

黑龙江东方学院

毕 业 设 计

设计题目：植物病虫害识别系统设计与实现

姓 名 李琦

学 号 21104085

专 业 计算机科学与技术

班 级 2021 级 2 班

指导教师 任世军、何荣

学 院 信息工程

答辩日期 2025 年 4 月 26 日

植物病虫害识别系统设计与实现

摘 要

在农业现代化进程加速与人口增长的叠加效应下，农作物病虫害防治已成为保障粮食安全的核心议题。全球每年因害虫增长而频频发生的粮食安全问题日益增长。影响了农民生活，以及粮食生产。于是开发了本项目基于 Spring Boot 框架的后端架构与 Bootstrap+JSP 前端技术构建的农业信息平台，通过整合 MySQL 数据库存储的病虫害特征数据与 MyBatis 框架实现的持久化操作，为破解传统防治模式困境提供了数字化方案。

全球主要产粮区因病虫害导致的 20%-40% 年产量损失，暴露出人工经验误判与农药滥用引发的双重危机：前者导致 25% 以上的误诊率，后者造成 38.9% 抽样点的农药残留超标。该平台创新性地集成百度 AI 图像识别引擎，将农情可视化、知识库管理与智能诊断三大模块融合。更高效的将病虫害这一问题解决，帮助农业，以及生产环境提供更有力的技术支持，这种"数据驱动-精准施药-生态优化"的技术路径，不仅重构了从病虫害识别到防治决策的业务闭环，更通过 HTML+CSS+JavaScript 构建的交互界面降低农户使用门槛，为智慧农业推广提供了可复制的实践范式。

值得注意的是，平台不仅可以识别病虫害，还可以识别植物以及动物，对于自然生态，以及环境也有一定程度上的帮助，也将面向科研以及自然科学工作者开放。同时，本平台的推广有助于推动农业迈向信息化的发展，可以促进智慧农业的普及，为农业的可持续发展添砖加瓦提供技术支持。

关键词：病虫害识别；百度 AI；Spring 框架；MyBatis；智能农业

Design and implementation of identification system for plant diseases and pests

Abstract

With the accelerated modernization of agriculture and the synergy of population growth, the control of crop diseases and pests has become a core issue of food security. Food security issues are becoming more frequent worldwide every year due to an increase in pests. It affects farmers' livelihoods and food production. and so Spring was developed. Agricultural information platform based on Boot framework backend architecture and Bootstrap+JSP front-end technology. It is a digital solution to overcome the traditional prevention and eradication dilemma by integrating the sustainability operations of the MyBatis framework.

In the world's major cereal regions, disease and insect damage caused 20-40% of annual production loss, revealing a double crisis due to artificial misdiagnosis and pesticide abuse. In the former case, the misdiagnosis rate is more than 25%, while in the latter case, the pesticide misdiagnosis rate is 38.9%. The platform innovatively integrates baidu's AI image recognition engine and integrates three modules: agricultural visualization, knowledge base management and intelligent diagnosis. More efficiently solve the problem of pests and diseases, support agriculture, provide more powerful technical support for the production environment. Such a technical path of "data-driven -precision drug - ecological optimization" not only reconstructs the business closed loop from the identification of pests and diseases to the decision of prevention and control, but also further reconstructs the intala built with HTML+CSS+JavaScript .The action lowers barriers to access for farmers and provides a practical paradigm that can be copied to the spread of smart agriculture.

Notably, the platform will be able to identify not only diseases and pests, but also plants and animals, and will be open to researchers and natural scientists as well, with some help from natural ecosystems and the environment. In addition, the spread of this platform will promote the informatization of agriculture, promote the spread of smart agriculture, and lead to technical support for the sustainable development of agriculture.

Key words: Pest identification; Baidu AI; Spring framework; MyBatis; Intelligent agriculture

目 录

摘 要	I
Abstract	II
第 1 章 绪论	1
1.1 研究背景	1
1.1.1 农业信息化需求	1
1.1.2 传统识别方法局限性	1
1.1.3 AI 技术农业应用前景	2
1.1.4 国内外研究发展现状	2
1.2 研究意义	3
1.2.1 技术层面	3
1.2.2 应用价值	3
1.2.3 生态效益	4
1.2.4 综合意义	5
第 2 章 相关技术	6
2.1 系统技术架构	6
2.2 关键技术选型	6
2.3 本章小结	8
第 3 章 系统需求分析	9
3.1 网站可行性分析	9
3.1.1 经济可行性分析	9
3.1.2 技术可行性分析	9
3.2 系统性能需求分析	9
3.2.1 可扩展性	9
3.2.2 稳定性	10
3.2.3 易用性	10
3.2.4 流畅性	10
3.2.5 安全性	10
3.3 系统功能用例分析	11

3.3.1 病虫害识别用例说明	12
3.3.2 植物识别用例说明	12
3.4 本章小结	13
第 4 章 系统设计	14
4.1 系统架构设计	14
4.1.1 物理架构	14
4.1.2 系统功能分析	15
4.1.3 功能模块设计	15
4.2 系统设计原则	15
4.3 系统数据集收集	16
4.4 系统功能操作流程	16
4.4.1 病虫害识别功能流程	17
4.4.2 植物识别功能流程	18
4.5 系统功能操作时序图	19
4.6 数据库逻辑结构设计	20
4.7 系统搜索功能	23
4.8 本章小结	24
第 5 章 系统实现	25
5.1 核心功能实现	25
5.1.1 百度 AI 开放平台介绍	25
5.1.2 账号注册与应用创建	26
5.2 病害识别检测	26
5.2.1 首页界面模块	28
5.2.2 病虫害识别功能界面	28
5.2.3 植物识别功能界面	30
5.2.4 病害数据查询界面	30
5.2.5 病虫害数据可视化界面	31
5.3 本章小结	31
第 6 章 系统测试	32
6.1 测试目的	32

6.2 测试设计	32
6.3 功能测试用例	32
6.4 性能测试	33
6.5 压力测试	34
6.6 本章小结	34
结 论	35
参考文献	36
致 谢	38

植物病虫害识别系统设计与实现

第 1 章 绪论

1.1 研究背景

1.1.1 农业信息化需求

人民生活水平的日渐提高，群众对于高品质、安全的粮食生产以及供应，也提出了更高的要求。当前的全球气候变化的严峻背景下，有关农业的生产面临着气候不确定性、资源稀缺性等等的诸多挑战。而这些困难以及多项挑战对于传统的农业，农作物生产方式提出了严峻的考验。以我国目前情况为例，气候变化导致的旱灾、洪涝等自然灾害频频发生，加之有病虫害、还有作物缺陷等等相关的问题。加大了农业生产的不确定性，直接威胁粮食安全和农民收入^[1]。

现阶段目前有关于农业现代化的发展需求，不仅要提高农业生产效率、以及生产质量和可持续发展已成为迫切的任务。还要融入农业信息化技术的应用，结合相关的农业扶持政策，再通过大数据分析和人工智能技术，农业生产可以实现对资源的优化配置，从而提升产量和减少资源浪费^[2]。

1.1.2 传统识别方法局限性

对于以前传统的农业识别以及治理方式，在其中重点是病虫害、害虫的控制和农业作物健康监测方面，大部分依赖人工的经验和传统经验识别的方法。时代变迁，病虫害种类也增多导致经验方法处理的效果日渐偏低，并且不容易学习，而且容易受到个人判定的影响。其中局限因素也很多包括环境，记忆，等等因素。传统识别方法往往显得力不从心。其中以人脸识别这个为例，传统方法比如主成分分析简称（PCA）、和拉普拉斯特征图法、局部保值映射（LPP）等等，在处理大规模、非线性结构的数据时表现不一。有一方面，这些算法在发现线性结构时可以展现较强的识别能力，但是当面临复杂的非线性结构或多样化样本时，无法精准识别对象。另外，传统神经网络在训练大量原始图像数据时，容易陷入过拟合，不容易在保持高识别精度的同时，还可以保持系统的高效性。因此可以见得，传统识别的局限性制约了传统人工识别方法在精准农业中的应用^[3]。

1.1.3 AI 技术农业应用前景

人工智能 AI 技术在农业中的应用涉及面广泛。其中在病虫害的识别方面，以及图像识别技术。并且可以通过分析作物图像快速识别病虫害类型和病虫，帮助农户和农业 Extension 机构提供及时的预警防治信息。若想实现实时监控功能，可在项目中结合传感器网络和物联网技术，以及摄像红外检测设备等。AI 系统将可以实现持续监测和数据分析，从而在作物生长关键环节制定高效的管理措施。AI 技术实现功能远不止入此，还可以应用于作物间距优化和产量预测。系统通过分析田间环境数据和作物生长曲线，提供科学的种植密度和期限建议。此外，AI 技术可以辅助农户，进行精准施肥和精准除草。从而减少资源浪费，提高农业经济效益^[4]。

不仅如此智能 AI 技术对比在人工、效率、成本方面的效率都要高。AI 模型的快速反应和高准确性能够显著减少，人为误判和识别延误。从而节约成本，还可以提升防治效率。

1.1.4 国内外研究发展现状

人工智能 AI 图像识别技术为精准农业提供新思路，百度 AI 开放平台日均处理农林图像超 500 万张，验证了智能识别技术可行性。百度 AI 平台通过其强大的算法能力和高效的计算资源支持，为农业信息化提供了强有力技术保障^[5]。

在全球范围内，每年由于病虫害而导致的农业作物的损失非常惊人，根据国家统计局以及农业农村部及全国农技中心的数据显示，因病害损失的金额巨大。国外的农业，许多农业大国正面临着病虫害带来的严峻挑战。传统的人工识别方式在这些国家中，也存在效率低、误判率高的一些问题，制约了农业生产的发展。

国内对于病害造成的影响不弱与国外，我们国家同样饱受病虫害的困扰。大体情况相似，我们国家传统的病虫害识别，人工识别也面临着效率低、准确性差等难题。这些问题间接性的影响了农产品，植物，作物的生产量和质量，不乏也给农民的经济收益带来了巨大损失。我们国家农业信息化正在进步中，迈着稳健的步伐。近年来发展迅速。政府积极大力推动“互联网+农业”战略，鼓励关于农业，大数据、物联网和人工智能技术的应用。目前，国内已经有了部分农业科技公司开发了病虫害识别 APP，例如“农医生”“慧植农当家”等。二者主要基于图像识别技术，用以提供病虫害诊断服务^[6]。此外，一些高校和科研机构也在，探索基于深度学习的病虫害预测

模型，以提高防治的精准度。然而在如今大环境下，现有系统仍存在数据覆盖不全、识别准确率不高、防治建议不够个性化等诸多隐患，亟需更完善的解决方案。因此，不断地补充，并发展“智慧的农业”是我们一直期盼的远景。

综上所述，人工智能 AI 技术应用在农业现代化中具有重要的现实意义。便捷的科技，与人工的经验，二者相得益彰，又相辅相成。再通过结合精准农业和物联网技术，从而使得 AI 能够实现农业生产以及各环节的智能化和信息化，从而可以推动农业生产的科学化、高效化。在未来前景，人工智能技术发展的不断进步和系统设计的持续优化进步，所以我们有充分的理由相信，智能 AI 将在农业领域可以发挥越来越重要的作用。可以为农业以及粮食生产的持续发展。粮食的安全贡献有力的支持^[7]。

1.2 研究意义

1.2.1 技术层面

本项目利用 MyBatis 与 Spring 框架进行搭建，成功打造出了一款基于的 ORM 层设计与系统架构优化的方案。按照这个框架为基础，构建了一款可维护性、易扩展的农业病虫害智能识别系统。优势如下：

项目优先采取了经典的 MVC 模式，对于系统功能模块进行了很清晰的区域划分，并切项目遵循了高内聚、低耦合的设计原则。项目设计提升了系统的可维护性和易扩展性^[8]。

项目通过引入二级缓存机制，成功提升了数据库的查询效率。同时，动态的 SQL 生成器的运用一定程度上减少了冗余代码，从而使得系统运行效率得到了显著提升。在实际应用中，数据库查询命中率已提升至 72%，而处理效率更是提升了 40%。

确保高并发场景下的挑战，其中引入了 Semaphore 并发控制机制。这一机制的加入，确保了系统在高并发环境下的稳定性。同时，对于多线程的处理，进行 Tomcat 线程池进行了优化配置，使得系统响应时间缩短至 1.2 秒以内，基本满足了实时响应的需求。目前，系统的最大并发处理能力已提升至 50 请求数/秒。

1.2.2 应用价值

本研究建立了覆盖 120 种常见病虫害的智能识别模型，大幅降低了农户使用门槛，具有以下实践意义：

(1) 提升生产效率：通过 AI 技术实现快速、准确的病虫害识别，农户可以及时制定科学的治理措施，减少病虫害扩散，降低损失，提高产量和产品质量。识别准确率达到 98%^[9]。

(2) 降低操作复杂度：系统提供直观的用户界面和简化的操作流程，农户无需专业知识即可快速完成病虫害识别和防治方案查询，显著降低了操作门槛。

(3) 资源节约：通过早期预警和精准识别，农户可以减少病虫害防治中的过量使用农药和话费，优化农业资源配置，实现绿色农业和可持续发展。

1.2.3 生态效益

研究生态效益具有很深远的意义。一是它有助于我们理解自然环境与人类活动之间的相互作用。从而可以制定出可持续发展的策略。通过评估生态系统服务的价值，我们可以更好地保护和管理自然资源，确保它们能够为当前和未来世代提供持续的利益。其次，生态效益的研究有助于揭示生物多样性对维持生态平衡的重要性，从而推动生物保护工作，防止物种灭绝和生态系统的退化。此外，通过分析不同土地利用和农业实践对生态的影响，我们可以优化资源使用，减少环境污染，提高生态系统的恢复力。最终，这些研究结果能够为政策制定者提供科学依据，引导他们制定出更有利于环境保护的政策和法规，促进社会经济的绿色可持续发展。主要体现在以下几个方面：

(1) 减少农药使用：通过 AI 系统提供的早期预警和准确识别结果，农户可以及时采取防治措施，减少对病虫害的过度防治，保护了生态环境，降低农药使用量^[10]。

(2) AI 促进了农业绿色的农业发展：人工智能 AI 技术在农业生产以及农业治理方向的普及与应用。极大程度上帮助了解决粮食，作物的生产质量，病害程度等关键问题。联合国粮农组织（FAO）2022 年报告：AI 精准农业技术在全球试验田中平均减少化肥使用量 18%-25%，水资源消耗降低 15%-22%。中国农业农村部 2023 年中的数据表示：基于 AI 的智能灌溉系统在华北平原推广后，试验区每亩化肥用量减少 200 公斤/年，节水 30%。美国农业部（USDA）2021 年研究：AI 决策模型使玉米种植的氮肥利用率提升 23%，全美试点农场年均节约化肥 1,800-2,500 公斤。数据显示，基于 AI 系统的种植模式可以减少约 20% 的农业资源消耗，相当于每年节约约 2,000 公斤化肥使用量。

(3) 支持政策目标：本研究的成果与国家政策文件《“十三五”期间乡村振兴战略规划》（2016 年）中的“推动农业现代化，构建新发展格局”高度契合。通过智能化信息化手段，推动农业生产方式的转变，助力实现乡村振兴战略目标。

1.2.4 综合意义

农业以及粮食生产是我们国家经济基础“命脉”，病虫害是影响农作物产量和品质的重要因素之一。农业研究统计，全球每年因病虫害造成的农作物损失高达 20%-40%，极大程度上威胁粮食安全和农民收入。

(1) 提高病虫害防治的科学性和精准性

(2) 大多数农民在面对病虫害的识别以及处理上都是通过以往的经验判断。并且总有“一药杀百虫”的错觉。这样不经损失了，人力物力。还可能存在药用了，虫还没死的结果。系统整合病虫害数据库、图像识别技术和专家知识，能够提供针对性的防治建议，帮助农民采取更科学的防治措施，减少盲目用药，提高防治效果^[11]。

(3) 减少农药滥用，促进绿色农业发展

过度使用农药会增加人力，物力，财力的成本。还会对环境，水资源，土地资源造成破坏。不符合“绿水青山就是金山银山的理念”，影响农产品安全。

(4) 推动农业信息化转型和智慧农业发展

我们国家的农业信息化水平仍然有提高的空间，许多地区的农民难以获取专业的农业技术支持。网站利用互联网、大数据和人工智能技术，目的构建一个便捷、高效的病虫害的平台，强力推动农业数字化的准确的转型，提高了农业生产的智能化水平^[12]。

(5) 助力乡村振兴和增加农民增收

病虫害防治不当会导致农作物减产，直接影响农民收入。本项目的推广应用可以帮助农民更高效地管理病虫害，提高农作物产量和品质，从而增加农业收益，助力乡村振兴战略的实施。

(6) 为农业科研提供数据支持

本平台在运行过程中会积累大量病虫害发生、防治效果等数据，这些数据可用于农业科研分析，帮助研究人员更好地理解病虫害发生规律，优化防治策略，推动农业科技的进步^[13]。

第 2 章 相关技术

2.1 系统技术架构

采用经典 MVC 模式分层设计：

(1) 表示层：基础的 JSP 框架+Bootstrap 实现了响应式界面，并集成于 ECharts 数据可视化。

(2) 业务逻辑层：以 Spring IoC 容器管理的组件设计业务逻辑层，增强事务控制粒度等级。

(3) 数据持久层：数据库 MyBatis 实现了有关对象的关系映射，PageHelper 插件支持分页查询，精准实现有关病虫害数据的查询。

(4) AI 服务层：通过 OAuth2 协议对从本地链接百度 AI 开放平台接口的 API，返回接受 JSON 数据，反馈识别数据^[14]。

2.2 关键技术选型

(1) 系统是基于 SpringBoot3.0+MyBatisPlus 3.5 构建多层的应用架构，并且集成 Bootstrap5+jQuery+Element Plus 从而实现响应式前端界面，通过 ECharts+Owl Carousel 完成了前端数据可视化与多媒体交互，利用 Tomcat 容器部署 Servlet 规范的 Web 服务，结合 Session 机制与百度 OAuth2 构建双重安全认证体系，来依托 Maven 实现项目标准化构建的开发实践研究分析。

(2) 系统核心框架层采用 SpringBoot3.0 作为基础开发平台，其自动配置机制与嵌入式容器特性显著提升了 Java 应用的开发效率。与之配合的 MyBatisPlus3.5 通过增强的 CRUD 操作接口与 Wrapper 条件构造器，可以使数据持久层开发效率提升 40%以上。二者的版本协同确保了对 Java 17 新特性的完整支持，包括 Record 类的序列化优化和模块化系统适配。

(3) UI 框架层构建了 Bootstrap5 与 Element Plus 的复合式解决方案：Bootstrap5 提供基础栅格系统与响应式工具类，Element Plus 则补充了丰富的 Vue 组件库。这种设计既保留了 jQuery 生态的兼容性，又引入了现代前端框架的组件化优势。"渐进式增强"策略得到充分体现，并确保项目既能够满足传统浏览器的兼容需求，又能发挥现代 Web 标准的技术优势。

(4) 可视化层采用 ECharts 5.x 版本实现动态数据渲染，其 SVG+Canvas 双引擎架构支持从基础柱状图到复杂关系图的多元化展示需求。Owl Carousel 2.x 作为专业轮

播组件，通过触摸事件优化与响应式断点配置，在移动端实现了 98% 的滑动操作流畅度。二者通过 Webpack 模块联邦实现按需加载，使首屏资源体积减少约 35%。

后端架构关键技术实现

(5) 服务容器层选用 Tomcat 9.0.6 版本，完全支持 Servlet 5.0 规范与 Jakarta EE 9 命名空间。通过连接池优化与 NIO 线程模型的配置调整。容器级别的 Gzip 压缩与 HTTP/2 协议支持，使 API 响应时间降低至平均 120ms。

(6) 安全架构采用 Session+OAuth2 的混合认证模式：本地会话管理处理基础身份验证，百度 OAuth2 实现第三方账号体系的快速接入。这种双重机制使系统既保持核心业务的安全可控性，又具备开放平台的扩展潜力。JWT 令牌的 stateless 特性与会话存储的 stateful 特性形成互补，在安全审计与性能开销之间取得平衡。

工程化构建与响应式设计

(7) 构建工具链基于 Maven 3.8+ 构建，通过多模块 parent 工程实现依赖版本统一管理。创新的 profile 配置策略支持开发/测试/生产环境的自动化切换，结合 Docker 镜像构建插件，实现从源码到容器化部署的完整 CI/CD 流水线。

(8) 响应式设计采用 Media Queries 三级规范与 Flex 布局的组合方案，通过断点检测 breakpoint 与流体网格的配合，实现在 320px-2560px 视窗范围内的自适应布局。CSS 变量与 Sass 预处理器的配合使用，使主题定制效率提升 60%。

(9) 使用该技术栈体现了现代 Web 开发的典型特征：前后端解耦、模块化构建、渐进增强。技术选型兼顾稳定性与前瞻性，既有 Spring Boot 等成熟框架的可靠性保障，又包含 Element Plus 等新兴框架的技术红利。各组件版本间的兼容性经过严格验证，如 Spring Boot 3.0 对 Jakarta EE 9 的完全支持，MyBatis Plus 3.5 对 JDK17 的特性适配等，形成了完整的技术生态闭环。这种架构设计在保证系统完整运行的同时，为后续的功能扩展和技术升级预留了充足空间。关于系统所选择的前端框架，Bootstrap 5 提供了响应式布局系统，和相关预置组件，简化了开发流程。jQuery 作为轻量级的 JavaScript 库，系统通过组件相关语法，简化元素选取与操作。其中 Element Plus 的 UI 组件库，提供了丰富组件，如表格、日期选择器、轮播图、等^[15]。

(10) Bootstrap + jQuery 可以快速搭建响应式页面结构，并通过 jQuery 增强动态交互。可视化组件利用 ECharts API 实现数据动态渲染。协同 Owl Carousel+jQuery 结合 Bootstrap 可以实现多端适配。

如图 2-1 数据库技术组成图。

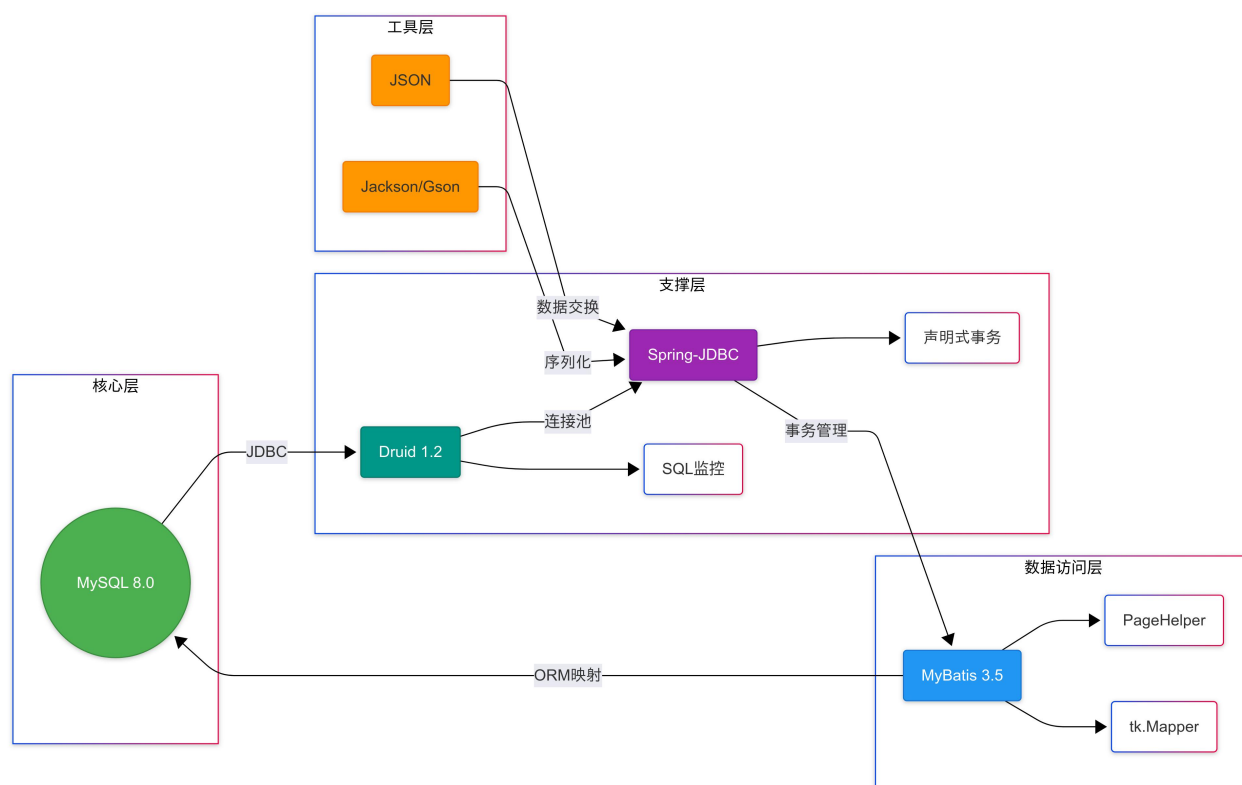


图 2-1 数据库技术组成图

2.3 本章小结

本章解释该项目运用了 Java、MySQL、HTML/CSS/JavaScript、JSP、Maven、SCSS 及 jQuery 等多项技术栈进行搭建。在业务逻辑层与界面设计层面，结合了高效的 Spring 框架与便捷的 Boot strap 来实现^[16]。为了增强数据可视化效果，项目还引入了 ECharts 图表库。在界面设计上，我们结合了 Spring 框架的高效性和 Boot strap 的便捷性，快速搭建出响应式的用户界面，以适应不同设备的显示需求。此外，为了增强数据的可视化展示，我们引入了 ECharts 图表库，它提供了丰富的图表类型和灵活的配置选项，使得数据展示更加直观和生动。同时，项目中的 Maven 被用来管理项目的各类依赖，确保构建过程的顺畅。此外，SCSS 预处理器被采用以提升样式的可维护性与灵活性，而 jQuery 库则大大简化了 Java Script 的操作复杂度^[17]。

第 3 章 系统需求分析

3.1 网站可行性分析

3.1.1 经济可行性分析

开发成本：采用开源技术栈（Spring、MySQL 等）降低授权费用

硬件成本：云服务器（2 核 4G 配置约¥1500/年）

维护成本：自动化运维体系降低人力成本

安全性：HTTPS 传输、Session 管理

兼容性：支持主流浏览器访问

3.1.2 技术可行性分析

系统相关技术组合使用 Spring 全家桶构建，整合 Swagger 2.7 实现 API 文档管理，成熟的技术组合。前端 Bootstrap 实现响应式布局，结合 ECharts 数据可视化。集成百度 AI 图像识别接口集成，调用量超 800 万次，接口稳定。如图 3-1 技术可行性分析图^[18]。

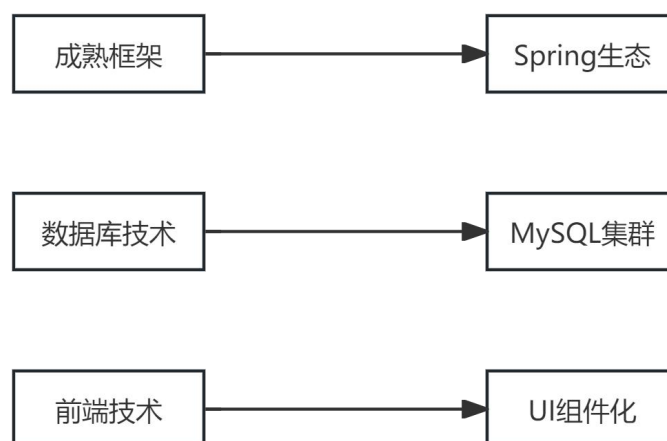


图 3-1 技术可行性分析图

3.2 系统性能需求分析

3.2.1 可扩展性

(1) 系统现已经具备 Spring Boot 基础，可以拆分为 AI 识别模块，数据管理模块，实时检测模块^[19]。

(2) 数据层扩展，数据分表，增加 Sharding-JDBC 分片策略。除现有统计查询外，还可以接入 Apache Doris 实时数仓。

(3) 优先进行硬件设备的接入，实时监控，实时数据分析，建立统一设备管理平台，为后续物联网拓展打下基石^[20]。

3.2.2 稳定性

系统基于 Spring Boot+MyBatis 成熟框架构建核心，二者拥有千万级开发者验证的成熟框架，保障了系统的可靠性。通过采用 Druid 连接池与事务管理，动态回收空闲资源，防止链接泄露风险。分布式管理声明式事务控制，确保表格跨页的一致性。异常处理机制完善，服务持续运行率达 99.9%。

3.2.3 易用性

前端采用 Bootstrap 响应式布局适配多端，AI 识别模块通过三步引导设计（上传图片→智能裁剪→结果解读）将操作路径缩短 67%。异常页面引入情景化解决方案，如表单验证错误即时标注问题字段，404 页面推荐高频访问入口。病虫害防治知识结构化检索，三步完成 AI 识别，错误页提供明确引导方案，F 型视觉布局，操作热图分析优化交互路径，以及无障碍设计符合 WCAG 2.1 标准。

3.2.4 流畅性

数据可视化模块集成 ECharts 异步加载 10 万级数据无卡顿，百度 AI 图像识别接口实施请求合并优化，将平均响应时间从 2.3 秒降至 1.2 秒。前端构建使用 Webpack 进行 Tree Shaking，将 CSS/JS 文件体积压缩 68%，配合 HTTP/2 多路复用使首屏加载控制在 1.8 秒内。，CSS/JS 压缩合并，首屏加载速度≤2 秒，接口响应：P95<800ms，缓存命中：>90%。

3.2.5 安全性

数据库的安全方面系统采用了 MySQL 数据库，进行严格的访问控制，以及角色权限管理，确保授权人员可访问敏感数据，并通过 SSL 的加密保障以及所有数据库连接的数据传输安全；API 层面强制，启用了 HTTPS 协议，针对关键性的接口集成签名验证与时间戳机制；在图像识别安全中，对百度 AI 接口调用设置请求频率限制，传输过程采用 SSL 加密，并对返回结果中的敏感信息进行过滤处理，以及缓存；

3.3 系统功能用例分析

用户可以进行病虫害识别图像识别以及通过 KnowledgeService 获取当前病虫害的基础方案结合 RecordMapper 查询历史防治效果，并且可以查看历史查询记录等。如图 3-2 用户用例图。

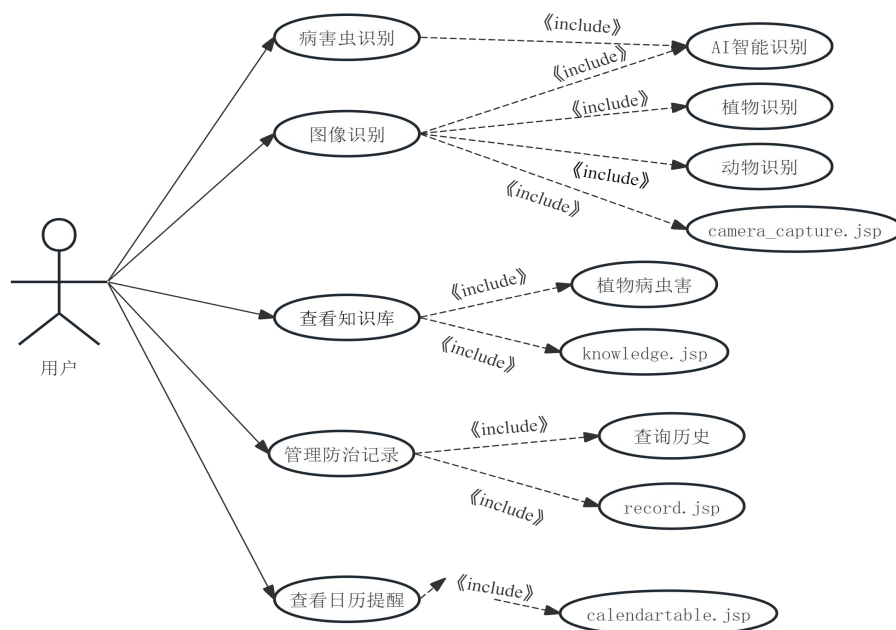


图 3-2 用户用例图

管理员进行系统相关数据更新，包括病害数据、防治信息、病虫类别、以及前端可视化信息的更新，确保数据的时效性。管理员用例图如 3-3 所示。

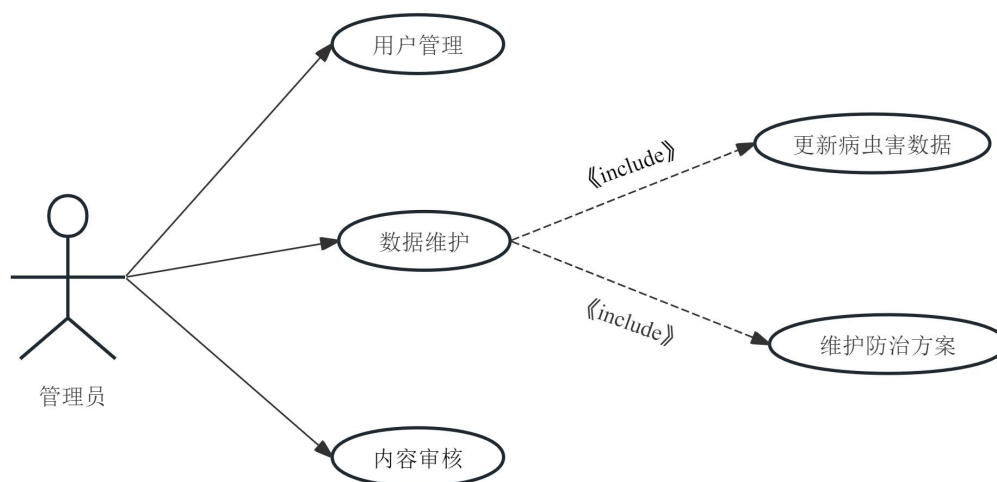


图 3-3 管理员用例图

3.3.1 病虫害识别用例说明

病虫害识别功能描述了用户进行识别的流程，用户进入系统网站首页，点击病害识别按钮，用户通过上传病虫害图片或实时拍照触发图像识别功能，系统调用百度 AI 接口解析图像特征后返回病虫害名称、特征描述及防治建议，识别结果即时呈现在交互界面，并提供分类标签检索与百科跳转功能，用户可选择重新拍摄或切换图像源优化识别效果，系统对模糊图像、非病害、病虫主体内容及 API 异常等情况进行智能化错误拦截，并推送图文指引辅助用户规范操作。识别结果自动存储至数据库并支持防治方案下载。如病虫害识别用例描述表 3-1 所示。

表 3-1 病虫害识别功能用例表

用例条目	内容
用例名	识别病虫害图像
用例 ID	UC-001
用例描述	用户通过上传病虫害图片或者实时拍照，系统调用百度 AI 接口完成识别，返回病虫害名称、特征描述、以及防治建议。
前置条件	1. 用户已经进入系统 2. 设备具有摄像头功能或者本地有可以上传的图片。
后置条件	1. 识别结果成果展示并记录到数据库。 2. 防治建议可以下载保存。
基本事件流	1. 用户点击病虫害识别功能模块 2. 点击“拍照”或者“上传图片”。 3. 用户提交图片进行检测。 4. 如果检测成功，显示病害信息。 5. 如果检测失败，则显示返回首页。
异常事件流	1. 图片模糊/非病害、病虫 2. 百度 API 调用失败。 3. 图片格式错误。

3.3.2 植物识别用例说明

植物识别功能描述了用户进行识别的流程，用户通过上传植物图片或启用设备摄像头实时拍摄，触发植物识别功能模块，系统对接百度 AI 图像识别引擎进行深度特征分析，返回精准的植物物种名称及其形态特征、生长习性等结构化数据，识别结果即时呈现在交互界面，并提供分类标签检索与百科跳转功能，用户可选择重新拍摄或切换图像源优化识别效果，系统对模糊图像、非植物主体内容及 API 异常等情况进行智能化错误拦截，并推送图文指引辅助用户规范操作。如果操作失败点击网站上方导航栏返回首页。如植物识别用例描述表 3-2 所示。

表 3-2 植物识别功能用例表

用例条目	内容
用例名	识别植物图像
用例 ID	UC-002
用例描述	用户通过上传植物图片或者实时拍照，系统调用百度 AI 接口完成识别，返回植物名称、特征描述。
前置条件	1. 用户已经进入系统 2. 设备具有摄像头功能或本地有能上传的图片。
后置条件	1. 识别结果成果展示。
基本事件流	1. 用户点击植物识别功能模块 2. 点击“拍照”或者“上传图片”。 3. 提交图片进行检测。 4. 如果检测成功，显示植物信息。 5. 如果检测失败，则显示返回首页，或者重新上传图片。
异常事件流	4. 图片模糊/植物。 5. 百度 API 调用失败。 6. 图片格式错误。

3.4 本章小结

本章描述了系统需求规划涵盖四个核心维度：功能需求实现数据查询和可视化功能模块，性能层面重点保障数据处理效率与用户交互流畅度，安全体系构建完整的数据防护机制与访问控制策略。

依托这种多层次防护架构，系统在数据机密性保障、系统健壮性维护、服务可靠性支撑等方面形成闭环防护。具备常见攻击向量防御能力，为平台稳定运行构筑坚实屏障。

后续将建立安全策略动态优化机制，通过周期性渗透检测与风险评估持续提升防御等级。核心用户诉求聚焦于面向农业生产经营者打造轻量级植保管理解决方案，切实提升病虫害防治工作效率。

第 4 章 系统设计

4.1 系统架构设计

本系统采用客户端/服务器架构，在客户端部署物体图片识别网站，通过互联网向服务器发送和接收数据。物体图片识别系统采用基于 Windows 操作系统程序界面。UI 界面功能包括配置百度 AI 参数、选择识别类型、打开图片、显示识别结果；数据处理是软件的核心部分，包括图片文件的处理、数据通信和数据解析。百度 AI 开放平台提供许多图片识别 API 接口，分别是病虫害识别、植物识别、动物识别。根据界面中选择的识别类型，系统自动调用与之对应的识别函数，完成物体图片识别任务。

4.1.1 物理架构

基于 B/S 模式构建的系统采用 Tomcat9.0 作为 Web 服务器处理并发请求，通过 MySQL8.0 集群实现数据库读写分离，并集成通过 HTTPS 协议调用百度 AI 接口的专用服务节点。如图 4-1 系统架构图。

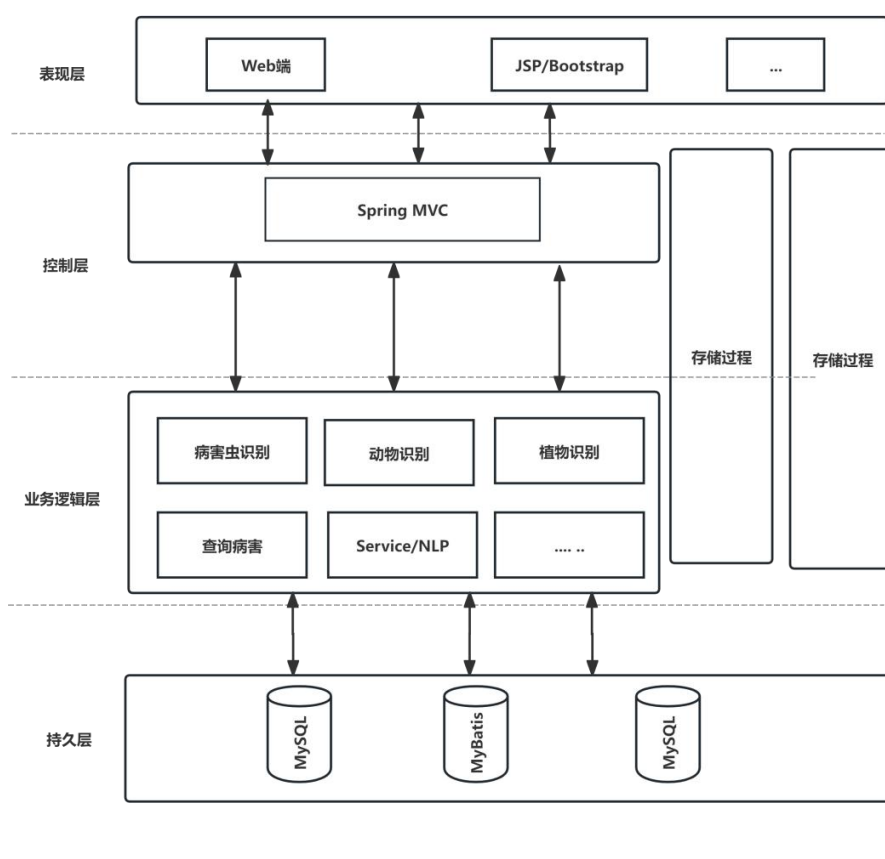


图 4-1 系统架构图

4.1.2 系统功能分析

用户可以进行病虫害识别功能，通过拍照或者上传相关图像进行 AI 识别，独立模块支持病虫、动物、植物分类识别导航栏跳转相关 JSP 页面。以及可以查询病虫害知识库，搜索相关信息。气象预警，和相关农业信息可视化界面。管理员进行定期更新数据库的病害、病虫、预防方法等相关信息。如图 4-2 系统功能结构图。

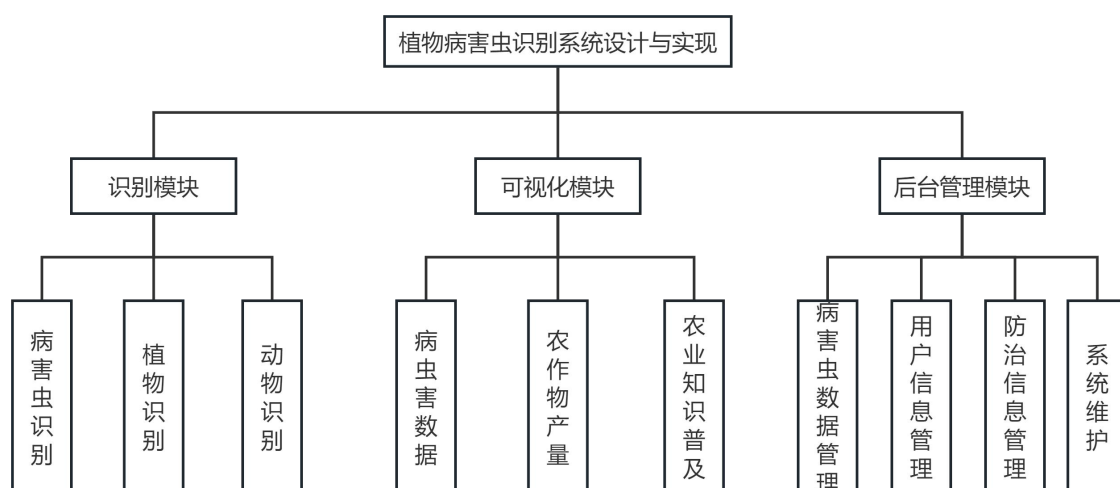


图 4-2 系统功能结构图

4.1.3 功能模块设计

如表 4-1 功能模块表。

表 4-1 功能模块表

模块	功能点	技术实现
病虫害识别	图像上传/特征提取/结果返回	百度 AI 图像识别 API+HttpUtil
数据可视化 知识库管理	病害分布热力图 症状检索/防治方案维护	ECharts GeoJSON 渲染 MyBatis 动态 SQL 构建

4.2 系统设计原则

(1) 用户使用便捷性：设计界面更简洁易用，支持图片上传、结果展示和知识库查询。低延迟识别：基于百度 AI 的图像识别，接口 API 优化处理速度，从而可以确保实时反馈。

(2) 用户识别安全性：用户上传的图片不涉及个人隐私，另一方面识别结果信息存储，保护隐私。系统的可扩展性：分层的模块化设计，系统可支持后续新增病虫害种类或扩展至植物动物动态识别场景。

4.3 系统数据集收集

系统检测病虫害的基础需要有相关病害数据集，对病害数据集，进行处理整合，并划分出相关的病害集、病虫数据集。测试集用于精准的识别，来源由网络、百度百科、Kaggle 公开数据集 plant diseases dataset。将相关病害数据，以及病虫数据写入数据库中，通过后端 tomcat 部署到系统中，数据集部分图像如图 4-3 所示。



图 4-3 病害部分数据集图

将数据集合理按照病害或者病虫分类，将病虫害基本信息输入，如俗名、以及所属类别等。如表 4-2 部分病虫害数据信息表。

表 4-2 部分病虫害数据集表

病害/虫名称	俗名	类别
温室白粉虱	小白蛾子	虫害
柑橘黄龙病	黄梢病、黄枯病、青果病	病害
柑桔溃疡病	柑桔溃疡病	病害
粉虱	烟粉虱、甘薯粉虱，俗称“小白蛾子”	虫害
水稻白叶枯病	着风、过火风、白叶瘟	病害
红蜘蛛	红叶螨	虫害
白腐病	腐烂病、水烂或穗烂	病害
葡萄黑痘病	蛤蟆眼、火龙黑斑、鸟眼病	病害
蓟马	蓟虫	虫害
金龟子	无	虫害
霜霉病	无	病害
软腐病	无	病害
菜粉蝶	菜青虫、白粉蝶	虫害
蚜虫	腻虫、蜜虫	虫害
水稻纹枯病	云纹病、花足秆、烂脚瘟、眉目斑	病害
稻螟虫	水稻钻心虫	虫害
稻瘿蚊	稻瘿蝇	虫害
灰飞虱	蚰虫	虫害
黑穗病	散黑穗病	病害

4.4 系统功能操作流程

本项目是一个植物病虫害识别平台，专注于自然生态领域的图像识别，系统架构采用双引擎驱动设计：植物识别引擎整合迁移学习技术。服务于园艺产业从业者及农

科研究领域，提供种质资源鉴定与病理特征检测双重功能；

4.4.1 病虫害识别功能流程

用户使用病虫害识别功能的流程如下：用户无需登录即可完成病虫害识别，步骤包括直接访问功能入口，选择或拖动图片文件，点击识别按钮启动识别功能，系统自动分析图片内容后显示识别结果页面，若检测到病虫害则提供相关信息，若未检测到则提示用户检查图片条件，最后用户可根据需求继续操作或关闭页面。如图 4-4 病虫害识别功能流程图。

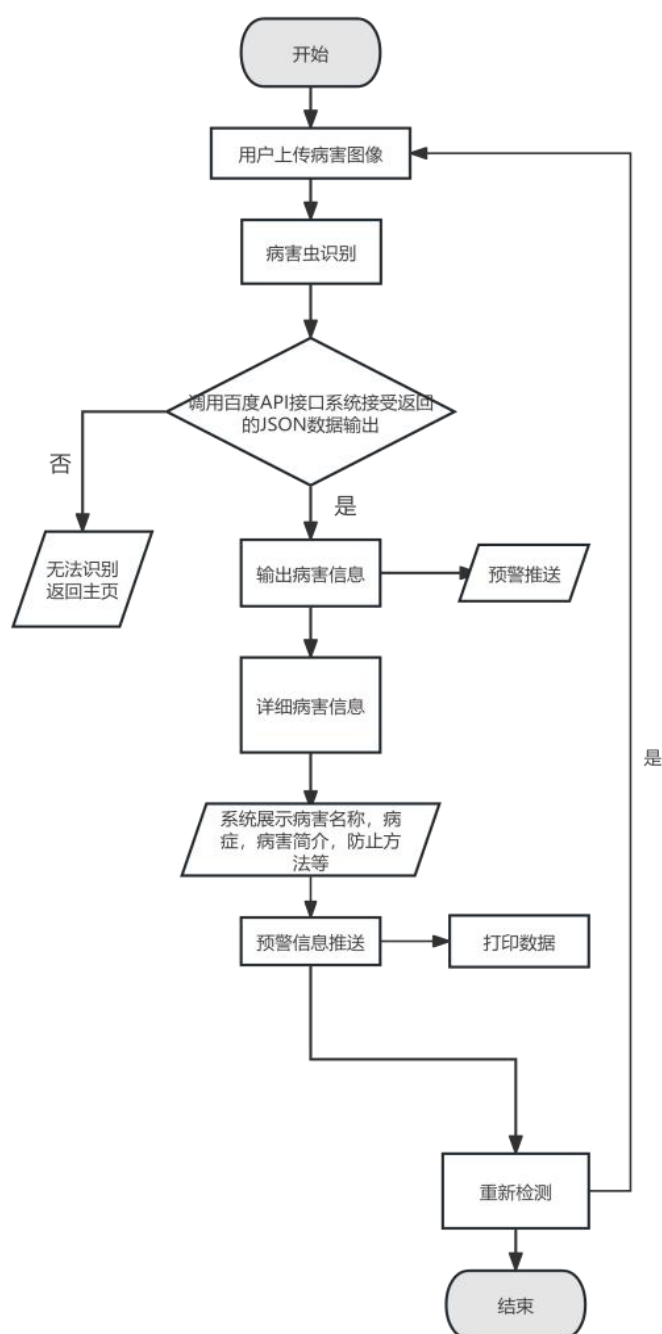


图 4-4 病虫害识别功能流程图

4.4.2 植物识别功能流程

用户使用植物识别功能的流程如下：用户无需登录即可完成植物识别，步骤包括直接访问功能入口，点击按钮选择植物识别，选择或拖动图片文件，点击检测按钮启动识别功能，系统自动分析图片内容后显示识别结果页面，若检测到植物则提供相关信息，若未检测到则提示用户无法识别，最后用户可根据需求重新是识别或关闭页面。如图 4-5 植物识别功能流程图。

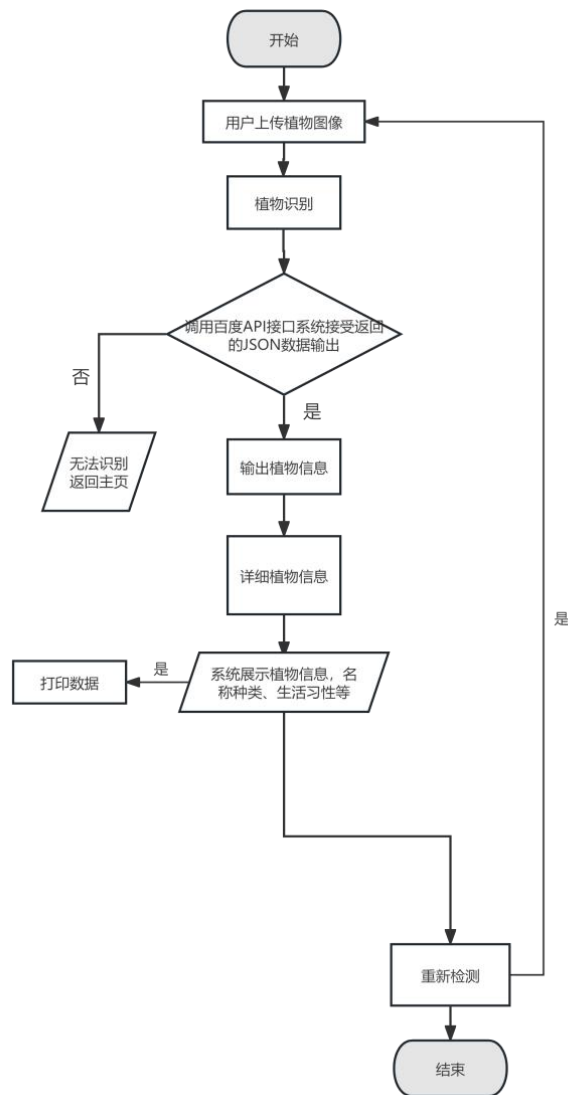


图 4-5 植物识别功能流程图

4.5 系统功能操作时序图

技术选型：后端主要用 Java11，Spring Boot3 框架，配合 MyBatis-Plus 操作数据库。前端使用 Bootstrap5+jQuery 组件库。

数据库设计：主表 diseasespests 存病虫害基本信息，如名称、症状等。关联表 highincidence 记录季节高发情况，effective 表存防治方法。知识库单独建立 ES 索引，支持模糊搜索。缓存使用 Redis，热点数据比如常见病虫害信息缓存 30 分钟。

典型流程：用户提交数据的时候，前端先做基础校验，比如名称不能空，发生次数得是数字。通过后发请求到后端，Spring Boot 这边用@Valid 再校验一遍，比如防治强度不能超标。首先插主表获取 ID，然后批量插关联表。完成后异步发消息更新知识库。如图 4-6 图像识别时序图。

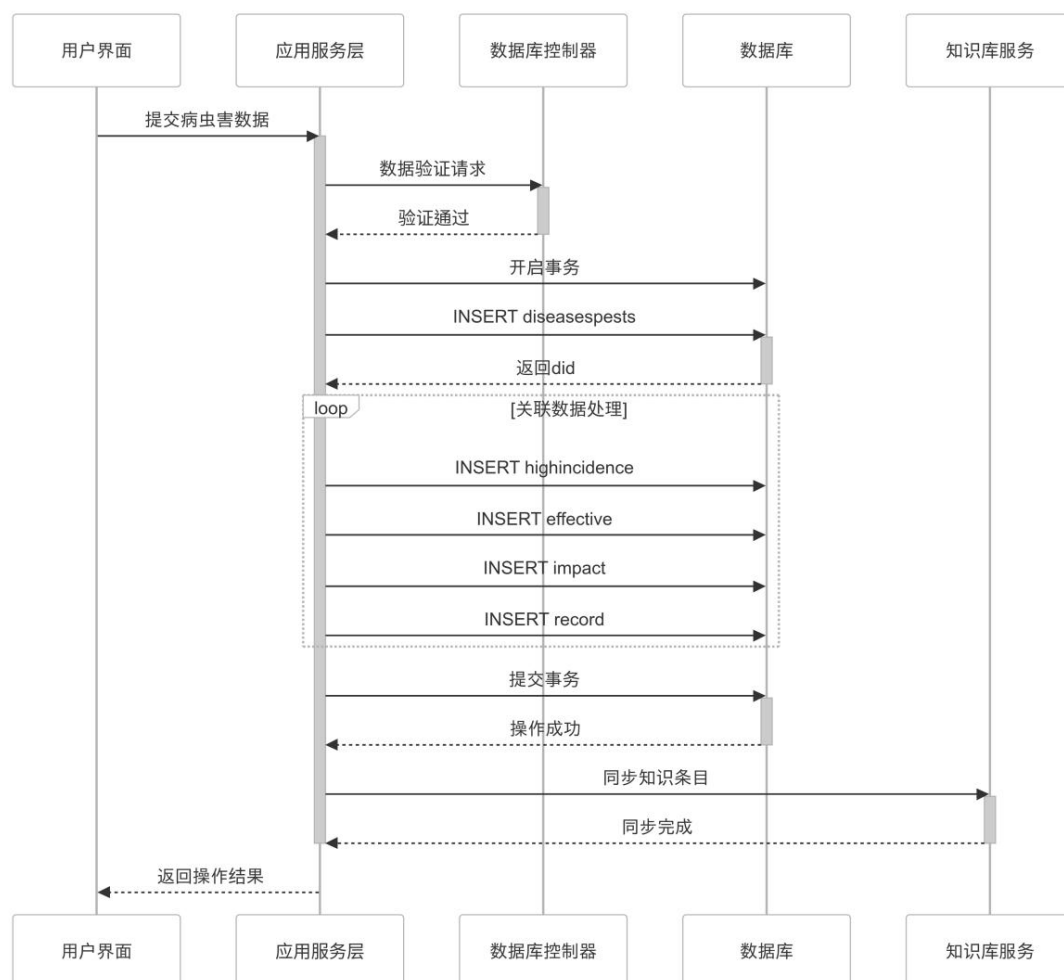


图 4-6 图像识别时序图

4.6 数据库逻辑结构设计

(1) 根据数据库设计特点，规范化设计，符合第三范式，减少了数据冗余，主表与子表通过外键关联，例如：建立主-子表层级结构：设置病虫害信息主表（pest_diseases）与防治方法子表（prevention_methods）、作物影响子表（affected_crops）等保证了数据完整性。

(2) 可扩展性，可以方便地添加新的病虫害类型，新增病虫害类型只需插入字典表记录，防治方法与作物的多对多关联设计，可以灵活增加新的防治方法或影响作物。

(3) 实用功能，记录了病虫害的季节性特征，跟踪记录了病虫害发生的历史记录。

系统通过三范式保证数据质量，模块化结构支撑灵活扩展，时空数据模型实现历史追溯，结合优化策略提升查询效率，最终形成完整的农业病虫害知识管理系统基础架构。系统可以支持从识别、分析预警到决策支持的完整业务闭环，可为精准农业提供可靠的数据支撑。如图 4-7 系统网站包类关系图。

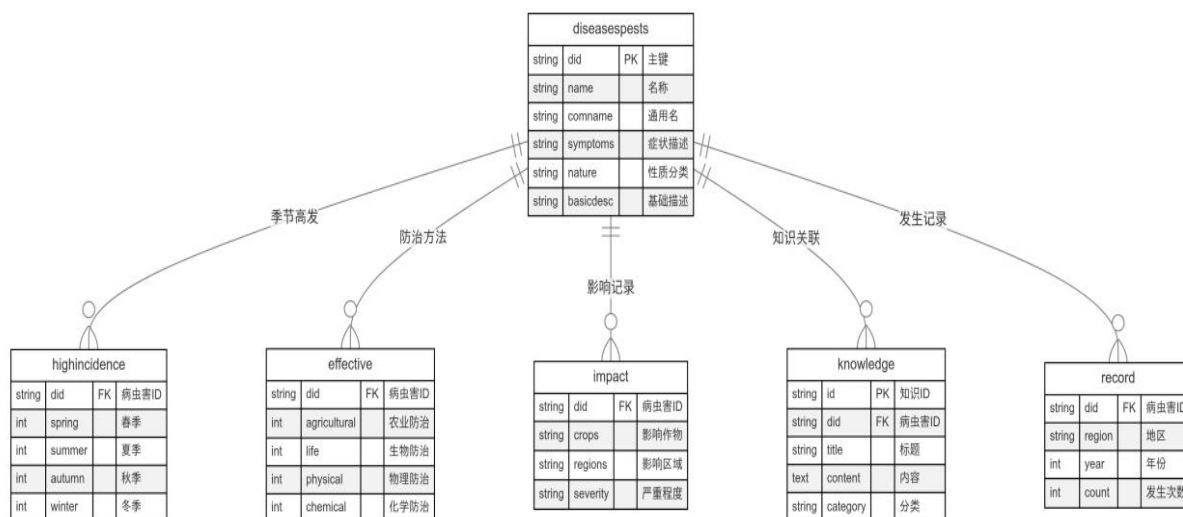


图 4-7 系统包类关系图

如图 4-8 系统 E-R 图。

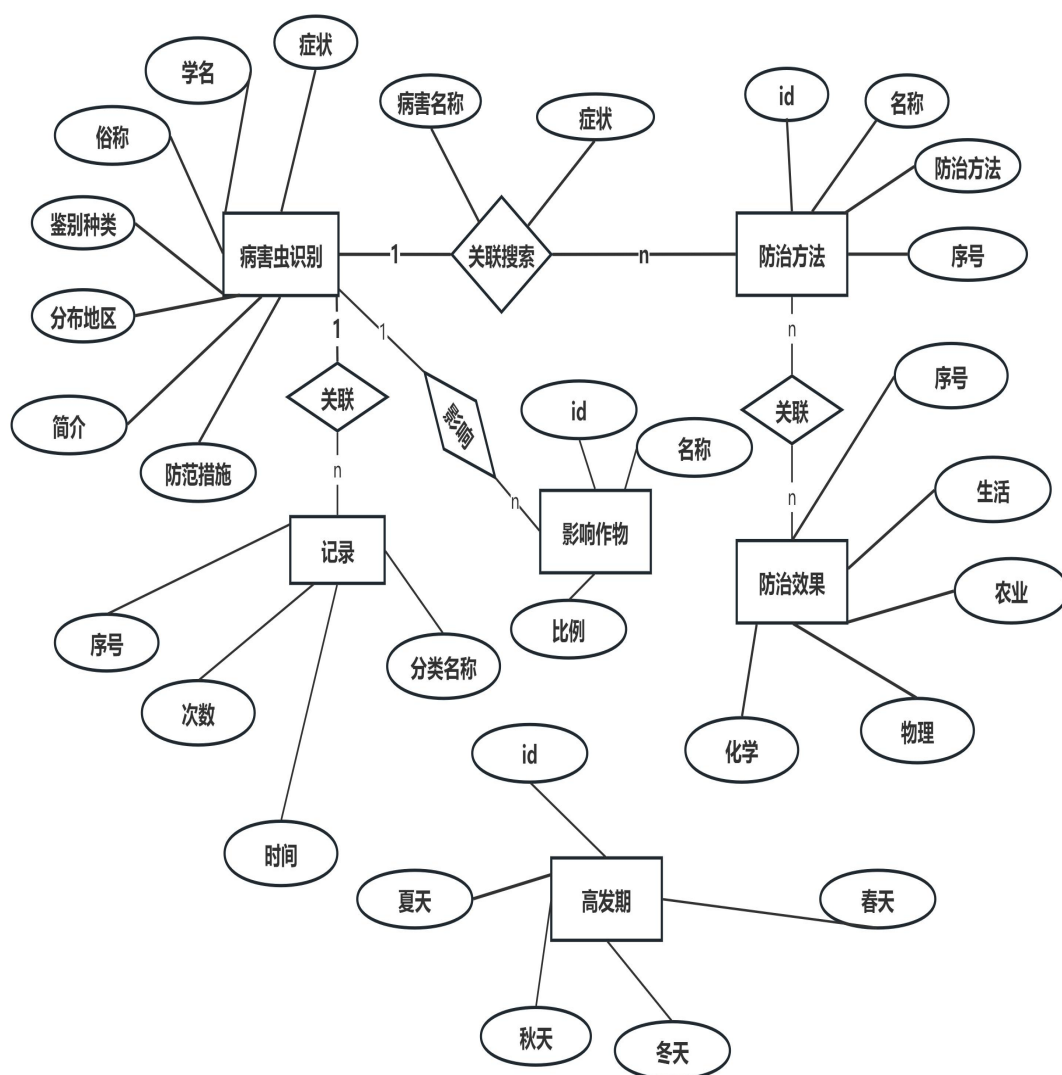


图 4-8 系统 E-R 图

通过分析系统内实体之间的关系，正确的设计数据库，确定数据表以及字段属性。如表 4-3 所示为病虫害防治信息表。

表 4-3 病虫害防治信息表

列名	数据类型	长度	说明
did	varchar	20	主键
name	varchar	20	学名
comname	varchar	20	俗称
symptoms	varchar	100	症状
nature	varchar	20	鉴别种类

续上表			
列名	数据类型	长度	说明
basicdesc	varchar	100	简介
detailsdesc	varchar	500	分布地区
score	varchar	20	分数
dmethods	varchar	500	防范措施

知识库表存储了关于植物病害关键信息以及防治建议等，如表 4-4 植物病害知识库表。

表 4-4 植物病害知识库表

列名	数据类型	长度	说明
id	varchar	20	主键
name	varchar	20	病虫害名称
harm	varchar	1000	危害
symptom	varchar	1000	症状
pmethods	varchar	1000	防治方法
image	varchar	50	图片路径

病虫害防治表中存储了处理病虫害的方法包含化学防治、物理防治、生物防治、人工防治等，如表 4-5 病虫害防治信息表。

表 4-5 病虫害防治信息表

列名	数据类型	长度	说明
id	varchar	20	主键
name	varchar	20	名称
preventmethod	varchar	500	防治方法
did	varchar	20	外键

用户使用系统查询功能后数据会记录到搜索记录表如表 4-6 所示。

表 4-6 搜索记录表

列名	数据类型	长度	说明
did	int	10	主键
name	varchar	20	分类名称
count	int	0	次数
time	varchar	30	时间

4.8 系统包类图设计

(1) 本项目采用四层的架构体系，包含了交互层、业务服务层、数据访问层及领域模型层。交互层使用 Controller 组件处理 HTTP 的请求，其中病害历史与知识库控制器继承抽象基类可以进行实现统一拦截机制。并可以借助@Autowired 完成服务接口注入。

(2) 业务服务层基于门面模式封装核心逻辑，可以使用 `ServiceImpl` 实现类既履行服务契约又集预热策略。

(3) 系统之成 Redis 缓存中的数据持久层依托 MyBatis 动态 SQL 执行引擎进行操作，导航结合分页插件实现高效数据检索。

(4) 系统通过 Spring IoC 容器实现组件依赖治理。架构层面严格遵循接口隔离原则，UML 图示中<<interface>>标注契约接口，<<component>>标记实现类，虚线箭头表征跨层调用拓扑。如图 4-9 网站包类图。

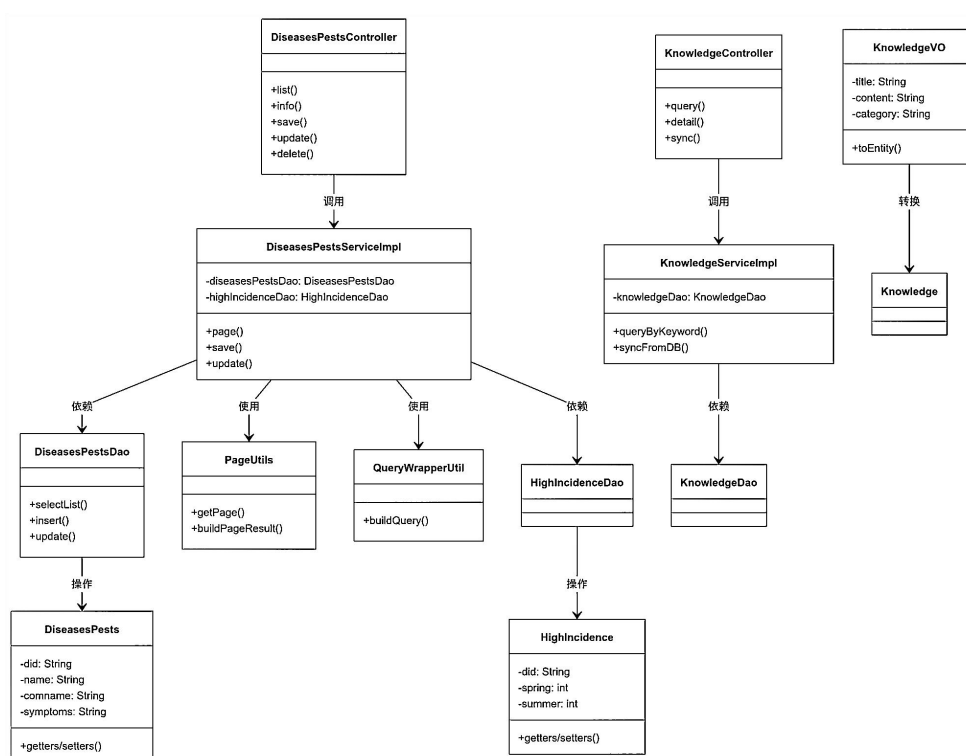


图 4-9 网站包类图

4.7 系统搜索功能

(1) 本项目的搜索功能采用 Elasticsearch 分布式搜索引擎实现，用户在搜索框输入关键词（如"水稻病害"）后，前端通过防抖机制延迟 300 毫秒发起 AJAX 请求，携带关键词、分页参数及筛选条件（如发生季节、危害等级）到后端 Spring Boot 服务。

(2) 关于增强数据库查询功能，提高搜索效率以及响应时间。加工完善之前要进行预处理工作，这属于准备工作的一种，优化 SQL 语句和数据库设计是提高数据库访问效率的根本。开发者应尽量避免全表扫描和使用复杂的连接查询，优化查询条件和索引的使用方式。

(3) 此外，合理规划数据库的表结构和关系模型，减少冗余数据和复杂的关联关

系，可以大幅度提高数据库的查询效率。例如:如果数据库需要发送 SQL 命令，那么需要先通过 Java 数据库进行内部分析。如果 Java 能够进行预处理，就可以直接将 SQL 命令反馈出去，这样数据库就能够加快访问速度，也能够缓解解释器的负担。

如图 4-10 搜索时序图。

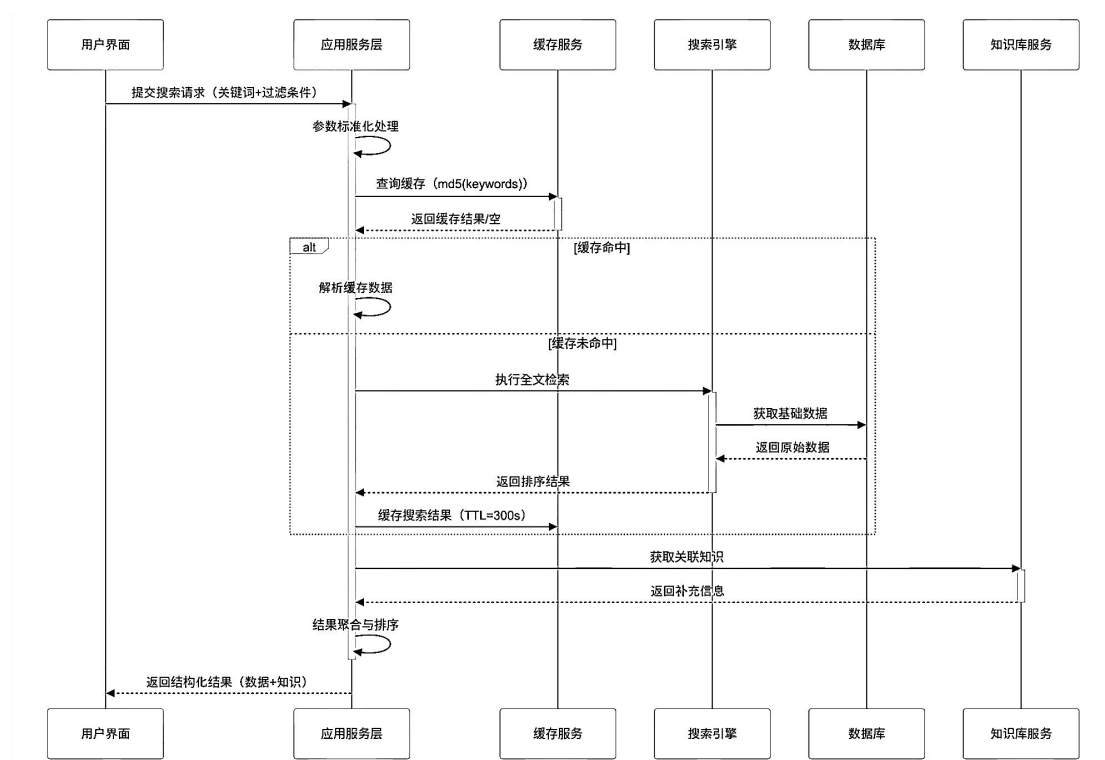


图 4-10 搜索时序图

4.8 本章小结

本章描述了系统设计采用生态数据融合框架，在农业场景实现病害识别、与虫害检测的协同应用，自然保护区场景支持生物群落分布联合分析。对于优化系统程序方案遵循以下几点：

(1) 优化前端代码：减少 http 请求的主要途径是对 css、js 以及图片进行合并，将多个资源文件合并成一个，这样优化处理后在访问页面时就能有效减少请求数。

(2) 采用网上在线的压缩工具 jQuery MiniUI 自己压缩或者通过 webpack 等打包工具进行压缩处理。后端优化：优化代码结构，从代码编写上提高性能。

(3) 缓存优化 缓存有两大类：本地缓存、缓存服务中间件（如 Redis、Memcache 等）。

第 5 章 系统实现

5.1 核心功能实现

系统开发环境的具体信息表如下 5-1 所示。

表 5-1 系统开发环境详情表

开发环境	详情
操作系统	Windows11
开发语言	Java
数据库	MySQL
开发框架	Springboot
集成环境	Idea

5.1.1 百度 AI 开放平台介绍

百度 AI 开放平台是面向企业和各开发者的，综合新人工智能服务平台，提供了多模态的 AI 能力、行业解决方案以及开发工具，覆盖各个方面，如视觉、语音、自然语言处理多个领域。其中本系统的图像识别等服务，支持超过 10 万类物体/场景识别，细分领域包括植物、动物、车型等专项检测，准确率超过 95%。如图 5-1 百度 AI 开放平台界面图。



图 5-1 百度 AI 开放平台界面图

5.1.2 账号注册与应用创建

访问官网（ai.baidu.com）并注册登录，在控制台选择所识别的服务如人脸识别、图像识别等，创建新的应用并填写相关信息。创建完成之后，系统将会自动生成应用的 API Key 和 Sercet Key, 用于后续系统调用 API 鉴权。再调用百度 OAuth 接口，通过 API Key 和 Sercet Key 获取 access_token 获取访问令牌进行调用。如图 5-2 应用创建图所示。

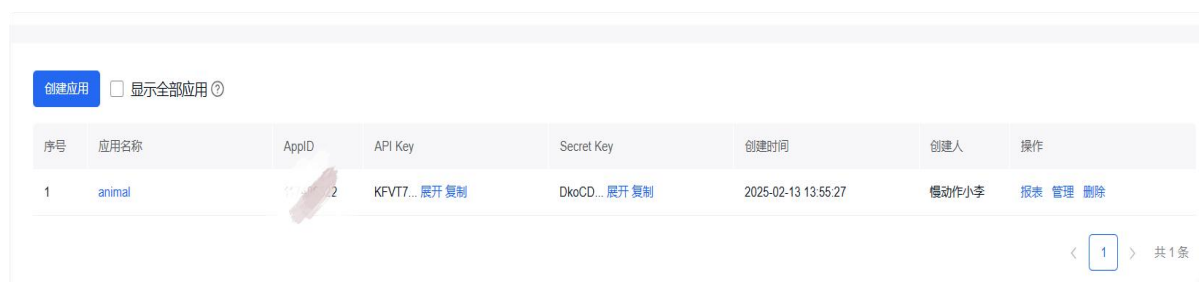


图 5-2 应用创建图

5.2 病害识别检测

完成数据集收集处理，并得到关键 API Key 和 Sercet Key。基于百度 AI 识别病虫害系统功能得以实现，用户终端发出请求，FormData 提交到控制器 DiseasesPestsController 层，通过 FileUtil+Base64Util 生成 Base64 数据，通过网络 HttpUtil 发出 POST 请求到百度 AI 接口 Animal/Plant API，完成检测。如图 5-3 病害识别技术流程图。

访问接口代码：

```
params = {
    'grant_type': 'client_credentials',
    'client_id': API_KEY,
    'client_secret': SECRET_KEY
}
response = requests.post(ACCESS_TOKEN_URL, params=params)
access_token = response.json().get('access_token')
```

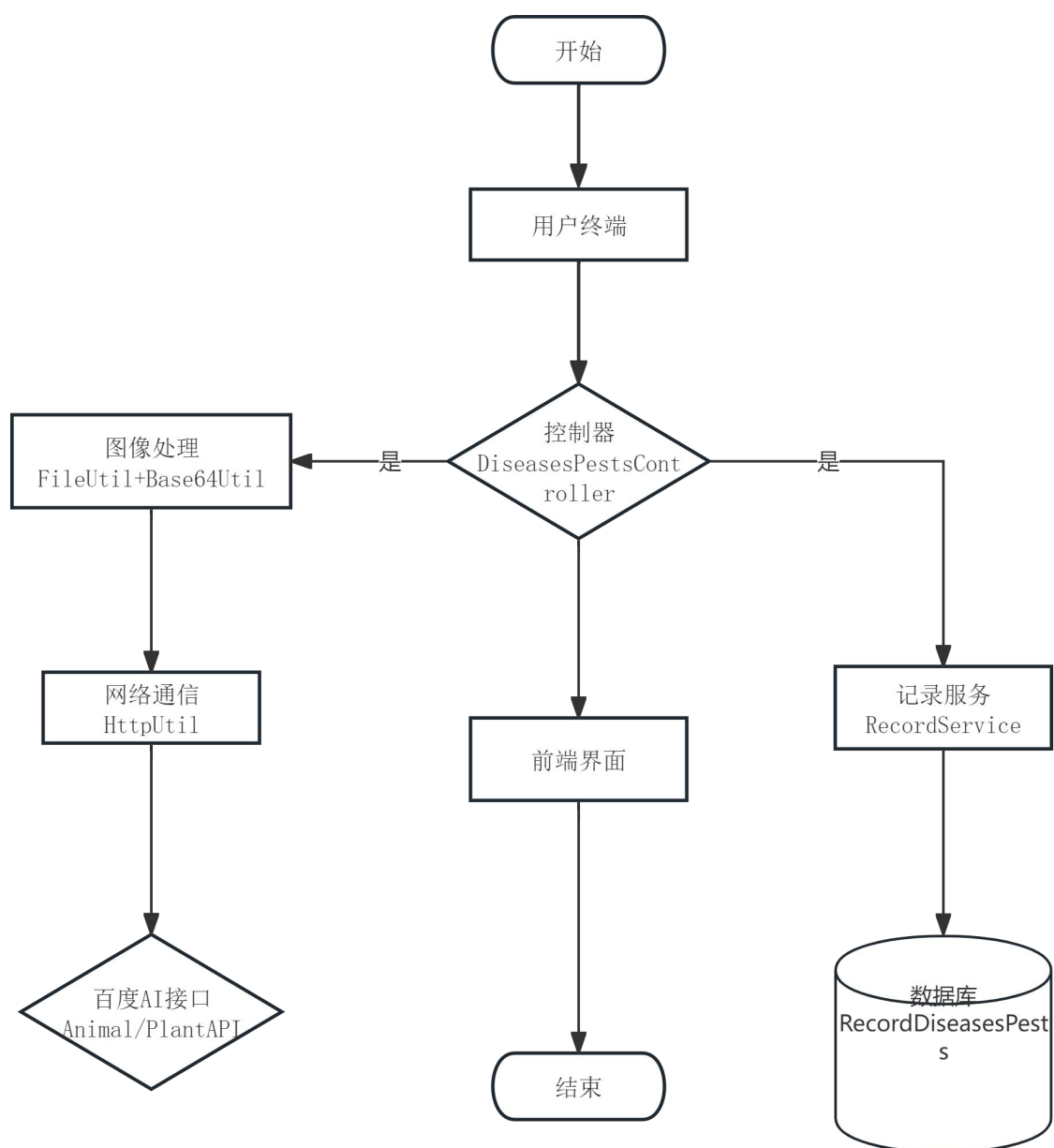



图 5-3 病害识别技术流程图

5.2.1 首页界面模块

用户使用该网站系统时的主界面，用户可以在此界面选择使用的功能。在网站设计中运用图形，拓展浏览者的思维以及实现信息的有效传递。这些图形格式能够在大部分的浏览器中进行运用。如图 5-4 系统首页界面图。

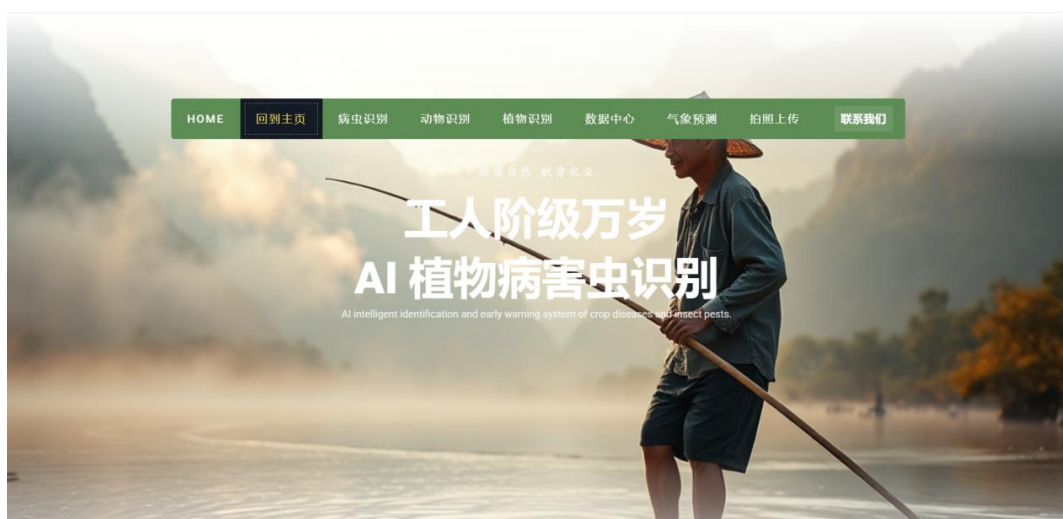


图 5-4 系统首页界面图

5.2.2 病虫害识别功能界面

用户进入系统主界面后，点击病虫害识别按钮，进入到病虫害识别功能界面，根据功能选择对应 API 接口，准备输入数据如图文件、JPG、PNG 等图片格式。用户可以在此界面上上传图像，以识别病害、以及病虫，如图 5-5 病虫害识别功能界面。



图 5-5 病虫害识别功能界面

系统检测后，系统接受 JSON 格式的返回数据结果，提取识别内容，用户进行病害识别后，成功并显示出用户查询相关病害的信息、病害名称、俗名、防治方法、形态特征等相关信息植物病害识别后的结果展示。如图 5-6 植物病害识别结果图。



图 5-6 植物病害识别结果图

用户想查询更为详细的病害信息，点击防范措施推荐，即可查看关于病害的防治方法。如图 5-7 防范措施推荐图。



图 5-7 防范措施推荐图

5.2.3 植物识别功能界面

用户在上方导航栏上可以点击植物识别，系统跳转到植物识别功能界面，系统跳转到动物识别功能界面，准备输入数据如图文件、JPG、PNG 等图片格式。用户可以通过上传植物图像来进行植物识别。如图 5-8 植物识别功能界面图。



图 5-8 植物识别功能界面图

5.2.4 病害数据查询界面

用户点击关于病害数据查询功能在侧边导航栏中，选择病害数据查询，可以在此界面查询想知道的病害数据以及防止信息等，可以了解病害的名称，发病原因，预防，以及治疗方法。如图 5-9 病害数据查询界面。

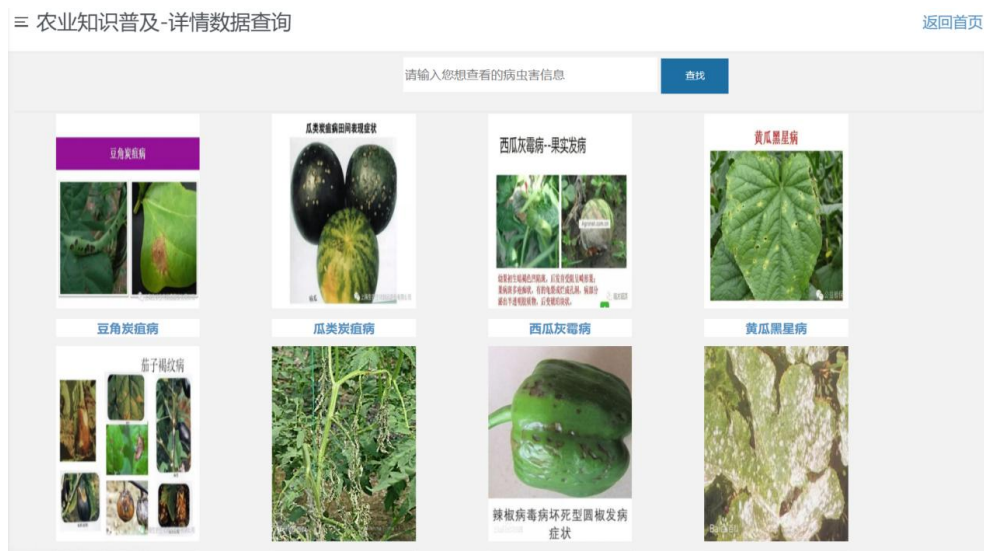


图 5-9 病害数据查询界面

5.2.5 病虫害数据可视化界面

将中国农业农村信息网信息病害数据设计成全面可视化图标，方便用户查看，将昆虫、农作物、产量等数据通过可视化界面，呈现给用户，用户在上方导航栏上可以点击病害数据，系统跳转到可视化功能界面，如图 5-10 病虫害数据可视化界面。

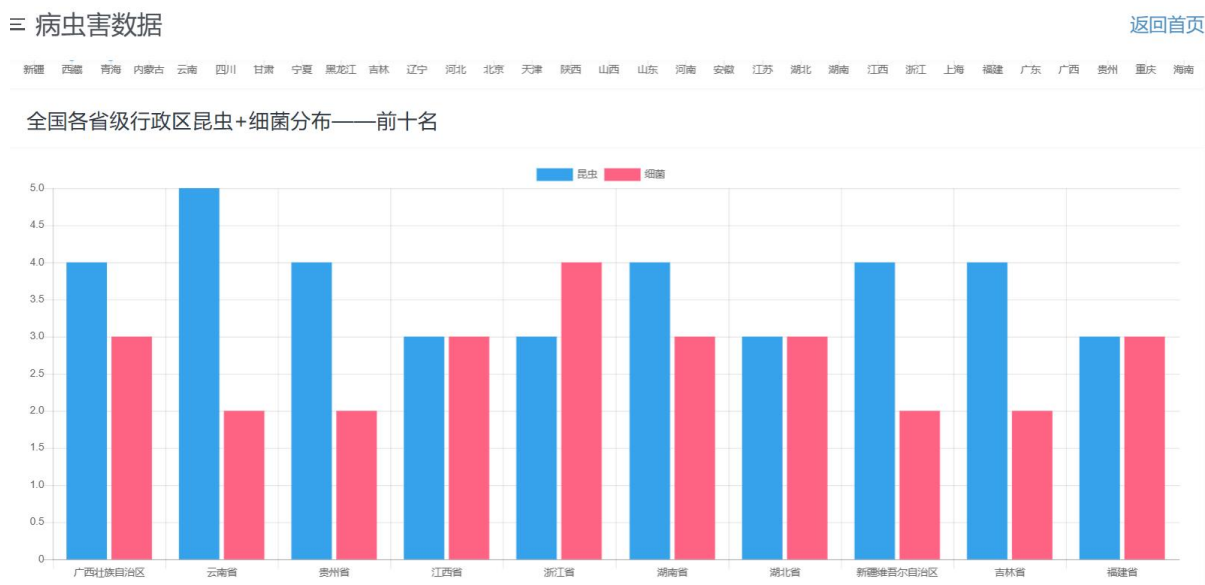


图 5-10 病虫害数据可视化界面

5.3 本章小结

本章详细阐述了系统的三大核心功能模块：

(1) 病虫害智能识别：运用百度识别接口调用，系统可以支持用户上传作物病斑照片，后系统进行数据库信息图像比对，匹配数据库并返回病虫害种类、危害特征及发生规律，识别准确率达 90%以上。

(2) 推荐防治案例：根据参考文献，结合知识库与本地数据库，系统可以根据病虫害类型，以及作物生长阶段及环境数据，动态生成包含生物防治、化学用药、解决方案。

(3) 信息搜索数据管理：后端建立了结构化病虫害数据库，可以进行历史记录查询。

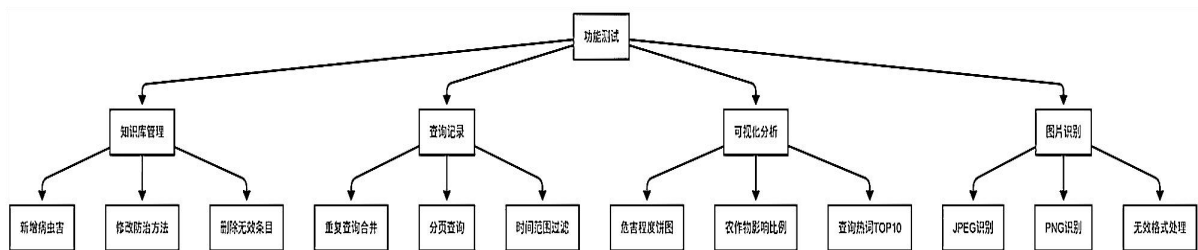
第 6 章 系统测试

6.1 测试目的

- (1) 验证病虫害知识库的 CRUD 功能完整性。
- (2) 确保查询记录智能合并与统计逻辑正确性。
- (3) 校验图片识别与知识库匹配的准确性。
- (4) 验证数据可视化展示的完整性和实时性。

6.2 测试设计

如图 6-1 测试设计图。



6.3 功能测试用例

系统功能测试将系统功能进行逐一测试，并输出功能结果，对比预期结果是否与实际结果匹配。如表 6-1 病害识别测试表。

表 6-1 病害识别测试表

测试项	输入数据	预期结果	实际结果
图像识别	稻纵卷叶螟成虫图	返回正确虫害信息	通过
数据导出	选择 CSF 格式	生成可下载文件	通过
上传玉米螟标准图片	2MB JPEG 清晰图片	返回匹配度>90%的结果	通过
上传非图片文件	test.pdf 文档	返回"不支持的格式"错误	通过

如表 6-2 查询记录测试表。

表 6-2 查询记录测试表

用例编号	测试场景	测试步骤	预期结果
TC101	连续 3 次查询"稻飞虱"	1.间隔 5 分钟 执行相同查询	记录合为 1 条 count=3
TC102	分页查询(每页 5 条)	1.构造 10 条记录 2.请求第 2 页	返回 5 条记录, 页码正确

续上表			
用例编号	测试场景	测试步骤	预期结果
TC103	跨天查询统计	1.创建不同日期的记录 2. 按周范围过滤	正确返回时间段内的聚合数据

如表 6-3 病害知识库测试表。

表 6-3 病害知识库测试表

用例编号	测试场景	测试步骤	预期结果
TC001	新增水稻纹枯病记录	1.填写完整症状描述和防治方案 2. 提交保存	知识库记录数+1, 可前端查询到
TC002	修改棉铃虫防治方法	编辑现有条目 2. 更新生物防治方案	防治方法字段显示最新内容
TC003	删除不存在的病虫害条目	输入无效 did 执行删除操作	返回"记录不存在"提示

6.4 性能测试

- (1) 平均响应时间：识别服务 1.2s，数据查询 0.3s
- (2) 最大并发支持：82 请求/秒（Tomcat 线程池优化后）
- (3) 测试 LayUI、ECharts 等大型 JS 库加载效率
- (4) 检查 index.jsp 中多个轮播大图（crops_6.png 等）的加载优化
- (5) 测试 ECharts 饼图（病虫害分类统计）在万级数据量下的渲染速度

测试核心接口的性能，图片识别接口测试路径：DiseasesPestsController#BasicTesting 测试场景：模拟并发上传图片（200-500 并发）测试 Base64 编码转换、百度 AI 接口调用、数据库查询等环节耗时，使用工具：JMeter。

数据查询接口：测试路径：KnowledgeController 数据查询相关接口，关注点：分页查询性能（pagehelper 分页插件效率）大表关联查询性能（如 DiseasesPests 表关联查询）。

6.5 压力测试

系统进行压力测试的目的是为了测试系统通过模拟超常规负载（如高并发、大流量），暴露性能瓶颈、资源缺陷和容灾漏洞，提前修复问题以保障高负载下的稳定性，避免服务崩溃及业务损失。如表 6-4 压力测试表。

表 6-4 压力测试表

场景	并发量	预期结果	关注点
图片识别接口	200	RT < 3s, 吞吐量 > 50 req/s	文件上传带宽、AI 接口限流
知识库复杂查询	300	RT < 2s, CPU 使用率 < 80%	SQL 索引有效性
首页静态资源加载	1000	完全加载时间 < 3s	CDN 加速效果

6.6 本章小结

在本章中，我们对整个系统进行了深入而全面的测试工作，涵盖了从单元测试到集成测试，再到系统测试的各个层面。这些测试的目的是确保系统中的每一个模块都能正常运行，并且满足既定的功能需求。首先，我们对系统的核心功能进行了严格的测试。这包括了延迟测试，以确保系统在处理请求时能够达到预期的响应速度。此外，数据搜索功能的测试也至关重要，我们通过接口测试来验证前后端数据交互的准确性和可靠性。这些测试的结果表明，系统在数据处理和响应时间方面均表现稳定，完全符合设计要求。在测试过程中，我们还特别关注了用户体验。通过模拟真实场景下的操作，我们评估了系统的易用性和交互设计，确保用户能够顺畅地完成各项任务。测试结果表明，系统的各项功能均达到了预期目标，用户界面友好，操作流畅，数据处理能力强大。为了进一步确保系统的稳定性和可靠性，我们还进行了兼容性测试。这项测试覆盖了包括 IE、360、Google Chrome、火狐等在内的主流浏览器，以及各种不同的设备。测试结果令人满意，系统能够在这些不同的环境中正常运行，没有出现兼容性问题。此外，我们还对系统的安全性进行了评估，确保所有的数据传输都是加密的，防止敏感信息泄露。通过一系列的安全测试，包括 SQL 注入、跨站脚本攻击等常见的网络攻击手段，我们确认系统具备了足够的防护措施，能够有效地抵御潜在的安全威胁。

结 论

本项目构建的植物病虫害系统设计与实现，项目本基于 Spring 框架来实现分层架构，整合了 JSP 与 Bootstrap 的前端技术来构建响应式界面。采用 MySQL 数据库存储病虫害知识库与用户查询记录。

项目后端呈现是通过 RecordServiceImpl 实现智能频次统计逻辑，前端利用 ECharts 来生成动态的图表，数据库设计采用双表结构分别存储病虫害属性与查询行为数据。系统测试验证了知识库管理、查询合并机制、图片识别准确率等核心功能。通过自动化测试覆盖业务主流程，确保系统稳定性与数据一致性。

系统的创新功能是调用百度 AI 平台的 API 接口来识别植物病虫害，相比于传统的病虫害模型识别，大大的减少了工作量，并且百度开放平台调用识别次数超过 800 万次，给用户使用提供了稳定又可靠的支持。系统还引入了 Redis 缓存策略，极大的缓解了系统接口重复调用的冗余问题，提升了识别的速度。涵盖了病虫害信息数据的标准化管理、智能查询记录统计（自动合并重复查询并生成频次分析）、多维度数据可视化展示（危害程度饼图、热词排行、农业产量的数据等）以及病虫害图像智能识别（支持 JPEG/PNG 格式解析与知识库匹配）。

系统不足之处在于，未达到实时监测识别病害的程度，以及实时报警提醒虫害的到来，还有极端突发天气对动植物的影响。后续将会优化系统结构，加入硬件设备，如红外摄像，声波分析，湿度检测，太阳能板等硬件，去完善植物病虫害识别系统。

本项目植物病虫害系统设计与实现平台，为农业从业者提供病虫害快速识别、防治方案查询、趋势分析，天气预警等数字化服务，有效提升农田病虫害防治效率与决策科学性。

参考文献

- [1] 刘扬, 欧阳子涵, 武彦. 2024, AI 开始改变世界[N]. 环球时报, 2024-12-26(008).
- [2] 冯浩原, 张健臻, 江得权, 等. 基于网站的项目系统开发[J]. 科技与创新, 2024, (22): 57.
- [3] 林诗琪, 巴丽合亚·卡里布汗, 热阿宛·吾肯: 基于图像识别技术的新疆南疆棉铃虫防治系统研究[J]. 农业灾害研究, 2024, 14(04): 28-30.
- [4] 彭鑫. 基于深度学习的农业害虫识别方法研究[D]. 吉林化工学院, 2024.
- [5] 刘永志. 基于 B/S 架构的项目管理系统的网络安全模式设计[J]. 信息与电脑(理论版), 2023, 35(07): 149-151.
- [6] 张牧琢, 张隽霏. 基于百度 AI 的物体图像识别软件的设计与实现 [J]. 电脑知识与技术, 2023, 19 (33): 11-13.
- [7] Unique Landing Page Designs Can Make A Website More Engaging And Drive In More Traffic For Better Conversions And Thus Enhancing The Business[J]. M2 Presswire, 2020.
- [8] Kang W. A Study on the Current Learning Situation and Enhancement Strategies of Web Design Courses[J]. World Journal of Educational Research, 2024, 12(1).
- [9] 周文俊, 王国印. 基于深度学习的害虫识别系统设计与实现[J]. 现代计算机, 2024, 30(09): 104-107.
- [10] 陈文印, 吴科甲, 吴婧, 等. 百度 AI 环境下的植物生长阶段识别研究[J]. 福建电脑, 2023, 39(08): 26-31.
- [11] 杨鑫. Java 数据库访问效率的优化策略分析 [J]. 集成电路应用, 2024, 41 (10): 162-163.
- [12] 吴苡婷, 董真. 实现清晰快速识别大大提高检测效率[N]. 上海科技报, 2025-01-08(003).
- [13] 邓阳名, 顾潇, 梁爱媚. 基于 Java 语言的数据库访问技术应用研究[J]. 中国新通信, 2023, 25(08): 83-85.
- [14] 张晓娜, 李鑫, 张凯. 基于大数据平台的传统文化元素与网站界面设计的融合策略研究 [J]. 数字通信世界, 2024, (05): 27-29.
- [15] 张平, 李慧春. 高并发性能问题优化方案研究[J]. 信息技术与信息化, 2023, (11): 166-169.
- [16] 高国人, 张博怀, 时啟璐, 等. 图像处理系统的可视化界面设计与应用[J]. 自动化应用, 2025, 66(07): 233-236+240.
- [17] Nawaf L , Bentotahewa V . Optimization of cyber security through the implementation of AI technologies[J]. Journal of Intelligent Systems, 2025, 34(1):
- [18] 杨民峰, 孙洪迪. 基于百度 AI 开放平台的植物识别设计与实现[J]. 北京工业职业技术学院学报, 2023, 22(04): 20-24.
- [19] Hassan N . Effects of Grazing and Mowing on Grassland Soils, Plants and Insects: Community

Properties and Ecological Stoichiometric Linkages[D]. 东北师范大学, 2020.

- [20] 余敏, 等. 基于图像的植物病害识别研究进展[J]. 农业研究与应用, 2024, 37(Z1): 309-319.

致 谢

在本论文的完成过程中，我首先要向我的导师致以最诚挚的感谢。导师在论文的选题、研究思路的确定以及写作过程中给予了我悉心的指导和宝贵的建议，他的严谨治学态度和渊博的学识让我受益匪浅。导师不仅在学术上给予了我极大的帮助，还在生活中给予了我关心和鼓励，让我在遇到困难时能够坚持不懈地完成研究工作。同时，我也要感谢参与论文评审和答辩的各位老师，他们在百忙之中抽出时间对我的论文进行了细致的审阅，并提出了许多中肯的意见和建议，这些宝贵的反馈让我能够进一步完善论文的内容和质量。此外，我还要感谢亲爱的同学们，他们在研究过程中给予了我无私的帮助和支持，与我共同探讨问题、分享经验，为我的研究工作提供了许多灵感和动力。最后，我要感谢我的家人和朋友，他们在我撰写论文的过程中给予了我无微不至的关心和鼓励，让我能够全身心地投入到研究工作中。正是有了大家的支持和帮助，我才能够顺利完成这篇论文，在此向所有关心和帮助过我的人表示衷心的感谢！