

ROS TensorFlow 功能使用与讲解

1. 功能简介

在 ROS 中使用主控与 TensorFlow 来进行机器学习和深度学习任务，TensorFlow 是一个流行的深度学习框架，用于构建、训练和部署深度神经网络模型。可以利用 TensorFlow 进行视觉感知、语音识别、自然语言处理等机器学习任务，从而实现物体识别、目标检测、手写数字识别等功能。物体识别功能主要是对实时图像进行识别，并在终端打印所识别到的物品种类以及对应的置信度。目标检测功能同样也是对实时图像进行识别，其与物体识别功能不同之处在于它还能将识别到的物体进行框选，并在图像上显示类别与置信度。手写数字识别功能主要是使用了机器学习中经典的 MNIST 数据集。

2. 使用方法

① 配置开启虚拟环境

在进行 ROS TensorFlow 功能的使用前，如果使用的 ROS 主控是 Jetson 系列（如 Nano、TX1、NX），我们需要先进入已配置好的虚拟环境中，虚拟环境的作用主要是为了不与 ROS 运行环境产生冲突。需要注意的是虚拟环境中 python 版本为 3.6 或以上，与 ROS 使用的 python2.7 不一致，这里针对 ros_tensorflow 功能包做了相应环境变量配置，建议不要在虚拟环境中运行其他 ROS 功能。

首先在 .bashrc 文件中配置开启虚拟环境，将用于整段注释的 EOF 语句注释掉。在终端运行 `gedit .bashrc`，修改为图中所示。注意，当正常运行其他不涉及虚拟环境的 ROS 功能时，需将此段对 conda 环境的配置整段注释掉，修改方法为去掉两个 EOF 语句前的“#”。

修改 bashrc 后需要 source 一下才能生效，运行 `source .bashrc` 进行环境变量配置更新。注意如果 source 后仍在进入虚拟空间时出现报错：conda command not found，可以重新开启一个终端运行。

```
#: <<EOF
source /home/wheeltec/catkin_workspace/install/setup.bash --extend
# added by Archiconda3 0.2.3 installer
# >>> conda init >>>
# !! Contents within this block are managed by 'conda init' !!
__conda_setup="$(CONDA_REPORT_ERRORS=false '/home/wheeltec/archiconda3/bin/conda' shell.bash
hook 2> /dev/null)"
if [ $? -eq 0 ]; then
    \eval "$__conda_setup"
else
    if [ -f "/home/wheeltec/archiconda3/etc/profile.d/conda.sh" ]; then
        . "/home/wheeltec/archiconda3/etc/profile.d/conda.sh"
        CONDA_CHANGEPS1=false conda activate base
    else
        \export PATH="/home/wheeltec/archiconda3/bin:$PATH"
    fi
fi
unset __conda_setup
# <<< conda init <<<
#: EOF
```

修改.bashrc 文件环境变量配置

```
wheeltec@wheeltec:~$ source .bashrc
```

环境变量配置更新

然后在终端运行 `conda activate wheeltec` 进入虚拟环境, 进入后终端用户名前会显示(wheeltec)字样, 表示已进入名为 wheeltec 的虚拟环境, 如需退出虚拟环境, 可在终端运行 `conda deactivate`, 此时终端用户名前会显示(base)字样, 表示已退出虚拟环境。

进入虚拟环境:

```
conda activate wheeltec
```

```
wheeltec@wheeltec:~$ conda activate wheeltec
(wheeltec) wheeltec@wheeltec:~$
```

进入 wheeltec 虚拟环境

② ROS TensorFlow 物体识别功能

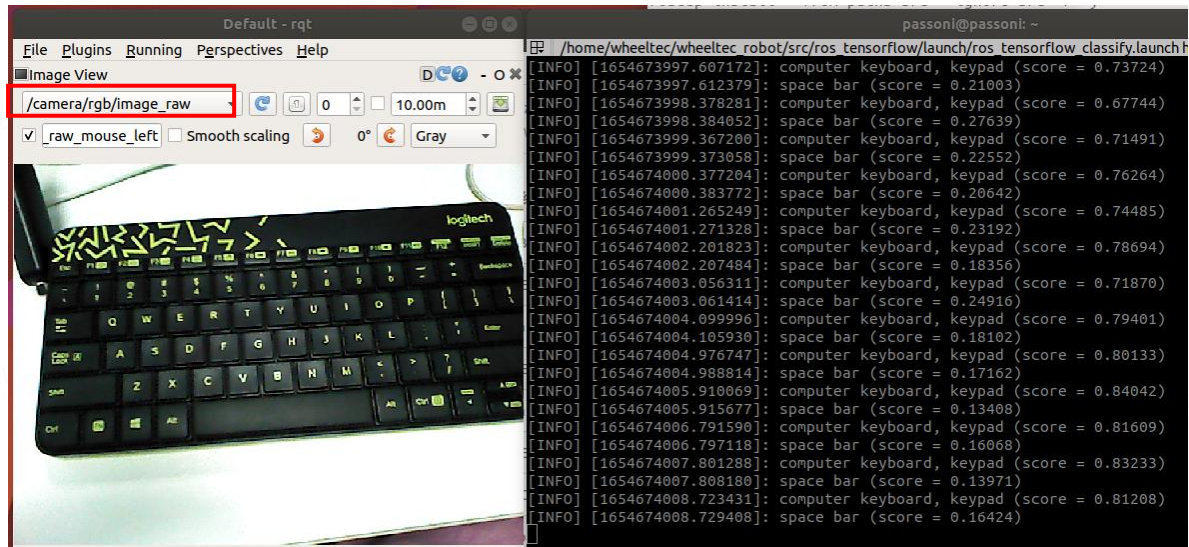
1) 首先运行 launch 文件启动物体识别功能:

```
roslaunch ros_tensorflow ros_tensorflow_classify.launch
```

2) 打开 rqt 工具查看实时图像:

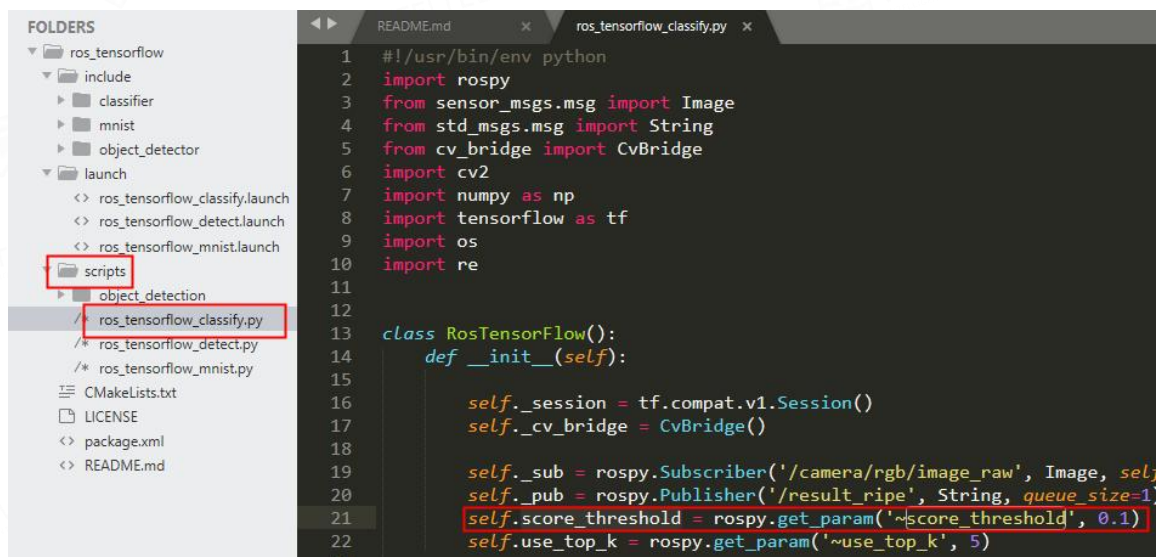
```
rqt_image_view
```

实时图像可选择/camera/rgb/image_raw/compressed 话题进行查看。



实时图像与物体识别结果输出

如图所示，终端将会打印所识别到的物体的类别以及置信度，由于该功能直接读取摄像头画面并进行处理，会有一定的延迟，具体延迟情况视 ROS 主控性能而定。从终端输出我们可以看到它会将识别到的所有物体分类别逐行打印，且源码中设定了对某类物体的识别置信度大于 0.1 就会在终端进行打印。



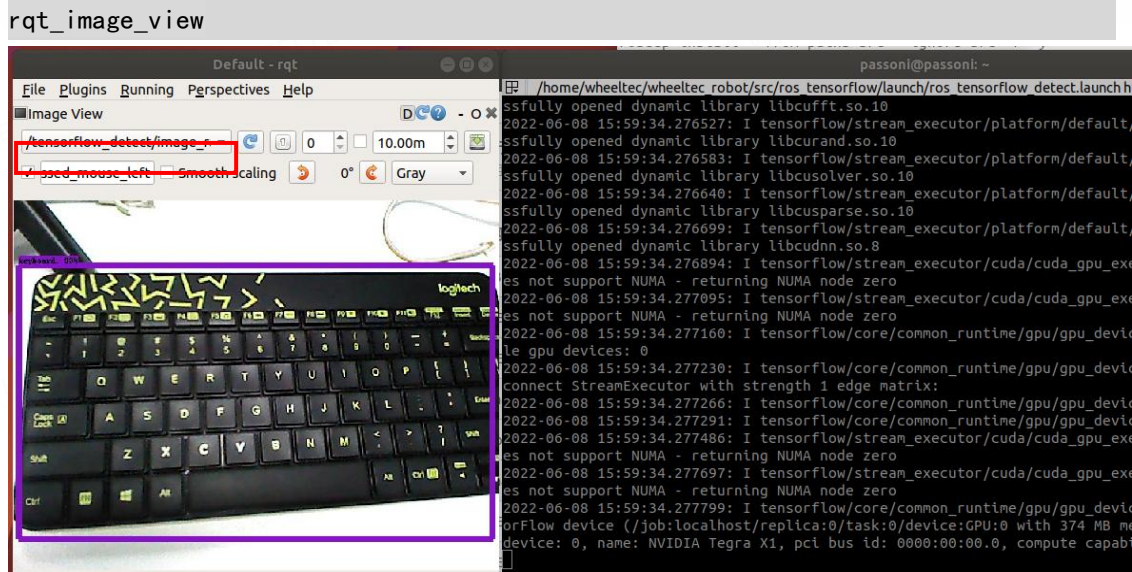
置信度设置

③ ROS TensorFlow 目标检测功能

1) 首先运行 launch 文件启动目标检测功能:

```
roslaunch ros_tensorflow ros_tensorflow_detect.launch
```

2) 打开 rqt 工具，选择/tensorflow_detect/image_raw 话题查看目标检测结果图像输出。



目标检测结果输出

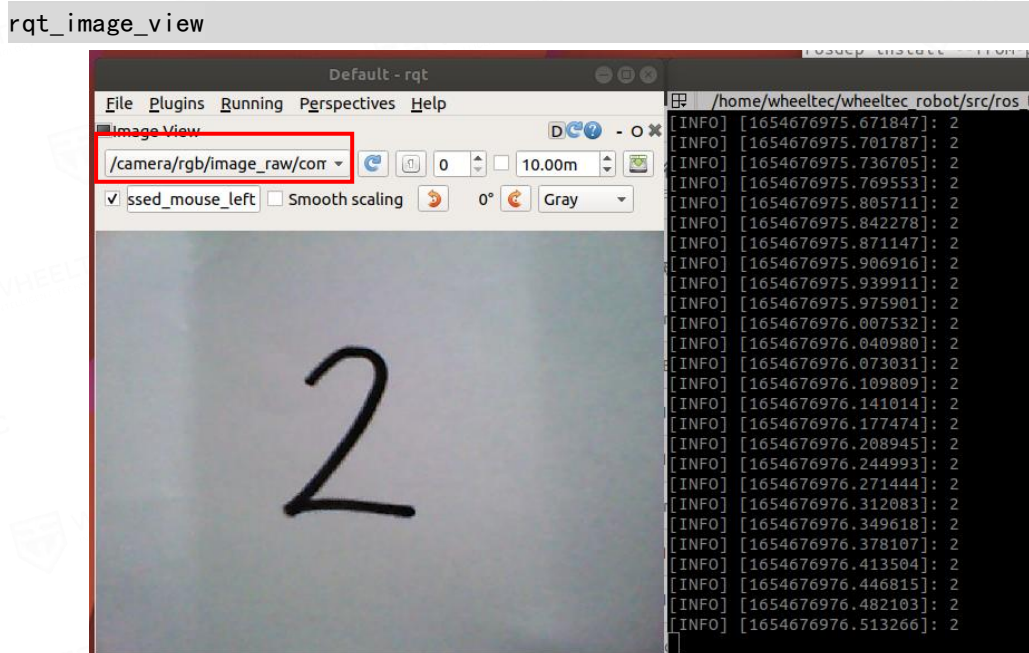
识别结果以图像形式输出，终端没有进行打印，可以看到被识别到的物体已被框选（如图所示的紫色框）并显示识别到的类别及置信度，这里同样需要提醒的是，由于是实时图像直接处理与输出，尽管图像已经进行了压缩处理仍存在一定的延迟，具体延迟情况视 ROS 主控性能而定。

④ ROS TensorFlow 手写数字识别功能

1) 首先运行 launch 文件启动手写数字识别功能：

```
roslaunch ros_tensorflow ros_tensorflow_mnist.launch
```

2) 打开 rqt 工具，选择 /camera/rgb/image_raw/compressed 话题查看实时图像



手写数字识别结果输出

如图所示，成功运行后，将相机朝着所手写的数字，终端将不断的打印所识别到的可能的数值，识别数字范围为 0-9。该功能直接使用官方数据集，识别准确度由数据集决定。若识别到多个数字，终端将会轮流打印识别结果。

3. 注意事项

① 物体识别功能模型路径

物体识别功能所使用模型路径如图所示，

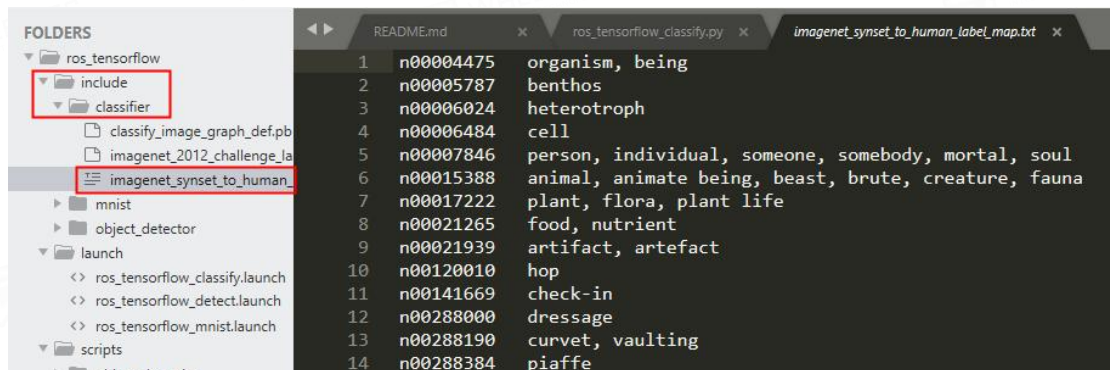


```

88 if __name__ == '__main__':
89
90     ROOT_PATH = os.path.abspath(os.path.join(os.path.dirname(__file__), os.pardir))
91     PATH_TO_CKPT = ROOT_PATH + '/include/classifier/classify_image_graph_def.pb'
92     PATH_TO_LABELS = ROOT_PATH + '/include/classifier/imagenet_2012_challenge_label_map_proto.pbtxt'
93     PATH_TO_UID = ROOT_PATH + '/include/classifier/imagenet_synset_to_human_label_map.txt'
94
  
```

物体识别所用模型的路径

其可识别物体种类文档路径位于：



FOLDERS

- ros_tensorflow
 - include
 - classifier
 - classify_image_graph_def.pb
 - imagenet_2012_challenge_label_map_proto.pbtxt
 - imagenet_synset_to_human_label_map.txt
 - mnist
 - object_detector
 - launch
 - ros_tensorflow_classify.launch
 - ros_tensorflow_detect.launch
 - ros_tensorflow_mnist.launch
 - scripts

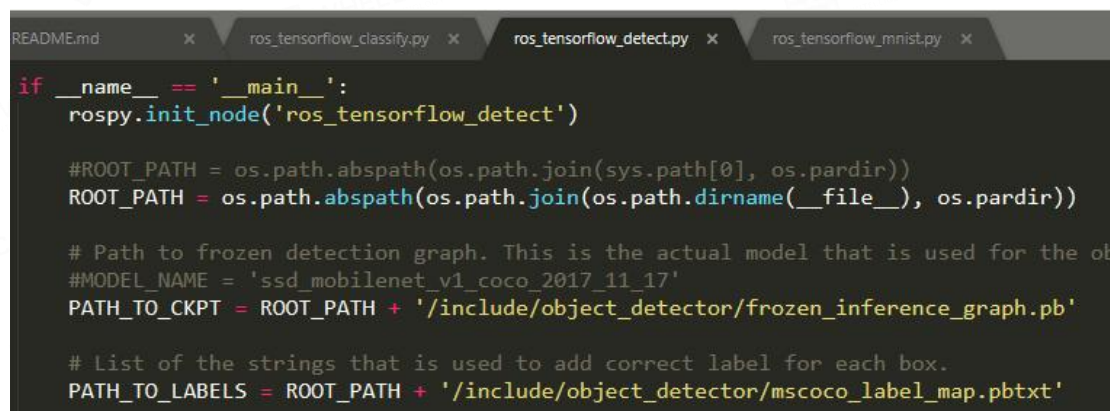
Object Categories (from imagenet_synset_to_human_label_map.txt):

1	n00004475	organism, being
2	n00005787	benthos
3	n00006024	heterotroph
4	n00006484	cell
5	n00007846	person, individual, someone, somebody, mortal, soul
6	n00015388	animal, animate being, beast, brute, creature, fauna
7	n00017222	plant, flora, plant life
8	n00021265	food, nutrient
9	n00021939	artifact, artefact
10	n00120010	hop
11	n00141669	check-in
12	n00288000	dressage
13	n00288190	curvet, vaulting
14	n00288384	piaffe

物体识别可识别物体种类

② 目标检测功能模型路径

目标检测所使用模型路径如图所示，



```

if __name__ == '__main__':
    rospy.init_node('ros_tensorflow_detect')

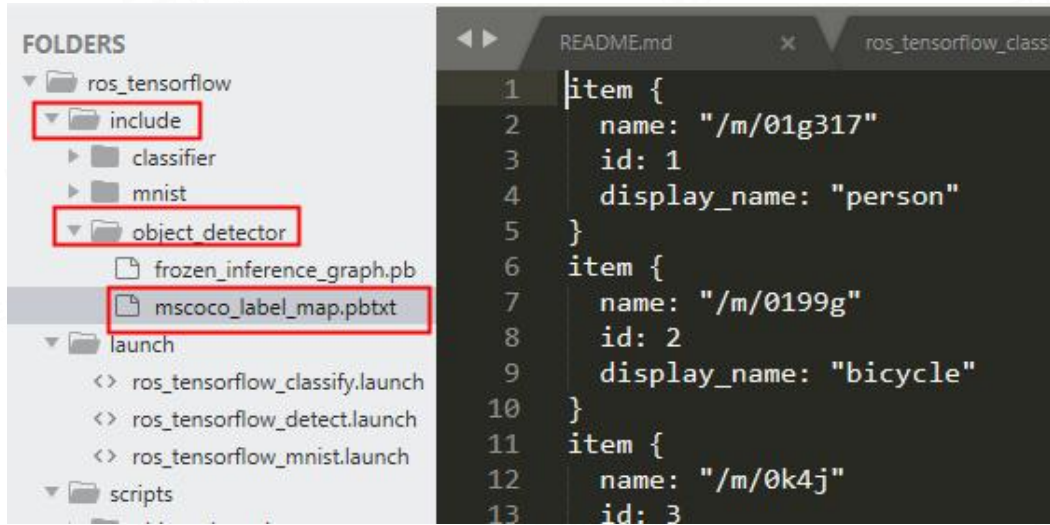
    #ROOT_PATH = os.path.abspath(os.path.join(sys.path[0], os.pardir))
    ROOT_PATH = os.path.abspath(os.path.join(os.path.dirname(__file__), os.pardir))

    # Path to frozen detection graph. This is the actual model that is used for the ob
    #MODEL_NAME = 'ssd_mobilenet_v1_coco_2017_11_17'
    PATH_TO_CKPT = ROOT_PATH + '/include/object_detector/frozen_inference_graph.pb'

    # List of the strings that is used to add correct label for each box.
    PATH_TO_LABELS = ROOT_PATH + '/include/object_detector/mscoco_label_map.pbtxt'
  
```

目标检测所用模型路径

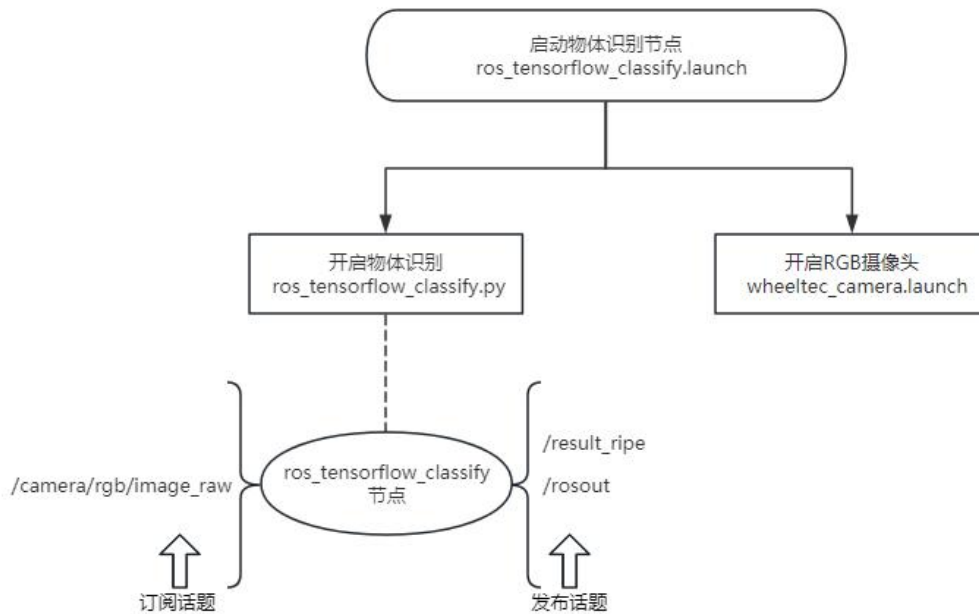
其可检测目标种类文档路径位于：



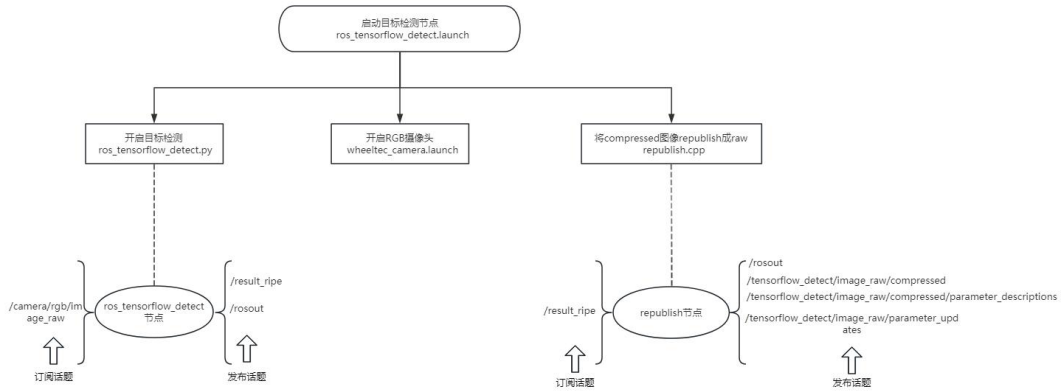
目标检测可识别物体种类

4. ROS TensorFlow 功能程序讲解

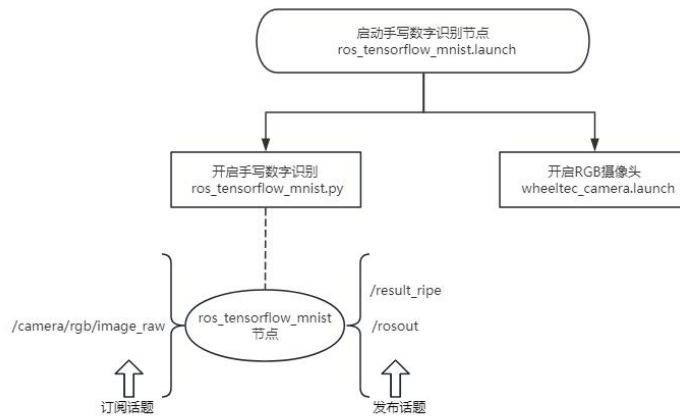
① ROS TensorFlow 各个功能启动框架：



ROS TensorFlow 物体识别功能启动框架图



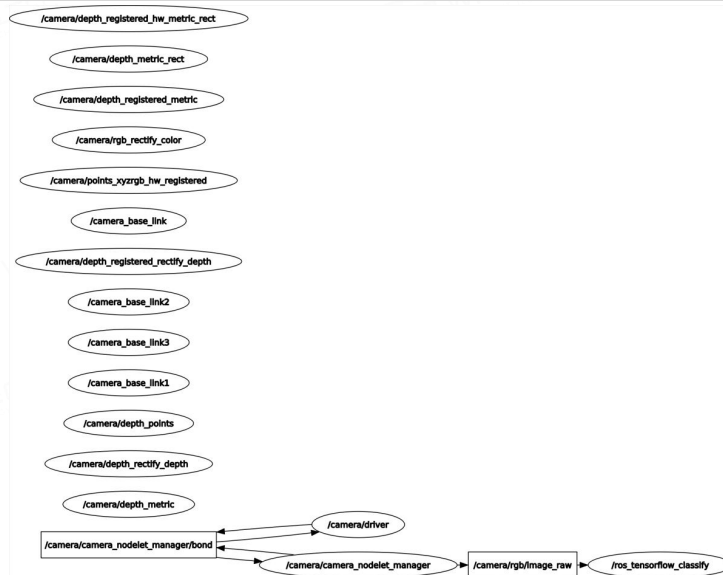
ROS TensorFlow 目标检测功能启动框架图



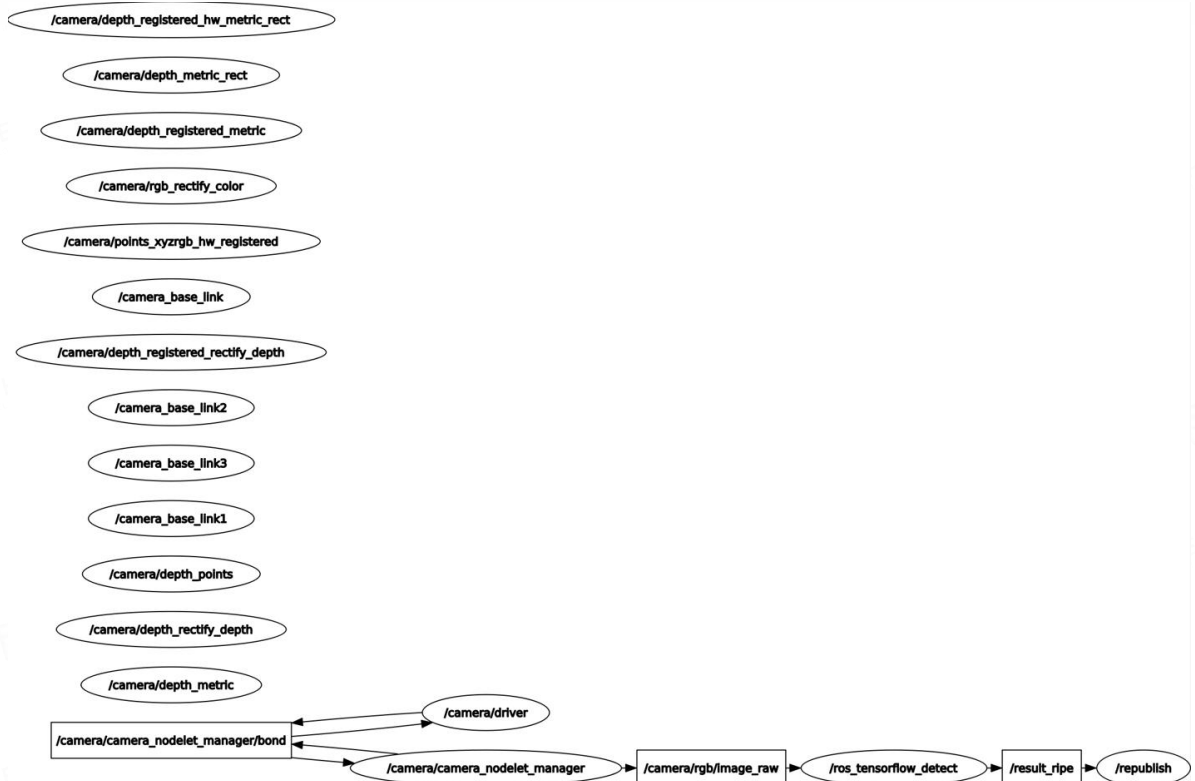
ROS TensorFlow 手写数字识别功能启动框架图

② 查看 ROS TensorFlow 功能节点:

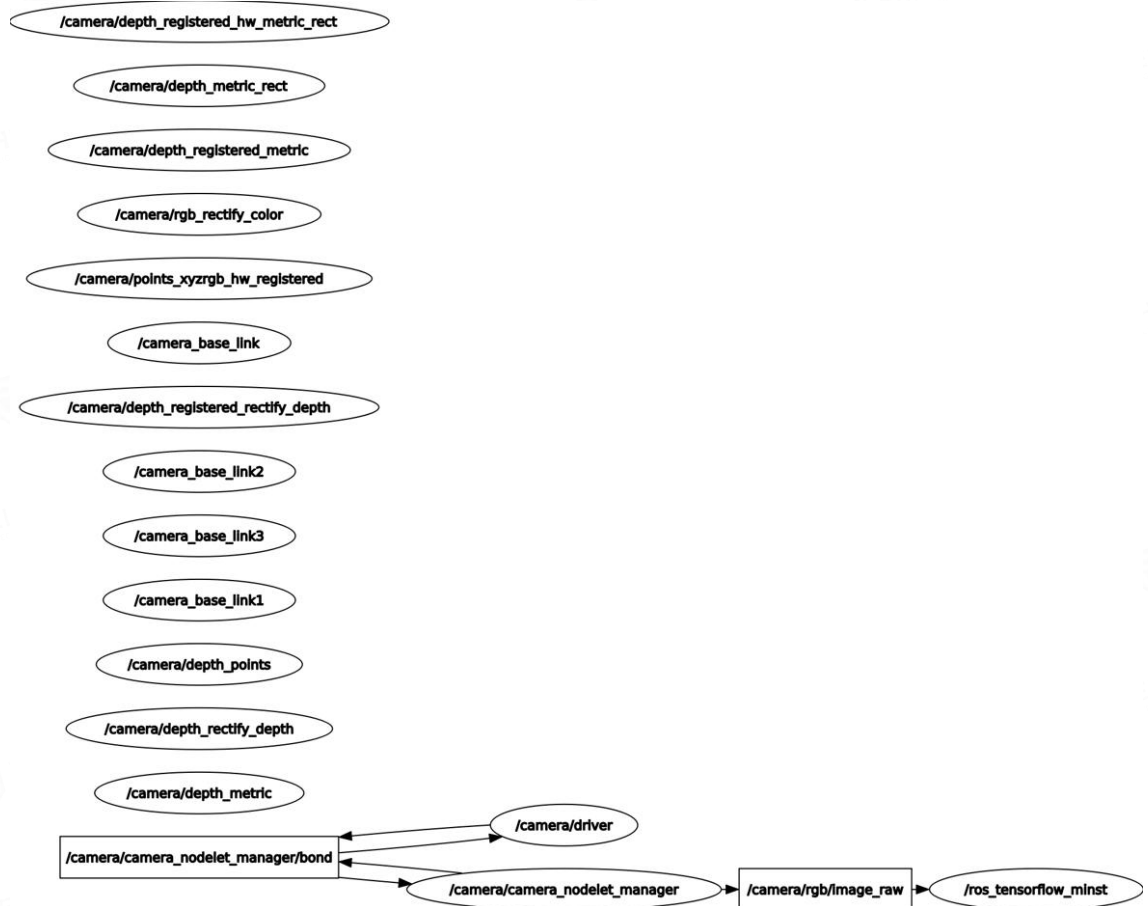
rqt_graph



ROS TensorFlow 物体识别功能节点关系图



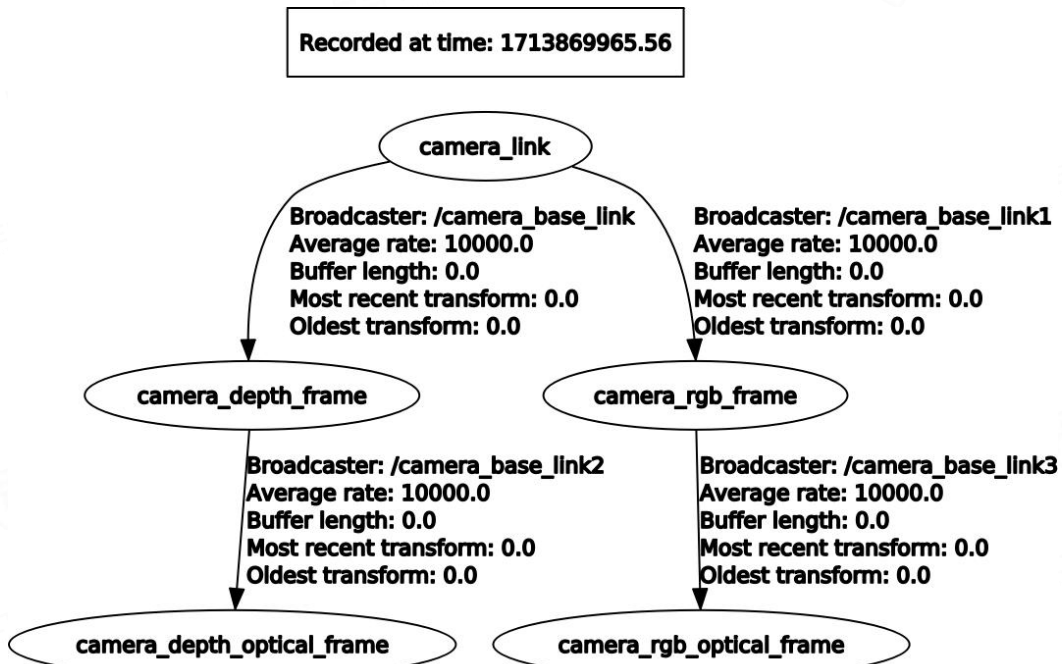
ROS TensorFlow 目标检测功能节点关系图



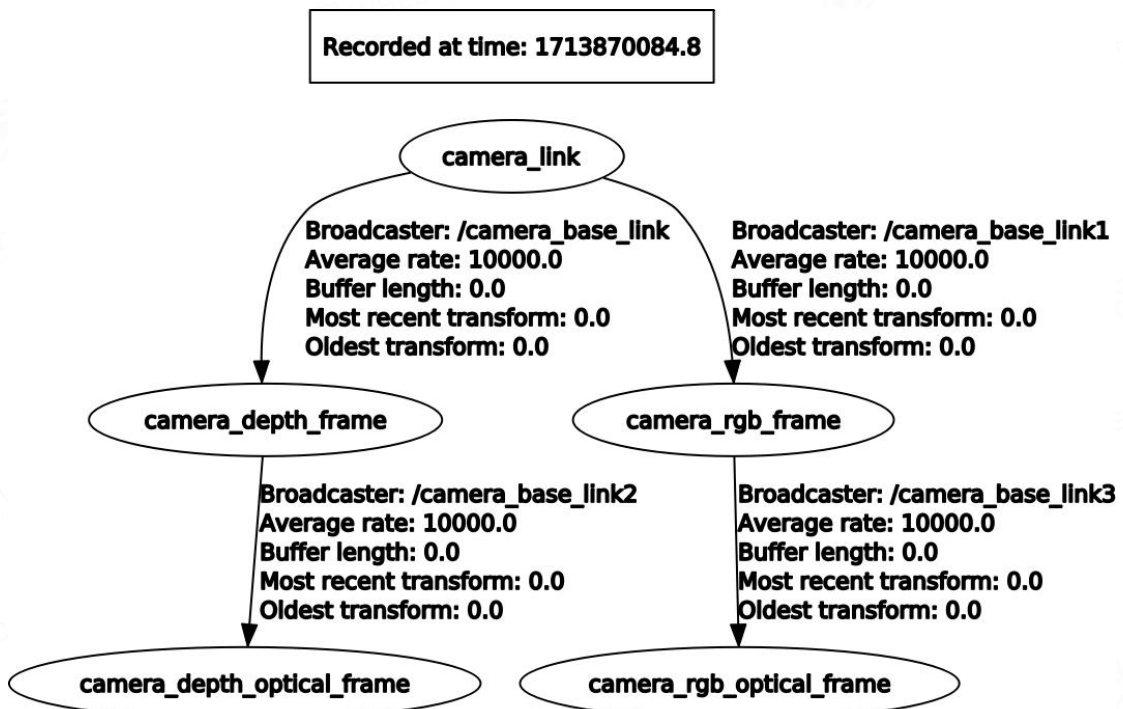
ROS TensorFlow 手写数字识别功能节点关系图

③ 查看 ROS TensorFlow 功能 TF 树:

```
roslaunch rqt_tf_tree rqt_tf_tree
```

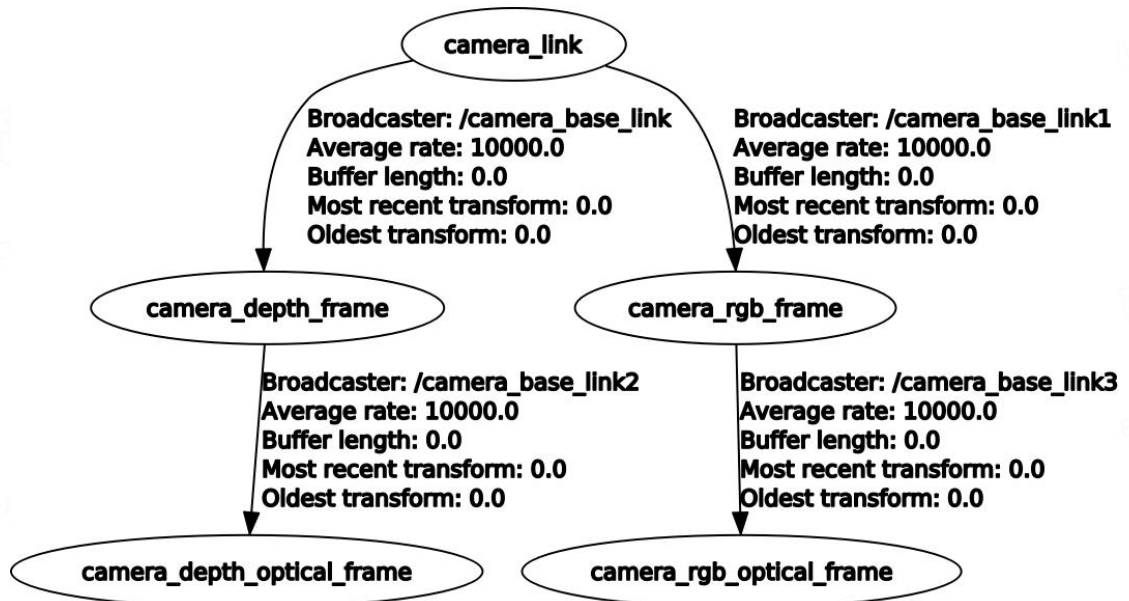


ROS TensorFlow 物体识别功能 TF 树



ROS TensorFlow 目标检测功能 TF 树

Recorded at time: 1713870185.29



ROS TensorFlow 手写数字识别功能 TF 树

④ ROS TensorFlow launch 文件解析:

此处仅借用 ROS 官方提供的 Tensorflow 功能包进行适配，相关机器学习内容及例程源码不进行详细讲解，本章主要讲解功能的使用。相关内容可在以下链接查看：

[GitHub - cong/ros_tensorflow: 本文介绍如何使用对象检测 API 将 Tensorflow 框架集成到 ROS 中。](#)

www.tensorflow.org