

**《软件项目管理》课程项目报告**



题 目  **林业害虫识别**

学 院  **软件学院**

专 业  **软件工程**

学生姓名  **王璐瑶**

学 号  **2018141463144** 年级 **2018级**

二Ο二一 年 六 月

1 绪论

* 1. 项目背景

昆虫是动物界中一个十分庞大的类群，全世界存在的昆虫总数可能超过一千万种。因为昆虫种类繁多，并且现从事昆虫鉴定的人员仅限于数量极有限的昆虫分类学专家，所以这对昆虫知识的普及，害虫的防治以及对昆虫学领域乃至自然界生物规律的探讨都是不利的。

基于图像的昆虫识别技术应用广泛，可以应用于植物病虫害预测预报及其防治。据统计每年因未及时发现植物病虫害，而导致损失很大，如果实现了基于图像的昆虫识别系统，那么林业工作者就可以应用该系统进行昆虫识别，从而实现病虫害的及时发现。基于图像的昆虫识别技术也能够帮助昆虫学家更快更好地判别昆虫种类，从而实现植物病虫害的及时防治。在我国社会经济快速发展的背景下，生态保护越来越重要，而林业有害生物防治工作在推动林业的可持续发展方面有非常重要的作用。

根据软件项目管理课程要求，本项目致力于通过图像识别进行智能分析，从而对林业有害生物进行侦查来促进林业生态环境建设。

1.2 项目的主要工作

项目的主要工作是搭建一个基于微信小程序的害虫识别系统，它主要服务于林业管理人员，方便他们在工作中识别一些虫类，判断林区的健康状况，同时也可以由管理员管理害虫信息库，项目具备在线学习的能力。

项目的预期结果是完成一个可供使用的微信小程序。为了完成项目，需要进行以下三个阶段的工作，首先，搭建一个具备害虫目标检测和识别功能的深度学习框架，其次，完成后端与数据库系统的设计搭建，最后，完成前端系统设计和代码编写，并将其与后端系统和深度学习板块进行连接调试。

1.3 项目组成员及分工

|  |  |  |
| --- | --- | --- |
| 姓名 | 角色 | 工作描述 |
| 王心雨 | 项目管理，前期分析，设计 | 分析系统需求，项目计划，项目团队管理，进度把握 |
| 王璐瑶 | 分析，设计，编码，文档编写 | 分析需求，代码模块编写，软件使用手册编写 |
| 李萌 | 设计，编码，文档编写，测试 | 代码模块编写，软件使用手册编写，软件测试 |
| 刘桐源 | 分析，设计，编码，测试 | 需求分析，代码模块编写，软件测试 |

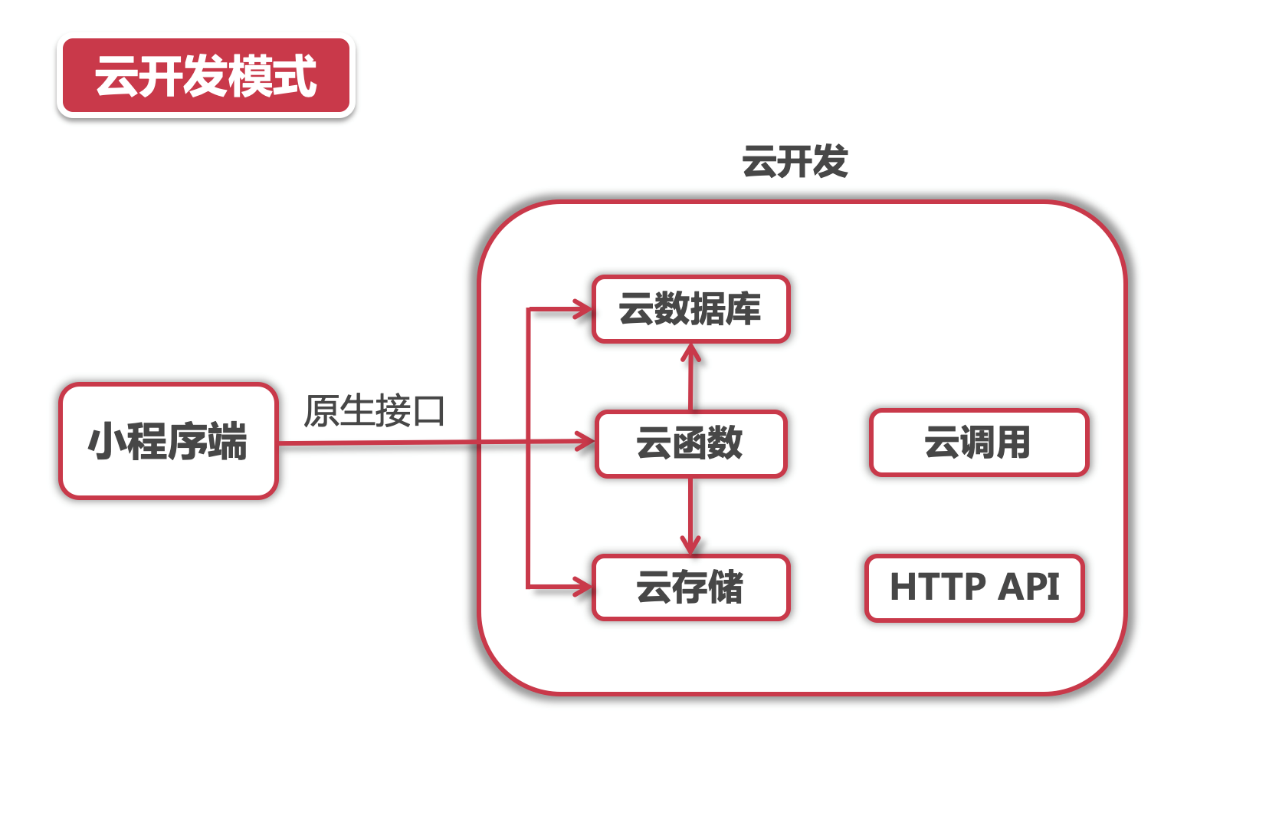
沟通方式如下：

|  |  |
| --- | --- |
| 沟通对象 | 团队内部成员，包括项目经理，设计和编码组，测试组所有成员 |
| 沟通方式 | 通过面对面会议，腾讯会议（网络），QQ聊天群和私聊方式开展沟通活动 |
| 协作模式 | 团队在不同时期有不同的分工，其中项目经理为王心雨同学担任，在设计阶段王璐瑶和刘桐源同学为设计和编码组成员，在编码阶段王璐瑶，李萌和刘桐源同学都为设计和编码组成员，在测试阶段团队所有成员都参与软件测试，由于团队人数较少，因此设计为扁平化管理 |
| 沟通内容 | 沟通内容包括角色临时分配，项目阶段现状沟通，测试内容沟通等 |

1. 相关技术介绍

## 2.1微信小程序云开发简介：

小程序云开发是腾讯云和微信团队联合开发的，集成于小程序控制台的原生 Serverless 云服务，为开发者提供完整的原生云端支持和微信服务支持，弱化后端和运维概念，无需搭建服务器，使用平台提供的 API 进行核心业务开发，即可实现快速上线和迭代。只需要一名开发人员就可以完成所有的工作。云开发核心能力包括：云存储、云数据库、云函数、云调用、HTTP API。



云开发为开发者提供完整的原生云端支持和微信服务支持，弱化后端和运维概念，无需搭建服务器，使用平台提供的 API 进行核心业务开发，即可实现快速上线和迭代，同时这一能力，同开发者已经使用的云服务相互兼容，并不互斥。

资源环境:一个环境对应一整套独立的云开发资源，包括数据库、存储空间、云函数等资源。各个环境是相互独立的，用户开通云开发后即创建了一个环境，默认可拥有最多两个环境。在实际开发中，建议每一个正式环境都搭配一个测试环境，所有功能先在测试环境测试完毕后再上到正式环境。以初始可创建的两个环境为例，建议一个创建为 test 测试环境，一个创建为 release 正式环境。

## 2.2卷积神经网络

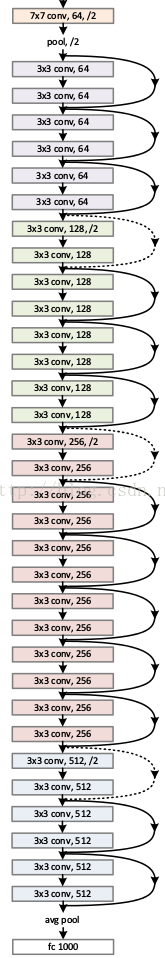
20世纪60年代，Hubel和Wiesel在研究猫脑皮层中用于局部敏感和方向选择的神经元时发现其独特的网络结构可以有效地降低反馈神经网络的复杂性，继而提出了卷积神经网络（Convolutional Neural Networks-简称CNN）。现在，CNN已经成为众多科学领域的研究热点之一，特别是在模式分类领域，由于该网络避免了对图像的复杂前期预处理，可以直接输入原始图像，因而得到了更为广泛的应用。 K.Fukushima在1980年提出的新识别机是卷积神经网络的第一个实现网络。随后，更多的科研工作者对该网络进行了改进。其中，具有代表性的研究成果是Alexander和Taylor提出的“改进认知机”，该方法综合了各种改进方法的优点并避免了耗时的误差反向传播。

一般地，CNN的基本结构包括两层，其一为特征提取层，每个神经元的输入与前一层的局部接受域相连，并提取该局部的特征。一旦该局部特征被提取后，它与其它特征间的位置关系也随之确定下来；其二是特征映射层，网络的每个计算层由多个特征映射组成，每个特征映射是一个平面，平面上所有神经元的权值相等。特征映射结构采用影响函数核小的sigmoid函数作为卷积网络的激活函数，使得特征映射具有位移不变性。此外，由于一个映射面上的神经元共享权值，因而减少了网络自由参数的个数。卷积神经网络中的每一个卷积层都紧跟着一个用来求局部平均与二次提取的计算层，这种特有的两次特征提取结构减小了特征分辨率。

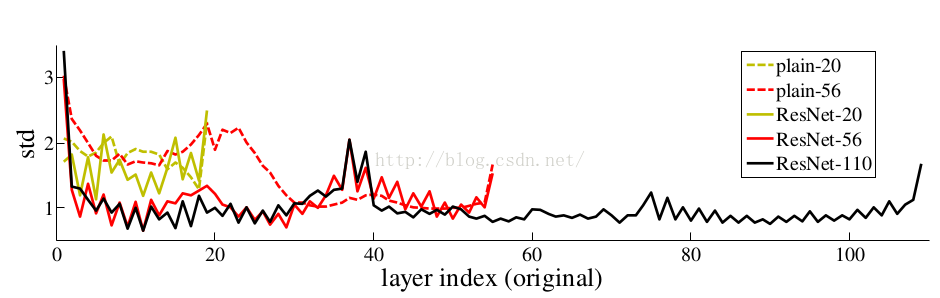
CNN主要用来识别位移、缩放及其他形式扭曲不变性的二维图形。由于CNN的[特征检测](https://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=6744362&ss_c=ssc.citiao.link)层通过训练数据进行学习，所以在使用CNN时，避免了显示的[特征抽取](https://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=8979813&ss_c=ssc.citiao.link)，而隐式地从训练数据中进行学习；再者由于同一特征映射面上的神经元权值相同，所以网络可以并行学习，这也是卷积网络相对于神经元彼此相连网络的一大优势。卷积神经网络以其局部权值共享的特殊结构在语音识别和图像处理方面有着独特的优越性，其布局更接近于实际的[生物神经网络](https://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=66512827&ss_c=ssc.citiao.link)，权值共享降低了网络的复杂性，特别是多维输入向量的图像可以直接输入网络这一特点避免了特征提取和分类过程中数据重建的复杂度。

## 2.3Resnet

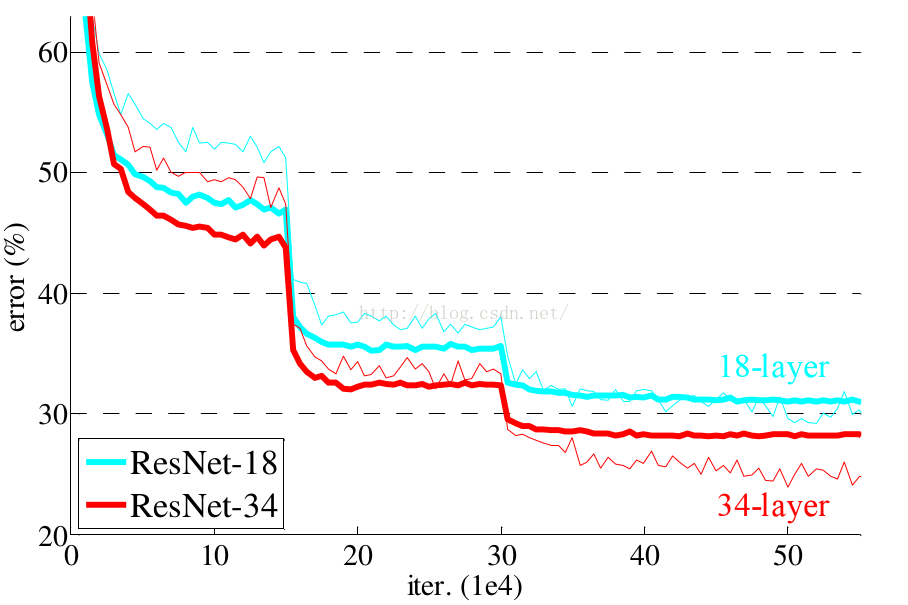
ResNet网络是参考了VGG19网络，在其基础上进行了修改，并通过短路机制加入了残差单元。变化主要体现在ResNet直接使用stride=2的卷积做下采样，并且用global average pool层替换了全连接层。ResNet的一个重要设计原则是：当feature map大小降低一半时，feature map的数量增加一倍，这保持了网络层的复杂度。ResNet相比普通网络每两层间增加了短路机制，这就形成了残差学习，其中feature map数量发生了改变。对于18-layer和34-layer的ResNet，其进行的两层间的残差学习，当网络更深时，其进行的是三层间的残差学习，三层卷积核分别是1x1，3x3和1x1，一个值得注意的是隐含层的feature map数量是比较小的，并且是输出feature map数量的1/4。下图是resnet的网络结构。



resnet学习的是残差函数F(x) = H(x) - x, 这里如果F(x) = 0, 那么就是恒等映射。事实上，resnet是“shortcut connections”的在connections是在恒等映射下的特殊情况，它没有引入额外的参数和计算复杂度。 假如优化目标函数是逼近一个恒等映射, 而不是0映射， 那么学习找到对恒等映射的扰动会比重新学习一个映射函数要容易。从下图可以看出，残差函数一般会有较小的响应波动，表明恒等映射是一个合理的预处理。



残差网络解决了退化的问题，在训练集和校验集上，都证明了的更深的网络错误率越小，如下图。



## 2.4Flask

Flask是一个基于Python开发并且依赖jinja2模板和Werkzeug WSGI服务的一个微型框架，对于Werkzeug本质是Socket服务端，其用于接收http请求并对请求进行预处理，然后触发Flask框架，开发人员基于Flask框架提供的功能对请求进行相应的处理，并返回给用户，如果要返回给用户复杂的内容时，需要借助jinja2模板来实现对模板的处理，即：将模板和数据进行渲染，将渲染后的字符串返回给用户浏览器

“微”(micro) 并不表示你需要把整个 Web 应用塞进单个 Python 文件（虽然确实可以 ），也不意味着 Flask 在功能上有所欠缺。微框架中的“微”意味着 Flask 旨在保持核心简单而易于扩展。Flask 不会替你做出太多决策——比如使用何种数据库。而那些 Flask 所选择的——比如使用何种模板引擎——则很容易替换。除此之外的一切都由可由你掌握。如此，Flask 可以与您珠联璧合。

默认情况下，Flask 不包含数据库抽象层、表单验证，或是其它任何已有多种库可以胜任的功能。然而，Flask 支持用扩展来给应用添加这些功能，如同是 Flask 本身实现的一样。众多的扩展提供了数据库集成、表单验证、上传处理、各种各样的开放认证技术等功能。Flask 也许是“微小”的，但它已准备好在需求繁杂的生产环境中投入使用。

1. 设计工作和进展情况

## 3.1 项目概况

### 3.1.1 项目特点

1）用户访问小程序即可使用系统提供的功能，访问途径方便，所需条件简单。

2）想法新颖，人工智能在害虫防治方面的应用较少，此项目将人工智能与害虫防治结合，是一个创新的想法。

### 3.1.2 功能需求

1）害虫智能识别

2）害虫库概览

3）害虫搜索

4）历史记录浏览

## 3.2 项目开发计划及阶段性完成情况

### 3.2.1 工作内容

1）确定选题

2）需求分析

3）系统设计

4）编写代码

5）编写文档

6）系统测试

本人工作内容主要分为两个方面，首先是参与讨论需求分析和系统设计，以及参与完成需求分析，开发计划文档，系统设计文档，和主要负责详细设计文档工作；其次是完成微信小程序云数据库建表，完成前端的图片上传、页面展示、昆虫搜索、历史记录等功能。

### 3.2.2 工作进度安排

|  |  |  |
| --- | --- | --- |
| 时间 | 工作内容 | 备注 |
| 1-3周 | 完成建组  确定团队成员和项目题目；  开会初步讨论项目需求等 | 确定项目题目为《林业害虫识别》，初步确定使用微信小程序开发前端页面 |
| 4-5周 | 讨论确定项目具体需求；  完成项目需求分析文档 | 完成需求分析文档 |
| 6-7周 | 参与完成项目设计文档；  搭建前端页面框架，初步确定界面功能。 | 画界面原型图，搭建前端页面框架 |
| 8-11周 | 建立微信小程序云数据库，导入相应的昆虫信息数据；  编写上传图片功能。 | 学习云数据库增删改查操作；上传图片包括手机拍照和相册选择两种方式 |
| 12-13周 | 完善项目详细设计文档；  完善个人项目报告；  完成微信小程序页面所有功能；  完善个人项目报告 | 昆虫识别，昆虫信息列表显示功能。 |

## 3.3 项目的开发内容和结果

### 3.3.1 系统需求分析

1）功能需求

1.上传害虫图片

系统应提供用户上传害虫图片功能，包括从图库中上传害虫图片和直接用手机相机拍摄害虫图片两种方式。

2.反馈害虫信息

系统应提供反馈图片中害虫详细信息功能，包括图片中害虫的名称、描述、危害等级、分布和消除办法等详细信息。

3.查看历史识别记录

系统应提供用户查看害虫识别历史记录的功能，包括每次上传图片的时间地点以及上传的图片**。**

4.害虫库概览

系统应提供用户查看数据库中所有害虫详细信息的功能，即用户可以浏览每种害虫的名称、描述、危害等级、分布和消除办法等详细信息。

5.搜索害虫

系统应提供用户搜索害虫库中特定害虫的功能，用户输入害虫名称，系统反馈害虫的详细信息。

2）性能需求

1.时间要求

用户上传图片后，等待系统反馈害虫信息的时间不超过3s。

2.准确率要求

在25种害虫的情况下，害虫种类识别结果top3正确率达到95%。

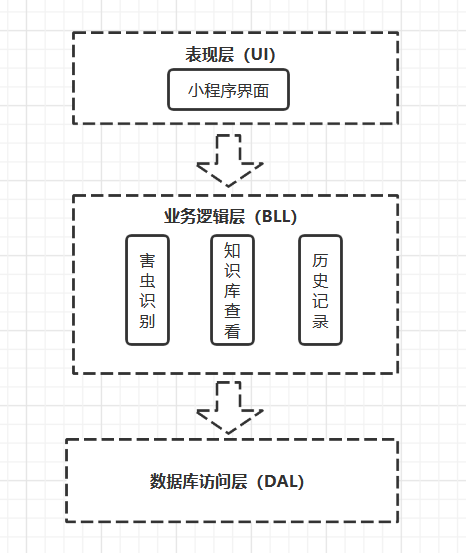
3）其他需求

程序通过微信小程序即可访问，用户通过使用微信小程序就可进行害虫智能识别。

### 3.3.2 系统设计

1）体系架构设计

统设计思路主要为分模块开发，为常见的B/S架构，将系统分为前后端分离开发，其中前端使用小程序框架，后端使用pytorch和flask框架进行开发，flask框架用于实现web访问，pytorch使用ResNet152实现图像分类及目标检测。



2）系统接口设计

1.外部接口设计

|  |  |  |
| --- | --- | --- |
| 接口类型 | 接口 | 功能说明 |
| 调用本地接口 | 系统拍照和图库功能 | 用于用户上传图片至云端服务器进行分析 |
| 调用云端接口 | 小程序云端数据库接口 | 用于存储用户数据，用户识别历史数据等 |

2.内部接口设计

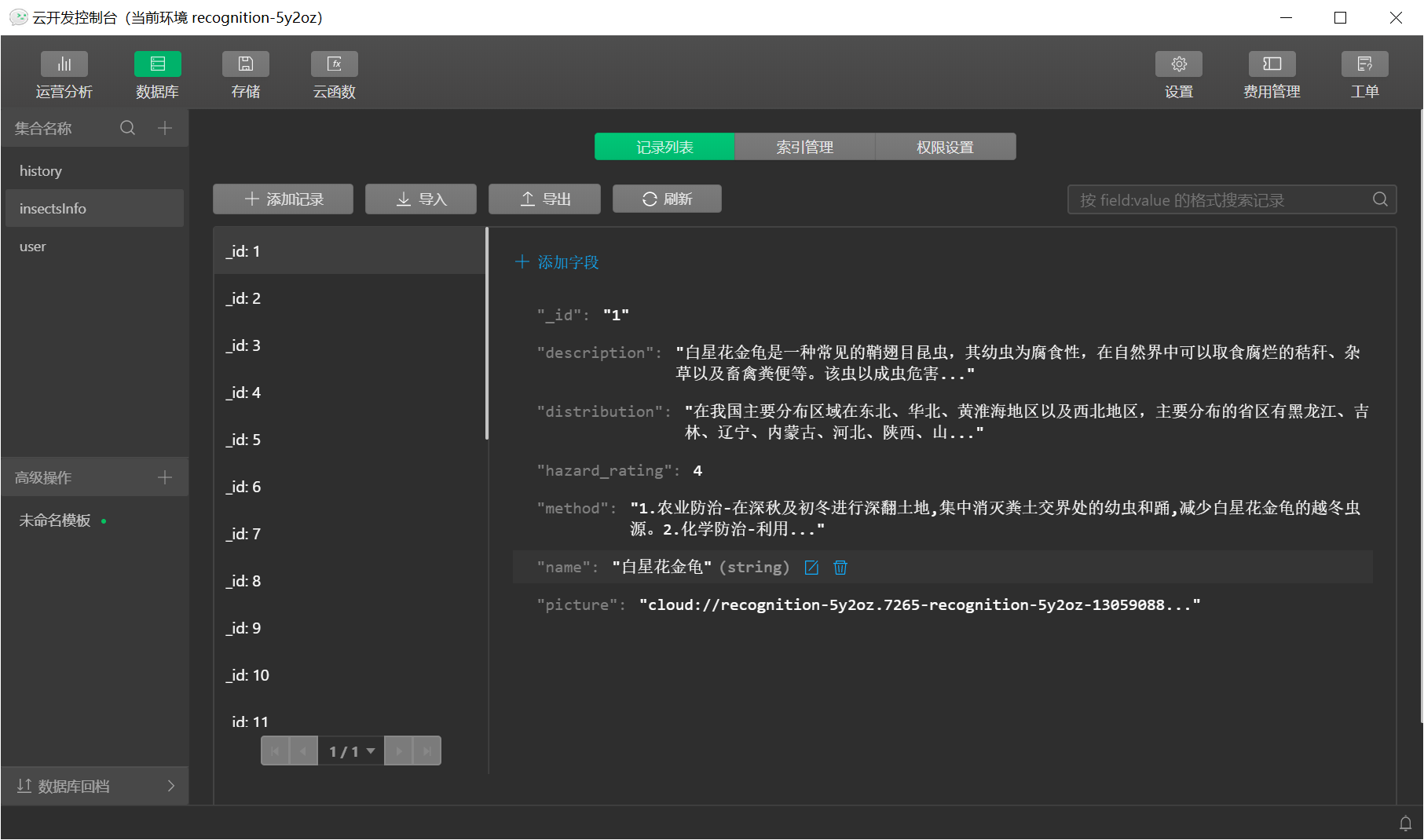
|  |  |  |
| --- | --- | --- |
| 接口说明 | 操作 | |
| 本地小程序端 | 服务器端 |
| 上传图片 | 将图片信息传送至服务器 | 接收图片信息，进行目标检测和识别 |
| 接收识别结果 | 接收识别结果并呈现 | 将识别结果返回至小程序端 |
| 发送查询识别历史信息请求 | 将查询请求发送至服务器端 | 根据请求进行查询 |
| 接收识别历史信息数据 | 接收历史识别信息的数据 | 将查询结果返回至小程序端 |

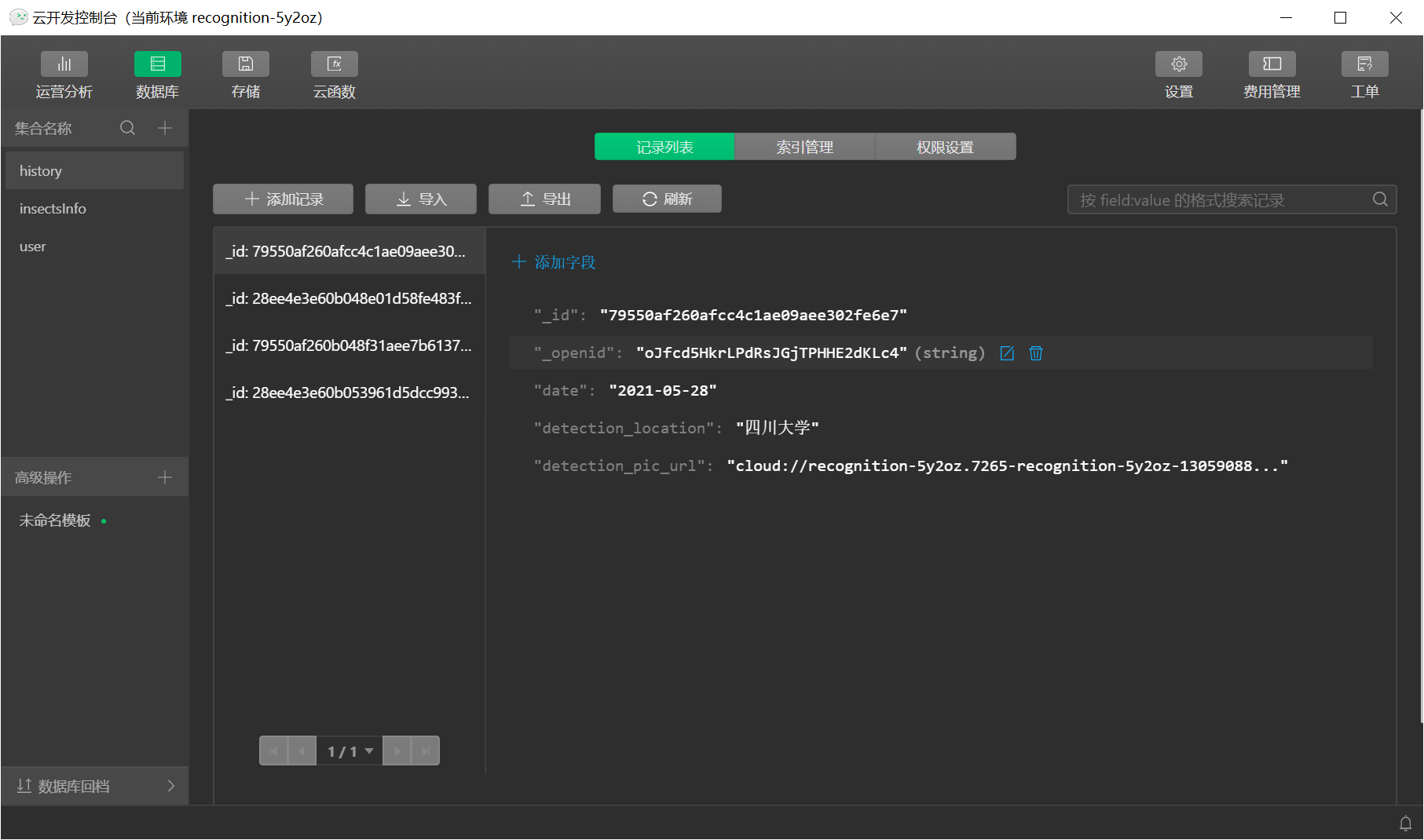
3.人机接口设计

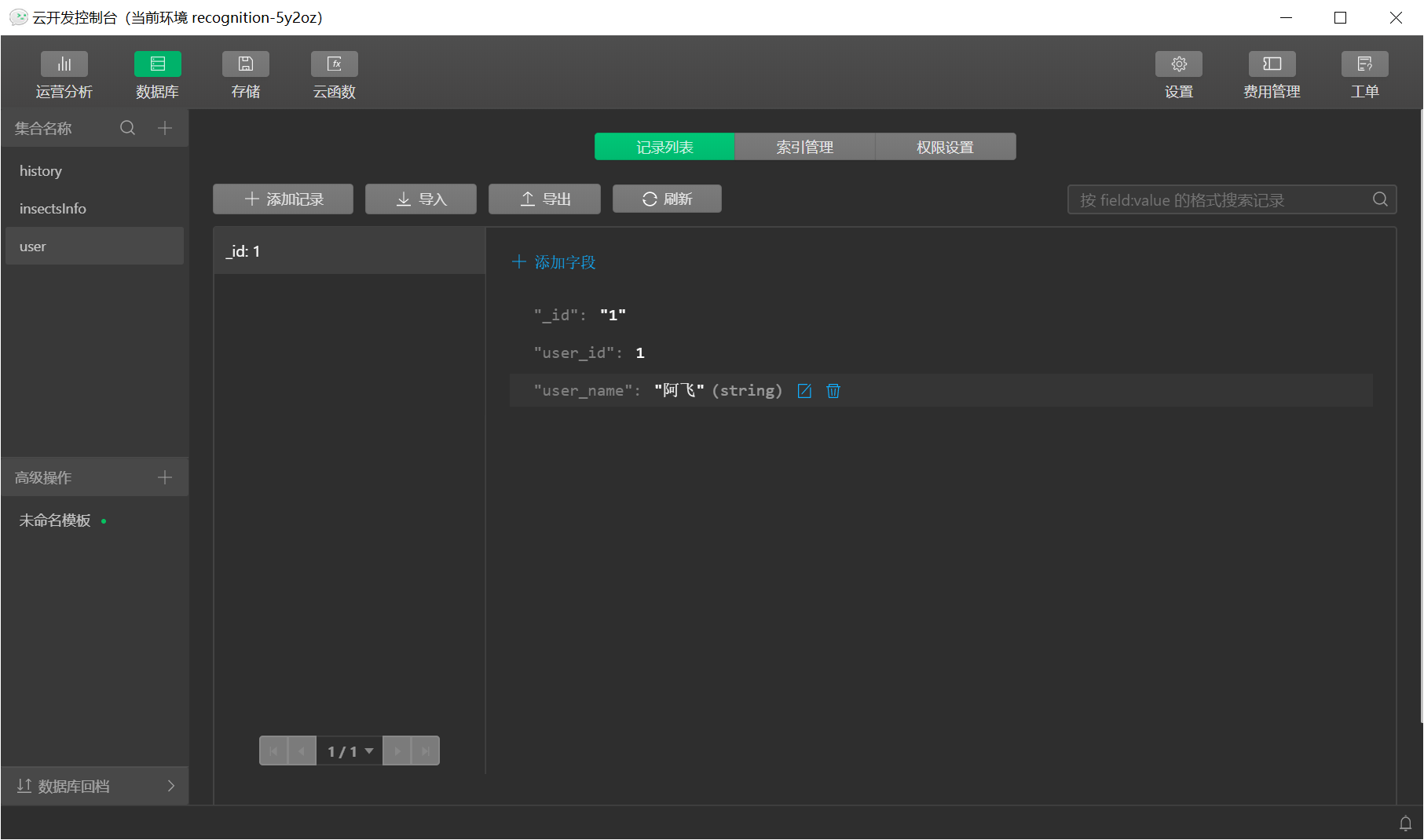
|  |  |  |
| --- | --- | --- |
| 接口描述 | 用户请求 | 小程序应答 |
| 上传图片 | 请求系统调用图库或者相机 | 弹出请求窗口，经过允许后打开相机或是图库 |
| 请求历史识别记录 | 请求查看历史记录 | 切换历史记录页面，进行历史识别记录的显示 |

### 3.3.3 具体工作内容

1. 建数据库和数据表

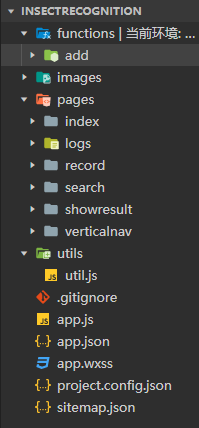




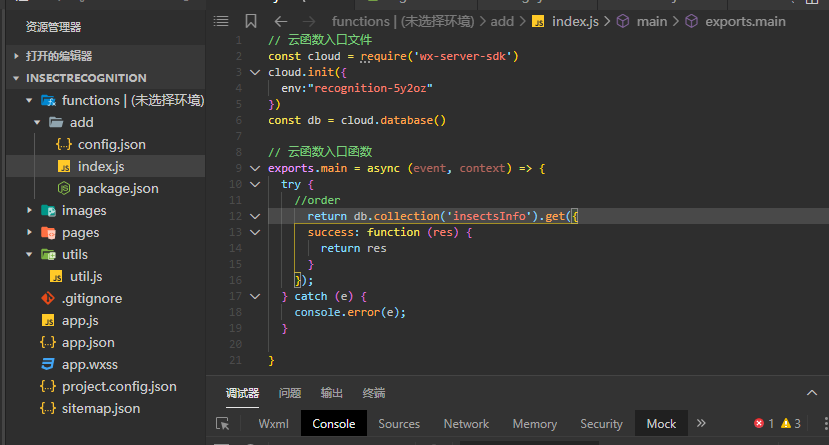


1. 页面搭建部分：

代码结构：



1. 云函数入口配置：



1. 主页面：

存放在Index文件夹中。用户选择或拍摄的图片上传到后端接口并接收后端返回的信息，再将返回的信息中的昆虫名称传给showresult页面。



1. 昆虫识别结果页面：

存放在showresult文件夹中，展示识别结果页面的代码，通过Index页面传来的害虫名称从数据库获取相应害虫的详细信息并在页面显示。



1. 害虫库概览页面：

代码存放在verticalnav文件夹中，从数据库中获取所有害虫的名称、描述、危害等级、分布、消除办法以及图片信息显示在页面，点击页面最上方的搜索框跳转到search页面。



1. 历史记录页面：

代码存放在record文件夹中，从数据库中获取用户曾经上传过的图片以及上传时间和地点，在小程序的历史记录页面中显示。



1. 讨论与体会

## 对项目过程的体会

在本次项目中，我所负责的是微信小程序功能。从最开始的小组讨论调研决定选题，到功能质量需求分析，再到系统概要详细设计，到最后的代码构建测试，我都认真参与用心负责，做好自己所分配的工作，完成好自己所负责的任务。

我意识到了团队合作的重要性，以及正确的项目管理方法对于项目开发来说是非常重要的。需求分析的过程做得越详细越全面，后续的开发就会少很多麻烦。对于我们团队来说，我认为组长的角色非常重要，不仅需要负责跟踪前后端的进度，还要分配任务，进行验收，就如同公司的项目经理，我体会到了编码知识只是开发中的一小部分，开发之前的准备工作也很重要。这次的文档写了很多很详细，到后面开发的时候就非常方便，希望以后能更加重视编码之前的需求分析和系统设计阶段。

## 4.2 对项目的评价

我们项目的总目标是搭建一个基于微信小程序的害虫识别系统，它主要服务于林业管理人员，方便他们在工作中识别一些虫类，判断林业的健康状况，同时也可以由管理员管理害虫信息库，具备在线学习的能力。项目的结果是一个可供使用的微信小程序。项目也致力于通过图像识别进行智能分析，从而对林业有害生物进行侦查来促进林业生态环境建设。

在开发的过程中，我们团队通力合作，合理分配任务，进展很顺利，最终完成了害虫智能识别、害虫库概览、害虫搜索、历史记录浏览等基础功能。但是由于开发周期不长，项目需求分析报告中的某些扩展功能未能完成，比如害虫库的维护和更新等。希望在未来可以把这个项目加以完善，让可用性更强。

5 小结

这学期的项目管理课让我学到了很多关于正式的企业项目管理流程，也让我意识到编码知识只是项目开发中的一小部分，开发之前的准备工作也很重要，比如需求文档、概要设计文档、详细设计文档的编写，把这些弄好之后，就差不多把项目要做的东西了解透彻了，之后的开发就更加方便和容易。希望自己以后在项目开发中也能更加重视编码之前的需求分析和系统设计阶段。

在林业害虫识别小程序的开发中，我了解了微信小程序云开发，也更加明白了团队合作的重要性，同时团队里面有一个leader也是很必要的，明确地分配任务，可以让开发效率更高。我们团队的每个人都认真地负责了自己的模块，基本实现了项目设计当初拟定的所有需求。不过在细节方面还有可以改善的空间，如昆虫库的更新和数据集的扩大等。希望以后能将这次学到的经验和感受铭记于心，不再犯相同的错误，在项目开发中做得更好。