

基于TensorRT加速的智能遥感图像处理系统

本仓库基于NVIDIA Jetson AGX Xavier实现基于TensorRT加速的远程遥感图像处理边缘部署。

实验中使用**MacBook Air M4**作为上位机，将遥感图像通过websocket发送至Jetson开发板，Jetson AGX Xavier作为边缘设备，使用TensorRT进行模型推理，将处理后的图像通过websocket发送回上位机。

1.运行仓库代码

1.1 网络连接

要实现两个设备时间数据互传，首先要保证两台设备之间能够ping通。

代码使用端口号为8765，**请确保防火墙允许该端口号的通信!!!**

1.2 代码运行

打开终端，输入以下命令下载仓库：

```
# 克隆仓库
git clone

# 进入目录
cd yourrepository
```

1.2.1 上位机

项目中 **transfer** 目录包含上位机所有代码。

安装依赖

```
# 进入目录
cd ./transfer

# 安装依赖
pip install -r requirements.txt
```

修改ip地址

安装完依赖后，首先修改代码中ip地址为自己主机与jetson开发板的ip地址。

```
if __name__ == "__main__":
    # 示例配置，根据需要修改
    my_ip = "10.194.151.116" # 上位机
```

```
my_port = 8765
server_uri = "ws://10.171.103.229:8765" # jetson开发板
```

指定大模型API

运行大模型需要指定环境变量

```
vim .env
# 写入你自己的大模型API
GPT_BASE_URL=YOUR_BASE_URL
GPT_API_KEY=YOUR_API_KEY
```

或者手动指定，需要删除代码中 `load_env(".env")`

```
export GPT_BASE_URL=YOUR_BASE_URL
export GPT_API_KEY=YOUR_API_KEY
```

运行代码

```
python main.py
```

大模型默认索引目录为 `./datas`，如要修改，请修改 `user_read_file`、`user_list_file`、`user_load_image` 中的 `base_path`。

1.2.2 开发板

开发板Jetpack系统版本为 **4.6.5**，支持最高Python版本为 **3.6.9**

安装依赖

```
# 进入目录
cd ./receive

# 安装依赖
pip install -r requirements.txt
```

修改ip地址

同理修改代码中ip地址

```
if __name__ == "__main__":
    transfer = RealTimeImageTransfer(
        "10.171.103.229", # jetson
        8765,
        "ws://10.194.128.156:8765", # 上位机
        './build/yolov5s-best.engine' # 引擎文件地址
    )
    transfer.run()
```

修改引擎文件（.pt文件）地址

请保证引擎文件地址或.pt文件路径正确！！！，不正确请修改

```
# 运行引擎文件
vim run_engine.py
# 修改第457行
'./build/yolov5s-best.engine'

# 运行.pt文件
vim run_yolo.py
# 修改第63行
img_process, detection_info = detect_image(img, weights="best.pt",
device="0")
```

运行代码

```
# 请先在代码中配置.engine文件（或.pt文件）的路径，并保证网络通畅
# 运行未加速的yolov5处理
python run_yolo.py

# 运行TensorRT加速的yolov5处理
python run_engine.py
```

2.低版本Jetson平台部署TensorRT指南

由于Jetson平台的版本较低，我们无法直接使用Yolov8库中自带的TensorRT。因此，我们需要手动部署TensorRT。

2.1 资源准备

假设你已经拥有使用 yolov5-v6.0 训练得到 .pt 权重文件（例如：yolov5s.pt）

```
# 克隆 YOLOv5 官方仓库（v5.0 分支）
git clone -b v5.0 https://github.com/ultralytics/yolov5.git
```

```
#具体训练步骤参考git官方步骤，通过训练后可以得到后续所需.pt格式文件。
```

```
# 根据提供的requirements.txt即可  
pip install -r requirements.txt
```

2.2 部署步骤

2.2.1 克隆代码仓库

```
# 克隆 TensorRTX 转换工具（适配 YOLOv5-v6.0）(本文件夹中的yolov5)  
git clone -b yolov5-v5.0 https://github.com/wang-xinyu/tensorrtx.git
```

2.2.2 生成.wts文件

```
# 进入 tensorrtx/yolov5 目录  
cd tensorrtx/yolov5  
  
# 运行转换脚本（需指定你的 .pt 文件路径）  
python gen_wts.py -w /path/to/your/yolov5s.pt -o best.wts
```

2.2.3 生成引擎文件

```
# 在 tensorrtx/yolov5 下创建 build 目录  
mkdir build && cd build  
  
# 复制上一步生成的 .wts 文件到 build 目录  
cp ../best.wts .  
  
# 编译项目  
cmake .. && make  
  
# 生成引擎文件（耗时约 5-20 分钟）  
sudo ./yolov5 -s best.wts yolov5s.engine s
```

2.2.4 使用.engine查看加速效果

```
# 使用Jetson开发板运行  
cd ./receive  
  
# 请先配置好引擎文件的路径，同时确保上位机和板子的ip正确  
  
python run_engine.py
```

2.2.5 使用.pt查看未加速对比

```
python run_yolo.py
```