.mp4
#num-sources=1
#gpu-id=0
#cudadec-memtype=0

[source1]
enable=1
#Type - 1=CameraV4L2 2=URI 3=MultiURI 4=RTSP 5=CSI
type=5
camera-csi-sensor-id=0
camera-width=1280
camera-height=720
camera-fps-n=30
camera-fps-d=1
```

2. If using a USB camera, you need to change type=5 to type=1

```
#[source0]
#enable=1
#type=3
#uri=file:///opt/nvidia/deepstream/deepstream/samples/streams/sample_1080p_h264.mp4
#num-sources=1
#gpu-id=0
#cudadec-memtype=0

[source1]
enable=1
#Type - 1=CameraV4L2 2=URI 3=MultiURI 4=RTSP 5=CSI
type=5 ———— type=1
camera-csi-sensor-id=0
camera-width=1280
camera-height=720
camera-fps-n=30
camera-fps-d=1
```