

step 0: 基本说明

注：使用框架切记修改框架，尽量使自身情况复合框架要求，以免带来不必要的麻烦

1. 模型说明

```
# 框架内的模型很多，尽量使用较为熟悉或者知名的
# 开始选取mask-rcnn，结果程序老是出错，主要这个模型也可以用来做分割，里面涉及做分割的模块，而我提供的数据集是做检测的，所以老是运行不成功
# 本篇使用faster-rcnn作为模型选择
```

2. 数据集说明

```
# mmdetection对coco格式的数据集支持较为良好，建议使用coco格式数据集，对于非coco格式数据集，建议转换成coco格式数据集，避免不必要的麻烦
```

step 1: 准备配置文件

注：前面说过，尽量不要再框架已有的文件上直接修改，而是建立自己的工作环境修改，避免不必要的麻烦

```
# 现在在mmdetection/下
cd configs
# 现在在configs/下，建立自己的模型配置文件目录
mkdir 个性化模型配置文件目录名
" 例如: mkdir Stepend (注: Stepend 是我的英文标识)"
cd 个性化模型配置文件目录名
# 创建配置文件，命名规则如下

"""
配置文件名称风格：
    我们遵循以下样式来命名配置文件。建议贡献者遵循相同的风格：
    {model}_{model setting}_{backbone}_{neck}_{norm setting}_{misc}_{gpu x
batch_per_gpu}_{schedule}_{dataset}

    1. {xxx} 是被要求的文件 [yyy] 是可选的。

    2. {model}: 模型种类，例如 faster_rcnn, mask_rcnn 等。

    3. [model setting]: 特定的模型，例如 htc 中的without_semantic, reppoints 中的
moment 等。

    4. {backbone}: 主干网络种类例如 r50 (ResNet-50), x101 (ResNeXt-101) 等。

    5. {neck}: Neck 模型的种类包括 fpn, pafpn, nasfpn, c4 等。
```

6. [norm_setting]: 默认使用 bn (Batch Normalization), 其他指定可以有 gn (Group Normalization), syncbn (Synchronized Batch Normalization) 等。gn-head/gn-neck 表示 GN 仅应用于网络的 Head 或 Neck, gn-all 表示 GN 用于整个模型, 例如主干网络、Neck 和 Head。

7. [misc]: 模型中各式各样的设置/插件, 例如 dconv、gcb、attention、albu、mstrain 等。

8. [gpu x batch_per_gpu]: GPU 数量和每个 GPU 的样本数, 默认使用 8x2。

9. {schedule}: 训练方案, 选项是 1x、2x、20e 等。1x 和 2x 分别代表 12 epoch 和 24 epoch, 20e 在级联模型中使用, 表示 20 epoch。对于 1x/2x, 初始学习率在第 8/16 和第 11/22 epoch 衰减 10 倍; 对于 20e, 初始学习率在第 16 和第 19 epoch 衰减 10 倍。

10. {dataset}: 数据集, 例如 coco、cityscapes、voc_0712、wider_face 等。
"""

假设我们使用的是faster-rcnn, 采用YZ的数据集, 我们在个性化模型配置文件目录下建立配置脚本 faster_rcnn_r50_fpn_giou_1x_coco_YZ.py, 内容如下

```
# faster_rcnn_r50_fpn_giou_1x_coco_YZ.py
# _base_ 是继承对象, 用来继承faster_rcnn_r50_fpn_1x_coco.py里的配置, 我们需要把改修改的部分修改即可

_base_ = '../faster_rcnn/faster_rcnn_r50_fpn_1x_coco.py'

# 我们选用的是YZ 0726的数据集(这里不懂问问师兄, 因为这个数据集没有说明文档)
# YZ 0726的数据集基本情况: 有3类, 分别是["person_water", "person_shore", "person_boat"]
# 所以我们要讲分类种类设置成3
model = dict(
    roi_head=dict(
        bbox_head=dict(
            num_classes=3,
            reg_decoded_bbox=True,
            loss_bbox=dict(type='GIoULoss', loss_weight=10.0)))

# 修改数据集相关配置
dataset_type = 'CocoDataset'
classes = ("person_water", "person_shore", "person_boat")
# 下面 img_prefix 指的是图片的目录, 目前在/data/A0I/YunZhou/YZSOD/0726/images/train/
# ann_file指的是annotations文件的位置, 即标注文件的位置
data = dict(
    train=dict(
        img_prefix='/data/A0I/YunZhou/YZSOD/0726/images/train/',
        classes=classes,

    ann_file='/data/A0I/YunZhou/YZSOD/0726/annotations/yzsod0726_train.json'),
    val=dict(
        img_prefix='/data/A0I/YunZhou/YZSOD/0726/images/val/',
        classes=classes,
        ann_file='/data/A0I/YunZhou/YZSOD/0726/annotations/yzsod0726_val.json'),
    test=dict(
        img_prefix='/data/A0I/YunZhou/YZSOD/0726/images/val/',
        classes=classes,
```

```
ann_file='/data/AOI/YunZhou/YZSOD/0726/annotations/yzsod0726_val.json'))
```

```
# 我们可以使用预训练的 faster R-CNN 来获取更好的性能
# 这里我们不实用预训练模型，所以注释即可
# load_from = 'checkpoints/mask_rcnn_r50_caffe_fpn_mstrain-poly_3x_coco_bbox_mAP-
0.408__segm_mAP-0.37_20200504_163245-42aa3d00.pth'
```

step 2: 训练配置模型

```
# 现在在mmdetection/下
python tools/train.py 配置文件路径
""" 例如：我的配置文件在configs/Stepend/下，
名为：faster_rcnn_r50_fpn_giou_1x_coco_YZ.py

那么，训练命令则为：
python tools/train.py configs/Stepend/faster_rcnn_r50_fpn_giou_1x_coco_YZ.py
"""
# 训练好的权重保存在mmdetection/work_dirs/目录下对应实验，认真查看训练日志即可看到
```

step 3: 测试模型

```
python tools/test.py 配置文件路径 权重文件路径 --eval bbox
```

step 4: 看看效果

```
# 现在在mmdetection/下
# 在demo/下建立测试脚本
cd demo
# 现在在demo/下
# 建立faster-rcnn-YZ-test-demo.py， 写入以下内容
```

```
# faster-rcnn-YZ-test-demo.py
from mmdet.apis import init_detector, inference_detector

config_file = 刚刚的配置文件路径
checkpoint_file = 刚刚训练的参数路径
device = 'cuda:0'
# 初始化检测器
model = init_detector(config_file, checkpoint_file, device=device)
# 测试图片路径
img = 测试图片路径
# 推理演示图像
result = inference_detector(model, img)
# 保存图片
model.show_result(img, result, out_file="result.jpg")
```

测试效果：

```
python demo/faster-rcnn-YZ-test-demo.py  
# 结果在mmdetection/下, 名为result.jpg
```