

项目启动流程指南

项目代码结构说明

```
1 | .
2 | └─ src/                                # 核心代码目录
3 |   └─ process.py                        # 🛠️ 图像处理模块
4 |   └─ requirements.txt                  # 📄 依赖
5 |   └─ test/                             # 🧪 测试目录
6 |       └─ imgs/                         # [ 待检测图片存放位置]
7 |       └─ end/                          # [ 推理结果输出目录]
8 | └─ best.onnx                           # 🤖 ONNX模型文件
9 | └─ main.py                             # 🚀 主执行入口
```

详细模型训练过程、代码解释将在doc文件夹中

模型技术实现

- 使用GAN生成球场反光、泥渍污染等特殊场景（GAN代码存放在doc文件夹中）
- 引入GhostNet模块替换部分卷积层，降低计算量
- 嵌入动态通道注意力机制（DCAM）
- 迁移学习：基于COCO预训练，用网球专用数据集微调。

实现

- 数据层：通过GAN生成特殊场景训练数据
- 模型层：使用YOLOv5框架+Ghost轻量化模块
- 注意力层：嵌入动态通道注意力机制
- 部署层：ONNX Runtime实现高效推理

二、环境准备步骤

1. 克隆项目仓库

```
1 | git clone --depth 1 https://github.com/TianZaishuiZhong/wq
```

2. 安装Python依赖

安装依赖前注意pip版本
过低可能报错，可以使用命令更新：

```
1 | pip install --upgrade pip
```

安装依赖

```
1 # 推荐方式（使用依赖清单）📦：
2 pip install -r src/requirements.txt
3
4 # 或手动安装：
5 pip install onnxruntime opencv-python numpy Pillow
```

💡 提示：

可以在python虚拟环境中安装依赖

python -m venv name

source name/bin/activate

建议用python 3.8.2，比较稳定

三、文件准备指引

1. 📁 将待检测图片放入指定文件夹位置：

src/test/imgs/

2. 📄 推理结果将输出到：

src/test/end/

四、执行命令

执行命令前，要在main.py文件的当前目录下

```
1 # 1. 单图片处理（指定置信度）🖼️
2 python main.py --image test/imgs/3.jpg --output test/end/result.jpg --
  confidence 0.5
3
4 # 2. 批量处理文件夹 📁
5 python main.py --folder test/imgs --output test/end --confidence 0.5
6
7 # 3. 使用自定义模型 📄
8 python main.py --image test.jpg --output custom_result.jpg --model
  custom.onnx
```

基础格式

```
1 python main.py [参数]
```

⚙️ 参数详解表

参数	必选	默认值	说明
<code>--image</code>	△	-	单图片路径（与 <code>--folder</code> 二选一）
<code>--folder</code>	△	-	📁 图片文件夹路径（与 <code>--image</code> 二选一）
<code>--output</code>	✓	-	输出路径（文件或目录）
<code>--model</code>	×	best.onnx	自定义模型文件路径
<code>--confidence</code>	×	0.05	📊 置信度阈值（0.0-1.0，值越大越严格）

 符号说明:

✓: 必需参数

△: 互斥参数 (与对应参数二选一)

×: 可选参数

五、最佳实践提示

1. 路径处理技巧

- **待检测图片**: 放在 `src/test/imgs/`
- **结果保存**: 到 `src/test/end/`
- **路径问题**: 推荐使用**绝对路径**避免歧义



```
1 # macOS/Linux示例
2 python main.py --image /User/project/src/test/imgs/1.png
3
4 # windows示例
5 python main.py --image D:\\project\\src\\test\\imgs\\1.png
```

3. 模型选择提示

- 默认使用当前目录的 `best.onnx`
- 自定义模型需提供完整路径:

```
1 python main.py --model models/custom.onnx
```

4. 其他注意事项

- 支持常见图片格式: JPG/PNG/BMP等
-  同名输出文件会被覆盖
-  `--image` 和 `--folder` 参数互斥, 不要同时使用!

可以创建python虚拟环境, 使用python3.8

 **问题排查**: 遇到错误时, 检查文件路径是否正确、依赖是否安装完整。