**石河子大学**

**本 科 毕 业 论 文（设 计）**

不同水体识别系统设计与实现

|  |  |
| --- | --- |
| 学生姓名: | 梁嘉成 |
| 学 号: | 20201108005 |
| 学 院: | 信息科学与技术学院 |
| 专 业: | 计算机科学与技术 （第二学士学位） |
| 年 级: | 2020级 |
| 指导教师: | 尹小君 |

中国·新疆·石河子

2022年6月

**摘 要**

计算机水体识别技术，是利用计算机对图像进行处理、分析和理解，以识别各种不同模式的水体的技术，是应用深度学习算法实现图像识别的一种实践应用。本设计选用目前研究水体识别方法前沿领域中的深度学习模型，通过不同水体提取系统实现数据采集、特征提取、不同水体提取、和算法评价等模块。

通过选用合理的深度学习模型，可以完成水体的模式识别和评价工作，通过与之低耦合的系统设计，进而实现不同水体识别系统。深度学习模型采用……。系统总体设计采用Web应用B/S结构实现，简化系统开发、维护和使用；前端选用Vue框架实现前后端分离的组件化编程，借助Element UI等工具快速搭建前端项目；后端采用Java语言开发，使用成熟的Spring Boot框架，整合MyBatis、Junit等其他开源框架，完成系统功能的实现；前、后端通信使用Axios框架完成异步通信工作，持久层采用Redis缓存技术和MySQL数据库完成数据持久化工作。

**关键词：**水体识别、深度学习、Vue、Spring Boot

目 录

[1. 绪论 1](#_Toc97566817)

[1.1. 研究意义 1](#_Toc97566818)

[1.1.1. 国外研究现状 1](#_Toc97566819)

[1.1.2. 国内研究现状 3](#_Toc97566820)

[1.2. 设计思路 4](#_Toc97566821)

[2. 相关技术原理 7](#_Toc97566822)

[2.1. 图像识别技术 7](#_Toc97566823)

[2.2. 具体实现 8](#_Toc97566824)

[2.3. 前端技术 8](#_Toc97566825)

[2.4. 后端技术 9](#_Toc97566826)

[3. 需求分析 13](#_Toc97566827)

[3.1. 可行性分析 13](#_Toc97566828)

[3.1.1. 技术可行性 13](#_Toc97566829)

[3.1.2. 经济可行性 13](#_Toc97566830)

[3.1.3. 操作可行性 13](#_Toc97566831)

[3.1.4. 时间可行性 13](#_Toc97566832)

[3.2. 系统功能性需求 13](#_Toc97566833)

[3.3. 非功能性需求 14](#_Toc97566834)

[4. 系统总体设计 15](#_Toc97566835)

[4.1. 系统总体设计原则 15](#_Toc97566836)

[4.2. 系统逻辑结构设计 15](#_Toc97566837)

[4.3. 整体结构设计 16](#_Toc97566838)

[4.4. 数据库设计 17](#_Toc97566839)

[4.5. 缓存设计 19](#_Toc97566840)

[5. 系统实现 20](#_Toc97566841)

[5.1. 系统框架 20](#_Toc97566842)

[5.2. 开发环境 20](#_Toc97566843)

[5.3. 前端实现 21](#_Toc97566844)

[5.4. 后端实现 21](#_Toc97566845)

[6. 系统测试 22](#_Toc97566846)

[6.1. 接口测试 22](#_Toc97566847)

[6.2. 单元测试 22](#_Toc97566848)

[6.3. 压力测试 22](#_Toc97566849)

[7. 总结与展望 23](#_Toc97566850)

# 绪论

## 研究意义

传统的对水体的识别方法中，模型结构层次较浅，计算量较小，计算时间较短，不需要以大量的图像为基础，即可完成图像识别的分析，但无法从原图像中获取更高层次的语义特征和深度特征，由于认为以及外界因素的干扰，图像识别率较低，面对大数据的情况下，无人为设计，无法获取图像特征，而采用深度学习的识别方法，能获取更深层次的图像特征，图像特征表达更为丰富无人为以及外界环境因素的干扰，图像特征提取更加准确图像识别正确率较高。

本设计选用目前研究水体识别方法前沿领域中的深度学习模型，通过不同水体提取系统实现数据采集、特征提取、不同水体提取、和算法评价等模块，对不同算法进行全方面评价，并应用于研究区域的不同水体提取。

### 国外研究现状

图像识别，是指利用计算机对图像进行处理、分析和理解，以识别各种不同模式的目标和对象的技术，是应用深度学习算法的一种实践应用。研究和图像识别技术对推动人工智能的发展具有重要的意义。

图像识别技术最早被提出约是源自二十世纪四十年代，但自提出以来直到九十年代，神经网络和支持向量机相结合，才促进了图像识别技术的发展，使得图像识别技术能够有更加广泛的应用，随着时间的推移，原来需要耗费大量人力对图像做预处理的工作，也逐渐由科学家们在研究中提出的深度学习模型取代，例如DBN、DFN、CNN、RNN等。

深度学习是图像识别领域的一种重要技术手段，其目的是通过构建一个多层网络，在这个网络上的计算机通过自动学习以得到数据隐含的内部关系，从而得到更加隐含和深刻的以数据形式表现的内容，因此深度学习在未来图像识别研究和应用中仍然是一个重要课题。

水体，水的集合体。水体是江、河、湖、海、地下水、冰川等的总称。水体识别系统的核心是通过深度学习的计算机图像识别技术对不同水体提取的相关算法和模型进行研究。目前该领域已经吸引了众多相关学者数十年的研究投入，存在众多不同的水体识别系统和算法，但大多都是针对某一特定水体或是某一特定功能的系统。

深度学习在图像识别中的模型有以下几类：

深层信念网络（Deep Belief Network，DBN）起源于人工神经网络，是一个概率生成模型，由多层受限玻尔兹曼机（RBM）和一层某种分类器组合而成，经典的DBN网络结构是由若干层RBM和一层BP 组成的一种深层神经网络。此深度模型广泛应用于图像分类识别，语音识别等领域。

卷积神经网络（Convolutional Neural Network，CNN）是一种特殊的深层前馈网络，CNN 模型主要包含输入层、卷积层、池化层、全连接层以及输出层。但是，在网络结构中，为了使输出更加准确，特征提取更加丰富，通常网络模型中使用多卷积层和多池化层相结合的网络模型，较为经典的CNN模型有LeNet-5、AlexNet、ZF-Net、VGGNet、GoogLeNet、ResNet以及DenseNet，上述CNN模型均是LeNet 的改进型模型。

循环神经网络（Recurrent Neural Network，RNN），又名时间递归神经网络，主要是用来解决序列数据问题。在RNN 结构模型中，网络会对之前时刻的信息进行记忆并且运用到当前的输出计算之中，相比于卷积神经网络、深度前馈网络，循环神经网络隐藏层之间的神经元是相互连接的，隐藏层中神经元的输入是由输入层的输出和上一时刻隐藏层神经元的输出共同组成。

生成式对抗网络（Generative Adversarial Network，GAN）是Goodfellow 等人于2014 年提出的一种生成式模型，通过在对抗过程中估计并生成模型的新框架，是近几年最成功的生成模型。GAN 主要由两部分构成：生成模型（G）和判别模型（D）。生成模型捕捉真实数据样本的潜在分布，并生成新的数据样本。判别模型是一个二分类器，判别区分输入的是真实数据还是生成的样本数据。判别模型输出是以概率值表示，概率值大于0.5 则为真，概率值小于0.5 则为假。当判别器无法区别出真实数据和生成数据时则停止训练，此时达到生成器与判别器之间判定误差的平衡，训练达到理想状态。

胶囊网络（Capsule Network，CapsNet）是Hinton等人在2017 年提出，是当前图像分类识别最前沿的技术之一。CapsNet 是在CNN的基础之上发展而来，解决了CNN对物体之间的空间辨识度差及物体大幅度旋转之后识别能力低下的两个缺陷。目前的CapsNet 结构较浅，是由卷积层、PrimaryCaps（主胶囊）层、DigitCaps（数字胶囊）层构成。

采用图像识别技术识别遥感图像是当前主流的水体识别分析方法，遥感图像作为良好、可靠、稳定的数据源，为图像识别技术提供了广泛的学习模型。遥感图像分类的主要依据是地物的波谱特征。地物波谱特征是指该地物对太阳辐射的反射, 散射能力随波长而变的规律, 地物波谱特征与地物的组成成份, 物体内部的结构关系密切。一般说来, 不同地物拥有不同的地物波谱特征, 据此可以将它们识别。

随着遥感技术被广泛应用于水体监测领域，水体信息提取方法成为热门研究方向。如Komeil 等利用Landsat TM、ETM+和OLI 遥感影像，模拟了2000~2013 年伊朗乌鲁米耶湖的时空变化；Adrian 等以澳大利亚东部的TM/ETM/OLI 影像为数据源，在比较七种水体指数的基础上，提出了一种简单精确的大范围水体自动分类方法。目前，用于水体信息提取的方法以单波段阈值法和多波段谱间关系法为主。单波段阈值法主要是利用水体与背景地物在遥感影像的某一波段反射率存在差异，能有效抑制背景地物，实现与背景地物相分离的目的，但单波段阈值法对不同时相、不同区域的水体需要设置不同的阈值，具有一定的局限性；多波段谱间关系法综合利用各波段信息，通过波段之间的组合，极大的增强了水体与其它地物反射率的差异，与单波段阈值法相比不受时空的影响，多波段谱间关系法中以水体指数法最为常见，如McFeeters提出了归一化差异水体指数（normalized difference water index, NDWI），能够抑制植被和土壤信息，实现增强水体信息的作用。

### 国内研究现状

徐涵秋针对NDWI 提取市区水体不理想，在NDWI 的基础上，提出了改进的归一化水体指数（modified normalized difference water index,MNDWI），陈文倩等基于高分一号4 个波段并结合决策树法，提出了阴影水体指数（shade water index，SWI），能有效的剔除阴影、裸地等背景地物对水体的影响，王瑾杰等在阴影水体指数的基础上，提出了改进的阴影水体指数（modified shade water index ,MSWI），进一步提高阴影与水体的分离程度，王小标等针对复杂环境下水体提取精度易受到低反射率地表影响的问题，利用ETM+影像，构建了多波段水体指数（multi-band water index，MBWI），但此方法很难提出与水体反射率接近的地物，王琳等利用Landsat8 影像，提出了双红外水体指数( doubleinfrared band waterindex，DIBWI)，用于准确识别蓝藻湖泊水体信息。

现阶段水体研究主要利用国外卫星数据，国外卫星数据存在时间/空间分辨率低、晴空数据少等问题，难以对水体进行精准监测，而我国的高分六号卫星,是我国自主研发的低轨光学遥感卫星，与国外数据相比，具有高时间分辨率、宽覆盖等特点，有利于湖泊水库的精细化动态监测。

## 设计思路

本系统总体设计采用Web应用B/S[[1]](#footnote-1)结构实现，简化系统开发、维护和使用；

系统中需要设计实现的模块功能如下图 1‑1。

系统设计遵循MVC模式[[2]](#footnote-2)，其优点有职责单一、互不影响，利于各功能模块解耦合、利于组件重用和二次开发。

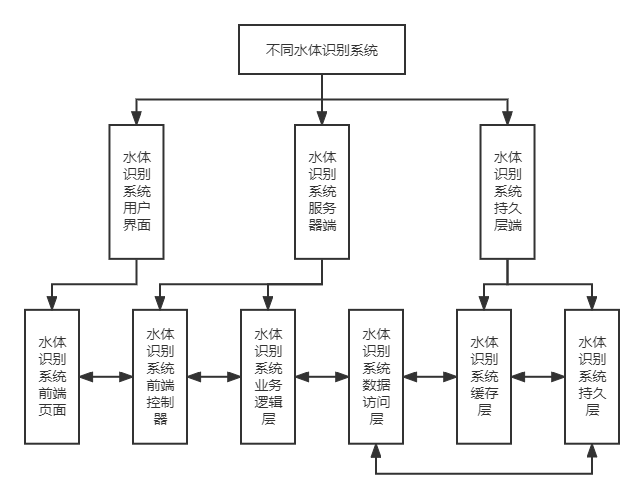


图 1‑1 不同水体识别系统设计思路图

### 水体识别系统用户界面

该模块的主要任务是实现与用户交互，完成用户的登录、注册、上传图像等交互功能。该模块设计的主要目的是实现水体识别系统前端页面的展示和构建、前端路由跳转的设计与实现、前端向后端发送请求和处理响应数据并展示的逻辑实现。

### 水体识别系统服务器端

该模块的主要任务是用于构建业务逻辑体系。

1. 前端控制器

前端控制器主要完成与前端的信息交互，实现处理前端请求并相应数据的功能。该模块设计的主要目的是处理前端请求数据，向下交付给业务逻辑层进行处理，并以响应形式返回业务逻辑层向上交付的数据给前端。

1. 业务逻辑层

该层是实现业务逻辑的核心层，主要包含两部分，设计的目的就是实现两部分功能。

一是实现核心功能，核心功能模块是水体识别的处理逻辑，这里是主要运用深度学习模型实现水体识别的环境，

二是实现用户功能，用户功能主要包括识别用户登录逻辑、识别用户权限逻辑等功能。

1. 数据访问层

该层是用于数据库和缓存交互，实现数据的增、删、改、查基本操作，并向上为业务逻辑层层提供服务，该层的主要任务是编写业务逻辑层需要请求数据库交互所需的全部接口。

### 水体识别系统持久层端

1. 缓存层

该层主要功能是处理一些经常被访问和操作的数据，以达到降低磁盘I/O操作导致的I/O阻塞平均时长，以加快程序运行速度。该层的主要任务是实现数据访问层提供的接口。

1. 持久层

该层是将数据进行持久化操作的一层，通过合理的技术选型和建库建表等操作可以大大降低占用的外存存储空间。该层的主要任务是实现数据访问层提供的接口。

# 相关技术原理

## 图像识别技术

深度学习模型存在各自优缺点，下表 2‑1阐述了不同深度学习模型各自存在的优缺点。

由于卷积神经网络训练参数少，模型的泛化能力更强，池化运算降低网络的空间维度，对输入数据的平移不变性要求不高，根据水体信息综合以上信息考虑，为解决遥感影像分辨率高、信息量大而导致信息识别提取精度不高的问题，本项目选用卷积神经网络进行水体识别系统的设计。

水体识别系统依赖的数据源可以选用北斗卫星导航系统。北斗卫星导航系统是中国着眼于国家安全和经济社会发展需要，自主建设、独立运行的卫星导航系统，是为全球用户提供全天候、全天时、高精度的定位、导航和授时服务的国家重要空间基础设施。

表 2‑1 经典网络结构对比分析

| 方法 | 主要任务 | 优点 | 缺点 |
| --- | --- | --- | --- |
| DBN | 特征 提取 | 能够反映同类数据本身的相似性 | 对于分类问题，分类精度不高  某些学习的复杂性较高  输入数据具有平移不变性 |
| CNN | 特征 提取 | 训练参数减少，模型的泛化能力更强  池化运算降低网络的空间维度，对输入数据的平移不变性要求不高 | 容易出现梯度消散问题  空间关系辨识度差  物体大幅度旋转之后识别能力低下 |
| RNN | 特征 提取 | 可以对序列内容建模 | 需要训练的参数较多，容易出现梯度消散或梯度爆炸问题  不具备特征学习能力 |
| GAN | 生成对抗样本 | 与其他模型相比，能够产生更好的样本  可以训练任何一种生成器网络  不需要设计遵循任何种类的因式分解的模型，任何生成器网络和任何鉴别器都会有用  避免了复杂的马尔科夫过程，回避了近似计算棘手的概率难题 | 模型的收敛性较差  训练过程中，容易出现崩溃问题  由于无需预先建模，模型过于自由不可控 |
| CapsNet | 特征提取 | 解决了CNN模型出现的空间关系辨识度差以及物体大幅度旋转之后识别能力低下 | 模型的网络结构较浅，在图像识别分类上的正确率和目前流行的CNN模型还是有很大差距 |

## 具体实现

系统实现选择前后端分离的开发模式，前后端分离模式可以避免在后端项目中添加前端页面，导致逻辑结构不清、维护和二次发开困难，有利于实现高并发、高性能、高可用，后端编写维护接口文档，在API变化时更新接口文档，后端根据接口文档进行接口开发，前端根据接口文档进行开发，开发完成后联调和提交测试。

## 前端技术

### Vue-CLI

前端选用Vue.js框架实现，Vue是一套用于构建用户界面的渐进式JavaScript框架。与其它大型框架不同的是，Vue 被设计为可以自底向上逐层应用。Vue 的核心库只关注视图层，不仅易于上手，还便于与第三方库或既有项目整合。另一方面，当与[现代化的工具链](https://cn.vuejs.org/v2/guide/single-file-components.html)以及各种[支持类库](https://github.com/vuejs/awesome-vue#libraries--plugins)结合使用时，Vue 也完全能够为复杂的单页应用提供驱动。

Vue的开发模式还借鉴了MVVM[[3]](#footnote-3)模型使得程序员可以将注意力完全集中在数据上，用数据驱动视图，而数据和视图间的维护工作则交给Vue进行。借助强大的Webpack[[4]](#footnote-4)打包工具实现的Vue-CLI，我们可以实现组件化模式的编程，以大大降低代码复用和模块耦合关系。

尽管截至2022年尤雨溪团队已经宣布Vue3成为Vue的默认版本，但是为了保证本项目的稳定性，本项目仍然采用Vue2实现前端应用。

### Vue-Router

采用Vue实现单页面应用时，需要进行的前端路由跳转可借助Vue-Router插件实现，这简化了路由跳转的开发难度，增加了视图的多样性，不仅可以给用户良好的体验，同时也实现了路由跳转逻辑和前端编码工作的解耦合，另外也可以借助路由守卫、JWT[[5]](#footnote-5)技术实现用户登录状态的认证，避免了采用Session认证的诸多弊端。

### Vuex

为了方便不同组件之间的消息和数据共享，可以借助Vuex插件实现。Vuex 可以帮助我们管理共享状态，并附带了更多的概念和框架。这是对短期和长期效益进行权衡后的选择。

### Axios

由于Vue只注重视图的展示，且是单页面应用，通信时主要采用AJAX，我们就需要一个AJAX框架以实现前后端的通信工作。这里我们选用Axios框架，Axios是一个基于 promise的HTTP库，可以用在浏览器和 node.js[[6]](#footnote-6) 中。

## 后端技术

### Java EE

后端选用Java作为开发语言，其优势有如下四点。

1. Java是全场景开发语言。采用Java开发一个比较方便的事情是技术选型可以完成从前端、移动端到后端的整体解决方案，这是目前不少开发团队选择Java语言的一个重要原因。
2. Java语言的生态体系比较健全。Java语言经过多年的发展，已经形成了一个健全的语言生态体系，这会在很大程度上降低程序开发的潜在风险，也能够在很大程度上保证项目的开发周期。随着Java语言在云计算和大数据领域的广泛应用，未来Java语言在生产环境下的应用依然有广阔的前景。
3. 稳定的性能和较强的扩展性。Java语言虽然在代码实现上相对繁琐一些（与Python相比），但是Java语言的性能却比较稳定，而且Java语言的扩展性也比较强，这些因素决定了大型互联网平台往往更愿意采用Java语言。
4. 开发人员基数大。目前IT行业内掌握Java编程语言的程序员非常多，这使得搭建Java开发团队非常方便，这也是不少开发团队愿意采用Java语言的一个重要原因。

Java EE是指Java Enterprise Edition，Java企业版，用于企业级开发，是Java的一套开发规范，包括Web开发等。目的是帮助程序员开发和部署可移植、健壮、可伸缩且安全的服务器端Java应用程序。

### Spring、Spring MVC

而选用Java就不得不提到Spring框架，Spring 是一款目前主流的 Java EE 轻量级开源框架 ，是 Java 世界最为成功的框架之一。其优势如下

1. 非侵入式设计

Spring是一种非侵入式（non-invasive）框架，它可以使应用程序代码对框架的依赖最小化。

1. 方便解耦、简化开发

Spring就是一个大工厂，可以将所有对象的创建和依赖关系的维护工作都交给Spring容器的管理，大大的降低了组件之间的耦合性。

1. 支持AOP

Spring提供了对AOP的支持，它允许将一些通用任务，如安全、事物、日志等进行集中式处理，从而提高了程序的复用性。

1. 支持声明式事务处理

只需要通过配置就可以完成对事物的管理，而无须手动编程。

1. 方便程序的测试

Spring提供了对Junit4的支持，可以通过注解方便的测试Spring程序。

1. 方便集成各种优秀框架

Spring不排斥各种优秀的开源框架，其内部提供了对各种优秀框架（如Struts、Hibernate、MyBatis、Quartz等）的直接支持。

1. 降低Java EE API的使用难度

Spring对Java EE开发中非常难用的一些API（如JDBC、Java Mail等），都提供了封装，使这些API应用难度大大降低。

Spring MVC 是 Spring 提供的一个基于MVC设计模式的轻量级Web开发框架，本质上相当于Servlet[[7]](#footnote-7)。

### Spring Boot

Spring Boot是所有基于Spring开发的项目的起点。Spring Boot的设计是为了尽可能快的跑起来 Spring 应用程序并且尽可能减少配置文件。简单来说就是Spring Boot其实不是什么新的框架，它默认配置了很多框架的使用方式。

### Maven

Maven是一个项目管理工具，可以通过一小段描述信息来管理项目的构建，报告和文档的项目管理工具软件；它包含了一个项目对象模型，一组标准集合，一个项目生命周期，一个依赖管理系统，和用来运行定义在生命周期阶段中插件目标的逻辑。

本项目采用Maven作为项目管理工具。

### Tomcat

Tomcat 服务器是一个免费的开放源代码的Web 应用服务器，属于轻量级应用[服务器](https://baike.baidu.com/item/%E6%9C%8D%E5%8A%A1%E5%99%A8)，在中小型系统和并发访问用户不是很多的场合下被普遍使用，是开发和调试JSP 程序的首选。

### MyBatis

MyBatis是一款持久层框架，它支持自定义SQL、存储过程以及高级映射。MyBatis免除了几乎所有的JDBC代码以及设置参数和获取结果集的工作。MyBatis可以通过简单的 XML或注解来配置和映射原始类型、接口和Java POJO[[8]](#footnote-8)为数据库中的记录。

### Redis

Remote Dictionary Server，即远程字典服务，是一个开源的使用ANSI [C语言](https://baike.baidu.com/item/C%E8%AF%AD%E8%A8%80)编写、支持网络、可基于内存亦可持久化的日志型、Key-Value数据库，并提供多种语言的API。

### MySQL

MySQL是最流行的关系型数据库管理系统，在WEB应用方面MySQL是最好的 RDBMS[[9]](#footnote-9)应用软件之一。

## 图像识别技术

### OpenCV

OpenCV是一个基于Apache2.0许可（开源）发行的跨平台[计算机视觉](https://baike.baidu.com/item/%E8%AE%A1%E7%AE%97%E6%9C%BA%E8%A7%86%E8%A7%89/2803351" \t "_blank)和[机器学习](https://baike.baidu.com/item/%E6%9C%BA%E5%99%A8%E5%AD%A6%E4%B9%A0/217599)软件库。

### PyTorch

## 其他技术

### Git

Git是一个分布式版本控制系统，旨在快速高效地处理从小型到超大型项目的所有内容。

# 需求分析

## 可行性分析

### 技术可行性

图像识别技术诞生80年来，已经形成了非常成熟的理论和技术基础，相关文献记载的关于水体识别技术的应用也不胜枚举，本系统图像识别算法借助相关文献和开源平台资料，选用现阶段比较成熟的开源算法作为水体识别的基础，以现阶段发展成熟的Java Web技术作为后端开发，选用Spring MVC框架，以易于上手的Vue作为前端框架，数据存储选用开源、高效且成熟的MySQL库保证了系统技术上的可行性。

### 经济可行性

本项目在研发阶段开发算法和环境均来自相关文献资料和开源社区，开发时间预计为半年以内，相关开发软件中仅有IntelliJ IDEA需要支付每年约￥1300费用，网络方面10Mbps带宽网络需要每年向电信运营商支付￥1000费用，经济上可行。项目投入运营阶段需要申请域名、服务器等相关服务和资源，运营维护阶段可根据访问量和需求进行系统迭代和变更，但核心算法框架一般情况下无需迭代升级，经济上完全可行。

### 操作可行性

本系统实现需要学习运用的知识面贯穿前端、后端、数据库和算法，是对本科阶段所学知识的一个综合应用，操作上可行。

### 时间可行性

通过制定充分的开发计划，结合项目管理技术的应用，项目在时间周期安排上可行。

## 系统功能性需求

### 数据库需求

数据库系统用以存储必要的数据，本系统中最重要的需求是存储图片在本地的URL和图片状态等必要信息。由于系统需要识别用户和用户状态等信息，所以数据库中也要存储用户信息。

### 前端交互

前端用于和用户直观交互，以实现一个亲和用户的平台。

### 后端业务逻辑

后端用来处理登录请求、图片识别请求等逻辑业务。

### 图像识别处理

用来处理分析图像，并标记特征并返回给后端业务逻辑，进行处理。

## 非功能性需求

### 仓库系统

用来存储用户图片的仓库。

### 收费系统

实现按图片处理数量收费。

### 系统展示

将系统功能进行展示。

# 系统总体设计

## 系统总体设计原则

抽象：描述物体本质的设计技术即为抽象，它忽略了不重要的层面。抽象将一些复杂事情简化到可以分析和理解的程度。在软件工程学科中，从定义软件到开发系统经历了许多步骤。每个步骤都可以看作是软件解决方案抽象层次的细化。抽象的底层是软件的源代码。在模块化设计中，也可以有多个抽象层次。抽象级别最高的模块一般用概括的办法描述问题的解决方案，抽象级别较低的模块详细描述了抽象较高的问题的解决方案。

模块化：程序中的模块总是程序对象的集合，例如数据描述和可执行语句，或者是单独定名和寻址的元素。模块化是将一个要开发的软件分成几个简单的小部件模块，每个模块可以独立开发、测试，最后拼装成一个完整的程序。使程序结构清楚，易于阅读、理解、测试和更改，则是模块化的主要目的。

封装：封装是程序结构开发中应用的规则。每个功能的组件都封装在一个模块中，在定义每个模块时，尽可能少地暴露内部处理。封装在提升软件的易修改性、易测试性和易移植性方面起着至关重要的作用。

可扩展性原则：系统设计还应考虑到未来可能会有新的需求，如扩展新功能、接入新系统、增加新设备等，因此系统应保留第三方接口，以方便其他系统的接入和并行使用。

系统实现选择前后端分离的开发模式，前后端分离模式可以避免在后端项目中添加前端页面，导致逻辑结构不清、维护和二次发开困难，有利于实现高并发、高性能、高可用，后端编写维护接口文档，在API 变化时更新接口文档，后端根据接口文档进行接口开发，前端根据接口文档进行开发，开发完成后联调和提交测试。

## 系统逻辑结构设计

不同水体识别系统的逻辑结构设计对该系统有着至关重要的作用，本系统公司信息展示模块、用户信息处理模块、图片仓库模块、图片处理模块如下所示。

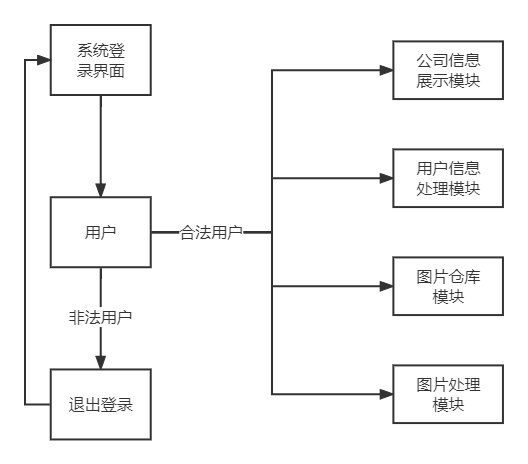


图 4‑1 逻辑结构设计

## 整体结构设计

系统设计采用模块化结构。各模块的设计应遵循高内聚、低耦合的思想，以改善各模块间的不一致性，尽可能减少模块间的数据耦合，最大限度地降低模块间的数据依赖性。系统总体结构设计如下图所示。

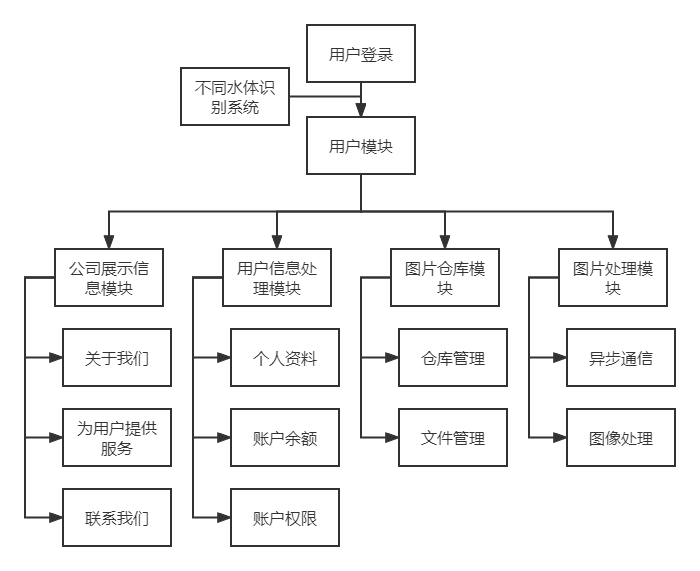


图 4‑2 系统结构图

## 数据库设计

### 概念设计阶段

根据需求分析，本设计预计需要如下的三张表和属性，E-R图如下。

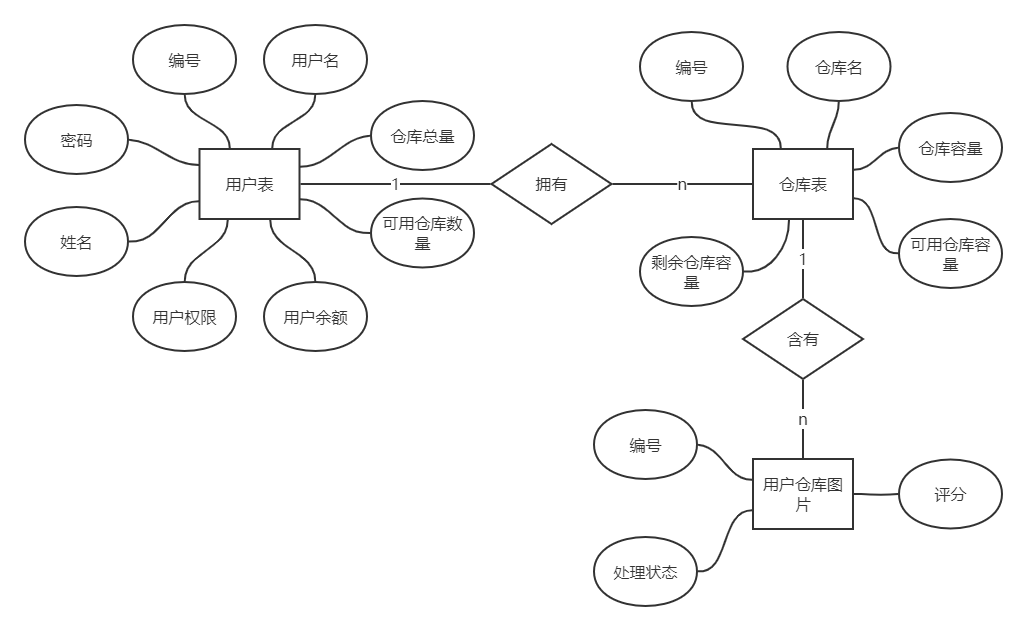


图 4‑3 概念设计E-R图

1. 用户表

编号：唯一编号作为该表主键；

用户名：登录的用户名；

姓名：用户姓名；

密码：用户登录密码；

用户权限：用户的使用权限；

用户余额：用户的使用余额；

仓库总量：用户仓库总量；

可用仓库数量：用户可用仓库数量。

1. 仓库表

编号：唯一编号作为该表主键；

仓库名：仓库名称；

仓库容量：仓库总容量；

可用仓库容量：可用仓库容量；

剩余仓库容量：剩余仓库容量。

1. 用户仓库图片

编号：唯一编号作为该表主键；

评分：该图片的评分；

处理状态：该图片的处理状态。

### 逻辑结构设计阶段

1. 表结构设计

表 4‑1 用户表结构设计

|  |  |  |  |
| --- | --- | --- | --- |
| 属性名 | 描述 | 数据类型 | 约束 |
| id | 编号 | int | 主键约束 |
| account | 用户名 | varchar(50) | 唯一约束、非空约束 |
| name | 姓名 | varchar(20) | 非空约束 |
| password | 密码 | varchar(20) | 非空约束 |
| permission | 用户权限 | enum('1','2','3') | 非空约束 |
| balance | 用户余额 | float | 非空约束、默认约束值为0 |
| available | 仓库总量 | int | 非空约束、默认约束值为10 |
| capacity | 可用仓库数量 | int | 非空约束、默认约束值为10 |

表 4‑2 仓库表结构设计

|  |  |  |  |
| --- | --- | --- | --- |
| 属性名 | 描述 | 数据类型 | 约束 |
| id | 编号 | int | 主键约束 |
| name | 仓库名 | varchar(50) | 非空约束 |
| count | 仓库容量 | int | 非空约束、默认约束值为10 |
| available | 可用仓库容量 | int | 非空约束、默认约束值为0 |
| remaining | 剩余仓库容量 | int | 非空约束、默认约束值为10 |

表 4‑3 用户仓库图片表结构设计

|  |  |  |  |
| --- | --- | --- | --- |
| 属性名 | 描述 | 数据类型 | 约束 |
| id | 编号 | varchar(128) | 主键约束 |
| state | 处理状态 | enum('处理中','处理成功','处理失败') | 非空约束 |
| score | 评分 | int | 非空约束 |

1. 表结构设计

### 数据库物理设计阶段

数据库物理设计采用MySQL关系型数据库，数据引擎采用INODB，原因是INODB支持事务处理，本设计中需要用到事务。

### 数据库实施阶段

### 数据库运行和维护阶段

## 缓存设计

缓存采用Redis数据库，

# 系统实现

## 系统框架

### 前端服务

由于前端部分采用浏览器，实现，固本设计选用Vue.js框架构建前端系统。

### 后端服务

后端采用开源的Spring框架实现，服务器采用Spring Boot集成的Tomcat实现。

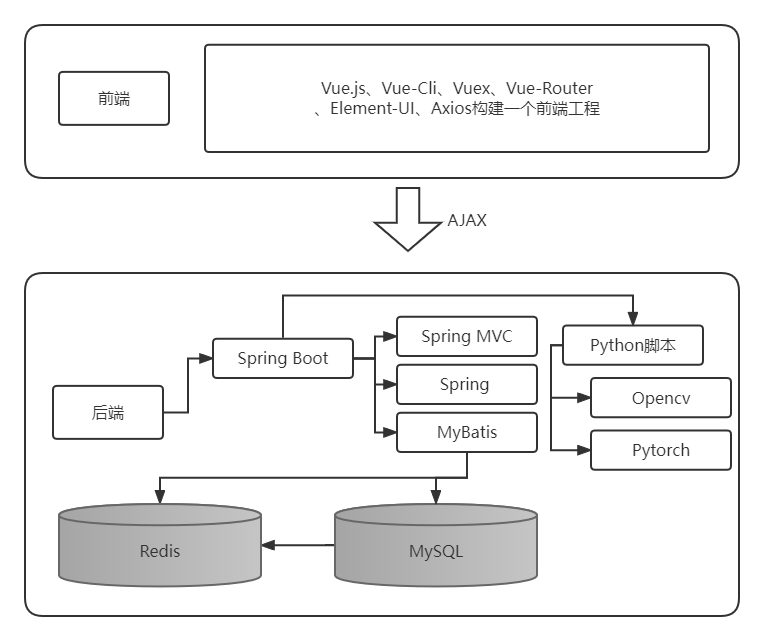


图 5‑1 项目整体架构图

## 开发环境

### 开发环境

前端：WebStorm、Vue-Cli；

后端：Intellij Idea、Spring Boot。

### 部署环境

可将项目部署在云服务器上，选用云服务器规格如下。

硬盘：高I/O，40GB；

内存：2GB；

操作系统镜像：CentOS 8.2 64bit。

## 前端实现

### 登录逻辑

用户只有通过登录模块的检验后，才可访问大学生综合测评系统。可以说，登录模块是整个大学生综合测评系统的门户。用户先访问登录页login，运行效果如图4- 1 所示。在用户名栏输入系统分配的用户名，在密码栏输入与用户名匹配的密码，设定要访问的模块，然后点击登录按钮就可以进入管理页面；如果用户名、密码填写错误，系统会转到错误信息页，给出相应的错误提示。

## 后端实现

# 系统测试

## 接口测试

## 单元测试

## 压力测试

# 总结与展望

1. Browser/Server，浏览器/服务器模式 [↑](#footnote-ref-1)
2. 一种分层开发的模式；M：Model业务模型，处理业务；V：View视图，界面展示；C控制器，处理请求，调用模型和视图 [↑](#footnote-ref-2)
3. Model-View-ViewModel 它本质上就是MVC 的改进版，是将其中的View 的状态和行为抽象化，让我们将视图UI和业务逻辑分开 [↑](#footnote-ref-3)
4. webpack 是一个用于现代 JavaScript 应用程序的静态模块打包工具 [↑](#footnote-ref-4)
5. Java Web Token 是为了在网络应用环境间传递声明而执行的一种基于JSON的开放标准（RFC 7519）。该token被设计为紧凑且安全的，特别适用于分布式站点的单点登录（SSO）场景 [↑](#footnote-ref-5)
6. 一个基于Chrome V8引擎的[JavaScript](https://baike.baidu.com/item/JavaScript/321142)运行环境，使用了一个[事件驱动](https://baike.baidu.com/item/%E4%BA%8B%E4%BB%B6%E9%A9%B1%E5%8A%A8/9597519)、非阻塞式I/O模型，让JavaScript 运行在服务端的开发平台 [↑](#footnote-ref-6)
7. Java Servlet的简称，称为小服务程序或服务连接器，用Java编写的服务器端程序，具有独立于平台和协议的特性，主要功能在于交互式地浏览和生成数据，生成动态Web内容 [↑](#footnote-ref-7)
8. Plain Old Java Objects，普通老式Java对象 [↑](#footnote-ref-8)
9. Relational Database Management System关系数据库管理系统 [↑](#footnote-ref-9)