

## Project 路径:

- 1、备份一：已经上传 gitlab。
- 2、备份二：10.1.2.200 服务器

**project 路径：** /home/datadisk3/pengyue/xpy/tensorflow/tf\_object\_det/RetinaNet/retinanet-tensorflow/

**目标检测数据路径：** /home/datadisk3/pengyue/xpy/tensorflow/tf\_object\_det/data/body\_detection\_data/

## 一、模型训练 Traing:

终端运行 python train.py 直接启动训练

训练配置如下:

train.py 文件中变量

model\_save\_dir: 用于保存 ckpt 模型文件的路径

summary\_dir: tensorboard 路径

```
457     args = parser.parse_args()
458     #####
459     model_save_dir = "./checkpoints/retinanet2_mojing2/lcls_224x384_6ssd_a7_1b_nop7_p045_n04_Fltf_Res50_rbw"
460     summary_dir = "./summary/retinanet2_mojing2/lcls_224x384_6ssd_a7_1b_nop7_p045_n04_Fltf_Res50_rbw"#lcls 2
```

--restore: 用于 restore 一个模型全部的变量 weights 进行 finetune

--backbone\_weight\_restore: 用于 restore 一个 backbone 的变量 weights 进行 finetune

```
443     parser.add_argument('--restore', '-r', action='store_true', help='whether restore all parameters weights or not')
444     parser.add_argument('--backbone_weight_restore', '-rbw', action='store_true', help='whether restore just backbone_weight o
```

## 二、模型 ckpt 文件 inference 测试:

终端运行 inference\_decode.py 文件

需要的配置有:

- 1、训练好的 ckpt 文件路径 model\_ckpt\_path
- 2、训练好的模型 ckpt 对应输入图像的大小 input\_size
- 3、待测试数据的路径 image\_test\_dir, 以及需要测试结果的路径 result\_dir

## 三、模型 pb 文件生成:

终端运行运行 pb\_save.py 文件

需要的配置有:

- 1、训练好的 ckpt 文件路径 model\_ckpt\_path
- 2、待保存的 pb 文件路径
- 3、待保存的 op 节点名称 (通过 print('op 节点')) 得到

```
66     with self.sess.as_default():
67         if args.restore:
68             model_path = './checkpoints/retinanet2_mojing2/lcls_448x672_5ssd_a4_1branch_nop3p7_data2_p047_n04_alph098_smooth10_tttttttttttt/model_24.ckpt'
69             self.saver.restore(self.sess, model_path)
70             print("\n Restore all weighes successful !!!")
71         print("\n Building session !!! \n ")
72         print(self.d_bboxes, self.d_cls_pred, self.d_score)
73
74         constant_graph = graph_util.convert_variables_to_constants(self.sess, self.sess.graph_def, ['images', 'traing_mode', 'concat_12', 'GatherV2_6', 'GatherV2_8'])
75         with tf.gfile.FastGFile(os.path.join(model_pb_dir, 'model.pb'), mode='wb') as f:
76             f.write(constant_graph.SerializeToString())
77         print("save as pb file over")
```

## 四、模型 pb 文件 inference 测试：

终端运行 pb\_inference.py 文件

配置同 ckpt 文件 inference 测试。

1、训练好的 ckpt 文件路径 model\_ckpt\_path

2、训练好的模型 ckpt 对应输入图像的大小 input\_size

3、待测试数据的路径 image\_test\_dir，以及需要测试结果的路径 result\_dir

注意：

pb 文件测试时关于 op 节点名称 (graph.get\_tensor\_by\_name('op\_name')) 的问题解决方案为：

在保存 pb 的文件 pb\_save.py 中直接 print 所需节点 op，得到 op\_name。例如\_bboxes、\_cls\_pred 如下图：

```
116         input      = graph.get_tensor_by_name('images:0')
117         is_training = graph.get_tensor_by_name('training_mode:0')
118         _bboxes = graph.get_tensor_by_name('concat_12:0')
119         _cls_pred = graph.get_tensor_by_name('GatherV2_6:0')
120         _score = graph.get_tensor_by_name('GatherV2_8:0')
```

## 五、文件描述：

(一) configuration.py 文件：

包含训练以及测试时的参数配置。通过以下形式调用，直接作为全局变量。

```
12     from configuration import conf
```

(二) inputs\_multi.py 文件：

主要为目标检测数据的读取以及制作 dataset，包含数据 xml 文件的读取，以及图像预处理的整体框架。

dataset\_generator() 为外部调用的接口函数，用于生成训练集、测试集迭代器。

```
138     def dataset_generator(mode,
139                           input_size=conf.input_size,
140                           num_epochs=conf.num_epochs,
141                           batch_size=conf.batch_size,
142                           buffer_size=conf.buffer_size,
143                           return_iterator=False,
144                           channels_first=False):
145         """Create dataset including [image_dataset, bboxes_dataset, labels_dataset]
146         Args:
147             mode: (str) 'train' or 'val'
148             input_size: (int) input size (h, w)
149             num_epochs: (int) nums of looping over the dataset
150             batch_size: (int) batch size for input
151             buffer_size: (int) representing the number of elements from this dataset
152                        from which the new dataset will sample, say, it
153                        maintains a fixed-size buffer and chooses the next
154                        element uniformly at random from that buffer
155             return_iterator: (bool) if false, return dataset instead
156         """
```

(三) loss.py 文件：

定义损失函数，包含 focal loss，smooth\_L1 loss，Iou\_loss(用于预测 proposal 与 Ground Truth 的 iou)。loss\_fn () 函数作为外部调用的接口。

```

288 def loss_fn(loc_preds, loc_trues, cls_preds, cls_trues, anchor_boxes, num_classes=20):
289     """Compute loss between (loc_preds, loc_targets) and (cls_preds, cls_targets).
290
291     Args:
292         loc_preds: (tensor) predicted locations, sized [batch_size, #anchors, 4].
293         loc_trues: (tensor) encoded target locations, sized [batch_size, #anchors, 4].
294         cls_preds: (tensor) predicted class confidences, sized [batch_size, #anchors, #classes].
295         cls_trues: (tensor) encoded target labels, sized [batch_size, #anchors].
296
297     loss:
298         (tensor) loss = SmoothL1Loss(loc_preds, loc_targets) + FocalLoss(cls_preds, cls_targets).
299     """

```

(四) encoder.py 文件：功能为：

1、anchor 的配置方式以及生成。

2、训练前，数据预处理中对数据进行 encoder，使得每个 anchor 都能匹配到一个与之 iou 最大的 GT，即生成正负样本。以及 GT 的 encoder 变换（如下图）。BoxEncoder () 类的 encode () 函数作为外部接口。

```

124 def encode(self, boxes, labels, input_size, pos_iou_threshold=0.5, neg_iou_threshold=0.4):
125     """Encode target bounding boxes and class labels.
126
127     We obey the Faster RCNN box coder:
128         tx = (x - anchor_x) / anchor_w
129         ty = (y - anchor_y) / anchor_h
130         tw = log(w / anchor_w)
131         th = log(h / anchor_h)

```

3、用于在前向测试时，对预测的结果进行 decoder，以及后期的候选框的去重处理 NMS 操作。

BoxEncoder () 类的 decode () 函数作为外部接口。如下：

```

181 def decode(self, loc_preds, cls_preds,
182            input_size=conf.input_size,
183            output_size = None,
184            cls_thred=conf.cls_thred,
185            max_output_size=conf.max_output_size,
186            nms_thred=conf.nms_thred,
187            return_score=True,
188            tf_box_order=conf.tf_box_order):
189     """Decode outputs back to bounding box locations and class labels.
190
191     We obey the Faster RCNN box coder:
192         tx = (x - anchor_x) / anchor_w
193         ty = (y - anchor_y) / anchor_h
194         tw = log(w / anchor_w)
195         th = log(h / anchor_h)

```

(五) ./retinanet2 文件夹

包含 RsNet50、ResNet101 等 backbone 的网络搭建。

包含 retinane 网络搭建。

包含 shuffleNet\_v2 的网络搭建。

(六) ./utils 文件夹

box.py 文件包含 encoder 过程中的 box\_iou 计算。preprocess.py 文件包含数据预处理如随机翻转、裁剪等。

(七) ./mAP 文件夹  
包含 mAP 的计算过程。

终端直接运行 python test.py 文件。运行时需要用到上述 configuration.py 文件的参数配置。  
mAP 计算过程主要包含于 eval.py 文件中。