**详细设计报告**

目录

[一．引言 2](#_Toc74468986)

[1.编制目的 2](#_Toc74468987)

[2.项目背景 2](#_Toc74468988)

[3.与其读者和阅读建议 2](#_Toc74468989)

[4.术语 2](#_Toc74468990)

[5.参考资料 3](#_Toc74468991)

[二．系统开发环境 3](#_Toc74468992)

[1.前端 3](#_Toc74468993)

[2.后端 3](#_Toc74468994)

[三．系统设计 4](#_Toc74468995)

[1.系统结构 4](#_Toc74468996)

[2.前端代码解读 5](#_Toc74468997)

[3.后端代码解读 6](#_Toc74468998)

[4.系统功能模块 7](#_Toc74468999)

[5.系统总体流程 7](#_Toc74469000)

[6.关键性技术 8](#_Toc74469001)

# 一．引言

## 1.编制目的

本项目计划书文档主要反映出林业有害生物智能识别系统的具体计划方案，引导软件开发人员进行后续的开发工作。此项目计划文档既可以作为软件开发工作的基础和依据，也可以作为此项目确认测试和验收的凭照，以便于保证项目团队按时保质地完成项目目标，便于项目团队成员更好地了解项目情况，使项目工作开展的各个过程合理有序。

## 2.项目背景

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。

## 3.与其读者和阅读建议

本文档预期面向多种读者：

（1）项目经理：项目经理可以根据该文档了解预期物品的功能设计，并据此进行项目管理。

（2）程序员：程序员可以配合需求规格说明书和此项目设计文档，了解所需要进行开发的功能，进行代码的编写。

（3）测试员：测试员可以根据需求规格说明书和此项目设计报告编写测试用例，并对系统进行功能性测试和非功能性测试。

（4）产品经理：产品经理可以根据需求规格说明书、此项目设计报告和开发出来的系统代码，撰写用户手册。

（5）其他人员：其他人员如部门领导、公司领导等可以据此了解系统的功能设计。

## 4.术语

* 深度学习

深度学习(DL, Deep Learning)是机器学习(ML, Machine Learning)领域中一个新的研究方向，它被引入机器学习使其更接近于最初的目标——人工智能(AI, Artificial Intelligence)

* 图像识别

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。现阶段图像识别技术一般分为人脸识别与商品识别，人脸识别主要运用在安全检查、身份核验与移动支付中；商品识别主要运用在商品流通过程中，特别是无人货架、智能零售柜等无人零售领域。

* 目标识别

目标识别是指一个特殊目标（或一种类型的目标）从其它目标（或其它类型的目标）中被区分出来的过程。它既包括两个非常相似目标的识别，也包括一种类型的目标同其他类型目标的识别

## 5.参考资料

* 菜鸟教程，runoob.com。
* 小程序开发指南，

<https://developers.weixin.qq.com/miniprogram/dev/framework/quickstart/>

# 二．系统开发环境

## 1.前端

前端使用微信小程序平台进行部署

配置版本为：微信开发者工具 1.04.2009152

云开发环境：recognition-5y2oz

## 2.后端

后端使用矩池云服务器进行部署

配置过程：

1. 上传源代码与模型至我的网盘
2. 租用服务器
3. 使用“pip install flask”命令安装依赖，使用“python web.py”命令执行

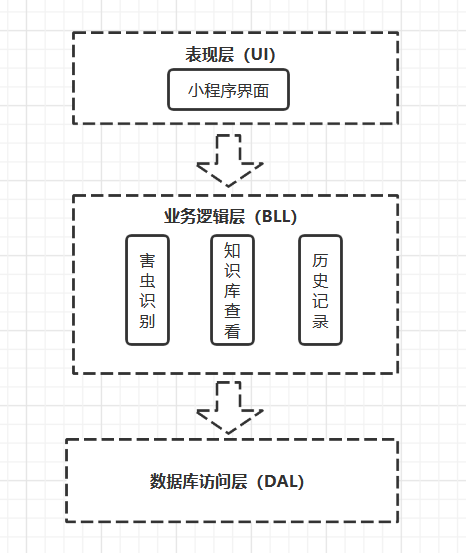
配置版本：

|  |
| --- |
| 环境版本 |
| Python 3.8 |
| CUDA 11.0 |
| cuDNN 8.0 |
| Pytorch 1.7.1 |
| flask 1.0 |

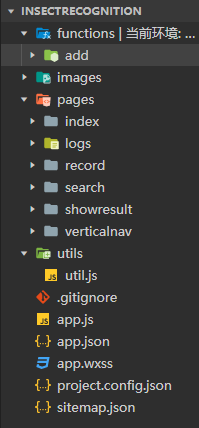
# 三．系统设计

## 1.系统结构

系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。



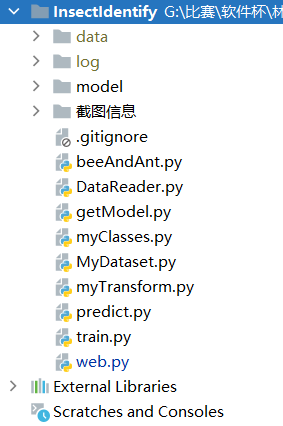
## 2.前端代码解读



文件及作用：

* + - 1. functions：云函数目录，用于存放云函数的文件夹
      2. add：用于读取数据库中的insectsInfo集合里的记录的云函数
      3. Images：用于存放小程序中要用到的图片的文件夹，例如菜单栏中的图标
      4. Pages：存放小程序的页面文件，书写各个页面代码以及组件
      5. Index：存放首页即识别昆虫页面的代码的文件，将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。
      6. Logs：存放运行日志输出的代码的文件。
      7. Record：存放历史记录页面的代码的文件，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。
      8. Search：存放搜索页面的代码的文件，通过用户输入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。
      9. Showresult：存放展示识别结果页面的代码的文件，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。
      10. Verticalnav：存放害虫库概览页面的代码的文件，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。

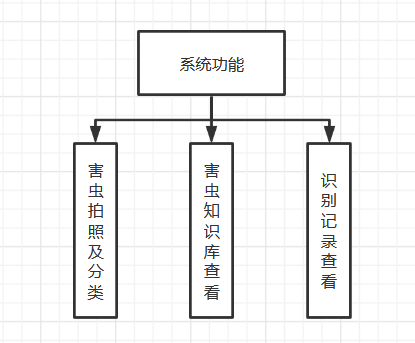
## 3.后端代码解读



文件及作用

1. myClasses.py：由于pytorch训练部分与数据库分离，故使用myClasses保存分类及id互相对应的信息，便于在训练时将分类转化为di，预测时将获得的id转化为名称。
2. getModel.py：用于在训练时加载预训练模型，预测时加载已经训练好的模型。
3. MyDataset.py：用于存储数据集合并将其作为参数生成Dataloader，便于训练。
4. DataReader.py：用于读取数据集对应的文件夹，生成数据集存入Dataset，并将其转化为Dataloader，以方便后续训练。
5. MyTransform.py：对图像数据进行的一些操作，如缩放，翻转等。
6. train.py：调用其他模块，进行训练。
7. predict.py：使用传入的模型进行图片分类。
8. web.py：使用flask框架，加载模型，等待http请求，调用predict进行分类并返回结果。

## 4.系统功能模块



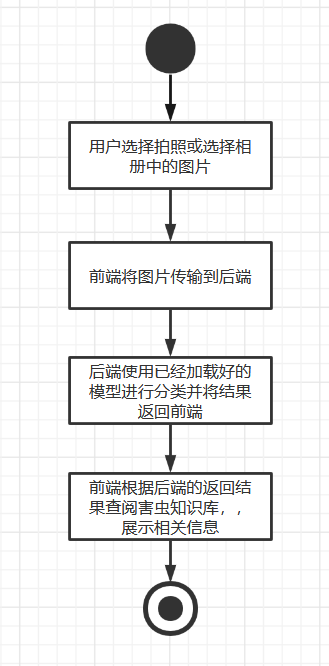
害虫拍照及分类：对用户所选择的害虫图片进行分类并快速获取相关信息予以展示。

害虫知识库：主要存储害虫的名称、分布、危害等级及防治方法，方便林业人员在需要时进行查阅。

识别记录：保存了用户上传的图片，当用户需要时，可以查看过往记录。

## 5.系统总体流程

系统的主要功能为害虫图片分类，图片数据由前端传输到后端，使用已加载好的模型进行分类获取害虫id和名称，前端通过拿到的id和名称获取害虫相关信息予以展示。



## 6.关键性技术

小程序开发：使用小程序的云开发功能在小程序端创建数据库，并通过云函数获取数据库的数据。

图像分类：采用ResNet152模型进行训练，修改fc层，使其符合应用需求。

目标检测（已实现但由于缺少数据集暂未启用）：