

step 0: 了解mmdetection/configs/文件目录

1. 该目录下有两种目录，一种是_base_，另一种是其他；前者下的组件组成的配置被称为原始配置(primitive)
2. 原始目录下有：dataset、model、schedule、default_runtime四个目录，这四个目录下定义了一个检测模型训练、测试以及使用的基本配置情况
3. 其他非原始配置目录代表了一类方法，他们可以继承原始配置的配置信息，也可以直接重写所有配置

step 1: 查看完整配置

```
# 现在在mmdetection/目录下
python tools/misc/print_config.py 配置路径
# 用上面的命令查看完整配置
```

step 2: 了解完整配置文件

一个完整的配置文件包含四个方面的信息：model、dataset、schedule、default_runtime，分别描述了用什么模型、用什么数据集、用什么优化方案、运行时该记录什么以及从什么地方开始运行，下面分别从四个方面的信息。

- model

```
model = dict(
    type="xxx", # 检测器名称
    backbone=dict(...), # 指定特征提取网络
    neck=dict(...), # 指定特征融合网络
    xxx_head=dict(...), # 指定回归网络
    train_cfg=dict(...), # 指定训练超参数
    test_cfg=dict(...) # 指定测试超参数
)
```

上面描述了一个目标检测模型的参数配置，每种模型的具体配置都有差别，总体结构是一样的。

- dataset

```
dataset_type = "xxx" # 指定数据集类型coco、pascal voc or others
classes = ("xxx", "yyy", ..., "zzz") # 目标类别
data_root = "xxx" # 指定数据集跟路径
"""
example:
对于coco2017来说，文件目录如下：
xxx/coco2017/:
    annotations/:
        train.json
    test/:
        1.jpg
    train/:
        1.jpg
```

```

    那么data_root = "xxx/coco2017/"
"""
# 图像标准化
img_norm_cfg = dict(
    mean=[xxx, xxx, xxx],
    std=[xxx, xxx, xxx],
    to_rgb=True
)
# 训练时图片预处理流程
train_pipeline = [...]
# 测试时图片预处理流程
test_pipeline = [...]
"""

    以上两项详见: Tutorial 5: 数据预处理流程
"""
data = dict(
    samples_per_gpu = x, # 单个GPU的batch size数, x是数字
    workers_per_gpu = x, # 单个GPU的加载线程数, x是数字
    train = dict(
        type="xxx", # 数据集类别
        ann_file="xxx/xxx.json", # 训练标注文件路径
        img_prefix="xxx/xxx/", # 训练图片的根路径
        pipeline = train_pipeline # 训练时图片预处理流程
    ),
    val=dict(...), # 和train差不多, 注意路径即可
    test=dict(...)# 和train差不多, 注意路径即可
)

```

- schedule

```

optimizer = dict(...) # 指定优化器和相关参数
optimizer_config = dict(grad_clip=None) # 优化器配置, 这个参数主要是对梯度的范围限制
lr_config = dict(...) # 学习率相关配置, 主要针对学习率可变的训练策略
runner = dict(...) # 训练回合指定

```

- default_runtime

```

checkpoint_config = dict(interval=1) # 每隔多少个epoch保存一次训练结果
log_config = dict(
    interval=50, # 日志打印间隔
    hooks = [...] # 指定日志记录器
)
dist_params = dict(backend="nccl") # 分布式训练参数
log_level = "INFO" # 日志级别
load_from = None # 指定加载路径作为预训练模型
resume_from = None # 指定恢复检查点
workflow = [("train", 1)] # runner的工作流程, [("train", 1)]表示只有一个工作流程且
#  workflow仅执行一次
work_dir = "work_dir" # 用于保存当前实验的模型检查点和日志的目录

```

