

deepstream集成yolov4

GUPAO TECH

我们的愿景

推动每一次人才升级

我们的使命

让每个人的职业生涯不留遗憾

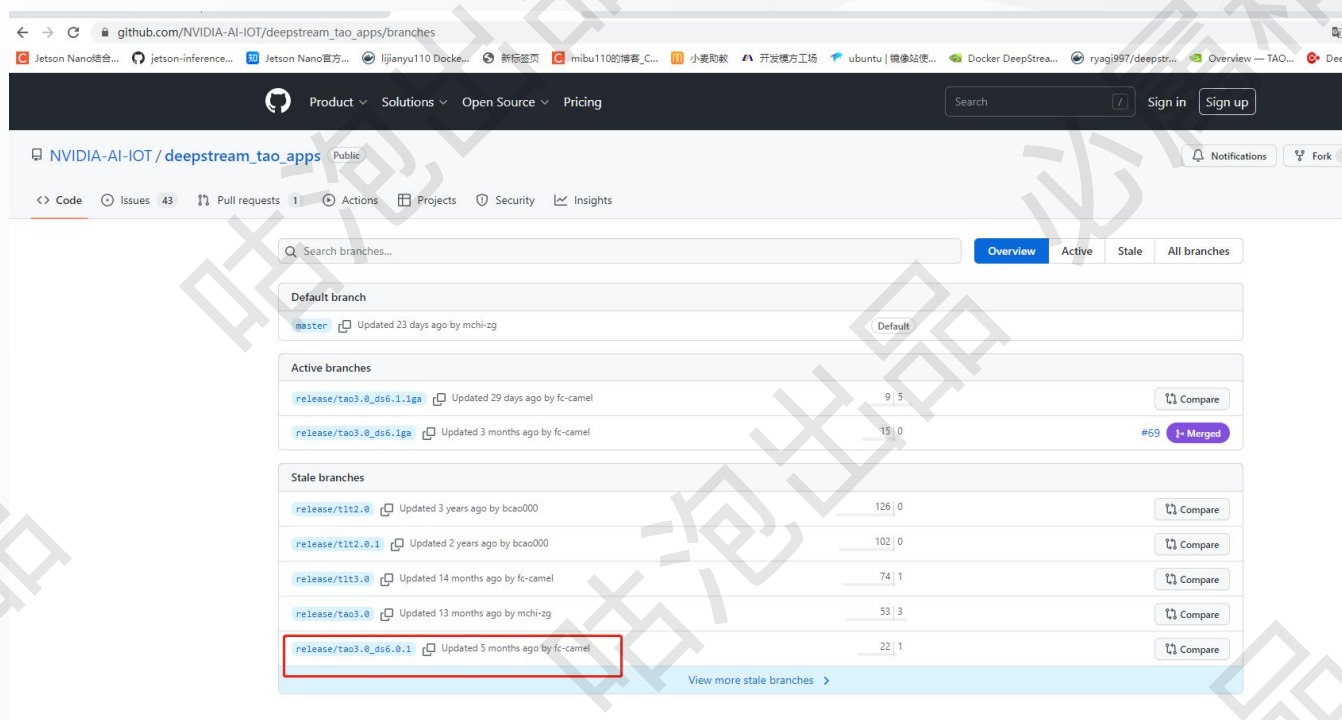
请在此处
输入文字

请在此处
插入二维码

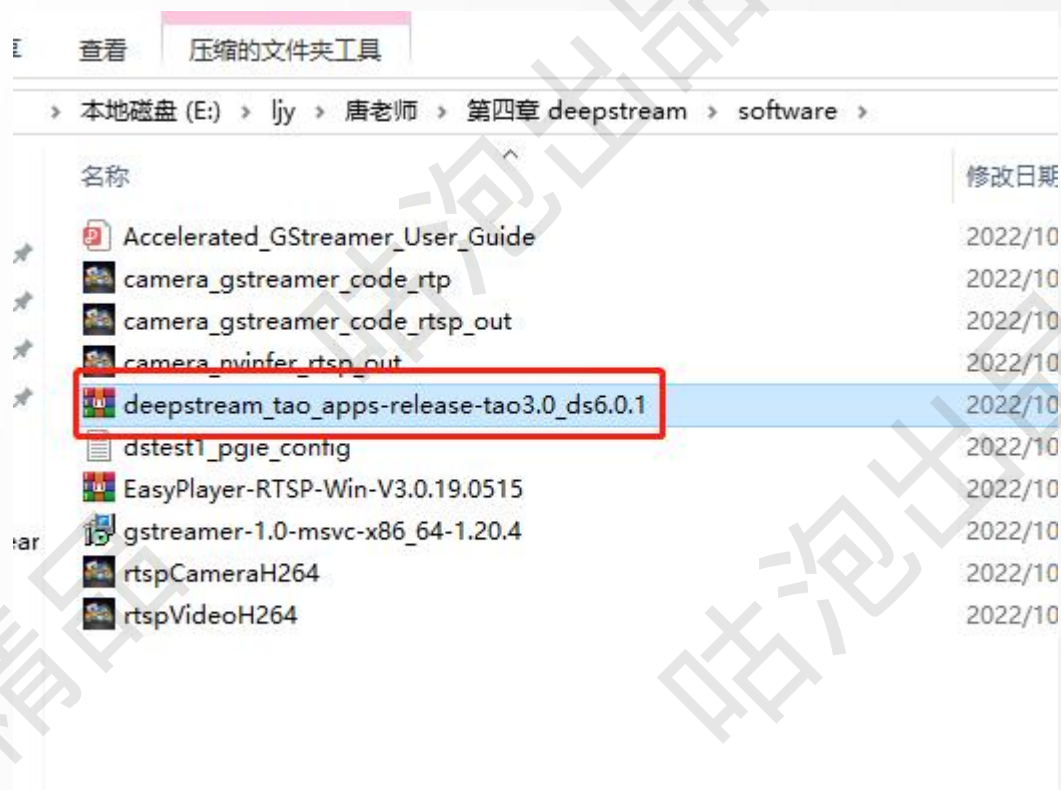
deepstream和tao训练模型集成

https://github.com/NVIDIA-AI-IOT/deepstream_tao_apps

下载6.01版本



下载6.01版本



安装集成步骤

- 第一、替换libnvinfer_plugin.so文件
- 第二、把tao导出的模型和配置文件下载到jetson nano
- 第三、修改配置 和 编译配置依赖的so文件
- 第四、上传文件运行

设置CUDA

```
export CUDA_HOME=/usr/local/cuda  
export PATH=/usr/local/cuda/bin:$PATH  
export LD_LIBRARY_PATH=/usr/local/cuda/lib64/:$LD_LIBRARY_PATH
```

查看系统几个核心组件版本

```
sudo apt-cache show nvidia-jetpack
```

```
nvcc -V
```

```
dpkg -l | grep TensorRT
```

安装步骤

一、替换 libnvinfer_plugin.so.8.2.1（在宿主机上操作）

```
sudo mv /usr/lib/aarch64-linux-gnu/libnvinfer_plugin.so.8.x.y ${HOME}
```

```
sudo cp $TRT_SOURCE/build/libnvinfer_plugin.so.8.m.n /usr/lib/aarch64-linux-gnu/
```

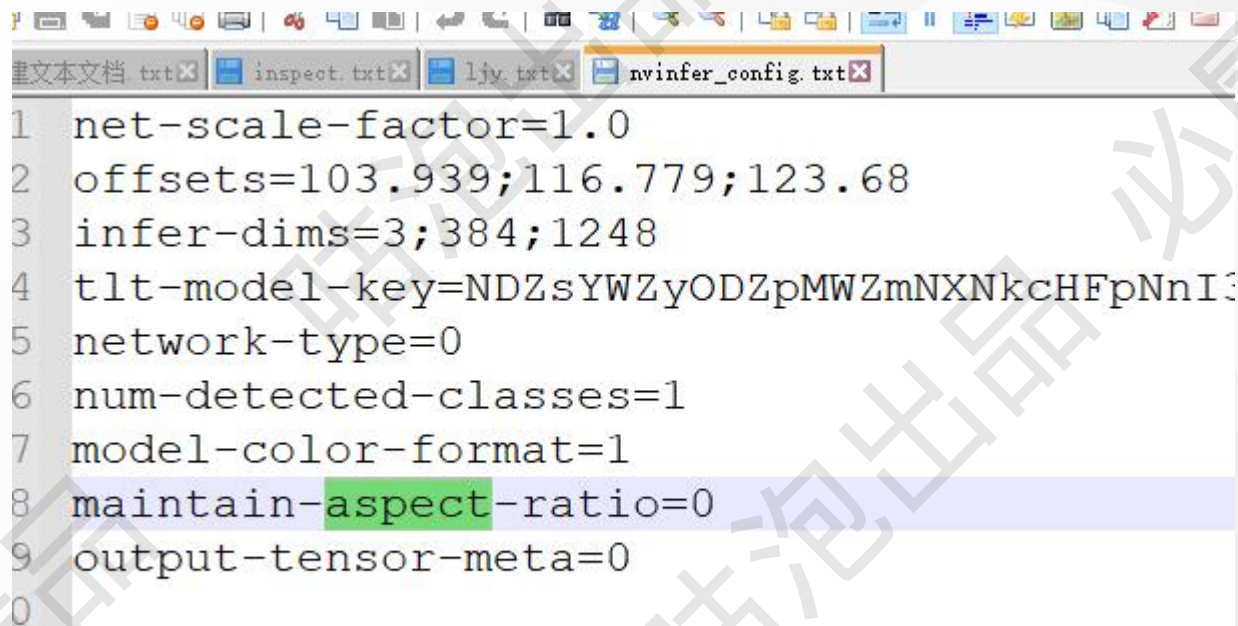
```
sudo ldconfig
```


第二步：拉取TAO训练的模型和配置文件

放到目录/GStreamer/



TAO生存的配置说明



```
1 net-scale-factor=1.0
2 offsets=103.939;116.779;123.68
3 infer-dims=3;384;1248
4 tlt-model-key=NDZsYWZyODZpMWZmNXNkcHFpNnI3
5 network-type=0
6 num-detected-classes=1
7 model-color-format=1
8 maintain-aspect-ratio=0
9 output-tensor-meta=0
10
```

yolov4 配置说明

tao生成的nvinfer_config.txt内容
替换到

pgie_yolov4_tiny_tao_config.txt 文件中

```
pgie_yolov4_tiny_tao_config.txt
1 [property]
2 gpu-id=0
3 net-scale-factor=1.0
4 offsets=103.939;116.779;123.68
5 model-color-format=1
6 labelfile-path=./export/labels.txt
7 model-engine-file=./export/trt.engine
8 #int8-calib-file=../../models/yolov4-tiny/cal.bin
9 tlt-encoded-model=./export/yolov4_cspdarknet_tiny_epoch_080.
10 tlt-model-key=NDZsYWZyODZpMWZmNXNkcHFpNnI3djBnNjk6M2ZkMWJlNG
11 infer-dims=3;384;1248
12 maintain-aspect-ratio=0
13 uff-input-order=0
14 uff-input-blob-name=Input
15 batch-size=1
16 ## 0=FP32, 1=INT8, 2=FP16 mode
17 network-mode=1
18 num-detected-classes=1
19 interval=0
20 gie-unique-id=1
21 is-classifier=0
22 #network-type=0
23 cluster-mode=3
24 output-blob-names=BatchedNMS
25 parse-bbox-func-name=NvDsInferParseCustomBatchedNMSTLT
```

进入docker

```
docker run -it --rm --net=host --runtime nvidia -e  
DISPLAY=$DISPLAY -v /tmp/argus_socket:/tmp/argus_socket --  
device /dev/video0 -v  
~/GStreamer:/opt/nvidia/deepstream/deepstream-  
6.0/sources/deepstream_python_apps/apps/GStreamer  
715a93f1d755
```

nvdsinfer_custombbboxparser_tao.cpp编译

```
export CUDA_VER=10.2
```

```
make
```

运行自己的模型在deepstream中

```
rm -rf ~/.cache/gstreamer-1.0/
```

```
cd /opt/nvidia/deepstream/deepstream-6.0/sources/deepstream_pytho  
pps/GStreamer
```

```
python3 camera_nvinfer_myEngin_rtsp_out.py
```


查看新生存的 Engine 文件

```
zlx@nano128G:~/GStreamer/export$ ls
labels.txt
libnvds_infercustomparser_tao.so
nvinfer_config.txt
yolov4_cspdarknet_tiny_epoch_080.etlt
yolov4_cspdarknet_tiny_epoch_080.etlt_b1_gpu0_fp16.engine
zlx@nano128G:~/GStreamer/export$
```

配置新生存的 Engine

```
model-engine-  
file=export/yolov4_cspdarknet_tiny_epoch_080.etlt_b1_gpu0_f  
p16.engine
```

这样后面不用在生存engine文件，启动就快了

谢谢观赏

GUPAO TECH



替换小标题文字，或简要说明

我们的愿景

推动每一次人才升级

我们的使命

让每个人的职业生涯不留遗憾

请在此处
输入文字

请在此处
插入二维码