

Yolo5+Tensorrt acceleration+DeepStream

Yolo5+Tensorrt acceleration+DeepStream

- 1.Precautions before use
- 2.instructions
 - 2.1 Model Transformation
 - 2.2 Deployment Model
 - 2.3 Modify the deepstream configuration file (this step can be omitted for YAHBOOM version images)
- 3.Compile Run

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

2.1 Model Transformation

```
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

```

4096 3月 31 12:00 ./
4096 3月 31 11:58 ../
4961 3月 31 11:58 CONTRIBUTING.md
4096 3月 31 11:58 data/
13305 3月 31 11:58 detect.py
2184 3月 31 11:58 Dockerfile
3702 3月 31 11:58 .dockerignore
28058 3月 31 11:58 export.py
15259 3月 31 11:59 gen_wts_yoloV5.py
4096 3月 31 11:58 .git/
75 3月 31 11:58 .gitattributes
4096 3月 31 11:58 .github/
3982 3月 31 11:58 .gitignore*
6445 3月 31 11:58 hubconf.py
35127 3月 31 11:58 LICENSE
4096 3月 31 11:59 models/
1554 3月 31 11:58 .pre-commit-config.yaml
15866 3月 31 11:58 README.md
926 3月 31 11:58 requirements.txt*
1272 3月 31 11:58 setup.cfg
33864 3月 31 11:58 train.py
56522 3月 31 11:58 tutorial.ipynb
4096 3月 31 11:59 utils/
18893 3月 31 11:58 val.py
6879 3月 31 11:59 yolov5n.cfg
4062133 3月 31 11:59 yolov5n.pt
16943740 3月 31 11:59 yolov5n.wts

```



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

```

4096 4月 2 16:48 ./
4096 4月 2 16:19 ../
621 4月 2 16:19 config_infer_primary.txt
623 4月 2 16:19 config_infer_primary_yolor.txt
622 4月 2 16:19 config_infer_primary_yoloV2.txt
619 4月 2 16:19 config_infer_primary_yoloV5.txt
863 4月 2 16:19 deepstream_app_config.txt
4096 4月 2 16:19 docs/
4096 4月 2 16:19 .git/
4096 4月 2 16:19 .github/
624 4月 2 16:19 labels.txt
4096 4月 2 16:19 nvdsinfer_custom_impl_Yolo/
19517 4月 2 16:19 readme.md
4096 4月 2 16:19 utils/
6879 4月 2 16:48 yolov5n.cfg
5943740 4月 2 16:48 yolov5n.wts
space/DeepStream-Yolo$

```

2.3 Modify the deepstream configuration file (this step can be omitted for YAHBOOM version images)

1. Modify Deepstream_app_Config.txt file
The modified content is as follows:
Comment on 70 lines, add a line after:
config-file=config_infer_primary_yoloV5.txt

As shown in the figure:

```
65 [primary-gie]
66 enable=1
67 gpu-id=0
68 gie-unique-id=1
69 nvbuf-memory-type=0
70 #config-file=config_infer_primary.txt
71 config-file=config_infer_primary_yoloV5.txt
72
```

2. Modify the second configuration file config_infer_primary_yoloV5.txt

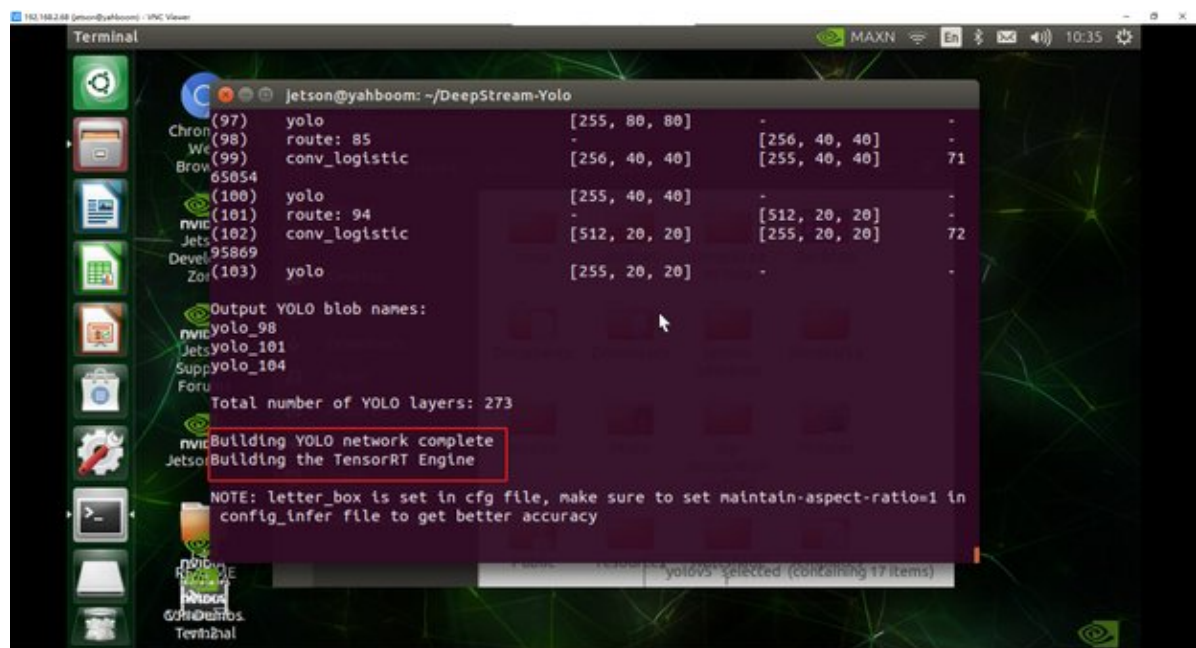
```
[property]
# omit ...
**model-engine-file=model_b2_gpu0_fp16.engine** # fp32->fp16
batch-size=2 # batch-size Change to 2, the speed will be faster
# omit...
**network-mode=2 *** 2:Force the use of fp16 inference
# omit ...
```



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 Run

```
cd nvdsinfer_custom_impl_Yolo/
CUDA_VER=11.4 make -j4 #Modify the numerical part of 11.4 based on your CUDA
version
cd ..
deepstream-app -c deepstream_app_config.txt
```



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

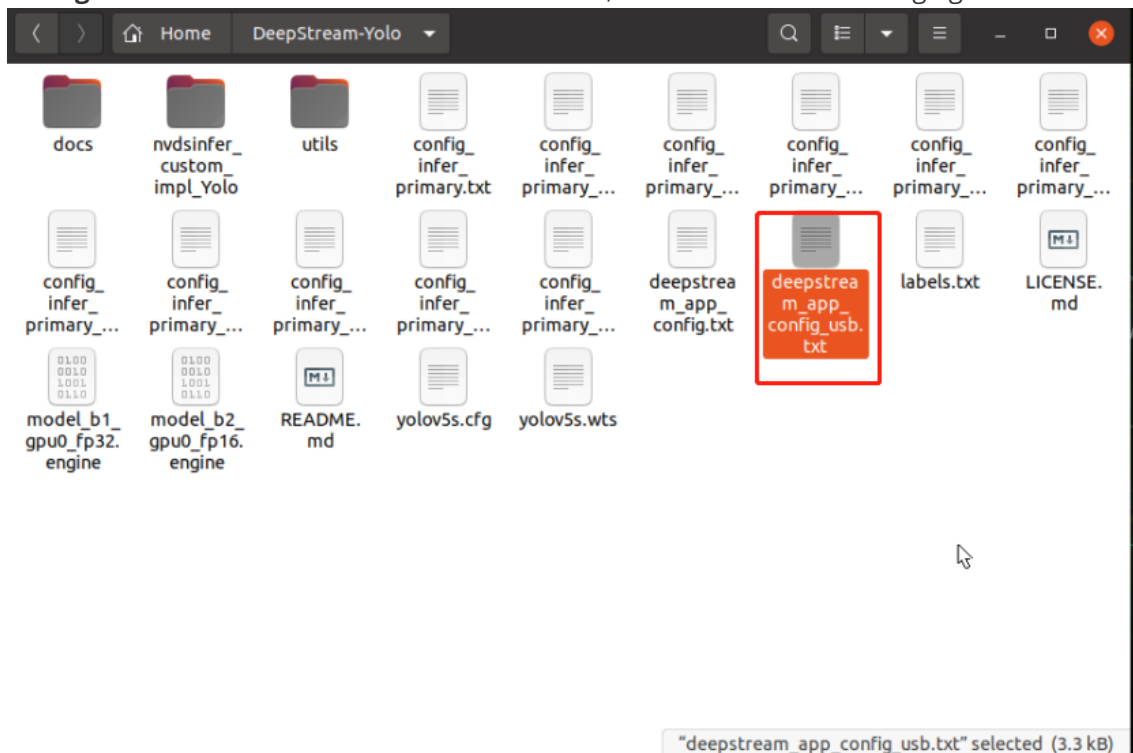
node

1. If you are using your own built image, you need to modify `deepstream_app_config.txt`. Make modifications 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 you are using a USB camera, you need to include the **deepstream** in the attachment of the document `_app_config_Upload the file USB. txt` to Jetson and `deepstream_app_Config.txt`. This file is located in the same location, as shown in the following figure:



then run

```
cd ~/DeepStream-Yolo
deepstream-app -c deepstream_app_config_usb.txt
```



Just a moment, we can achieve the detection

deepstream The parameter description for file configuration and related reference connections are as follows:

https://blog.csdn.net/weixin_38369492/article/details/104859567

