**概要设计文档**

**仰望星空团队**

目录

[**1 实现的场景及功能** 2](#_Toc108184367)

[**1.1 登录** 2](#_Toc108184368)

[**1.2 注册** 2](#_Toc108184369)

[**1.3 单文件推理（以变化检测为例）** 3](#_Toc108184370)

[**1.4 批量推理（以变化检测为例）** 4](#_Toc108184371)

[**1.5 查看历史记录** 5](#_Toc108184372)

[**1.6 删除历史记录** 5](#_Toc108184373)

[**1.7 下载批量推理结果** 6](#_Toc108184374)

[**2 总体架构设计** 7](#_Toc108184375)

[**3 视图层相关设计** 9](#_Toc108184376)

[**4 业务层相关设计** 9](#_Toc108184377)

[**4.1 Spring Cloud 框架设计** 9](#_Toc108184378)

[**4.2 Web api设计及调用主体流程** 10](#_Toc108184379)

[**4.2.1单次上传主体流程** 10](#_Toc108184380)

[**4.2.2批量上传主体流程** 11](#_Toc108184381)

[**4.3 数据库设计** 12](#_Toc108184382)

[**4.3.1 oss对象存储** 12](#_Toc108184383)

[**4.3.2mysql关系存储** 13](#_Toc108184384)

[**5 模型层相关设计** 13](#_Toc108184385)

[**5.1 模型训练** 14](#_Toc108184386)

[**5.1.1 变化检测** 14](#_Toc108184387)

[**5.1.2 目标检测** 14](#_Toc108184388)

[**5.1.3 地物分类和目标提取** 14](#_Toc108184389)

[**5.2 模型部署** 14](#_Toc108184390)

[**5.2.1 变化检测** 14](#_Toc108184391)

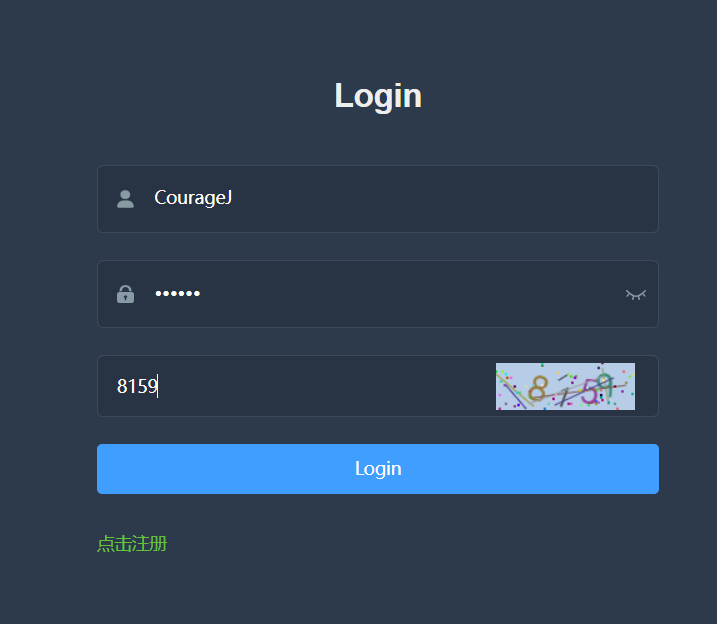
[**5.2.2 地物分类** 14](#_Toc108184392)

[**5.2.3 目标检测和目标提取** 15](#_Toc108184393)

**1 实现的场景及功能**

**1.1 登录**

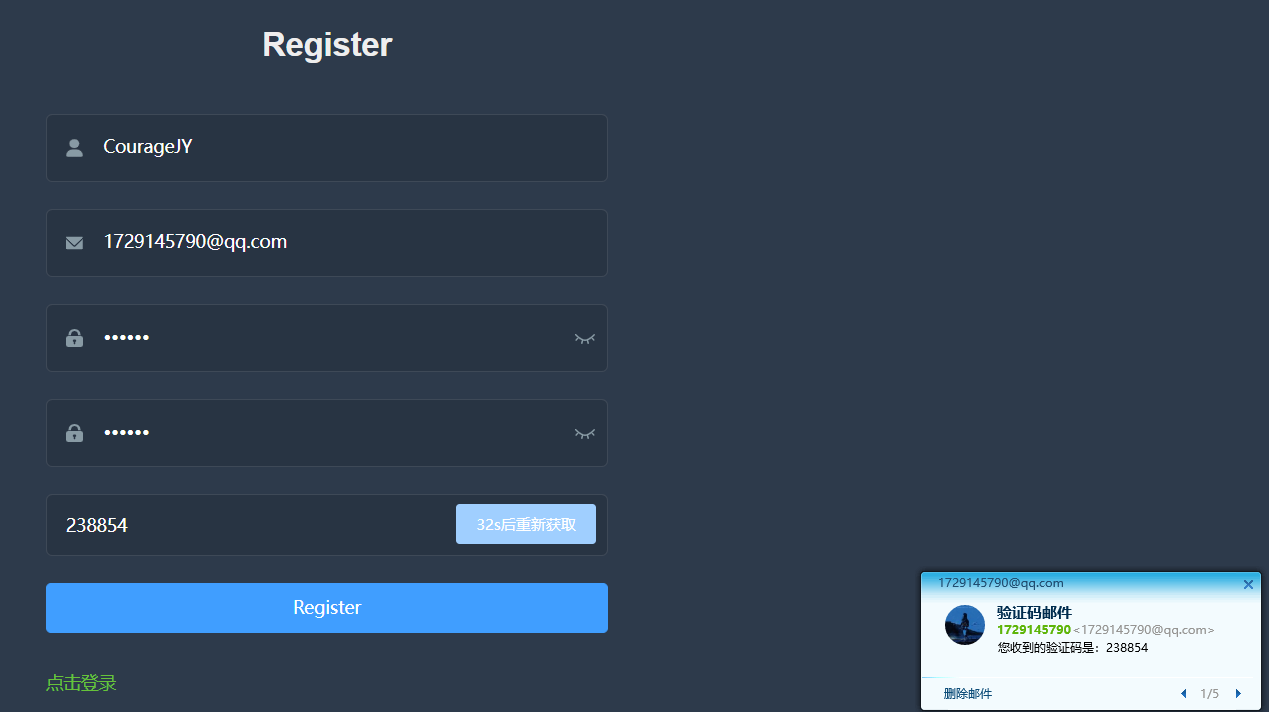
系统首先会进入登陆页面，用户可通过用户名或邮箱进行登录，并通过验证码进行人机验证。



**1.2 注册**

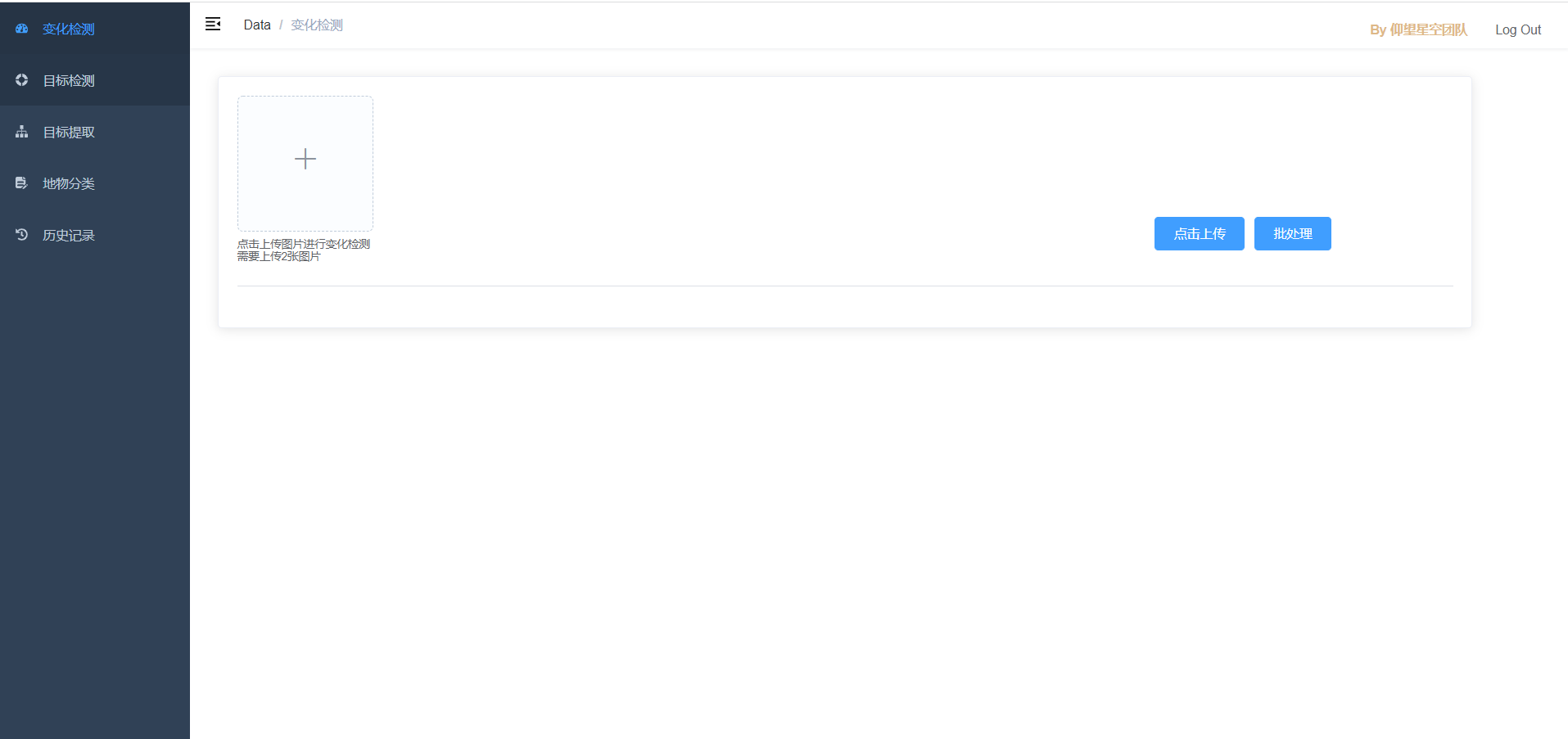
用户通过点击登录页面左下角的”点击注册“，可进入注册页面。

在注册页面填写用户名、邮箱、密码、重新填写密码后，用户可以获取验证码，系统会将验证码发送至用户邮箱，用户填写正确验证码后，即可进行注册，如下图所示：

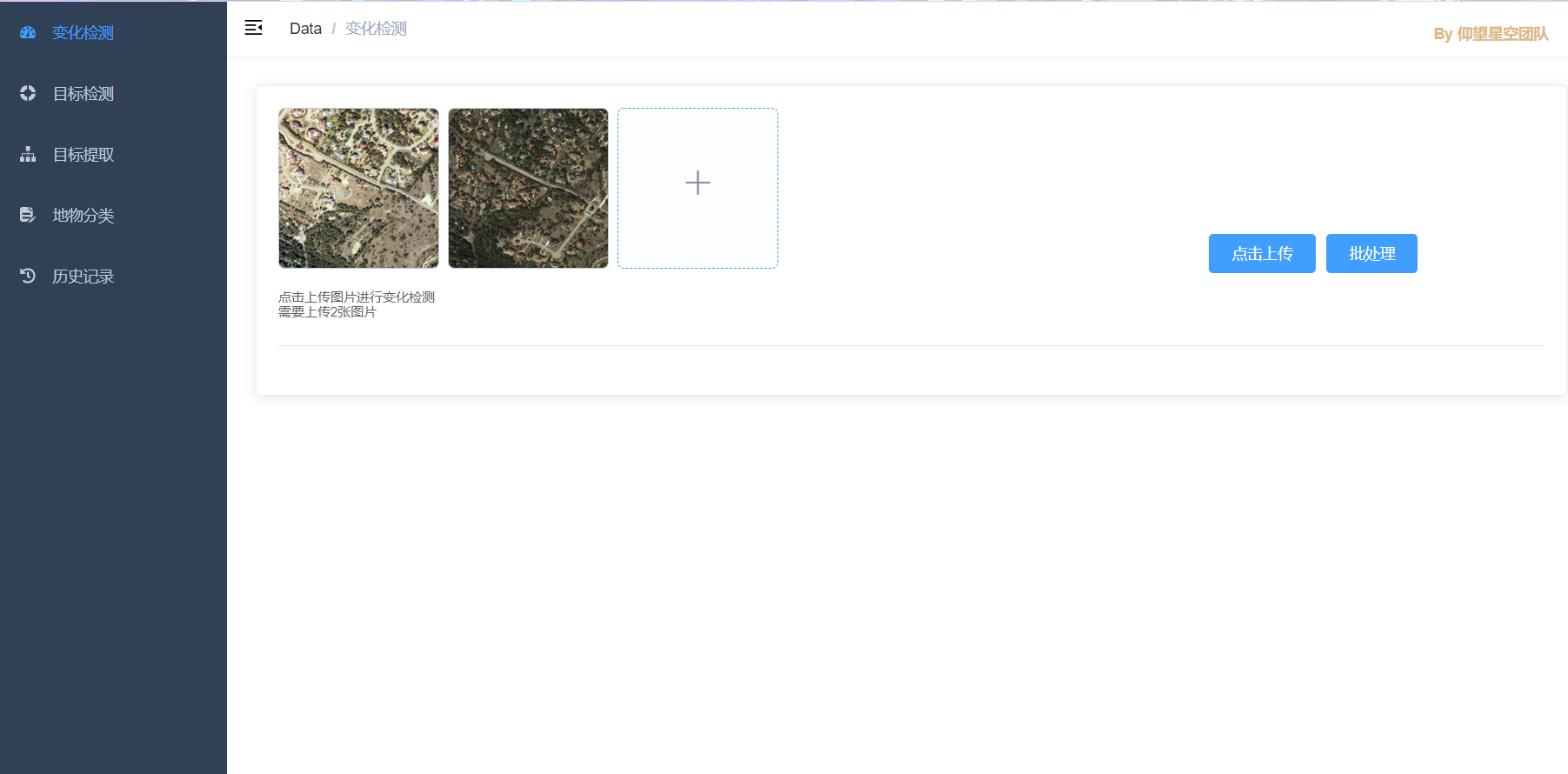


**1.3 单文件推理（以变化检测为例）**

用户进入系统后可以选择自己需要的遥感服务，首先会展示单文件推理的服务，此处以变化检测为例。



用户首先需要点击左上方上传图标上传图像，结果如下图所示：

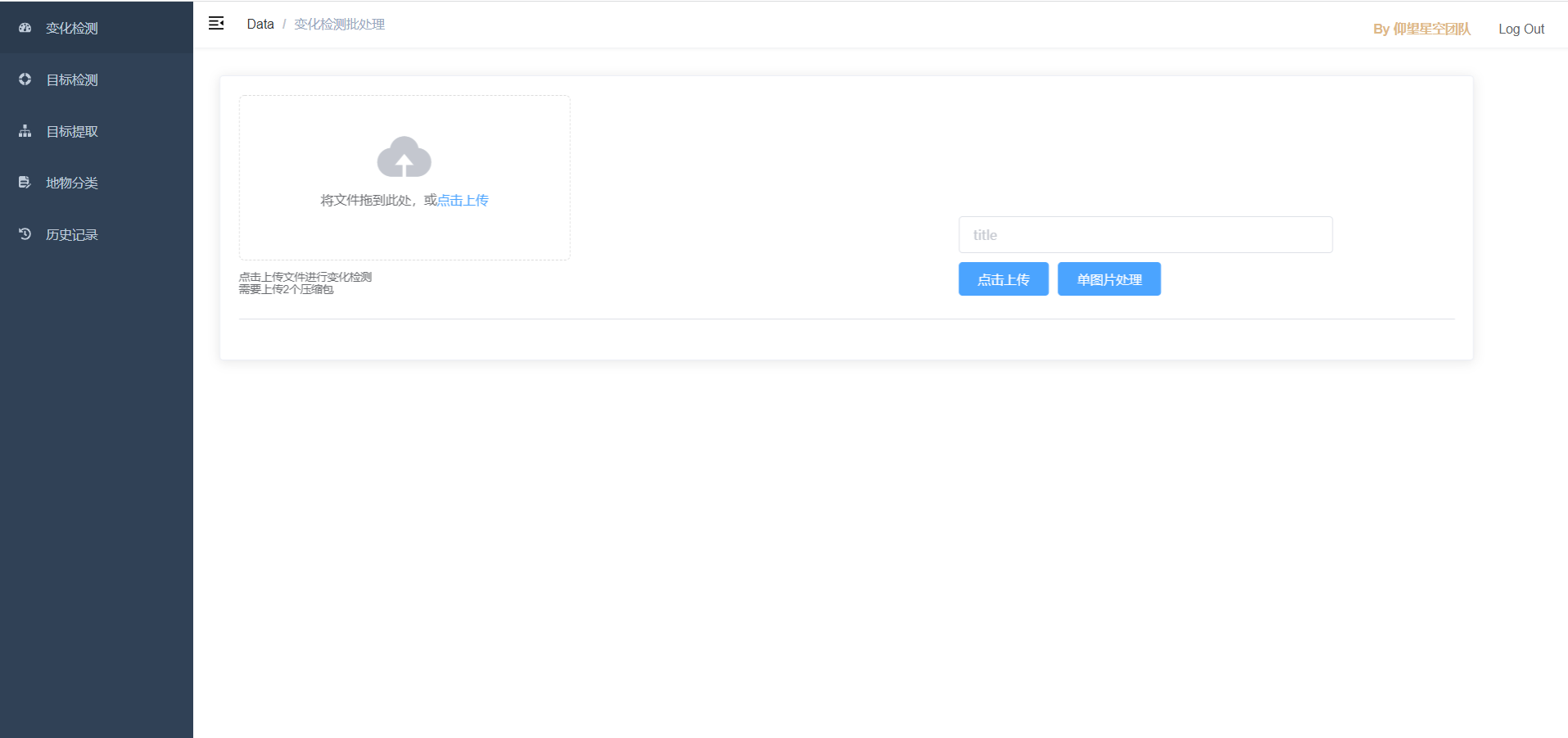


用户点击左上方上传图标上传图像后，系统会自动进行推理，对于变化检测还会自动计算变化率，结果如下图所示：



**1.4 批量推理（以变化检测为例）**

用户点击单文件推理页面中右侧的批处理按钮，可进入批量推理页面，此处以变化检测为例。



用户首先需要根据提示上传压缩包数据，对于变化检测要上传两个压缩包，两个压缩包内对应变化前后的两个图片的文件名必须相同。

用户选择本地的数据集并填写标题后，即可点击上传按钮进行上传：

上传成功后用户可点击处理按钮对数据集进行处理，系统会自动进行批量推理，在推理完成后用户可以在历史记录中对结果进行查看和下载。

**1.5 查看历史记录**

用户可以点击历史记录来查看自己进行过的批量推理的记录，如下图所示：



**1.6 删除历史记录**

用户可以点击删除按钮，对历史记录进行删除，删除前历史记录如下：

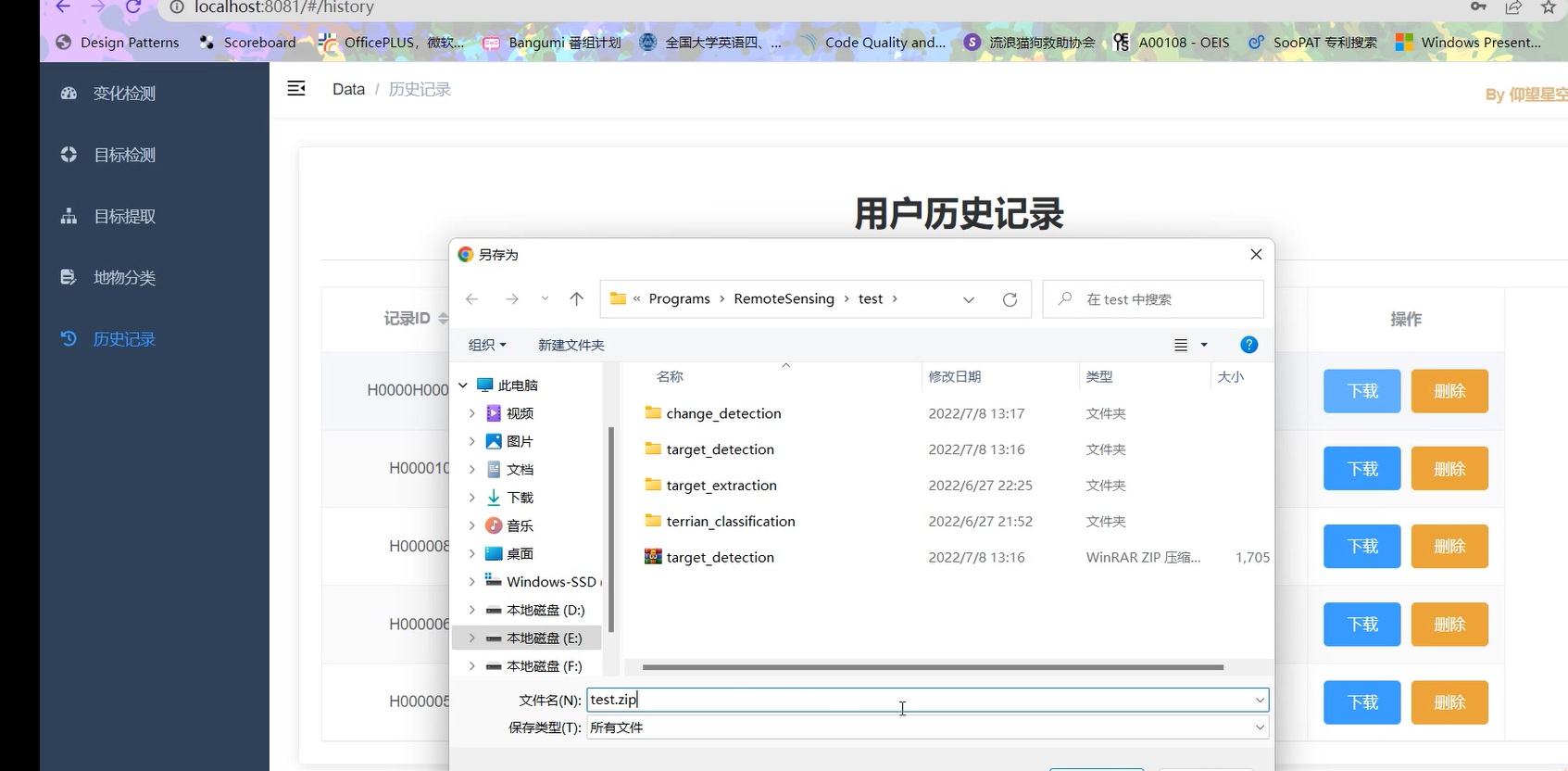


删除 后的历史记录如下：



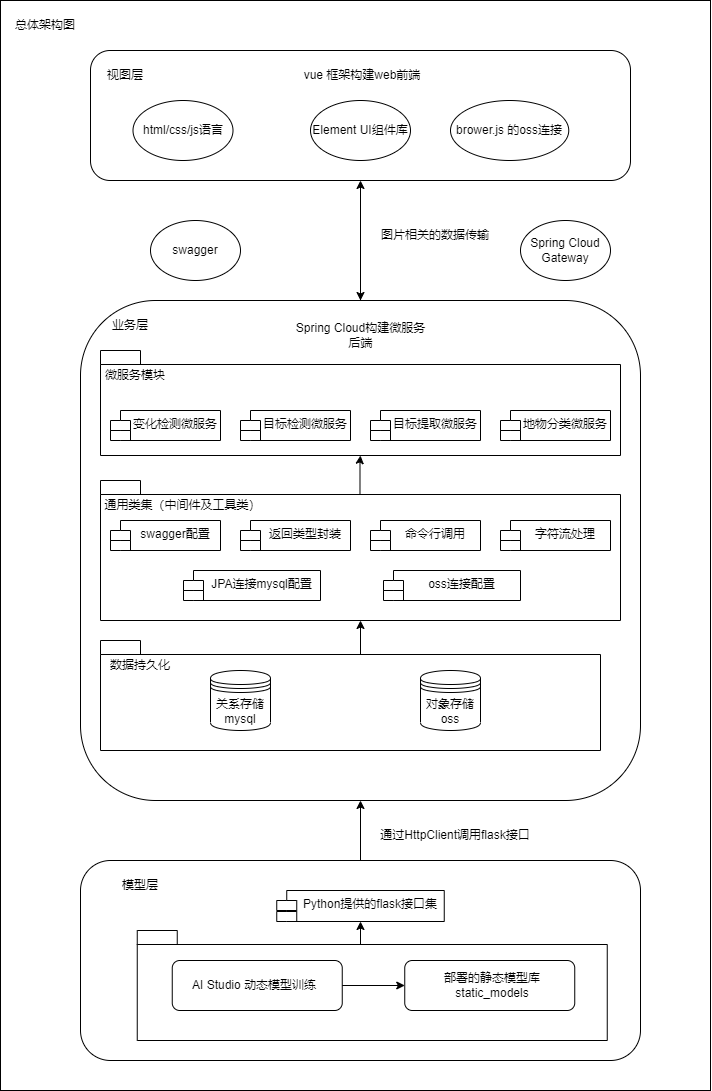
**1.7 下载批量推理结果**

用户可以点击下载按钮，下载对应的推理结果压缩文件，如下图所示：



**2 总体架构设计**

本系统在逻辑上可分为视图层、业务层和模型层，各个层次所使用的技术以及相关设计如下图所示：



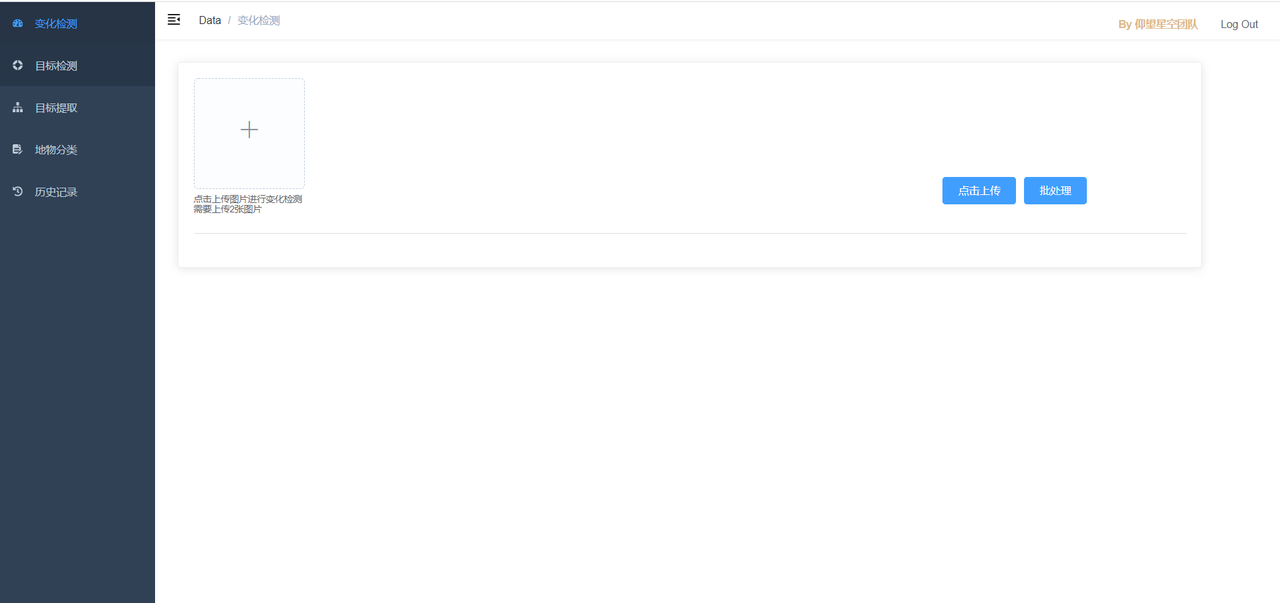
**视图层：**即web网站前端的设计。本系统采用vue框架构建web前端，采用Element UI组件库构建页面组件。在接口方面，除了使用后端Spring boot提供的接口外，还调用了外部oss对象存储的brower.js相关接口以完成对存储对象的上传和下载操作。

**业务层：**即系统总体后端设计。考虑到本系统的四种摇感服务都有着推理时间较长、性能消耗较高的特点，本项目采用Spring cloud构建微服务框架，将四种服务识别为四种微服务以提供服务的分布式部署与管理，并设计了诸如Jpa配置等中间件以及返回类型封装等工具类。此外本项目在功能上实现了单次处理以及批处理的业务需求，考虑到批处理的流程较长，我们使用了阿里云oss对象存储以提供大文件的长期存储，并使用了关系型数据库mysql以实现对用户相关信息的持久化存储。

**模型层：**即四种摇感服务的模型训练与推理过程。训练过程主体在百度AI Studio上完成动态模型训练，推理过程则使用的是部署之后的静态模型。为了避免Java调用python脚本相关类库导入时过长的问题，在模型层我们使用了flask框架以提供业务层所需要的推理接口。

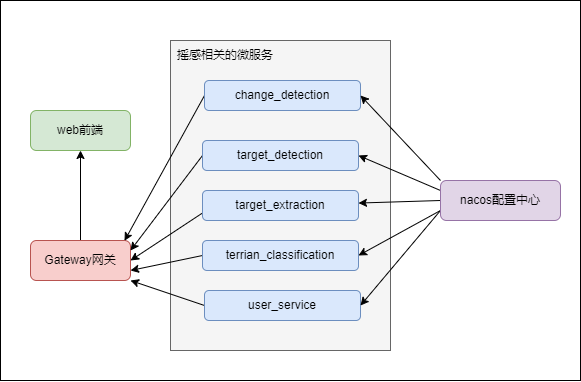
**3 视图层相关设计**

视图层采用Vue + ElementUI开发。风格上沿用传统的后台管理前端风格，侧边栏导航的使用使业务逻辑清晰明了；ElementUI组件库的使用增强了界面的美观性，大大提升了用户体验，视图层主体界面如下：



**4 业务层相关设计**

**4.1 Spring Cloud 框架设计**



考虑到四种摇感需求彼此之间较为独立，且推理模型不易统一，本项目将四种需求划分为了4种微服务，并额外设计了用户微服务以管理用户信息等相关业务。我们使用阿里云的nacos配置中心对4种服务进行了统一的配置管理，然后使用spring cloud gatway完成对网关的统一配置。相关微服务设计说明如下：

**change\_detection：**提供变化检测的单次处理与批处理接口。

**target\_detection：**提供目标检测的单次处理与批处理接口。

**target\_extraction：**提供目标提取的单次处理与批处理接口。

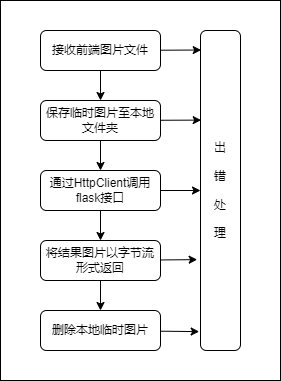
**terrian\_classification：**提供地物分类的单次处理与批处理接口。

**user\_service：**提供用户登录、注册，以及批处理历史记录的管理接口。

**4.2 Web api设计及调用主体流程**

本系统在功能上实现了四种摇感服务的单次处理以及批处理功能，并在后端提供了相应的接口。下面是四种摇感服务的单次上传以及批处理的主体流程设计。（详细代码设计请详见《详细设计文档》）

**4.2.1单次上传主体流程**



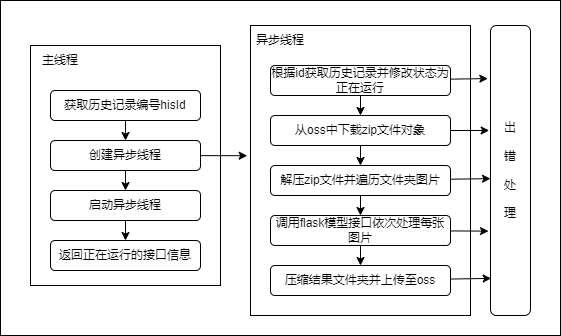
单次处理后端先是接收前端图片传输的POST请求，然后将临时图片保存至本地文件夹，进而通过java的HttpClient工具调用模型层的flask接口完成单次推理，最终将结果图片以字节流的形式返回，并最后删除临时图片。由于主体流程较长，在代码中多处使用了try\_catch\_finally等语句完成了对异常的捕获以及进一步处理，以保证系统的健壮性。

**4.2.2批量上传主体流程**

本项目批量处理需要用户上传待处理图片集的压缩文件夹，在整体上可分为：上传、运行、下载三步。

**上传**：前端利用brower.js的oss接口完成zip压缩包的oss分片上传，并利用后端接口在mysql中新增一条历史记录。

**运行**：前端调用后端的批处理api接口，为了避免运行接口迟迟无法返回（运行时间太长）的问题，此接口使用异步调用的方法分为主线程和异步线程。在主线程中程序根据传入的历史记录编号hisId获取待处理的上传记录，随后创建并启动异步线程，并在不阻塞的情况下返回“正在运行”的接口信息。在异步线程中则完成的是运行的主体逻辑：首先在mysql中修改该条历史状态为“运行中”，然后调用接口从oss中下载待处理zip文件，随后解压zip文件并遍历文件夹所有图片，对待处理图片依次使用flask模型层接口完成推理，最后将结果文件夹压缩并上传至oss并更新该记录的mysql数据库状态为“运行成功”；若在上述过程中程序出现了异常，则会终断线程，更新该记录的mysql数据库状态为“运行失败”并写入出错原因。具体流程可详见下图：



**下载**：用户在历史记录页面点击下载按钮后，前端将根据后端提供的结果文件对象名利用oss接口直接进行分片下载，结果文件以压缩后的zip文件夹形式呈现。

**4.3 数据库设计**

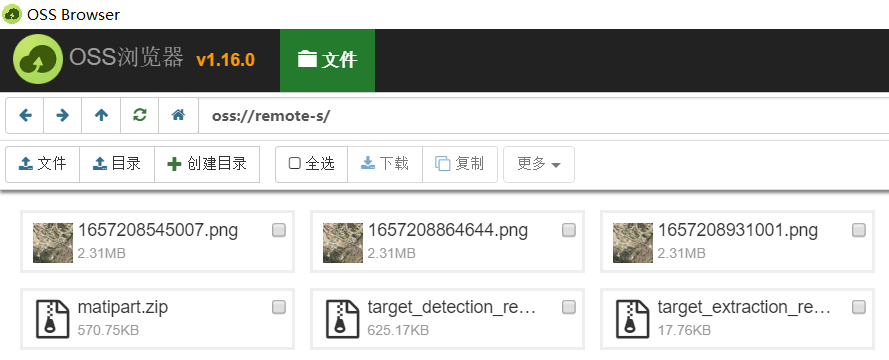
考虑本系统提供的批处理功能中用户往往需要上传较大的待处理图片集（几百M或是几G），本系统设计并使用了阿里云oss的的对象存储来存储用户上传的大文件以及处理之后的结果文件。另外为了配合管理用户批处理历史记录相关信息，我们还设计了关系型数据库mysql用于对用户相关信息的持久化存储。

**4.3.1 oss对象存储**

本系统使用了oss对象存储，并利用了oss提供的大文件的分片上传和下载功能来提高传输速率以及避免文件过大传输失败的问题 。有关oss的相关参数配置如下：

|  |
| --- |
| Properties region:oss-cn-hangzhou endpoint:oss-cn-hangzhou.aliyuncs.com key:LTAI5tFayF7LGkUjbqhPNDu8 secret:rAHGulFmT45UXtiWFJ3QP1xiA33i4m |

对于对象文件还可使用oss的可视化管理工具，如下图所示：



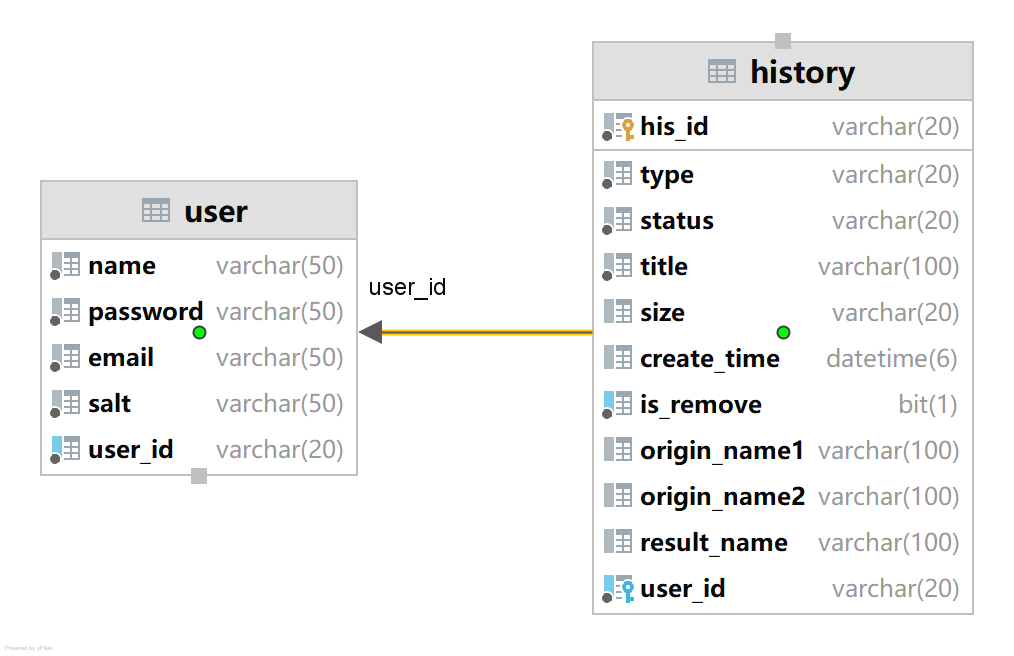
**4.3.2mysql关系存储**

由于本系统有关用户的业务逻辑不太复杂（主要包含用户登录注册以及对于历史记录的管理），我们使用设计了两张表用于存储用户相关信息：用户表user和批处理历史记录表history。

**user**：提供了对于用户id、昵称、密码（salt加密）、邮箱信息的存储。

**history**:提供了对于类型（四种摇感服务）、状态（上传成功、运行中、运行成功、运行失败）、待处理对象名origin\_name、结果对象名result\_name等相关信息。

数据库详细设计如下图关系图所示：



**5 模型层相关设计**

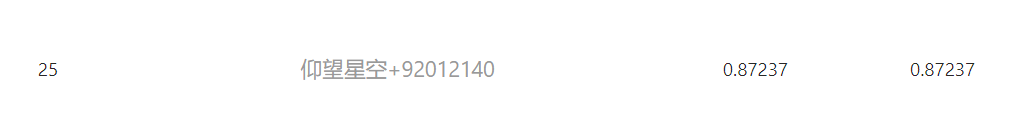
**5.1 模型训练**

**5.1.1 变化检测**

变化检测使用官方的BIT模型进行训练，为提高准确率采取了如下策略：

1. 数据增强：颜色抖动、随机裁剪、随机翻转；
2. 模型修改：采用hrnet18异构高级模型；
3. 超参数调整：修改起始学习率为0.0004、修改优化器为adamw；
4. 后处理调整：采取翻转预测的策略。

最终得分为0.87237，在复活赛榜单中排名第25名。



**5.1.2 目标检测**

目标检测中官方提供代码仅可以检测操场（playground）一类目标，修改代码扩展分类，使得模型可以识别数据集中的操场（playground），飞机（aircraft），oiltank（储油罐），overpass（立交桥）四种目标。

在模型方面依旧选用官方的PP-YOLO模型。

**5.1.3 地物分类和目标提取**

在官方提供算法基础上未作改变，使用DeepLab V3+模型。

**5.2 模型部署**

模型部署参考官方开源代码，先将模型训练得到的动态模型转为静态模型，再构建预测器进行推理。为提高推理精度，还对不同遥感功能进行了滑窗推理、翻转推理等不同尝试。最后对推理得到数据进行分析，以提供数据分析服务，详情如下：

**5.2.1 变化检测**

为提高推理精度，选用翻转推理策略。对于推理结果，计算变化区域和未变化区域像素点个数比例，以计算变化率。

**5.2.2 地物分类**

计算大地区域和其他物体区域像素点个数比例，以计算地物比率。

**5.2.3 目标检测和目标提取**

目前仅进行部署，未提供数据分析服务。