

基于百度飞桨的遥感图像智能解译平台

详细设计文档

目录

- 1 概述..... 1
 - 1.1 编写目的 1
 - 1.2 项目背景 1
 - 1.3 术语和缩略词 1
 - 1.4 参考文献 1
- 2 系统设计概述..... 2
 - 2.1 需求概述 2
 - 2.2 软件运行环境描述 3
 - 2.3 详细设计方法和工具 3
- 3 系统总体方案确认..... 3
 - 3.1 系统总体框架 3
 - 3.2 系统模块划分 4
 - 3.3 应用系统和支撑系统划分 5
- 4 系统详细设计..... 5
 - 4.1 网络端程序详细设计 5
 - 4.1.1 基于深度学习的变化检测简介..... 5
 - 4.1.2 实验环境..... 6
 - 4.1.3 数据集..... 6
 - 4.1.4 模型介绍..... 6
 - 4.1.5 模型优化..... 6
 - 4.1.6 模型训练与推理..... 7
 - 4.1.7 模型部署..... 7
 - 4.1.8 性能评价..... 7
 - 4.1.9 效果展示..... 8

4.2 平台详细设计	9
4.2.1 前后端数据交互设计.....	10
4.2.2 平台功能详细设计.....	11
4.2.3 平台界面详细设计.....	12
4.2.4 数据库设计.....	18
5 关键技术.....	19
5.1 模型优化方法	19
5.1.1 网络结构改进.....	19
5.1.2 训练技巧改进.....	19
5.2 数据分析可视化实现	19
5.2.1 指标评价体系的构建.....	19
5.2.2 可视化图表的渲染.....	20
5.3 图像操作功能实现.....	21

1 概述

1.1 编写目的

本文档主要对平台进行详细的功能设计和技术使用说明。从模型训练与部署、平台界面和功能详细设计等方面描述系统的设计思路、设计上的关键技术。

1.2 项目背景

随着遥感技术的不断发展，其在国土空间资源领域的应用也在不断扩大。由于传统的目视解译技术难以满足与日俱增的遥感数据分析应用需求，运用人工智能技术对图像进行高效、准确的解译成为大势所趋。借助基于深度学习的遥感图像识别技术对国土空间资源进行监测，对国家建设与规划意义重大。

因此，本项目拟开发一个基于百度飞桨的在线智能遥感图像解译平台，使用户能够借助相应的解译算法、通过简单的操作获取遥感图像的有效信息，以掌握国土空间资源的分布与变化情况，并为城市规划提供决策依据

1.3 术语和缩略词

序号	术语和缩略词	解释
1	PaddlePaddle	深度学习框架飞桨
2	PaddleRS	基于飞桨的遥感图像处理开发套件
3	AI Studio	基于飞桨的人工智能学习与实训社区
4	AHP	定性和定量相结合的、系统的、层次化的分析方法
5	TOPSIS	组内综合评价方法
6	Axios	基于 <code>promise</code> 的 HTTP 库
7	Flask	使用 <code>Python</code> 编写的轻量级 Web 应用框架
8	MySQL	关系型数据库

1.4 参考文献

[1] 陕永杰, 魏绍康, 苗圆, 原卫利. 基于 PSR-TOPSIS 模型的“晋陕豫黄河金三角”地区土地生态安全评价[J]. 生态经济, 2022, 38(07): 205-211

[2] 臧晓敏, 李志鹏, 胡家昕. 基于孪生网络的遥感图像变化检测方法[J]. 信息与电脑(理论版), 2021, 33(18): 56-58.

[3] 谢萍. 国土资源生态环境遥感监测与综合评估方法——以广东省为例[J]. 安徽农业科学, 2020, 48(11): 77-81+84.

- [4]陈丽甜. 农村公路网适用性评价及建设资源优化配置方法研究[D]. 华南理工大学, 2019.
- [5]王宏亮, 郝晋珉, 高艺宁. 基于 AHP-TOPSIS 模型的建设用地利用强度评价及灰色关联分析: 以内蒙古为例[J]. 北京师范大学学报(自然科学版), 2018, 54 (03) :277-283.
- [6]Zhao Zi-Tong, Cheng Hou-Ming, Wang Sheng, et al. SCC-UEFAS, an urban-ecological-feature based assessment system for sponge city construction[J]. Environmental Science and Ecotechnology, 2022, 12.
- [7]Life Science Research - Ecology; Investigators from Fujian Agriculture and Forestry University Release New Data on Ecology[J]. Ecology Environment & Conservation, 2019.

2 系统设计概述

2.1 需求概述

经过网络调查对现有的遥感图像解译平台进行分析, 主要存在以下问题:

- 1) 现有的遥感图像解译平台采用的模型优势不明显, 在识别的准确率和速度方面仍有较大的提升空间, 尤其是解译的速度较慢, 用户体验感较差;
- 2) 现有的遥感图像解译平台中, 主要面向于专业的技术人员, 需要一定的基础知识, 同时对遥感图像的格式限制严格, 用户友好度不足, 普适性和推广性较差;
- 3) 现有的遥感图像解译平台功能不够全面, 大多仅仅停留在单纯的解译上, 缺乏对解译结果的分析及可视化处理, 无法充分利用提取的遥感图像信息帮助用户做出决策。

针对当前传统系统存在的若干问题, 在提高图像解译准确率和速度、优化设计解译系统等方面, 本平台所具备的功能如下:

- 1) **登录注册:** 用户既可以直接使用平台享受快速的解译体验, 也可以通过登录认证保存自己的解译记录, 构建个性化数据库。
- 2) **图像上传:** 平台允许用户上传一般格式的遥感图像进行解译, 并可对图片进行各种操作以便确定最佳的解译区域。
- 3) **图像解译与可视化数据分析:** 基于 PaddleRS 的遥感图像解译算法, 极大的提高了解译准确率和速度。基于 AHP-TOPSIS 等数学模型构建的评价体系, 在数据分析与处理中发挥巨大作用。用户可以根据需求选择不同的算法对图像进行解译, 获得解译后的图像以及各类指标的可视化图表。
- 4) **结果下载:** 平台支持用户对解译后的图像进行下载与保存, 以便用户对图像进行进一步的处理。

5) **查看解译记录：**登录后的用户可以对其解译记录进行查询，直接调取解译结果，无需进行二次解译。

2.2 软件运行环境描述

本平台分为前端和后端两个程序，部署在本地服务器上，以网站的形式展示。数据库采用 MySQL，网站可兼容性支持主流浏览器。

2.3 详细设计方法和工具

在软件设计、测试、管理阶段所使用的方法和工具详细如下：

- 1) 使用 draw.io 绘制 E-R 图、各种流程图及功能模块图，完成对平台的总体设计；
- 2) 使用 Mockingbot 对平台进行原型设计，描述平台功能与内容的逻辑关系；
- 3) 使用数据库可视化工具 Navicat 来管理数据库相关的表及表中的数据；
- 4) 使用 postman 对后端接口进行测试，并且使用 postman 中的循环测试工具来生成大量的测试数据；
- 5) 使用 Git 进行版本控制并将代码托管到 Github 平台，方便后期项目以及代码管理。

3 系统总体方案确认

3.1 系统总体框架

基于百度飞桨的遥感图像智能解译平台是针对传统的遥感图像解译平台图像解译效果不佳、缺乏后续的数据分析等一系列问题所开发的一个平台。平台所具有的主要功能有：登录注册、图像上传、图像解译与可视化数据分析、结果下载、查询记录，意在帮助用户更好地获取遥感图像的信息，对城市规划、用地监测、环境保护具有一定的参考价值。平台的整体框架如下：

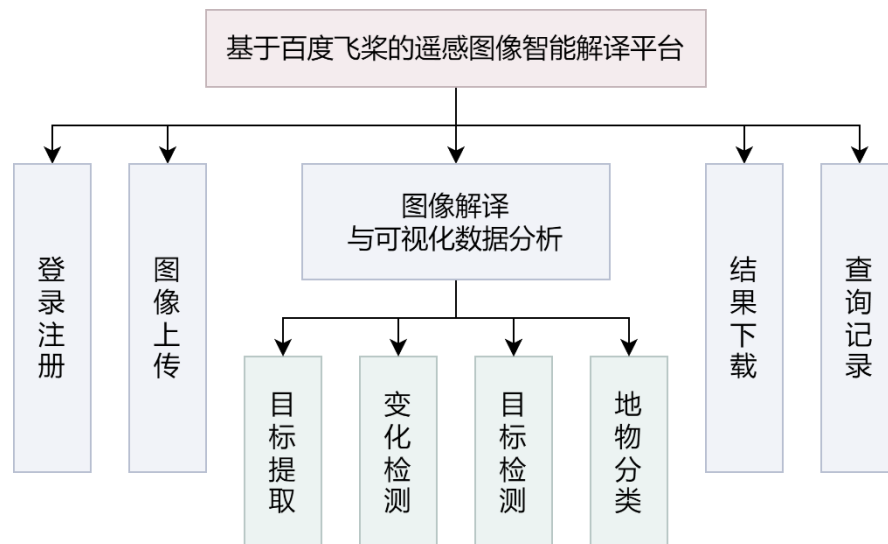


图 1 系统结构图

3.2 系统模块划分

本遥感图像智能解译平台主要划分为以下几个模块：

1) **登录注册**：实现用户信息的录入及身份验证，用户既可以不经登录注册直接使用平台在线解译，也可登录注册后进行图像解译，以便查询历史的解译记录。

2) **图像上传**：将用户的本地图片上传至服务端，并以 url 的形式返回客户端显示，用户可以在页面对图像进行移动、缩放等基本操作。

3) **图像解译和可视化数据分析**：将用户上传到服务端的图片，按照用户需求进行裁剪，将部分或全部图像运用用户选择的算法进行解译，并将结果从服务端传送至客户端。根据算法的不同，本模块可细分为目标提取、变化检测、目标检测和地物分类四个子模块：

① **目标提取**：运用目标提取算法对图像进行解译，将道路区域标亮，同时计算道路的覆盖率，评估路网的规模、通达性及与城镇化的匹配度，生成雷达图并得出路网的建设水平。

② **变化检测**：运用变化检测算法对图像进行解译，将建筑的变化区域标亮，计算建筑用地的变化率，评估城市扩张的范围和强度，生成折线图展示城市化程度。

③ **目标检测**：运用目标检测算法对图像进行解译，将检测出的运动场区域框出，评估该项城市资源的分布范围和分配密度，生成雷达图并得出资源分配的合理性。

④ **地物分类**：运用地物分类算法进行解译，将不同区域用不同颜色标出，计算建筑、道路和林地的覆盖率，用饼状图直观展示各区域所占比率，评估用地规划的合理性和土地利用程度。

4) **结果下载**：将显示在客户端上的图片下载到用户本地。

5) **查询记录**：实现已登录注册的用户对其解译记录的查询，从数据库调取相应的记录返回客户端。

3.3 应用系统和支撑系统划分

应用系统分为前端和后端，支撑系统作为一个模块嵌入到后端中。

1) 应用系统前后端数据交互通过 Axios 请求和 Json 数据格式来实现，后端为 Falsk 框架，用于接收并返回前端所需的数据；

2) 用户进入前端界面，通过请求与后端和数据库进行交互，实现解译算法的选择，图像的上传、解译，结果的下载与查询等操作；

3) 支撑系统为 PaddlePaddle 与 PaddleRS，内嵌到应用系统后端中，实现与应用系统的对接。应用系统接收到用户上传的待解译图像和范围后会传 PaddleRS 中根据相应的算法进行解译，所得到的结果将记录入数据库并返回给应用系统前端；

4) 应用系统后端进行数据分析并与前端数据可视化板块进行对接，将数据分析结果可视化的呈现在页面上。

4 系统详细设计

4.1 网络端程序详细设计

说明：由于目标提取、目标检测、地物分类三大功能可参考对应的套件，本文档仅针对变化检测功能展开论述。

4.1.1 基于深度学习的变化检测简介

利用深度卷积神经网络进行变化检测，使用人工标记好的数据对网络进行大量的训练，使神经网络自动学习遥感图像中的相应特征并进行比对，从而得出前后两个时相的遥感图像的变化区域。在本平台的使用中，用户通过上传图像，对图像的解译区域进行选择，后端将待解译的全部或部分图片保存并传入已训练好的神经网络模型中进行解译，得到带有标记的已解译图像。

4.1.2 实验环境

基于 Python 有封装好的神经网络函数库，PaddlePaddle 框架的高性能与 PaddleRS 特有的遥感图像处理模块，我们选择其作为语言和框架基础，并在 AI Studio 上对模型进行训练。

	具体描述
代码语言	Python
深度学习框架	PaddlePaddle, PaddleRS
GPU	NVIDIA Tesla V100 32GB
CPU	Intel(R) Xeon(R) Gold 5320 CPU @ 2.20GHz

4.1.3 数据集

本平台网络端程序进行深度学习训练时采用的遥感图像建筑变化检测数据集由第十一届 “中国软件杯” 大学生软件设计大赛遥感解译赛道提供。

4.1.4 模型介绍

本平台网络端程序进行深度学习训练时采用的模型参考自第十一届 “中国软件杯” 大学生软件设计大赛遥感解译赛道提供的基线。

4.1.5 模型优化

1) 模型骨干网络更改与数据增强

由于 resnet 的层数越多，深度越大，拟合度越高，因此我们将模型的骨干网络从 resnet18 更改至 resnet34。为了避免过拟合的发生，我们采用随机交换、随机模糊、随机水平翻转、随机垂直翻转对数据集进行增广处理，以获得更多的数据支撑训练。

2) 优化器算法与学习率衰减策略

使用 Momentum 优化器代替 Adam 优化器，收敛速度较慢但结果更加精确。设置 `weight_decay=0.0005` 和 `momentum=0.99`，采用自定义步长衰减策略，使学习率下降更加缓慢，利于模型的收敛。

3) 注意力模块和视觉野

将 Transformer 编码器使用的注意力模块个数从 1 增加至 2，将 `cropsizes` 从 256 增加至 512 以增强视觉野，使神经网络对图像的特征学习更加准确。

4.1.6 模型训练与推理

1) 数据预处理

解压数据集并按比例划分为训练集和验证集，将划分数据集训练集与验证集的比例从 0.95 增加到 0.985，增加数据量减少因增加 resnet 深度导致的过拟合。

2) 模型构建

使用 PaddleRS 构建数据集与模型。使用自定义步长衰减策略与构建 Momentum 优化器使模型效果更好。

3) 模型训练

使用 AI Studio 中 Tesla V100 环境配置训练模型，增加训练回合，使模型最后能达到收敛。

4) 模型推理

使用与模型训练同样的环境配置进行推理。将模型推理的固定阈值从 0.5 更改至 0.41，使变化的像素点识别更加准确；采用滑窗推理，使模型适用于不同大小的图像，提高模型性能。

4.1.7 模型部署

1) 动态图转静态图

使用 PaddleRS 中的 `export_model` 工具将推理得到的动态图模型转换为静态图模型，提高推理效率。

2) 图像尺寸拟合

基于 `paddlers.deploy.Predictor` 构建新的 `Predictor` 类，将输入的图片通过裁剪、分割、填充操作拟合模型的输入尺寸，分别推理后重新组合成完整结果。

3) 结果处理

将 `Predictor` 推理得到的结果进行分析，并依此绘制图像，返回便于服务器处理的数据格式以作进一步的数据分析。

4.1.8 性能评价

变化检测模型的优劣可用平均交并比 $mIoU$ 与精确率和召回率的调和平均值 $F1$ 来衡量， $mIoU$ 与 $F1$ 的定义如下：

$$mIoU = \frac{1}{2} \left(\frac{TP}{FN + FP + TP} + \frac{TN}{FP + FN + TN} \right)$$

$$F1 = \frac{2 \cdot TP}{2 \cdot TP + FN + FP}$$

其中， TP 表示预测为变化且实际为变化的样本数， TN 表示预测为不变且实际为不变的样本数， FN 表示预测为不变但实际为变化的样本数。

经过训练，我们模型的 $mIoU$ 变化如下：

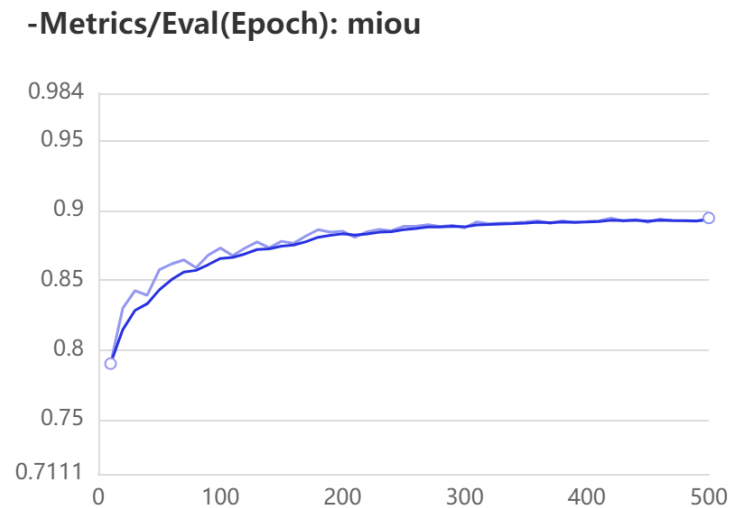


图 2 mIoU 变化图

$F1$ 分数达到 0.86755，模型的准确度较高。

4.1.9 效果展示

模型训练的结果展示如下：

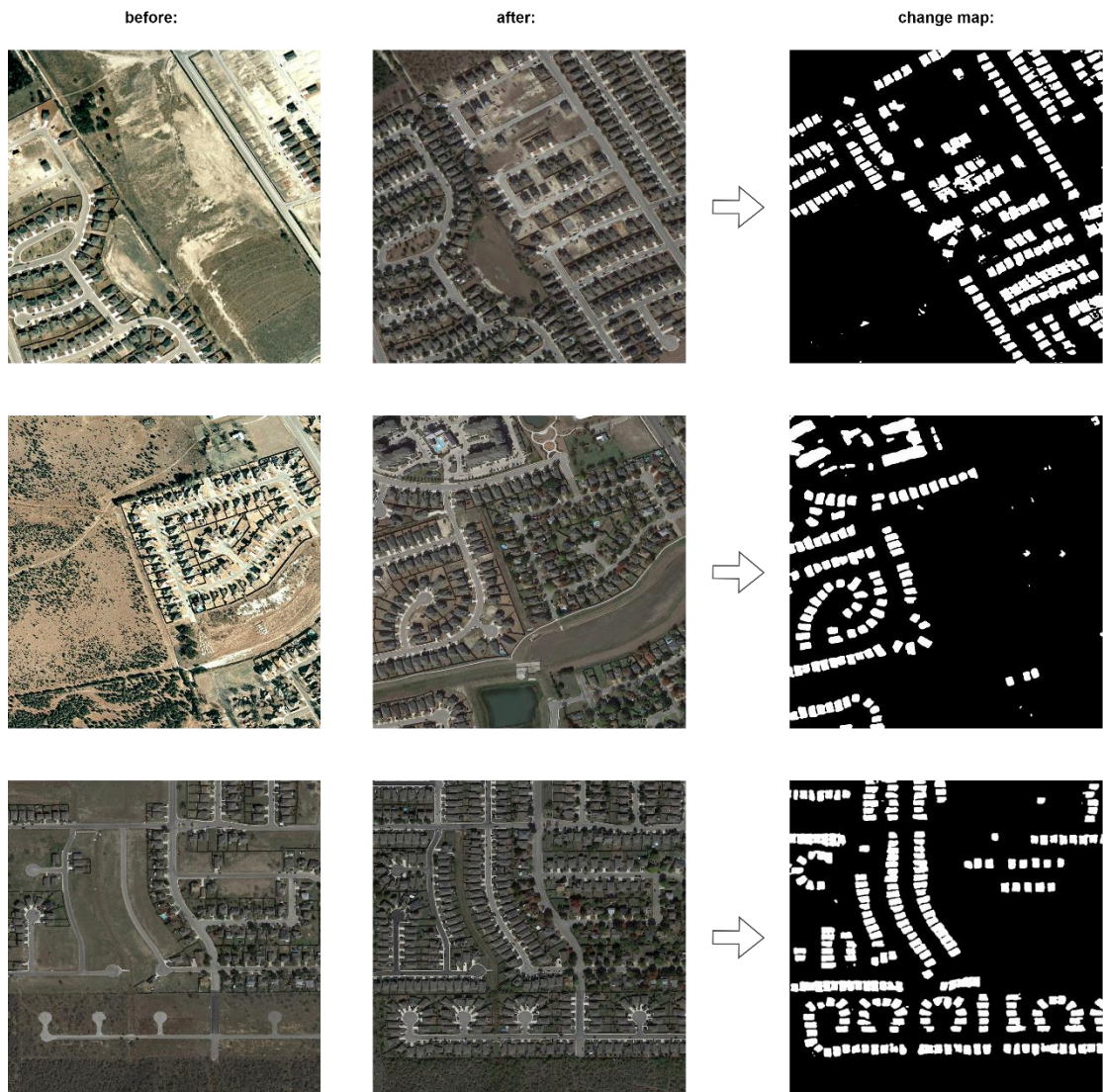


图3 训练效果图

4.2 平台详细设计

平台所使用的技术如下图所示：

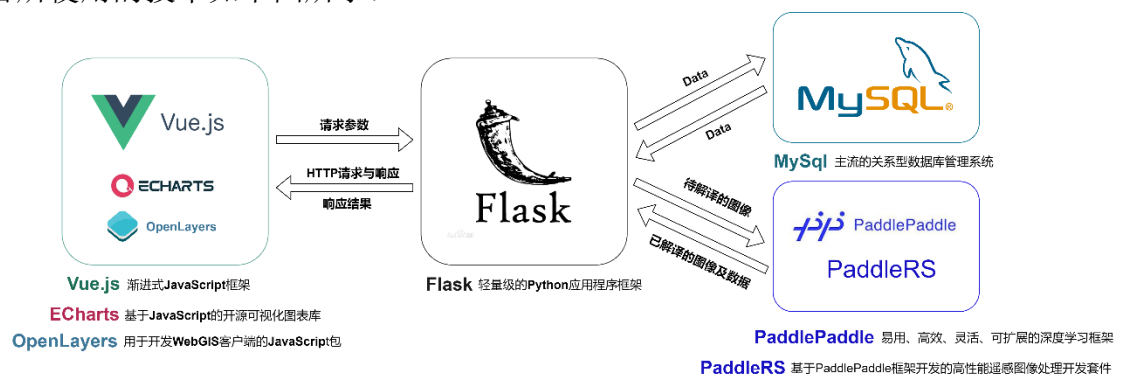


图4 使用技术图

4.2.1 前后端数据交互设计

说明：鉴于本平台的主要功能是遥感图像的智能解译，故相比于传统的 web 通信方式以及后端搭建，本系统采用简单方便的 Axios 以及轻量级的 Flask 进行数据通信交互，减少解译结果返回所需时间一次加快图像解译过程。

Axios 简介：

Axios 是一个基于 promise 的 HTTP 库，可应用在浏览器和 node.js 中，以实现网络请求的便捷发送。Axios 本质上是将 XMLHttpRequest 请求进行了二次封装，并支持 Promise API，符合最新的 ES 规范，便于开发人员更好地处理异步任务的执行结果以提高代码的可读性；它支持自定义拦截的请求和响应，便于开发调试；对浏览器的兼容性良好，可支持 IE7 等老版本浏览器；它还支持防御 XSPF 攻击，提高系统的安全性及健壮性。

Flask 简介：

Flask 是一个使用 Python 语言编写的轻量级可定制框架，较同类型的框架更为轻巧、简洁。该框架的环境部署简单，运行时不需要借助任何其他软件，只需安装 Python 的 IDE 即可；对于运行时所需的相应环境，只需要在 Python 中导入相应的包即可满足需求；它的配置自由、灵活、可拓展性强、第三方库的选择面广，便于开发人员更好地结合实际情况进行开发；相比起经典的 Python Web 框架 Django，Flask 在灵活性和自由度方面都占有很大的优势。

服务端操作：

- 1) 创建 Flask 服务等待客户端的请求；
- 2) Flask 服务接收客户端的请求并解析，获得请求参数，处理业务后返回结果给客户端；
- 3) 客户端关闭后断开 Flask 的连接。

网络端操作：

- 1) 网络端程序集成在服务端中，随服务器同时启动；
- 2) 服务端需要进行遥感图像解译时会待解译的图像传入网络端，网络端进行解译与数据分析后将解译后的图像及数据返回给服务端；
- 3) 等待下一次解译。

客户端操作：

- 1) 用户使用客服端访问服务端;
- 2) 客户端发起 http 请求并接受服务端响应数据解析后展示给客户;
- 3) 客户端关闭后断开连接。

4.2.2 平台功能详细设计

1) 登录注册模块

为提升用户体验,平台设计了该模块用于认证用户以便用户对解译数据的个性化管理。当用户从首页进入遥感图像解译界面时,若 `sessionStorage` 中留有 `token` 信息,则该用户的解译操作会携带此 `token`,便于将解译记录写入数据库;若 `token` 失效,用户则需要登录。用户将在登陆注册界面输入手机号及密码进行登录请求,对于尚未注册的用户,则需要先进行注册以将用户信息写入数据库,前端首先会对手机号及密码的长度进行检验,当数据符合要求后方可发送登录请求至后端。后端验证手机号于密码正确后,生成登录凭证 `token` 并返回给前端。前端将后端返回的 `token` 信息存入 `sessionStorage` 中,用于后续携带 `token` 的请求发送。

用户可通过界面显示的按钮自行决定是否登出。当点击登出后,后端会清空该 `token`,使之无效;前端则会清除 `sessionStorage` 中的 `token` 信息。

2) 图像上传模块

该模块主要实现将用户的本地图像上传至服务端进而进行解译。用户首先在路径选择窗口选择或拖动将要解译的图像至图像上传的板块,前端将根据用户选择的算法对图像的数量进行检验,当数量符合要求后将图像发送至后端。后端接收到图像后将其保存在服务端并为每个图像生成一个 `url` 返回给前端。前端利用后端返回的 `url` 在页面上显示图像,同时使用 `openlayers` 加载图像,以便对图像进行放大、缩小、移动、绘制等操作。

3) 图像解译和可视化数据分析模块

该模块分为图像解译和可视化分析两个部分。用户上传图像至服务端后,在客户端选择部分或全部图像进行解译,此时若用户通过登录认证,在发送解译请求的同时将携带 `token` 以便对解译记录进行保存。后端将待解译的图像传入与用户选择的算法对应的已训练好的神经网络模型中进行解译,在解译过程中获取有用的信息将通过 `Topsis` 等数学模型构建的评价体系做进一步的数据分析,提取出相应的指标数据与解译后的图像 `url` 一同返回给前端。前端将图像

渲染至页面显示，并使用 ECharts 将指标数据绘制成可视化图表供用户参考。

4) 结果下载模块

该模块主要实现将客户端上显示的解译后的图像下载至本地。用户通过前端发送下载请求，后端接收到请求后根据前端请求参数中的图像 url 返回相应的图像文件给前端。用户在客户端的路径选择窗口选择相应的路径对图像文件进行保存，以便进一步的分析处理。

5) 查询记录模块

平台提供了查询记录模块以便用户对已解译数据进行管理。基于用户已登录认证的前提，用户将通过客户端发送携带有 token 的请求至后端。后端接收到用户的请求内容后，对 token 进行解析以识别用户的身份，从数据库中调取相应的记录返回给前端。前端接收到解译记录后将其以列表的形式展现，用户可选择相应的记录进行查看，前端将在页面渲染相应的图像及图表，避免了不必要的再次解译。

4.2.3 平台界面详细设计

1) 首页



图5 首页

2) 图像处理页

① 用户引导

平台采用 Vue-tour 对用户进行引导，提供用户相应的操作提示

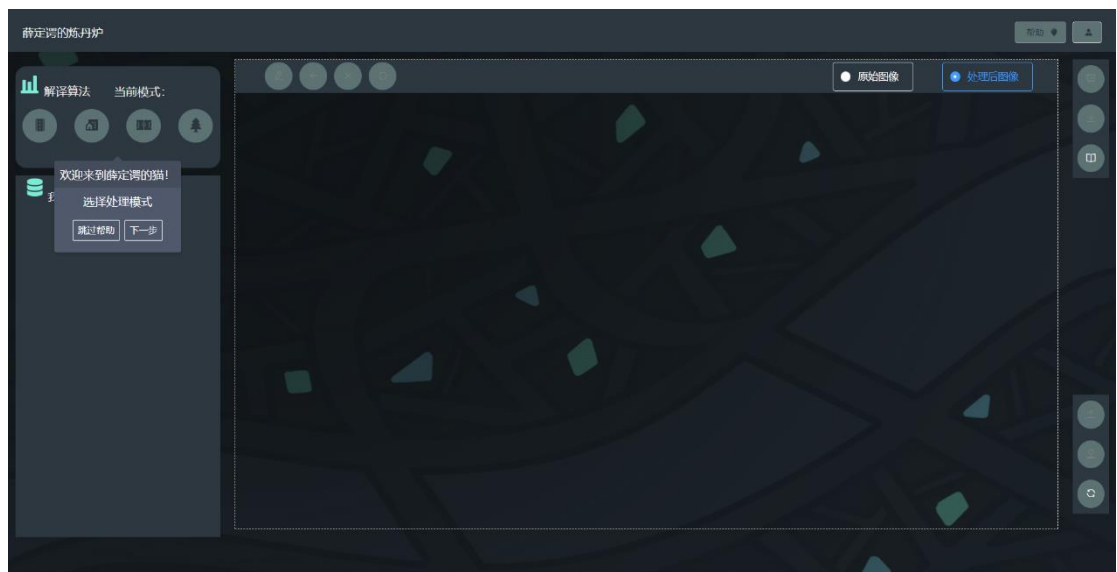


图 6 用户引导

② 选择解译算法

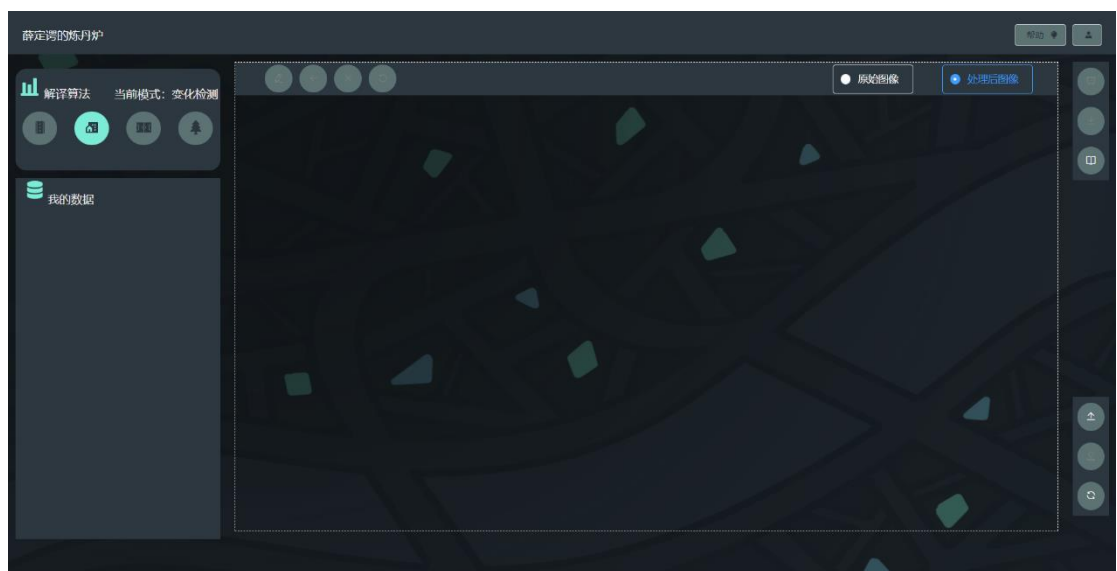


图 7 选择解译算法

③ 上传图像

格式与数量正确，上传成功：

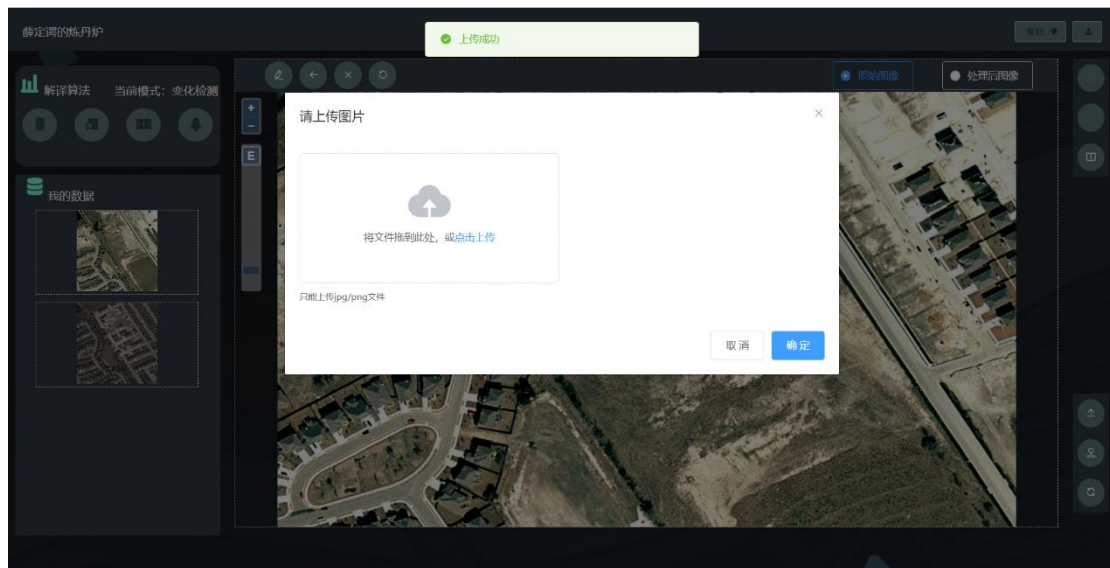


图 8 上传图像成功

格式或数量错误，上传失败：

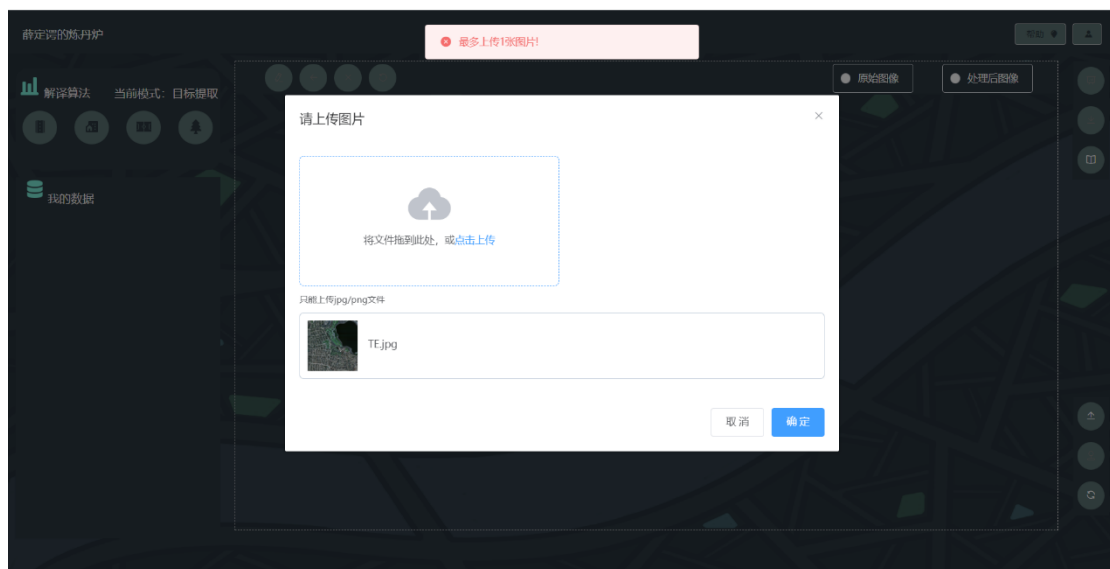


图 9 上传图像失败

④ 解译图像

解译完成：

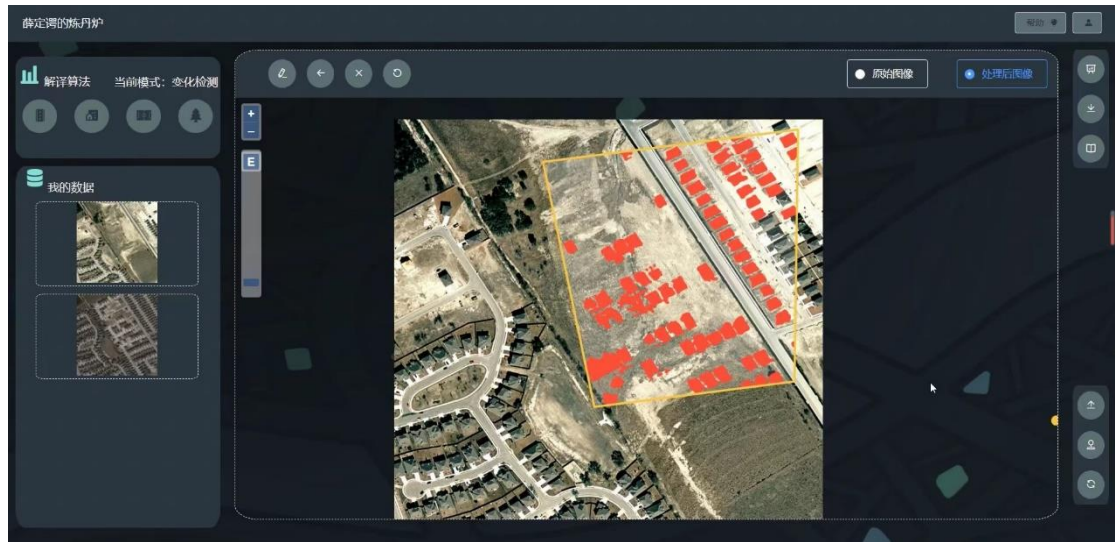


图 10 解译完成

查看图表：

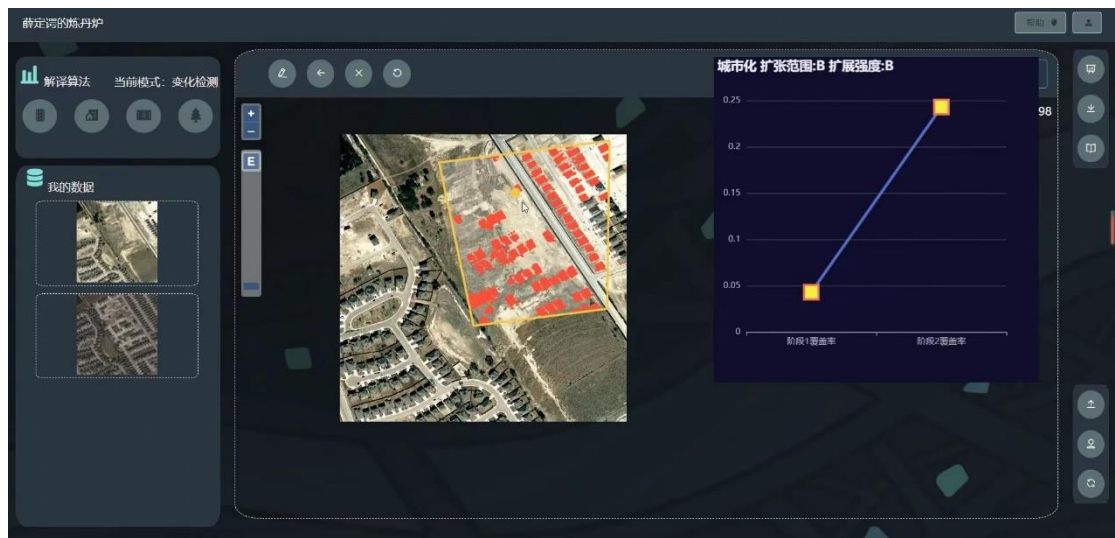


图 11 查看图表

⑤ 下载结果

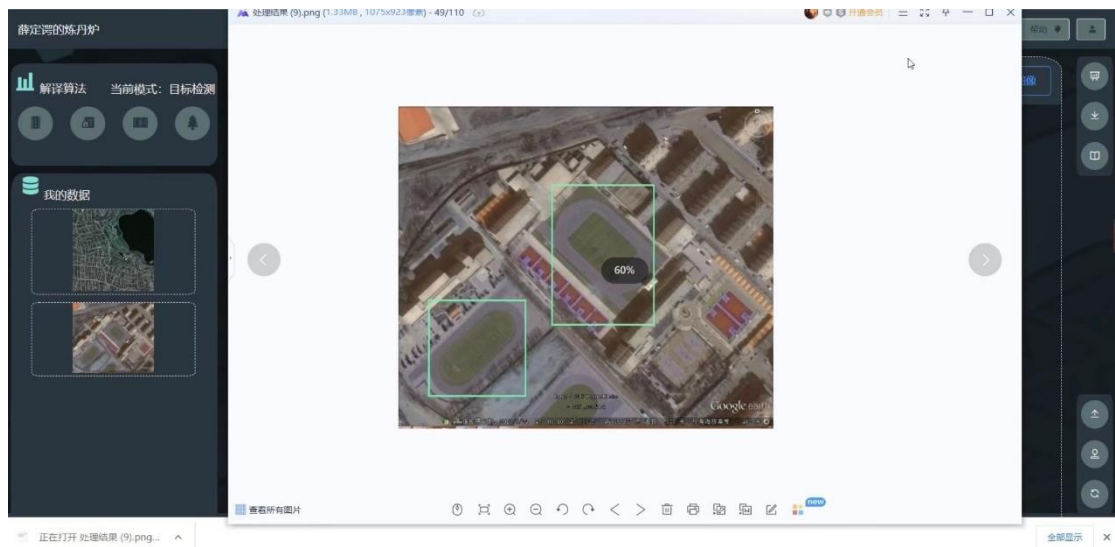


图 12 下载完成

⑥ 查看解译记录

已登录，查看成功：

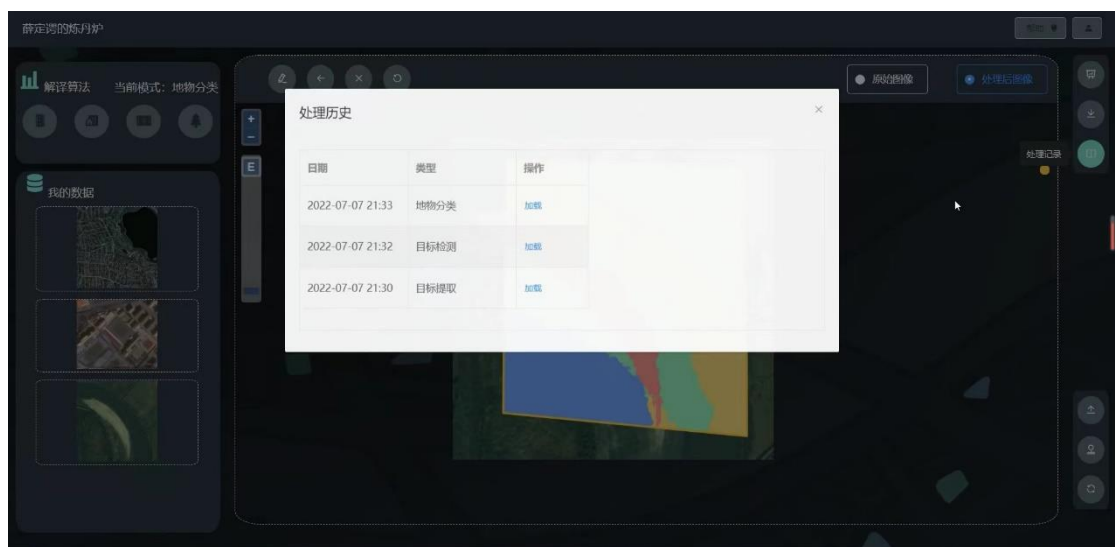


图 13 查看记录成功

未登录，查看失败：

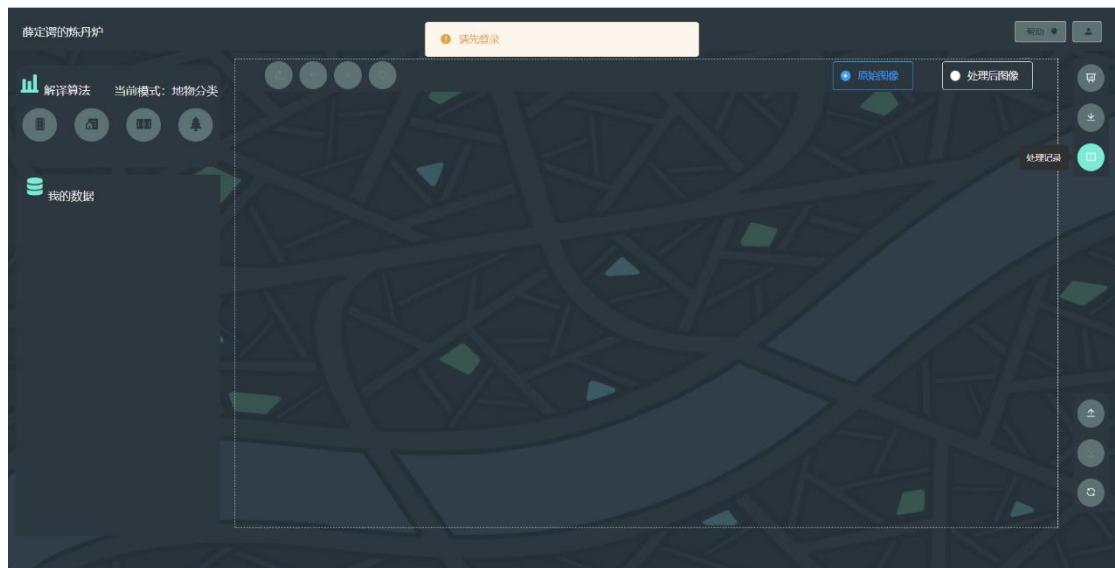


图 14 查看记录失败

3) 登录注册页

信息正确，注册/登录成功：

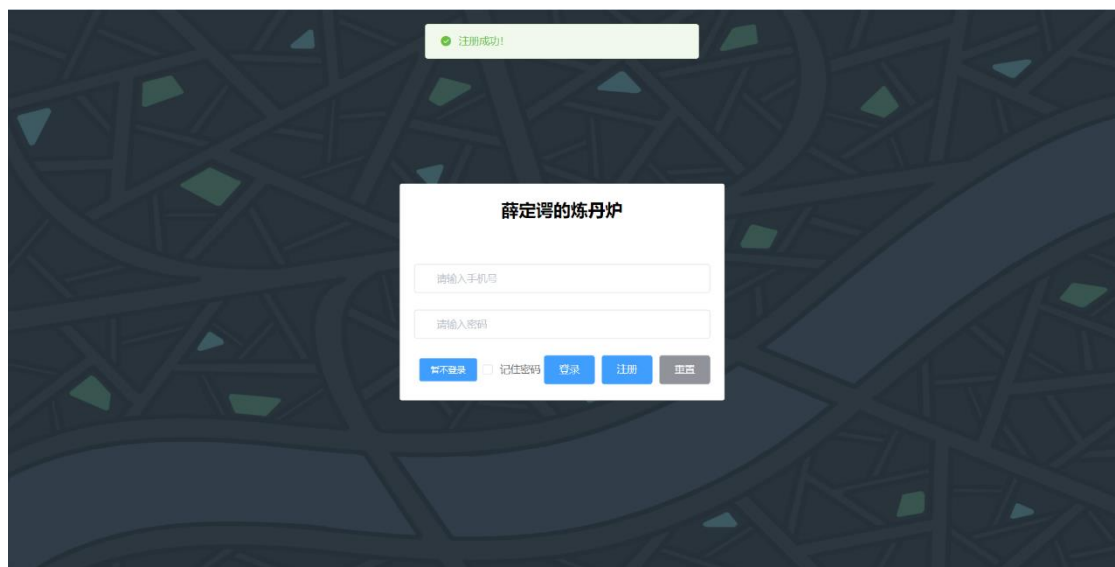


图 15 注册成功

信息错误，注册/登录失败：

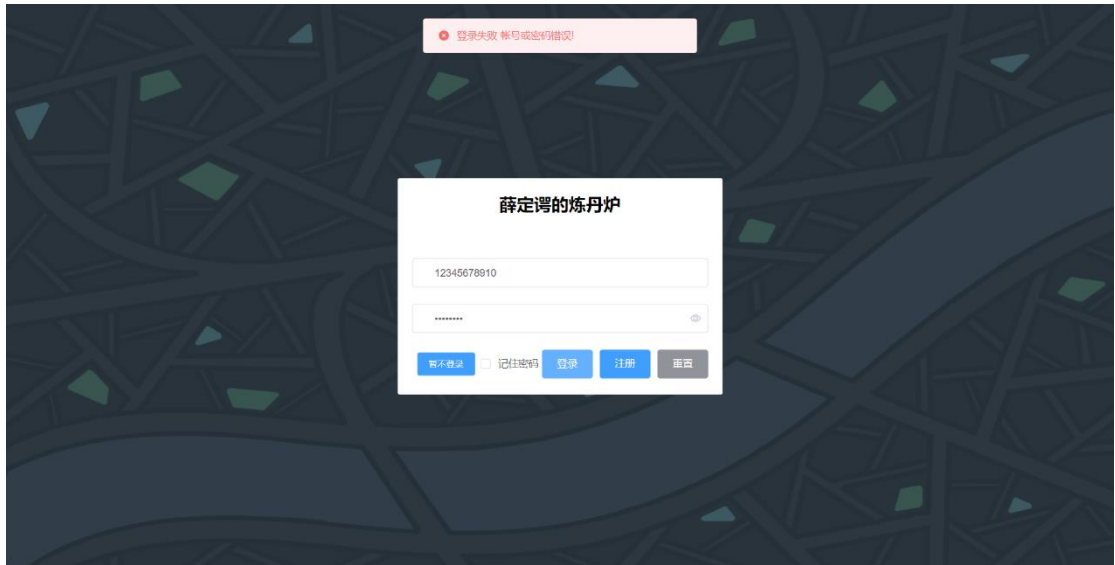


图 16 登录失败

4.2.4 数据库设计

本平台采用 MySQL 关系型数据库对数据的进行保存。。

MySQL 简介：

MySQL 是目前最主流的、使用次数最多的关系型数据库管理系统之一，支持多线程，具有体积小、速度快、性能高、使用灵活、免费开源等优点。

1) 数据库选型

平台使用关系型数据库——MySQL，其支持部署在 Windows、Linux 等操作系统上。

2) 数据库逻辑结构

列举出所使用的每个数据项、记录和文件的标识、长度及它们之间的相互关系，如下表所示：

user 表（用户表）：

序号	属性	类型	长度	备注
1	id	Integer	11	后台自动分配 id
2	name	String	20	用户名
3	telephone	String	20	手机号
4	password	String	50	密码

record 表（记录表）：

序号	属性	类型	长度	备注
1	id	Integer	11	后台自动分配 id
2	user_id	Integer	11	用户 id
3	type	Integer	11	解译类型
4	image1_url	String	100	第一张原始图像的 url
5	image2_url	String	100	第二张原始图像的 url
6	result_url	String	100	解译后的图像 url
7	index	String	100	解译后的指标数据
8	time	Time	20	解译时间

5 关键技术

5.1 模型优化方法

5.1.1 网络结构改进

将变化检测模型使用的骨干网络从 resnet18 更改成 resnet 34，增加 resnet 深度，提高模型拟合度。

5.1.2 训练技巧改进

- 1) 采用数据增广技术，使用随机交换、随机模糊、随机水平翻转、随机垂直翻转来扩大数据集，防止数据集较小导致过拟合；
- 2) 学习率采用自定义步长衰减策略，减慢学习率衰减速度，助于模型收敛；
- 3) 将 Transformer 编码器使用的注意力模块数从 1 增加至 2，提高模型特征学习的准确度；
- 4) 将 cropsize 从 256 增加至 512 以增强视觉野，提高模型性能。

5.2 数据分析可视化实现

5.2.1 指标评价体系的构建

将数学建模方法应用于遥感数据的分析，利用 AHP-TOPSIS 等模型构建评价体系，将解译过程中得到的数据提炼整合成更具有价值的信息，为可视化提供数据支撑。具体步骤为：

- (1) 将图像分割成多个样本，提供更多参考以提高准确度
- (2) 将解译过程中得到的原始信息，如像素点的覆盖、分布、密度等参考相应文献抽象成原始的指标数据，利用层次分析法确定各指标的权重；
- (3) 依次对原始的指标数据矩阵进行正向化、标准化以消除指标类型、量纲的影响，将权重加入计算各原始指标的得分并进行归一化得到最终结果。
- (4) 基于解译过程得到的数据可能存在的不完整性，采用三次埃尔米特插值算法对缺失数据进行补充，提供足够的数据支撑分析。

5.2.2 可视化图表的渲染

将分析得出的数据利用 ECharts 进行前端渲染，通过雷达图、折线图、饼状图等方式将分析的结果可视化地展示到页面上，提高解译数据的可读性与使用价值。

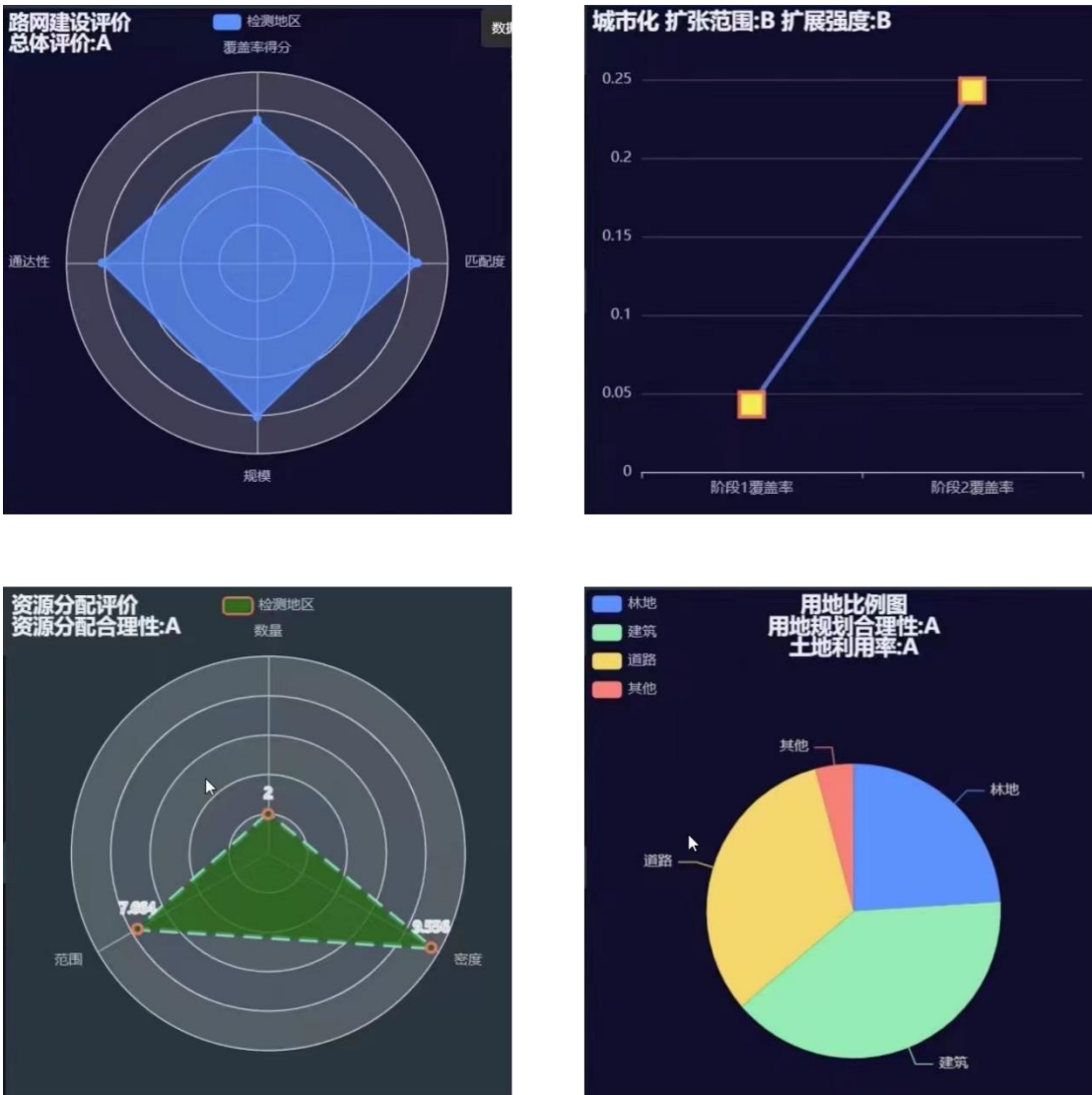


图 17 可视化图表的渲染

5.3 图像操作功能实现

将用户上传的遥感图像用 OpenLayers 加载显示，通过多种地图控件的运用，使用户能方便地对图像进行放大、缩小、移动，并且能够通过绘制区域划定合适的解译范围，提高解译的针对性与精确性。