

使用说明

文件&代码说明

- `original data` 原始的训练数据，手动由标注好的图像整理得到。VOC格式，Annotation为标注，JPEGImages为图像，共242张6k*6k大小图像
- `test` 为测试数据，暂时为7张（01_27~29三个场景）图像
- `voc_type` 划分为voc格式patch图像。`lffd` 和 `mmdetection` 表示要转化到哪种模型（框架）的输入格式；其中，`300_15` 表示300*300大小的patch，只保留最长边大于15个像素的目标，其他类似文件夹同理；`test_crop` 为手动裁剪出机场的测试集；以上均由代码 `voc2patch.py` 得到
- `mmdet_type` 适用于mmdetection的coco格式标注。由代码 `patch2coco.py` 得到
- `lffd_type` 适用于lffd的txt格式标注。由代码 `patch2lffd.py` 得到
- `dota` dota的原始数据，coco格式，已用官方代码拆分为1024*1024大小。由代码 `dota2lffd.py` 可转为lffd的输入格式，也可用于coco到lffd格式的转换
- `test2lffd.py` 仅将测试集从voc转换为lffd格式

使用流程

lffd

1. Run `voc2patch.py`
代码中可能需要修改的项
 - `to_where = 'lffd'`
 - 输入图像大小 - patch大小ps以及patch滑动间隔po
 - 目标的最小尺寸pixel_low_bound
2. Run `patch2lffd.py`
代码中可能需要修改的项
 - `patch_path`填上一步保存在voc_type中的文件路径
 - `save_path`为'lffd_type/XXX'，XXX与patch_path的文件夹同名
3. 将 `lffd_type/XXX` 中的train文件夹复制到 `lffd/face_detection/data_provider_farm/data_folder/train/` 中，先删掉原有的train文件夹（其中只有图像），再复制；将 `lffd_type/XXX/Annotations` 下的train_gt.txt覆盖lffd里train/split下的同名文件；删除neg_images文件夹
4. 运行data_provider_farm下的 `pickle_provider.py`
`picture_type`对应输入图像的后缀，main函数里参数为'train'
运行后生成数据文件train_data.pkl
5. 运行config_farm下的 `config_farm/configuration_10_320_20L_5scales_v2.py`，训练模型
大小16的batch size，训练200w个iters大概需要5天
若要微调模型，在 `param_pretrained_model_param_path` 设置加载的模型的路径
6. 运行test2lffd得到测试集，同步骤3和4，复制到data_folder/test下，并生成pkl数据

7. 运行accuracy_evaluation下的 `evaluation.py`

mmdetection

1. 运行 `voc2patch.py`
2. 运行 `patch2mmdet.py`
3. 在mmdetection/configs/aerial_plane下faster rcnn或yolov3配置文件中, `data_root` 的路径设置为数据的保存位置 `mmdet_type/XXX`
4. 测试集同test2lffd.py一样单独写一个test2mmdet, 仿照patch2mmdet实现 (暂未实现), 处理好的数据保存到 `mmdet_type/XXX`