

step 0: 基本说明

使用框架和自己造轮子的区别在于如何使用已有工具实现自身目的，既然已经走到这就是决定使用框架了，所以，如果没有必要，坚决不改动已有框架，本着尽量满足框架要求的原则实现自身目的，这一节正好以数据集为例，mmdetection支持coco数据集、pascal voc数据集以及一种基本格式，对于一种全新的检测数据集，他应该包含目标检测的要素，也就是能够满足以上三种数据集的基本要求，能够使框架正常运转，所以我们要做的就是将现有数据集转换成以上格式，这样工作成本和维护成本是最低。

step 1: 目标检测coco数据基本格式

```
'images': [
  {
    'file_name': 'COCO_val2014_0000000001268.jpg',
    'height': 427,
    'width': 640,
    'id': 1268
  },
  ...
],

'annotations': [
  {
    'segmentation': [[192.81,
      247.09,
      ...
      219.03,
      249.06]], # 如果有 mask 标签
    'area': 1035.749,
    'iscrowd': 0,
    'image_id': 1268,
    'bbox': [192.81, 224.8, 74.73, 33.43],
    'category_id': 16,
    'id': 42986
  },
  ...
],

'categories': [
  {'id': 0, 'name': 'car'},
]
```

在 json 文件中有三个必要的键：

- `images`: 包含多个图片以及它们的信息的数组，例如 `file_name`、`height`、`width` 和 `id`。
- `annotations`: 包含多个实例标注信息的数组。
- `categories`: 包含多个类别名字和 ID 的数组。

在数据预处理之后，使用现有的数据格式来训练自定义的新数据集有如下两步（以 COCO 为例）：

1. 为自定义数据集修改配置文件。
2. 检查自定义数据集的标注。

其中第二点极其重要，我们一般会忽略这一点，对于公开数据集可能已经有人检查过了，但是对于自定义数据集这很难说，尤其是现在数据驱动的情况下这点更重要。

写个转换脚本实现转换即可，这个不在赘述，不会的可以搜索询问。