

Readme

一、数据集

1. 目录说明

数据集位于dataset/HongKong_download文件夹中，包含以下子目录：

目录名	内容描述
depthmap	存储街景深度图
gsv_cut	存储切分后的街景图像
metadata	存储街景元数据信息
streetview	存储街景全景图像
signboard_cut	存储切分后的招牌图像
signboard_detect	存储招牌检测和OCR结果

2. Signboard_Dataset招牌检测数据集说明

数据集位于dataset/signboard_dataset，以YOLO格式存储。图像文件存储在images文件夹中，标注文件存储在labels文件夹中，每个图像对应一个同名TXT文件，每行格式：`类别ID 中心点X 中心点Y 宽度 高度`（所有值为归一化坐标，范围0-1）。

3. 招牌检测结果格式说明

signboard_detect目录中的文件采用JSON格式，存储了招牌检测和OCR的结果。每个文件对应一张街景图像的招牌检测结果，格式如下：

代码块

```
1  {
2      "0_back_0": { //招牌ID, 命名格式为"图像ID_序号"
3          "board_contour": [ //招牌在图像中的边界框坐标
4              71,          // xmin
5              537,         // ymin
6              243,         // xmax
7              572          // ymax
8          ],
9          "confidence": 0.740544855594635, // 招牌检测置信度
10         "ocr_result": [
11             "HAPPY SQUARE" // OCR识别文本结果
```

```
12     ]
13     }
14 }
```

3. Benchmark Dataset集说明

(1) 路径: dataset/gsv_label

目录名	内容描述
label_voc	原始标注文件,voc格式
signboard_entity	汇总后的标注文件
osm_poi	参考OSM POI
metadata	街景元数据

(2) 标注文件格式说明

原始标注文件用labelme标注软件标注，格式如下，其中name代表POI名称

代码块

```
1  <?xml version="1.0" ?>
2  <annotation>
3      <folder>img_gsv_choose</folder>
4      <filename>397_leftfront.jpg</filename>
5      <path>D:\subject\data\signboard detection
dataset\img_gsv_choose\397_leftfront.jpg</path>
6      <source>
7          <database>Unknown</database>
8      </source>
9      <size>
10         <width>1024</width>
11         <height>1024</height>
12         <depth>3</depth>
13     </size>
14     <segmented>0</segmented>
15     <object>
16         <name>北京同仁堂</name>
17         <pose>Unspecified</pose>
18         <truncated>0</truncated>
19         <difficult>0</difficult>
20         <bndbox>
21             <xmin>711</xmin>
22             <ymin>482</ymin>
23             <xmax>774</xmax>
24             <ymax>525</ymax>
```

```
25         </bndbox>
26     </object>
27 </annotation>
28
```

汇总后的标注文件保存在/dataset/gsv_label/signboard_entity/area_id/signboard entity.json文件中，包含"sign"和"poi"两个字段，格式为：

代码块

```
1  {
2      "sign": [
3          {
4              "name": "adidas",
5              "image_name": "783_back.jpg",
6              "board_contour": [xmin,ymin,xmax,ymax],
7              "coordinate": [lat,lon,alt],
8              "sign_id": 0
9          }],
10     "poi": [
11         {
12             "name": "adidas",
13             "match_type": "manual",
14             "sign_group": [],
15             "poi_id": 0}]
16 }
```

4. 生成结果说明

路径：poi_dataset

其中包含4个文件夹，分别是总的生成结果和3个子区域的文件夹，其中

merged_poi_with_name.json是最后的生成结果，格式为：

代码块

```
1  {
2      "0": {
3          "text_content": [...], //OCR结果
4          "coordinate": [lat,lon,alt], //生成坐标
5          "sign_id": [
6              "155_rightback_0" #关联的招牌示例id
7          ],
8          "top_k_sign_id": [
9              "155_rightback_0"
10         ],
11         "group_id": 401,
12     }
13 }
```

```
12         "name": "xxx" //POI名称
13     },
```

dev.json是用来做POI名称识别的chatglm标准格式json文件，generated_predictions.txt是POI名称识别的chatglm标准格式输出，poi_match.json是与OSM-POI匹配的结果，在smerged_poi_with_name.json的基础上加入了字段"match_POI"。

二、代码说明

1. 前向推理文件路径：/tools

文件名	内容描述
det_predict.py	使用YOLO模型检测图像中的招牌并保存检测结果
ocr_predict.py	对招牌检测结果进行OCR文本识别并更新JSON文件
poi_cluster.py	创建招牌示例并聚类
poi_processor.py	合并多个组下的POI文件，生成用于POI名称提取的数据集
chatglm_predict.py	chatglm生成POI名称
read_predict.py	从预测结果JSON文件中读取POI数据， 将name字段添加到JSON文件中形成最终结果
tools/mycode/poi_filter.py	根据子区域bbox过滤POI
main.py	主函数

2. 预处理文件路径

文件名	内容描述
Equirec2Perspec-master/Equirec2Perspec.py	将街景全景图切分自然视角图像
tools/mycode/poi_match.py	与真实值匹配并计算指标
tools/mycode/match_pair.py	使用SuperGlue算法匹配两张图像之间的特征点， 整合所有关键点和匹配信息到文本文件
ools/mycode/colmap_process.p	colmap三维重建并转换坐标系

3. config文件说明

路径：/config

其中包含6个文件，config_signboard_entity.py是总区域的config文件，config_signboard_entity_{area_id}.py是子区域的config文件，det_config.py是目标检测的config文件，ocr_config是文本识别的config文件。

三、注意事项

1. 安装mimcv时候不能直接安装，需要先安装mim

代码块

```
1 pip install -U openmim
2 mim install mimcv
```

2. 如果遇到报错：FatalError: `Segmentation fault` is detected by the operating system.

在终端中输入

代码块

```
1 conda activate your_envs
2 export LD_LIBRARY_PATH=/home/moss/miniconda3/envs/your_envs/lib:$PATH
```

3. 有的库用 pip -r requirements安装会报错（例如coloredlogs），手动安装就好。

4. paddleOCR需要安装cudnn

安装命令参考：[cudnn安装](#)

Select Target Platform

Click on the green buttons that describe your target platform. Only supported platforms will be shown. By downloading and using the software, you agree to full terms and conditions of the [NVIDIA Software License Agreement](#).

Operating System	Linux	Windows				
Architecture	x86_64	arm64-sbsa	aarch64-jetson			
Distribution	Tarball	Debian	OpenSUSE	RHEL	Rocky	Ubuntu
Version	20.04	22.04	24.04			
Installer Type	deb (local)	deb (network)				

四、下载说明

由于数据集过大，数据集可在百度网盘上下载，请自行下载并解压到项目路径下：

链接: <https://pan.baidu.com/s/1le89G7O5Y3DPuUiXk0iCcQ> 提取码: hcxj