训练模型

1. 数据集图片打标签

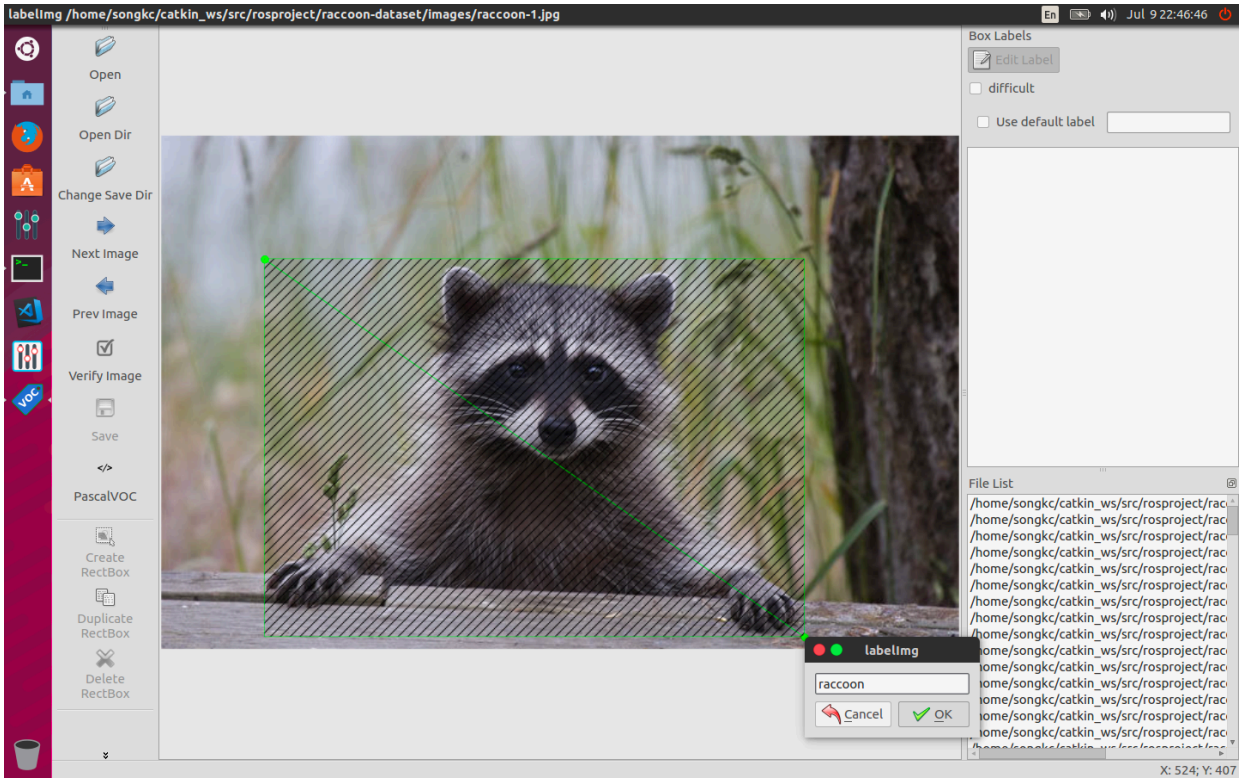
- 浣熊数据集来源于 Raccoon Detector Dataset
- 图片打标工具 LabelImg (请自行配置)

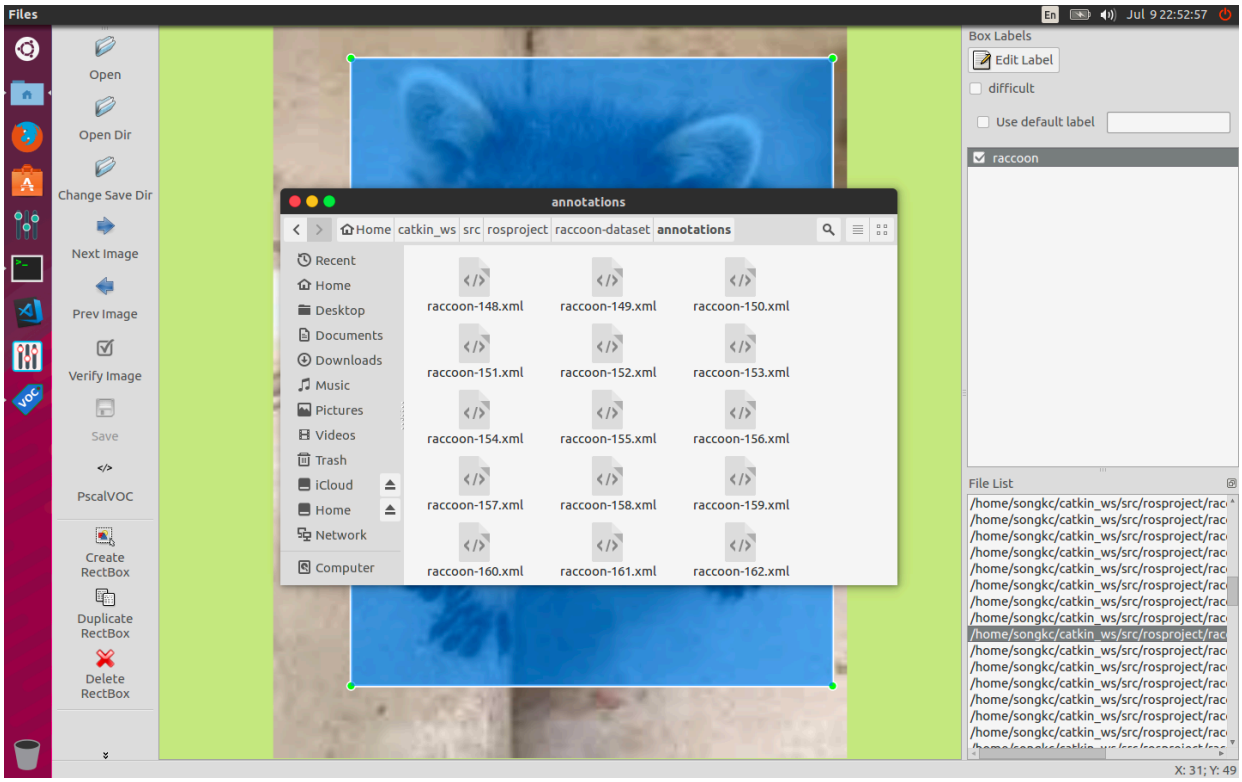
数据集文件结构：

```
raccoon_dataset/
    images/                             # 图片文件夹
    annotations/                        # 存放 xml 格式的标记文件夹
    ssd_mobilenet_v1_coco.config        # 训练脚本配置文件
```

使用 LabelImg 对图片打上标签可以得到 PASCAL VOC 格式的 XML 文件

设置好集图片标记文件(XML)的默认存储路径后，打开数据集所在文件夹即可开始图片标记





2. 编写训练集文本和验证集文本

在这之前还需要定义好类别 ID 与类别名称的关系，通常用 pbtxt 格式文件保存，在 **raccoon_dataset** 文件夹下建立名为**raccoon_label_map.pbtxt** 的文本文件。

因为只有一个类别，所以这里就只需要定义 1 个 item，若你有多个类别，就需要多个 item，注意 **id** 从 1 开始，**name** 的值要和标注文件里的类别 **name** 相同，即你在图像打标的时候标记的是 **raccoon**，这里就要写**raccoon**，不能写"浣熊"。

```
item {
  id: 1
  name: 'raccoon'
}
```

接着需要用两个文本文件，来告诉转换脚本，哪些图片文件用来做训练集，哪些图片文件用来做验证集。这里我们取前 160 张作为训练集，后 40 张作为验证集。在 **raccoon_dataset** 文件夹下新建一个名为 **train.txt** 的文本文件，内容如下：

```
raccoon-1
raccoon-2
raccoon-3
...
raccoon-160
```

在 **raccoon_dataset** 文件夹下新建一个名为 **val.txt** 的文本文件，内容为

```
raccoon-161
raccoon-162
raccoon-163
...
raccoon-200
```

以上两个文本都可以通过脚本生成。

3. 生成 TFRecord 格式的训练集和验证集文件

在 **rosproject/scripts** 文件夹下运行已编写好的图片和标记转换 TFRecord 格式脚本，生成 TFRecord 格式的训练集文件。

```
$ python object_detection/dataset_tools/create_dataset_tf_record.py \
        --
data_dir=/home/songkc/catkin_ws/src/rosproject/raccoon_dataset/images \
        --
set=/home/songkc/catkin_ws/src/rosproject/raccoon_dataset/train.txt \
        --
output_path=/home/songkc/catkin_ws/src/rosproject/raccoon_dataset/train.rec
ord \
        --
label_map_path=/home/songkc/catkin_ws/src/rosproject/raccoon_dataset/raccoo
n_label_map.pbtxt \
        --
annotations_dir=/home/songkc/catkin_ws/src/rosproject/raccoon_dataset/annot
ations
```

如果输出 **if not xml**，说明执行成功，**raccoon_dataset/train.record** 就是我们需要的 TFRecord 格式的训练集文件。

在 **rosproject/scripts** 文件夹下运行已编写好的图片和标记转换 TFRecord 格式脚本，生成 TFRecord 格式的验证集文件。

```
$ python object_detection/dataset_tools/create_dataset_tf_record.py \
        --
data_dir=/home/songkc/catkin_ws/src/rosproject/raccoon_dataset/images \
        --set=/home/songkc/catkin_ws/src/rosproject/raccoon_dataset/val.txt
\
        --
output_path=/home/songkc/catkin_ws/src/rosproject/raccoon_dataset/val.recor
d \
        --
label_map_path=/home/songkc/catkin_ws/src/rosproject/raccoon_dataset/raccoo
n_label_map.pbtxt \
        --
annotations_dir=/home/songkc/catkin_ws/src/rosproject/raccoon_dataset/annot
ations
```

如果输出 **if not xml**，说明执行成功，**raccoon_dataset/val.record** 就是我们需要的 TFRecord 格式的训练集文件。

4. 配置训练脚本并训练

在 **raccoon_dataset/ssd_mobilenet_v1_coco.config** 中将训练输入和验证输入路径修改为前面生成的 TFRecord 格式的训练集和验证集文件的路径即可。

接着在 **rosproject/scripts** 文件夹下执行下列命令开始训练模型。

```
$ python train.py --logtostderr \--
pipeline_config_path=/home/songkc/catkin_ws/src/rosproject/raccoon_dataset/
ssd_mobilenet_v1_coco.config \--
train_dir=/home/songkc/catkin_ws/src/rosproject/raccoon_dataset/train
```

将检查点文件导出为冻结的模型文件，TensorFlow 网络中含有大量的需要训练的变量，当训练结束时，这些变量的值就确定了，我们可以用下面的方法将训练的检查点文件里的变量替换为常量，导出成用于推断的模型文件。注意 75895 部分根据你自己的最后一个检查点的编号来调整。

```
$ python export_inference_graph.py \
    --
pipeline_config_path=/home/songkc/catkin_ws/src/rosproject/raccoon_dataset/
ssd_mobilenet_v1_coco.config \
    --
trained_checkpoint_prefix=/home/songkc/catkin_ws/src/rosproject/raccoon_dat
aset/train/model.ckpt-75895 \
    --
output_directory=/home/songkc/catkin_ws/src/rosproject/raccoon_dataset/trai
n
```

raccoon_dataset-master/train 下的 **frozen_inference_graph.pb** 文件最终要的模型文件。

修改 **rosproject/scripts** 文件夹下的 **obj_detect.py** 文件中的第 44 行和 46 行，分别对应上生成的模型文件 pb 路径和之前定义的 pbtxt 文件即可使用新训练的模型进行识别。