

文件编号: MALLOC(0)-SWC2018-20180081

受控状态: ☒ 受控 ☐ 非受控

保密级别: ☐ 公司级 ☐ 部门级 ☒ 项目级 ☐ 普通级

采纳标准: CMMI DEV V1.2



速写思维导图

SketchMind

项目开发文档

Version 0.2

2019/2/20

Written by malloc(0)



All Rights Reserved

目录

1	引言	1
1.1	编写目的	1
1.2	项目概述	1
1.3	项目背景	1
1.4	术语和缩略语	1
1.5	参考资料	1
1.6	项目定位	2
1.6.1	应用场景	2
1.6.2	目标人群	2
1.7	项目目标	2
1.8	项目价值	2
1.9	创新点	2
1.9.1	功能性创新	2
1.9.2	非功能性创新	2
1.10	解决思路	3
1.11	系统亮点	3
1.11.1	功能性亮点	3
1.11.2	非功能性亮点	3
2	开发计划	3
2.1	最终呈现形式	3
2.2	主要功能描述	3
2.3	运行环境	3
2.4	验收标准	4
2.5	关键问题	4
2.6	进度安排	4
2.7	开发预算	4
3	可行性分析	5
3.1	市场可行性分析	5
3.1.1	与现行软件对比	5
3.2	技术可行性分析	5
3.2.1	功能简述	5
3.2.2	技术要素	5
3.3	资源可行性分析	5

4	需求分析	6
4.1	数据描述	6
4.1.1	静态数据	6
4.1.2	动态数据	6
4.1.3	数据词典	6
4.1.4	数据采集	6
4.2	功能需求	7
4.2.1	功能模块结构图	7
4.2.2	核心功能模块描述	7
4.3	非功能性需求	8
4.3.1	扩展性和可维护性	8
4.3.2	易用性	8
4.4	性能需求	8
4.4.1	时间特性	8
4.4.2	适应性	8
4.5	运行需求	9
4.5.1	用户界面	9
4.5.2	硬件接口	11
4.5.3	软件接口	11
4.5.4	故障处理	11
4.6	用例图	11
4.7	核心模块用例规约	12
4.8	其他需求	12
5	概要设计	12
5.1	处理流程	12
5.2	总体结构和模块设计	13
5.3	功能分配	14
5.4	接口设计	14
5.4.1	外部接口	14
5.4.2	内部接口	14
5.4.3	用户界面设计	14
5.5	数据结构设计	15
5.5.1	逻辑结构设计	15
5.5.2	物理结构设计	15
5.5.3	数据结构与程序的关系	15
5.6	运行设计	15
5.6.1	运行模块关系	15
5.6.2	运行控制	16
5.6.3	运行时间	16
5.7	错误/异常处理设计	16
5.7.1	错误/异常输出信息	16
5.7.2	错误/异常处理对策	16
5.8	系统配置策略	16
5.9	系统部署方案	16

5.10	代码规范	17
5.11	其他相关技术与方案	17
6	数据库设计	17
7	详细设计	17
7.1	功能模块	17
7.1.1	功能描述	18
7.1.2	性能描述	18
7.1.3	输入	18
7.1.4	输出	18
7.1.5	算法	18
7.1.6	程序逻辑	18
7.1.7	接口	19
7.1.8	存储分配	19
7.1.9	限制条件	19

记录更改历史

[illegible]

1 引言

1.1 编写目的

为了对 malloc(0) 小组在初赛准备阶段所完成的工作进行一份完整、翔实的记录，本文档将会主要记录初赛准备阶段的市场调研、可行性研究、竞品比较；效果演示；开发计划、整体架构等重要项目组成部分。

1.2 项目概述

SketchMind 可以通过一张手绘的思维导图生成精美的思维导图电子文稿格式。此项目意在以较高的容错率将小组讨论中信笔划下的思维草稿转化成精美易读且容易编辑的电子版思维导图，并以多种通用格式输出作为选择给予用户。

1.3 项目背景

掏出纸笔，圈圈点点写写画画是小组讨论中经常能遇见的场景。然而，最终得到的结果往往还需要手动打开 XMind、Visio 等思维导图软件再进行一次誊抄，费时费力。我们希望能够通过这份粗糙的手稿，最大精度地还原思维的流向，减少无谓的重复劳动。

1.4 术语和缩略语

[1] 思维导图：又称导图，Mind Map。一种图像式思维的工具以及一种利用图像式思考辅助工具来表达思维的工具。

[2] XMind：一个由香港 XMind 公司开发的脑力激荡法和心智图的软件工具，其主要用途为帮助用户捕捉想法，组织各类报表。

1.5 参考资料

[1] 心智图. 维基百科. <https://zh.wikipedia.org/wiki/心智图>

[2] XMind. 维基百科. <https://zh.wikipedia.org/wiki/XMind>

1.6 项目定位

1.6.1 应用场景

本应用场景较为贴近日常:在某次线下头脑风暴之后写下的草稿思维导图可以利用本程序直接转化为电子版可编辑的精美导图;或是制作专业思维导图时遇到繁琐的小部件的定位问题的时候,可以先手绘出大体结构,简化第一步繁杂的布局工作。

1.6.2 目标人群

本项目适用于任何需要使用导图的用户。无论是专业的导图制作者,还是初级导图使用者,都可以使用本项目来简化复杂的工作。

1.7 项目目标

我们希望这个项目能降低制作思维导图的门槛:只要能用纸笔画出导图大意,就能轻松生成美观的电子导图。

1.8 项目价值

此项目可以节约导图制作过程中由想法(手稿)到成品(电子文稿)的工作,使得这步转换更为轻松。

1.9 创新点

1.9.1 功能性创新

新的导图输入形式。目前市面上的大部分导图软件仍然是传统的「拖拉到位」式设计式制作方针,从门槛上来说学习成本较高,从用户体验上来说仍然过于繁琐。现在,只需要输入一张简单的蓝图手稿,再通过精心设计的前端页面控件,仅做简单的微调就能生成一幅精美导图,并可导出为多种行业标准的导图文稿格式。

1.9.2 非功能性创新

本项目从日常生活中遇到的不便之处出发,寻找着眼点,再反作用于日常生活中,为用户带来便利。另外,本项目规避了大多数软件制作导图的基本思路,使用户的导图制作过程更加地快速与流畅,这样便可大大提高用户的作图效率,同时也是导图制作模式上的一大创新,能为之后的项目设计开发提供借鉴。

1.10 解决思路

导图一般分为元素与关系两部分。本项目通过对用户所给的手绘图进行分析，从而得出导图各个元素之间的关系。得出关系后，后台会对各个元素的内容进行识别（以图像文字识别为主），进而得出每个元素的内容。之后后台将生成一个与原绘稿贴近的导图展示给用户，并将每个元素内容的图片与识别结果一同放入元素的区域中以供用户检查。用户可以对元素的识别内容进行修改，并且可以任意添加、删除、修改组件。另外用户还可自行选择导图的背景。待用户确认无误后，后台将生成一个最终版的导图文件并提供给用户。

1.11 系统亮点

1.11.1 功能性亮点

从基本模式上对导图的制作流程进行了创新与优化，功能上更加便捷与高效。

1.11.2 非功能性亮点

能让用户在制作过程中将更多的注意力集中在导图本身的逻辑结构与布局美观和清晰度上，使得最终的结果更加符合用户的需求。

2 开发计划

2.1 最终呈现形式

本项目最终将用网页的形式来呈现给用户。用户的操作以及用户与程序的交互均在网页上进行。

2.2 主要功能描述

本项目的主要功能是通过用户提供的手绘版导图的照片进行识别与分析，通过人工智能自动生成与之相匹配的电子版导图图片，并提供初稿给用户进行确认与调整，最后在网页上给出用户可下载的电子版导图最终结果。

2.3 运行环境

由于本项目最终由网页进行呈现，故在当前市面上的大多数浏览器均可运行，兼容度高，可移植性强。

2.4 验收标准

完成项目所需的代码以及相关的文档与视频等材料，以及项目的训练与根据用户需求的优化调整。

2.5 关键问题

[1]设计合适的匹配机制，将用户的手绘元素转换成电子版后能够保持原版的风格。

[2]通过训练让后台能够生成合适的布局风格，使得电子版能够对简略的手绘版的布局进行优化调整，最终结果更加美观悦目。

[3]由于不调用用户端的硬件，因此不存在设备问题导致的影响。

[4]若网络状况不良好，可能会导致项目运行与反应缓慢。此时只能由用户调整网络环境来改善状况。

2.6 进度安排

[1]初赛完成项目开发文档、项目测试文档、技术性研究报告与作品创新性分析报告，并完成有关人工智能的学习以及项目的初步设计与整体构建。最后制作相关的项目视频。

[2]复赛之前进一步项目的设计与完善，并继续学习相关的知识。同时为项目的训练进行素材的规划与收集。

[3]复赛完成项目各部分的制作与调试。根据项目人工智能的神经网络设计对训练素材进行挑选，并完成对项目的训练。之后对项目开发文档、项目测试文档、技术性研究报告与作品创新性分析报告进行调整更新，并针对新进度的制作项目视频。

[4]决赛之前对项目进行用户调研分析，并根据用户需求进行功能优化。项目成品诞生后，完成项目开发文档、项目测试文档、技术性研究报告与作品创新性分析报告的最终版本，并制作项目的使用手册。最后制作相关的介绍 PPT 与项目视频。

2.7 开发预算

本项目完全由本小组自行设计与制作，因此项目本身并不需要资金投入。在训练素材收集阶段，由于需要大量素材，因此可能会请其他人员提供初稿等材料，因此可能需要一定的报酬。但绝大部分材料的提供者是小组成员与学校同学，因此此部分预算并不很多。

主要的花费在服务器的托管上，因此采用价格较为便宜的国内云服务提供商。

3 可行性分析

3.1 市场可行性分析

3.1.1 与现行软件对比

当前市面上有导图制作功能的软件基本全部采用组件式制作法,在所需导图规模较小时比较适用。但若导图规模很大,则组件法将需要大量的时间与精力来进行组件的布置与布局的调整。而本项目则通过手绘来传入导图的信息,在所制作导图的规模达到中等或以上时便会体现出巨大便利性,另外还会对元素的布局进行合理的调整,使得整个导图更加美观与清晰。

根据发放的问卷及回收情况来看,大部分(超过 80%)的人愿意试用这一新产品,且超过九成的受访对象表示自己会在学习、生活和工作中随手画出思维导图。这其中超过半数的人表示,不知道如何整理归纳这些信手画出的思维导图。因此,这一需求真实存在。

而目前通行的很多思维导图软件,例如 Microsoft Visio、XMind 等具有很高的门槛,主要针对需要制作精美流程图的办公人群,无法覆盖学生或是日常使用者这一要求和技术水平都不高的群体。因此,我们的产品具有一定的生存空间。

3.2 技术可行性分析

3.2.1 功能简述

本项目通过对用户提供的手绘版导图进行识别与分析,利用人工智能生成对应的电子版导图。

3.2.2 技术要素

[1]对用户提供的手绘版照片进行识别与逻辑判断

[2]人工智能根据元素逻辑对结构进行调整优化

3.3 资源可行性分析

本项目的设计与开发由本小组成员进行,因此有足够的人力资源。由于本项目并无太多预算,因此资金方面也不存在问题。另外比赛官方为项目提供了运行设备,因此物资方面资源充足。综上所述,本项目在资源方面完全可行。

4 需求分析

4.1 数据描述

为了节约带宽、加快程序响应速度，且保持一定的灵活性，后端不应直接返回渲染好的电子版思维导图，而应该返回可编辑的数据结构。因此我们将设计一套数据接口用于前后端之间的通信。

4.1.1 静态数据

训练好的模型不会轻易改动，是为了确保结果的稳定性和可预测性。因此这部分置于服务器上的模型数据应为静态的，且不需要传递给前端页面。

4.1.2 动态数据

每一次请求产生的 JSON 数据都是不同的，都是后端根据不同的传入图片动态生成的。这一数据集满足一定的规则，方便前端做进一步的处理。

4.1.3 数据词典

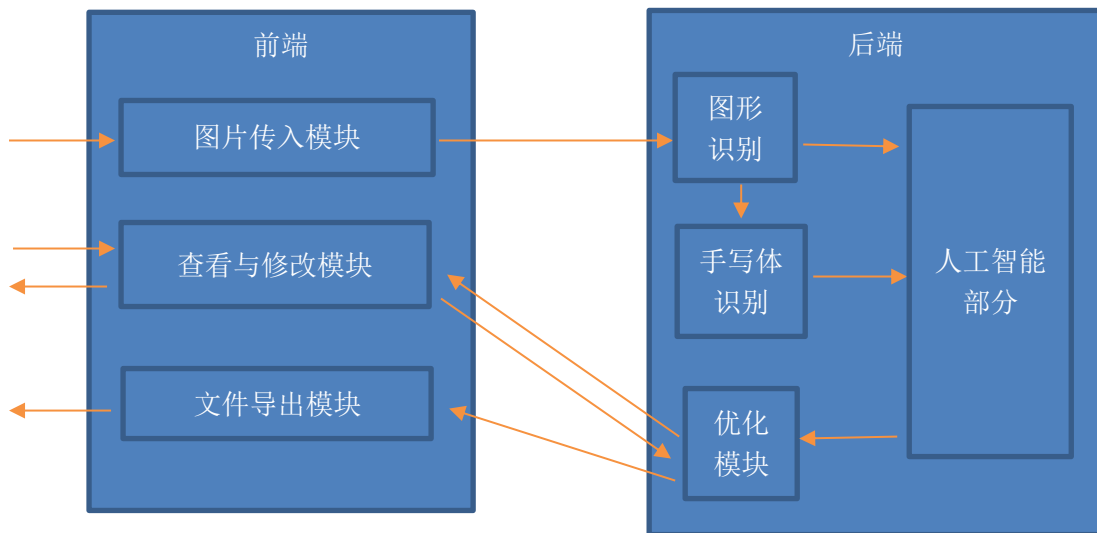
为了后续迭代的方便及紧急状态下的回滚，每一次新增训练所用到的训练集和训练结果都被存档在服务器上，按照最终的识别正确率来衡量改版的成功与否。

4.1.4 数据采集

为了确保数据的真实性，大部分手写体、箭头的训练集都来自作业笔记、信手涂鸦。少部分来自互联网上已有的数据集。

4.2 功能需求

4.2.1 功能模块结构图



4.2.2 核心功能模块描述

表 功能模块描述

功能模块	功能	功能描述	优先级
前端	图片传入模块	用户在此上传手绘版的导图以及其他图片	前端最高优先
	查看与修改模块	用户在此查看所给导图的初稿，并且可以任意添加、更改和删除导图分支、可以改变文字的样式与内容以及可以选择导图的背景图，包括用户上传的图片和系统提供的单色	前端的次高优先
	文件导出模块	用户在此将所给的导图成品导出到用户端	前端的最低优先
后端	图形识别模块	对手绘版导图进行识别，判断其元素组成以及位置逻辑关系	后端的最高优先
	手写体识别	对手绘版导图的元素内容进行识别	后端的次高优先
	人工智能部分	对图形识别与手写体识别结果进行分析并生成适当导图	后端的次低优先
	优化模块	对人工智能部分生成的初稿进行结构和布局的优化	后端的最低优先

4.3 非功能性需求

4.3.1 扩展性和可维护性

[1]从拓展性的角度来看,本项目除了简单地根据用户提供的手绘图生成导图之外,还为用户提供了修改与美化导图的权限,提高了本产品的拓展性。另外本项目可以通过稍加修改部分程序以及改变训练素材来适用于除导图之外的一些手绘图识别,因此本项目的拓展性是非常高的。

[2]从可维护性的角度来看,本项目前后端各个部分组件分工明确,便于对其进行维护与更新。另外本项目在网页进行呈现,各个功能效果十分明显,易于检查与维护。因此本项目的可维护性是十分可观的。

4.3.2 易用性

本项目可以说是几乎无门槛。由于是在网页进行呈现,因此任何懂得网页基本操作的人均可使用本项目进行电子版导图的生成。另外交互界面简洁易懂,因此易用性十分的高。

4.4 性能需求

4.4.1 时间特性

前端图片初步处理时间应控制在一秒以内,否则影响用户体验。

后端处理时间较为难以预测,和用户网络情况、图片复杂度等均有要求。平均应在五秒以内做出响应。

后续的编辑、输出均为本地代码执行,操作延迟应该控制在不被发现的范围内。

4.4.2 适应性

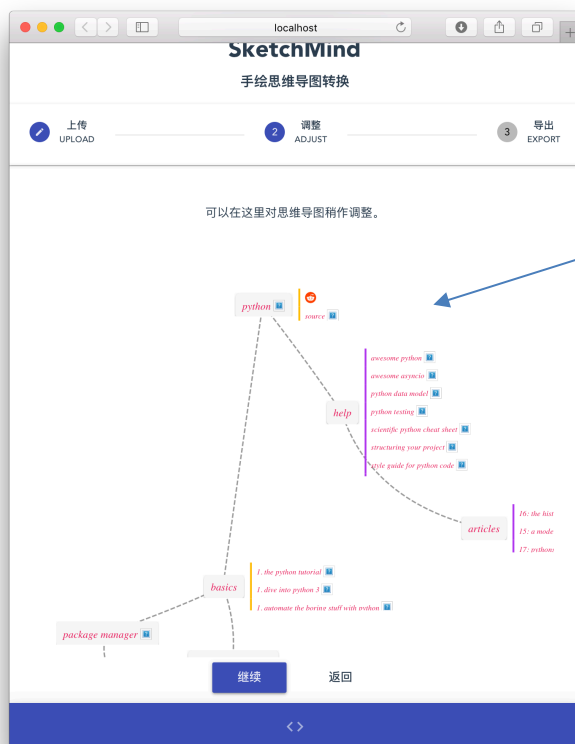
本项目应支持多种常见图片格式的输入输出,如 PNG、JPEG、GIF、BMP 等。

在思维导图工程文件的输出方面,目前仅支持开放 API 的 FreeMind 格式。后续可能会借由厂商帮助实现更多商业专有格式(如 XMind、ZenMind Pro 等)的输出支持。

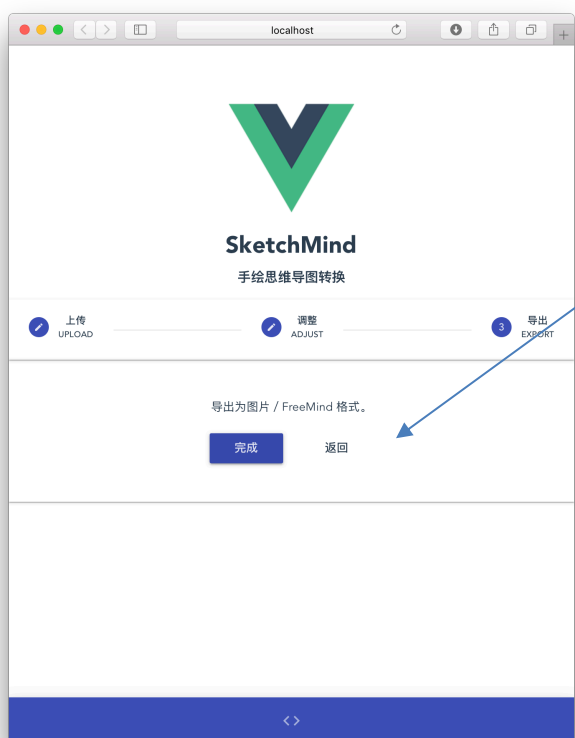
4.5 运行需求

4.5.1 用户界面





可以在此处预览思维导图效果，并进行编辑和纠错。



将最终的思维导图导出为图片格式或可编辑的工程文件。

4.5.2 硬件接口

无。本软件不需要任何特殊硬件接口支持。

4.5.3 软件接口

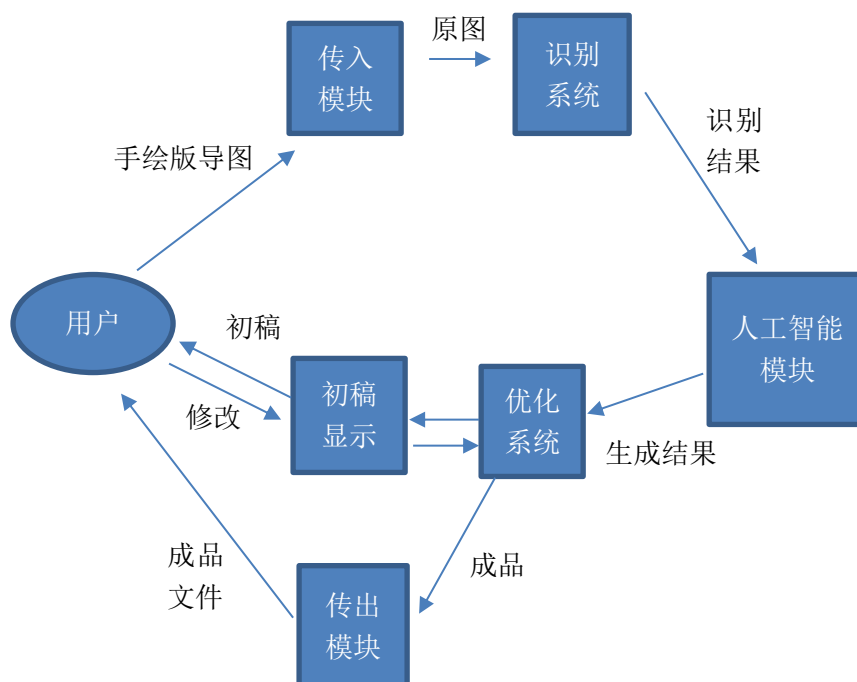
本软件需要的软件接口包括：

- Wikipedia（维基百科）词条查询 API
 - 主要用于思维导图编辑工具中的联想词功能
- 上传图片所需要调用的系统接口，用于访问本地图片文件
- 导出思维导图文件需要调用的系统接口，写入用户磁盘

4.5.4 故障处理

本项目在程序出错时，首先会自动将网页刷新，并重新将用户输入的图片进行处理，若仍有重复报错，则将弹出错误信息并自动将匿名错误信息反馈给开发小组。

4.6 用例图



4.7 核心模块用例规约

表 SketchMind 用例规约

用例名称	
功能简述	
用例编号	
执行者	
前置条件	
后置条件	
涉众利益	
基本路径	
扩展路径	
字段列表	
设计规则	
未解决的问题	
备注	

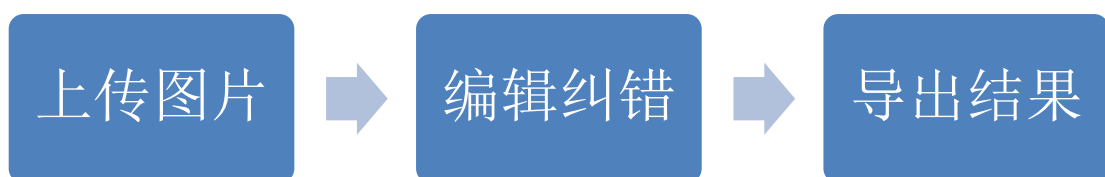
4.8 其他需求

本项目核心部分采用了人工智能,因此需要使用大赛官方提供的硬件设施进行训练与运行。而在此基础上,本项目以网页为呈现形式使得绝大多数的浏览器均可运行本产品,可移植性很高,因此在更换环境时没有额外的需求,也更利于维护。另外,本项目以人工智能为基础,安全性与稳定性更有保障。

5 概要设计

5.1 处理流程

使用本 Web 应用相对简单,不需要太多基础知识。界面提示布局较为合理,没有过高的技术门槛。基础操作仅仅分为三个步骤,相对较直观。

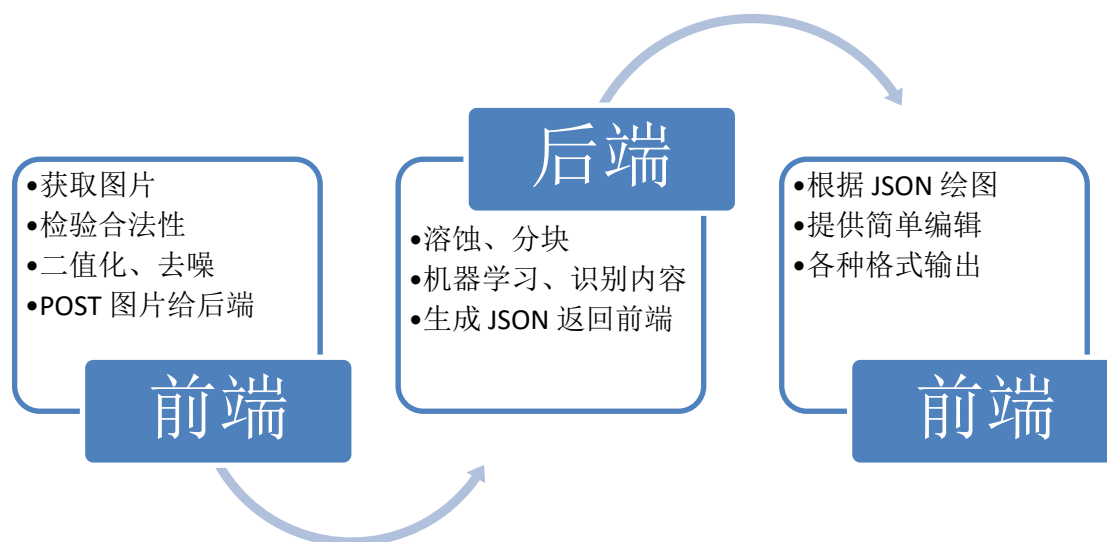


至于数据的实际流动见下页图：

为了减缓后端压力，部分较为简单的图片操作都被分工给前端网页，利用 Web 2.0 的思想，内嵌了图片预处理 JavaScript 脚本的网页可以提前对图片进行检测和处理。

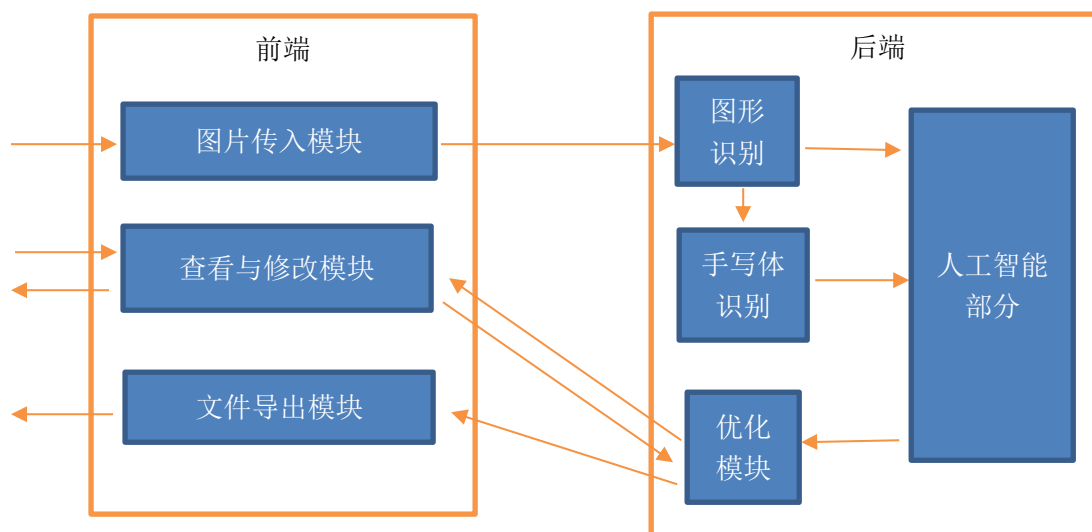
思维导图以 JSON 格式在前后端之间流动，相比于直接传输图片更为节约带宽且更为灵活，又比二进制格式易于维护，因此作此选择。

在得到 JSON 之后，前端可以直接在网页中绘制出思维导图，并进行一系列编辑、导出操作，这一部分无需服务器参与，编辑更为高效。

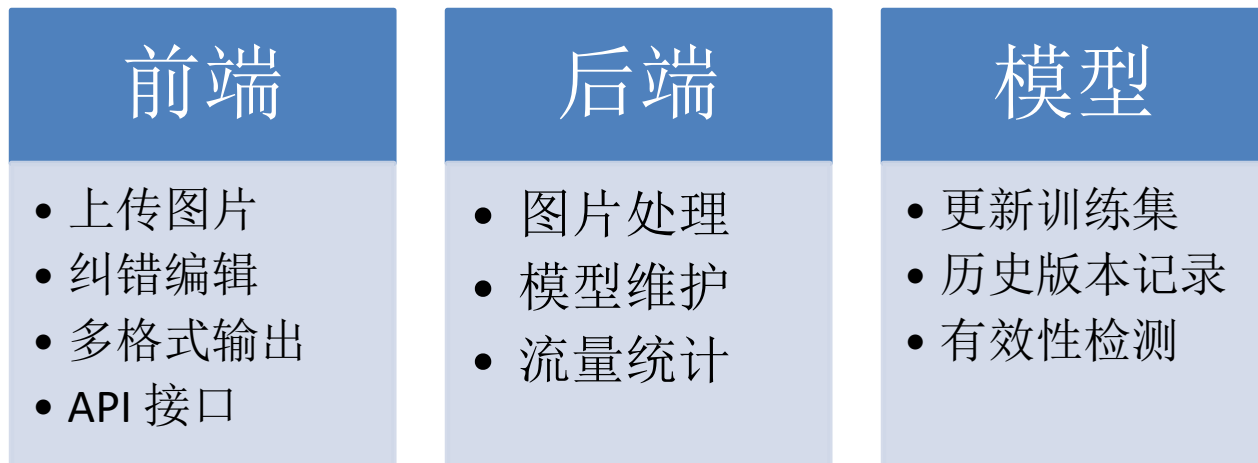


5.2 总体结构和模块设计

参照模块结构图，总体系统的结构和模块设计如下图所示：



5.3 功能分配



其中前端功能和 API 接口直接暴露给用户，后端管理系统和模型管理系统都用于内部维护使用，不对用户开放。

5.4 接口设计

5.4.1 外部接口

为了提升流量，吸引用户使用，此应用还提供公开的 API 方法调用，只需 POST 图像数据到指定地址就可以返回对应的 JSON 数据集合。

为了方便使用，后续本应用还可能支持与摄像头、扫描仪和打印机的互通，进一步方便纸质材料电子化的整理。

5.4.2 内部接口

前后端之间的接口已经在前面讨论 JSON 格式时详述过。

后端与模型服务器的联系：在后端执行正常操作时，不去更新模型，但偶尔核查模型版本的更新。在预定的维护时间段，进行模型数据的更新与有效性检验，确保系统安全稳定运行。

5.4.3 用户界面设计

由于本应用的针对人群不是专业人士，而是学生和日常使用者等不具有很高的技术力的群体，因此我们尽量简化用户界面的设计，提供了直观可交互的进度条、应用了谷歌提出的 Material Design 设计模式，为用户提供了直观准确的操作反馈。

5.5 数据结构设计

5.5.1 逻辑结构设计

逻辑上返回的 JSON 文件，我们以 Node 节点为基本元素，每个 Node 包含以下信息：

- 相对 x、y 位置（在原图片中），方便重建思维导图
- 文本信息
- 该节点所包含的子节点

子节点的格式同父节点相同。节点可嵌套。

由于 JSON 格式的特性，构造这一数据结构相较于邻接矩阵等方式更直观易懂，也方便绘制。

5.5.2 物理结构设计

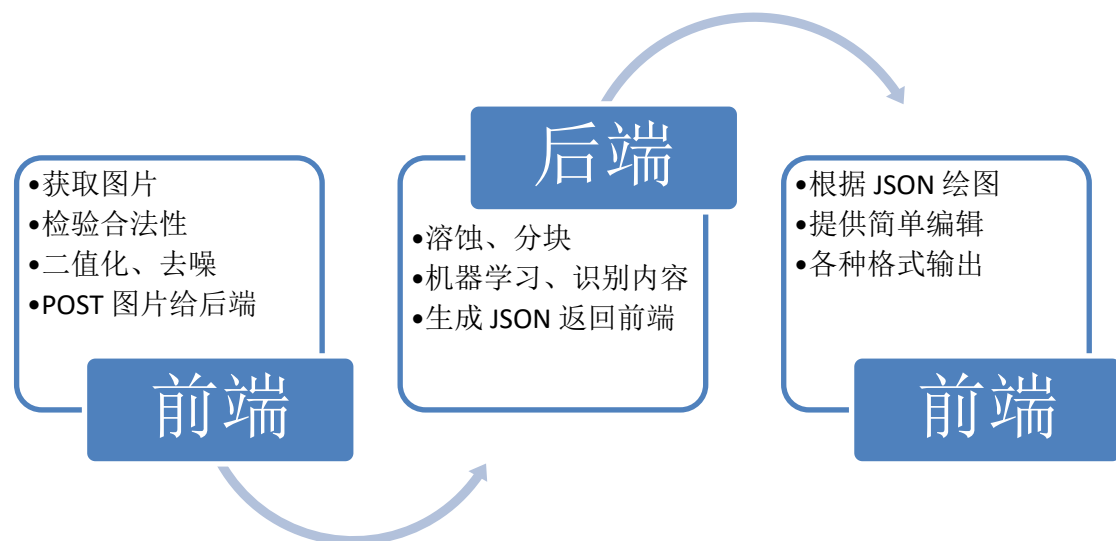
物理上我们将后端服务器托管在阿里云上，在国内拥有较快的访问速度。模型的训练则采用 Intel AI DevCloud 进行训练和有效性检测。

5.5.3 数据结构与程序的关系

这一数据结构是后端产生后返回给前端的，而前端也按照此格式进行后续的显示和输出。因此这一数据结构在整体程序的各个部件中都具有重要的意义，应该预先研究确定，不可轻易更改。

5.6 运行设计

5.6.1 运行模块关系



参照上图，前端利用嵌入网页的 JavaScript 脚本，提供了初步的图像处理检验和结果的绘制。而后端仅仅生成所需要的 JSON 格式数据，不直接参与界面的操作。因此就实现了数据处理和界面显示的分离，便于后续分别维护升级。

5.6.2 运行控制

程序主要由后端服务器发送的指令控制。在后端服务器不工作时，前端拒绝给出结果并提出警告。因此应该将控制权放在后端服务器上，以应对突发的情况。

5.6.3 运行时间

前端图片初步处理时间应控制在一秒以内，否则影响用户体验。

后端处理时间较为难以预测，和用户网络情况、图片复杂度等均有要求。平均应在五秒以内做出响应。

后续的编辑、输出均为本地代码执行，操作延迟应该控制在不被发现的范围内。

5.7 错误/异常处理设计

5.7.1 错误/异常输出信息

由于有效的数据均来自服务器，因此同样可以利用 JSON 来携带错误信息。只要没有提供有效合法的数据结构都算做是意外情况，并提供有意义的错误信息和代码。与此同时，应该在后端服务器中记录这一失败或异常，以便后续修复或改进。

鉴于此产品的大众化、非专业化的产品定位，不应该将详细的错误信息和调试信息展示给用户。相反应该强调问题反馈的醒目度，促使用户主动反馈问题，解决问题。

5.7.2 错误/异常处理对策

在发现异常或错误的情况下，应该首先调取发生问题的测试用例并尝试复现这一问题。随后，尽快制作出 patch，进行全面测试之后在维护阶段进行上线。

5.8 系统配置策略

在上线初期用户量比较小的情况下，首先不宜提供过高的服务器容量，而应将经费尽可能花在用户反馈和迭代改进上。在系统较为成熟之后，再进行服务器容量的扩大，此时的系统也能达到一个较高水平。

5.9 系统部署方案

由于本系统主要面向国内用户，因此系统部署在中国境内的阿里云计算服务器上。

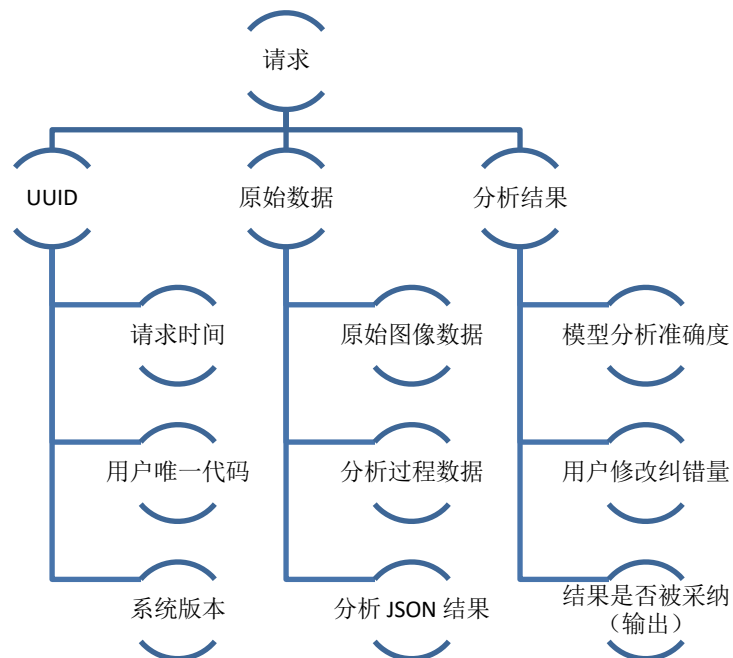
5.10 代码规范

由于本系统初期开发过程中的人数有限，因此提交高质量的代码很有必要。经过我们的协商讨论，JavaScript 编码部分的标准我们统一采用了 Airbnb JavaScript Style Guide。详细信息可以访问该项目的仓库地址 <https://github.com/airbnb/javascript>。

5.11 其他相关技术与方案

6 数据库设计

数据库中主要内容为每次用户所提起的请求的完成情况、后台分析数据等等信息。具体的 ER 图如下：



7 详细设计

7.1 功能模块

复赛阶段我们只提供所有功能模块中的图形识别模块作为关键模块进行分析设计。

7.1.1 功能描述



7.1.2 性能描述

根据上面的性能要求,图形识别模块应该在一秒内完成此部分工作。经过实际测试发现,在一台普通的笔记本式计算机上能够实现在一分钟内处理数十张一般大小的图片。因此在服务器上此目标应该是现实的。

7.1.3 输入

输入数据为前端初步处理后的图片内容。已经预先做过降噪处理和尺寸压缩。

7.1.4 输出

输出一系列四元组,其中每一个四元组表示一个矩形区域。每一个元组还包括识别后的类型信息。

7.1.5 算法

本程序主要采用了跨平台计算机图形学库 OpenCV 中的 Erode 和 findContours 方法来实现这一功能模块。具体代码参见 SketchMind/SketchMind.ML/parsing/ 目录中的 Python 脚本。

7.1.6 程序逻辑

此部分程序首先对给出的图片进行二值化、并施加 Erode 算法。随后,再使用 findContours 方法来分离各部件。此步骤将会循环进行,每次都改变 Erode 算法的参数,最终取中位数个

分块大小作为结果。

7.1.7 接口

此部分模块仅提供内部传入图片和输出元组的接口。不暴露外部接口。

7.1.8 存储分配

此模块不采用任何本地存储。

7.1.9 限制条件

对于部分底色不纯或带有干扰性底纹的图片可能具有很差的效果。
对于部分较大较复杂的图片处理时间可能会过长。