



回家! GIS 温暖回家路

中国地质大学（武汉）

项目开发文档

报 名 号：_____高级开发组_____K20175020

作品名称：_____回家_____

指导老师：_____郭明强 方芳_____

团队成员：_____余列冰 王俊珏 杨美娟_____

学 校：_____中国地质大学（武汉）_____

二零一七年 九月



目录

| | |
|---------------------|-----------|
| 目录 | 1 |
| 1.系统概述 | 3 |
| 1.1 编写目的 | 3 |
| 1.2 项目背景 | 3 |
| 1.3 需求分析 | 4 |
| 1.4 系统说明 | 6 |
| 1.4.1 运行环境 | 6 |
| 1.4.2 开发环境 | 6 |
| 1.5 技术分析 | 6 |
| 1.5.1 MapGIS 10 | 6 |
| 1.5.2 IntelliJ IDEA | 6 |
| 1.5.3 MySQL | 7 |
| 2. 系统设计 | 7 |
| 2.1 架构设计 | 7 |
| 2.1.1 架构分析 | 7 |
| 2.1.2 设计思想 | 7 |
| 2.1.3 架构体系 | 7 |
| 2.2 功能模块设计 | 12 |
| 2.2.1 功能模块汇总表 | 12 |
| 2.2.2 功能模块关系图 | 14 |
| 2.2.3 功能模块详述 | 15 |
| 3. 模型设计 | 18 |
| 3.1 路段失踪风险评估模型 | 18 |
| 3.1.1 算法原理 | 18 |
| 3.1.2 模型构建与可视化 | 19 |
| 3.2 区域失踪风险评估模型 | 20 |
| 3.2.1 算法原理 | 20 |



| | |
|---|-----------|
| 3.2.2 模型构建与可视化 | 21 |
| 3.3 走失点聚类模型 | 22 |
| 3.3.1 算法原理 | 22 |
| 3.3.2 模型构建与可视化 | 23 |
| 4. 数据库设计 | 24 |
| 4.1 逻辑设计 | 24 |
| 4.2 物理设计 | 25 |
| 4.2.1 表汇总 | 25 |
| 4.2.2 表[1]:[user_table]表 | 25 |
| 4.2.3 表[2]:[missing_people_data_table]表 | 26 |
| 4.2.4 表[3]:[user_attention_table]表 | 28 |
| 4.2.5 表[4]:[clue_table]表 | 30 |
| 5. 系统亮点与技术特色 | 32 |
| 5.1 系统亮点 | 32 |
| 5.1.1 人脸搜索的运用 | 32 |
| 5.1.2 建模思维的运用 | 32 |
| 5.1.2 数据分析功能的添加 | 32 |
| 5.2 技术特色 | 33 |
| 5.2.1 MapGIS Web 端插件式开发 | 33 |
| 5.2.2 实时通信系统 | 33 |
| 5.2.3 爬虫技术 | 33 |
| 6.界面设计 | 34 |
| 6.1 设计原则 | 34 |
| 6.2 界面布局 | 34 |
| 6.3 界面展示 | 35 |
| 7. 项目总结 | 36 |



1. 系统概述

1.1 编写目的

项目开发文档能提高软件开发的效率，保证软件的质量。为了详细说明“回家”——寻亲系统的设计与开发过程，达到指导，帮助，解惑的作用，同时便于开发人员的相互交流，以及系统的维护，我们编写了此文档。

本文档面向开发人员、项目指导老师、大赛评委老师、测试人员及最终用户编写，是了解“回家”——寻亲系统的导航。

1.2 项目背景

全国两会上，利用“互联网+”创新打拐工作引起全国人大代表和有关专家的热议。随着“互联网+”与大数据时代的到来，互联网正在改变寻亲的方式——“精准推送，全民助力”即通过互联网强大的时事通信功能进行实时消息的推送，通过互联网的广大用户发挥全体网民的集体力量。互联网公益寻亲的话题也正受到越来越广泛的关注。

寻亲的关键是失踪人员地理位置的确定、相关信息的匹配（照片人脸、特征描述等）以及线索消息的推送，然而目前国内流行的寻亲网站都是单一的相关信息登记与展示，没有与地理位置相结合，更没有一个强大的消息系统提供线索的广播以及用户之间的实时交流功能。“百度寻人”等网站结合深度学习技术，运用人脸搜索将线索人物照片与失踪人员信息库中的失踪人员照片进行比对把数据库中相似度较高的失踪人员信息返回给用户来达到寻人的目的，但是失踪人员信息中最为重要的地理信息数据却任然没有被利用。

另一方面，目前国内的寻亲网站也缺乏对这些大量的失踪人员地理位置信息分布的统计与研究。据不完全统计，全国每年的失踪人口高达 800 万，加之往年



未找到的失踪人口，每年在寻人口高达 1000 万，失踪人口已经成为一个严重的社会问题。如何探寻这些大量的失踪人员信息背后的规律与奥秘，从而辅助公安机关等相关政府机构采取相应的措施来减缓甚至解决失踪人口问题，已经成为了一个迫切需要解决难题。

基于上述现状，我们打算开发一个能够弥补传统寻亲网站不足的，具有地理空间数据分析能力的，功能完善的互联网公益寻亲系统——“回家”。

我们所开发的寻亲系统，利用了 MapGIS 10 平台良好的空间数据的存储、查询、展示、分析以及服务发布等强大的功能。以地理位置信息为核心，结合相应的实际需求，增加人脸搜索，全网搜索，实时消息推送等功能。建立一个功能完善的寻亲系统，充分发挥 MapGIS 在地理位置方面强大的功能，为互联网寻亲提供一个完整的需求分析与技术支持。

1.3 需求分析

随着“互联网+”时代的到来，互联网寻亲愈来愈体现出重大的社会意义。互联网寻亲系统可以充分利用网络的时效性，发挥广大互联网用户的力量以实现更加精准高效的寻亲。然而，目前还没有一个功能完善的寻亲系统，为了提供一个完善的互联网寻亲系统，为寻亲所涉及的广大用户提供一个高效科学的平台，我们开发了“回家”——寻亲系统。

寻亲系统的用户主要可分为以下三类：

(1)寻亲者：急需寻找与自身直接相关的失踪人员。需要在寻亲网站上发布失踪人员的相关信息，并接收相关线索。

(2)志愿者：不需寻找与自身直接相关的失踪人员，但是乐于帮助寻找网站上已经登记的失踪人员。需要在寻亲网站上提供相应线索，集合其他志愿者开展相关公益活动。

(3)决策者：公安机关等相关政府机构。需要查看失踪人员的统计数据以及

失踪人员地理位置的时空分布。

用户与相关用例之间的详细关系如图 1 所示：

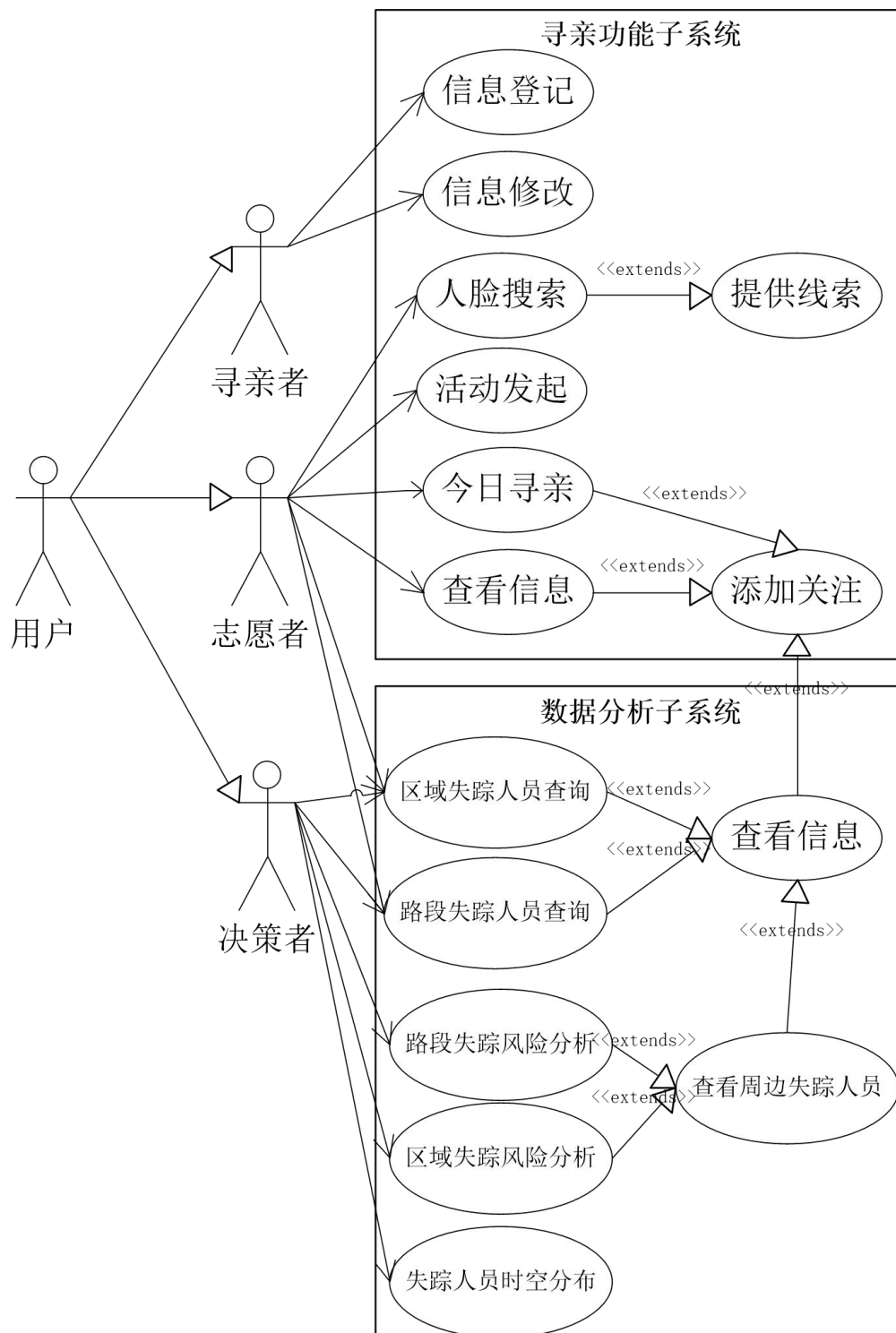


图 1 系统用例图



1.4 系统说明

1.4.1 运行环境

- (1)软件：Microsoft Windows 7/Windows 8/Windows 10, Mac OS;
- (2)硬件：2GB 以上内存、2GB 以上剩余硬盘空间

1.4.2 开发环境

- (1)操作系统：Windows10;
- (2)开发语言：前端，Html，CSS，JavaScript；后台，Java;
- (3)开发环境：IntelliJ IDEA 2017，MapGIS Server，Tomcat8.5;
- (4)数据库：MapGIS 10 空间数据库，MySQL;
- (5)SDK：zongdyClient.js，ol.js，JDK1.8。

1.5 技术分析

1.5.1 MapGIS 10

MapGIS 10 云 GIS 软件可由用户根据自己所需聚合、重构成各种 GIS 工具，最终迁移形成自己所需的行业应用；同时提供了完备的地图编辑数据工具、海量地理数据的可视化管理以及组件式开发模式，可进行数据的专业构建分析，为用户能够快速搭建数据服务体系。开发人员可以很方便地在此基础上二次开发，构建多种应用功能。

1.5.2 IntelliJ IDEA

IntelliJ IDEA 简称(IDEA)，是 Java 语言开发的集成环境，IDEA 是 JetBrains 公司的产品，支持 HTML，CSS，JavaScript，MySQL，Python 等。IDEA 是一个完整的开发工具集，它包括了整个软件生命周期所需要的大部分工具。由于失踪人员数据的爬取模块利用 Python 语言编写，IDEA 对 Python 的支持大幅增加了后台编码效率。同时 IDEA 对前端语言的强大支持能力也使得所开发的系统兼容更多的浏览器。因此，利用 IDEA 对系统进行标准化的架构和开发能够增强工



作效率，同时使得开发系统的兼容性更高。

1.5.3 MySQL

MySQL 是一个关系型数据库管理系统，由瑞典 MySQL AB 公司开发，目前属于 Oracle 旗下产品。MySQL 是最流行的关系型数据库管理系统之一，在 WEB 应用方面，MySQL 是最好的关系数据库管理系统应用软件。因此，选用 MySQL 存储与地理位置数据无关的业务数据。

2. 系统设计

2.1 架构设计

2.1.1 架构分析

“回家”——寻亲系统的开发涉及大量的 GIS 服务、人脸搜索相关的业务逻辑以及大量的地理信息数据展示。因此采用一个合理的架构，能够将各个模块进行解耦实现前后端的分离，在提高开发效率的同时也能体现软件结构体系的科学性。

2.1.2 设计思想

项目借鉴传统的三层架构（即数据访问层、业务逻辑层和界面层）。在传统的三层架构之上增加数据爬取层和数据处理层，通过轮询存储将空间数据存储于 MapGIS 10 空间数据库，利用 MapGIS IGServer 发布的服务连接请求处理层与空间数据库。数据访问层用于处理业务数据库的增删查改，业务逻辑层调用 Face++ 第三方接口，负责人脸检测和人脸搜索相关逻辑。

2.1.3 架构体系

系统对当前现有寻亲网站的失踪人员信息进行爬取，通过正则表达式匹配失踪人员失踪地点后进行逆地理编码得到具体的经纬度坐标，并将失踪人员照片加入项目对应的 Face++ 人脸照片集合，用于之后的人脸搜索与匹配。经过逆地理



编码之后的数据通过轮询存储插入 MapGIS 10 空间数据库。我们基于数据建模，采用核密度分析、缓冲区分析、聚类分析等 GIS 分析技术分析失踪人员数据的时空特点，通过数据可视化手段展示失踪数据背后的规律与含义。

系统整体架构主要可分为空间数据部分和业务数据部分，空间数据部分，数据爬取层负责历史数据与实时数据的爬取，数据处理层负责相应数据的处理，并存入空间数据库，通过 MapGIS IGServer 发布相应的服务，最终用于客户端的展示；业务数据部分由数据访问层负责业务数据的访问，业务逻辑层负责业务逻辑的处理，最终用于客户端的展示。系统的整体架构图如图 2 所示：

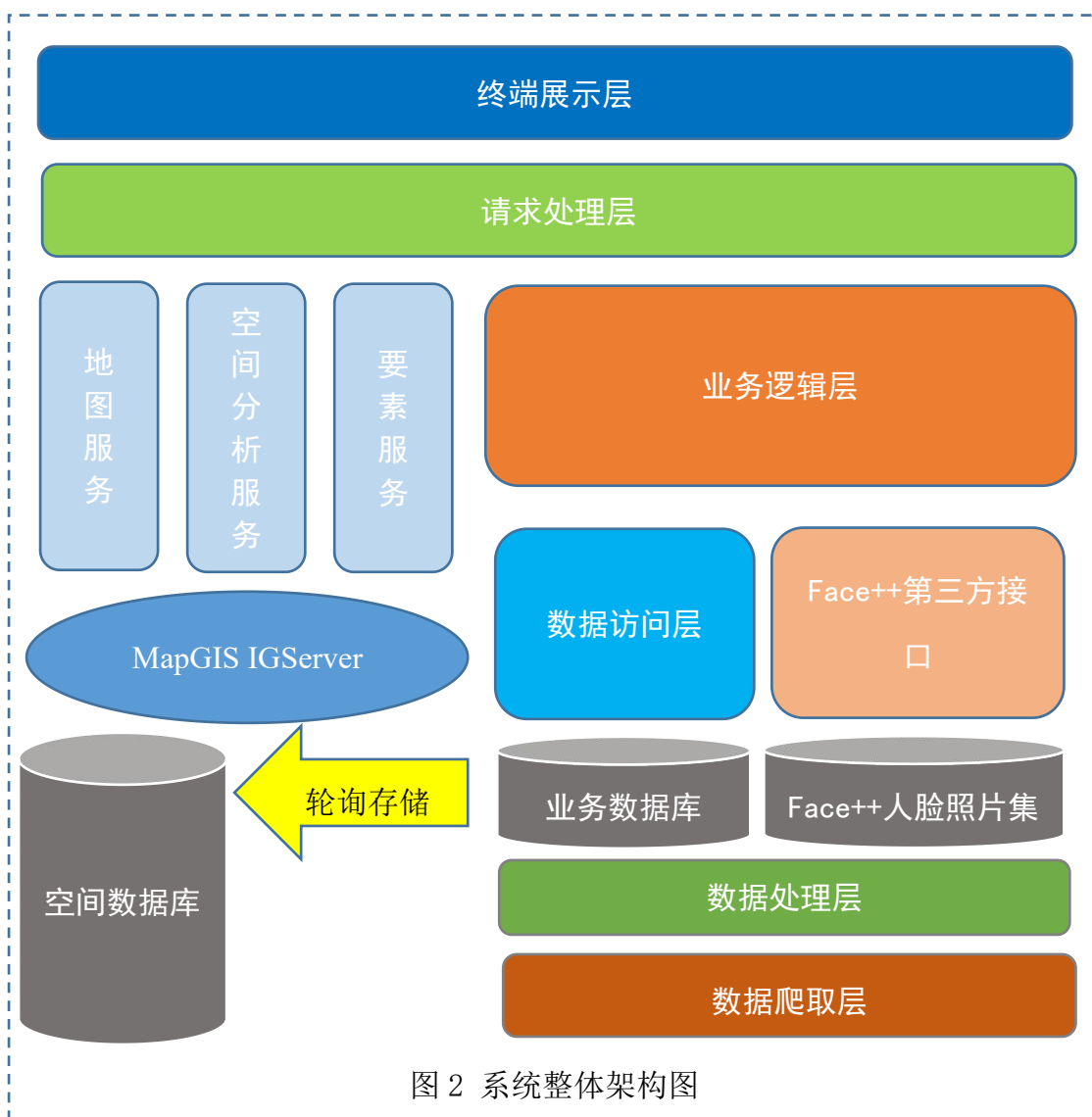


图 2 系统整体架构图

系统的后台架构主要可分为数据爬取层、数据处理层、服务发布层、数据访问层和业务逻辑层。

数据爬取层：我们利用 python3 原生 urllib 库编写的爬虫框架如图 3 所示：

- ①将网站主页 URL 插入 URL 队列；
- ②读取 URL 队列中元素发送至页面下载器；
- ③对获取到的 HTML 文本进行解析，得到的 URLs 插入 URL 队列、Items 进入项目管道；
- ④遍历项目管道，将数据插入从数据库中；
- ⑤（从②）重复直到 URL 队列为空。

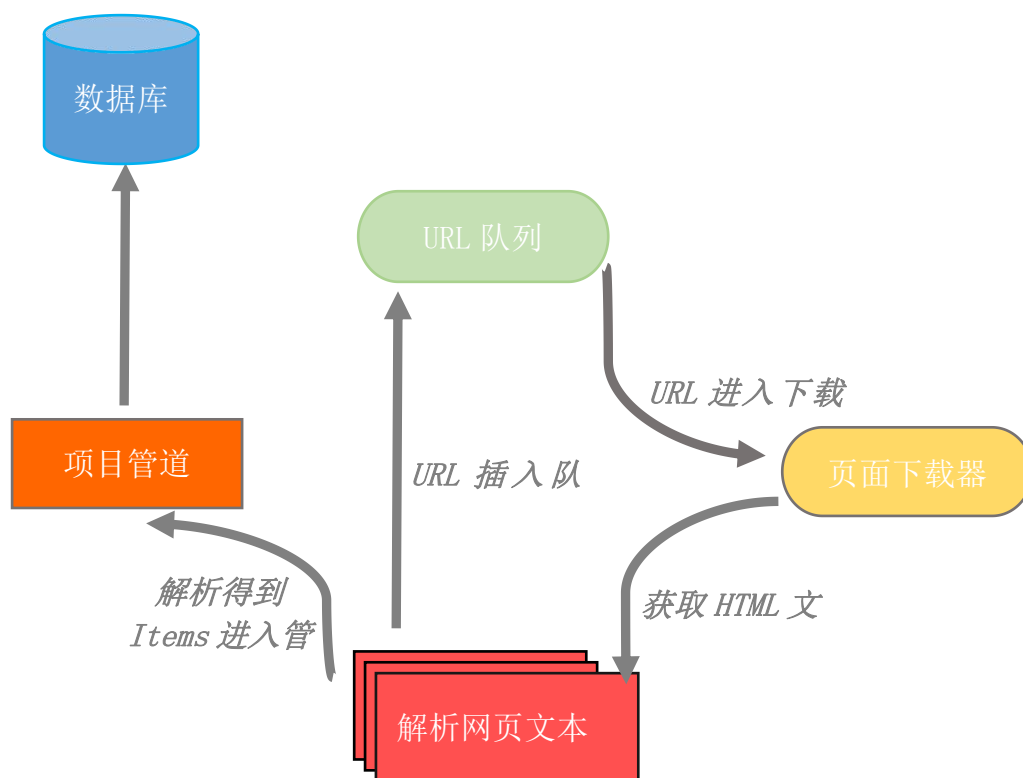


图 3 爬虫框架图

数据处理层：将数据爬取层得到的原始数据进行逆地理编码获得失踪人员所在地点的经纬度坐标，并将结构化数据存入 MySQL 数据库，将含有失踪人员人脸的照片加入项目对应的 Face++ 人脸照片集合。在数据分析上我们专注于 GIS 时空数据的分析。将经过逆地理编码后的数据进行聚类分析获得失踪人员分布密



度最大的点，并利用各个类簇的失踪人员个数计算该点的影响力范围。在易失踪路段分析上，我们采用核密度分析法，利用现有的失踪人员数据与路网数据计算武汉市各路段的失踪风险，并对结果进行分类评估。

服务发布层：我们使用了 MapGIS IGServer 进行地理服务的管理。其中失踪人员数据通过要素服务进行发布，轮询其 Rest 接口进行地理数据的增删查改；空间分析服务通过要素缓冲分析查询缓冲区失踪人员。

数据访问层：数据访问层与业务数据库中的表相对应，负责业务数据库的增删查改。其类图如图 4 所示：

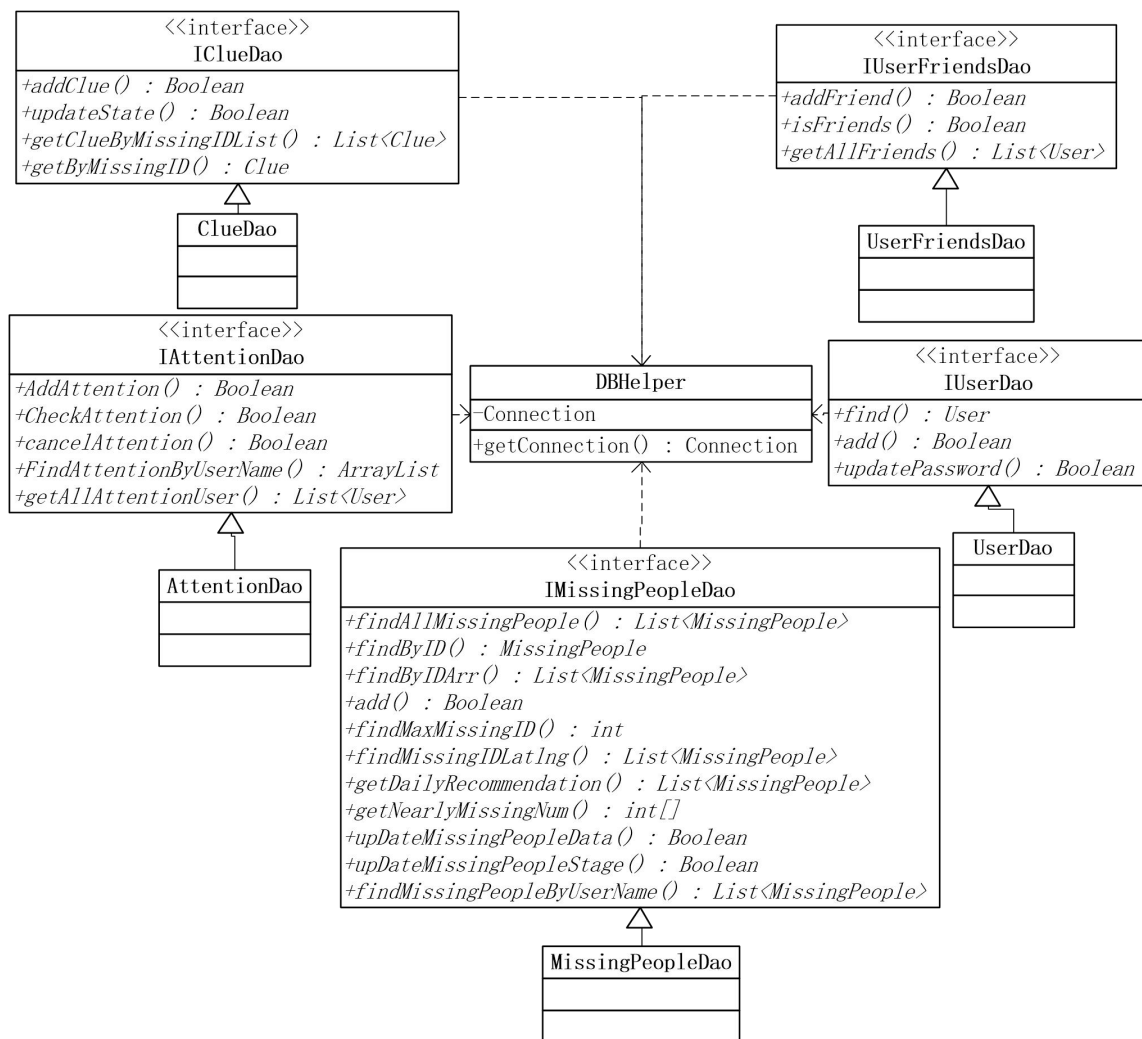


图 4 数据访问层类图



其中类 DBHelper 负责数据库的连接工作，类 UserDao 负责系统用户数据的处理，对应数据库中的表 user_table；类 MissingPeopleDao 失踪人员信息的数据处理，对应数据库中的表 missing_people_data_table；类 AttentionDao 负责用户关注的失踪人员信息处理，对应数据库中的表 user_attention；类 ClueDao 负责失踪人员先关线索的处理，对应数据库中的表 clue_table；UserFriendsTable 负责系统用户之间好友关系的处理，对应数据库中的表 user_friend_table。

业务逻辑层：业务逻辑层可分为人脸搜索模块和实时消息推送模块。人脸搜索模块主要负责调用 Face++ Rest 接口进行人脸检测，人脸搜索，人脸相似度计算，其框架如图 5 所示：

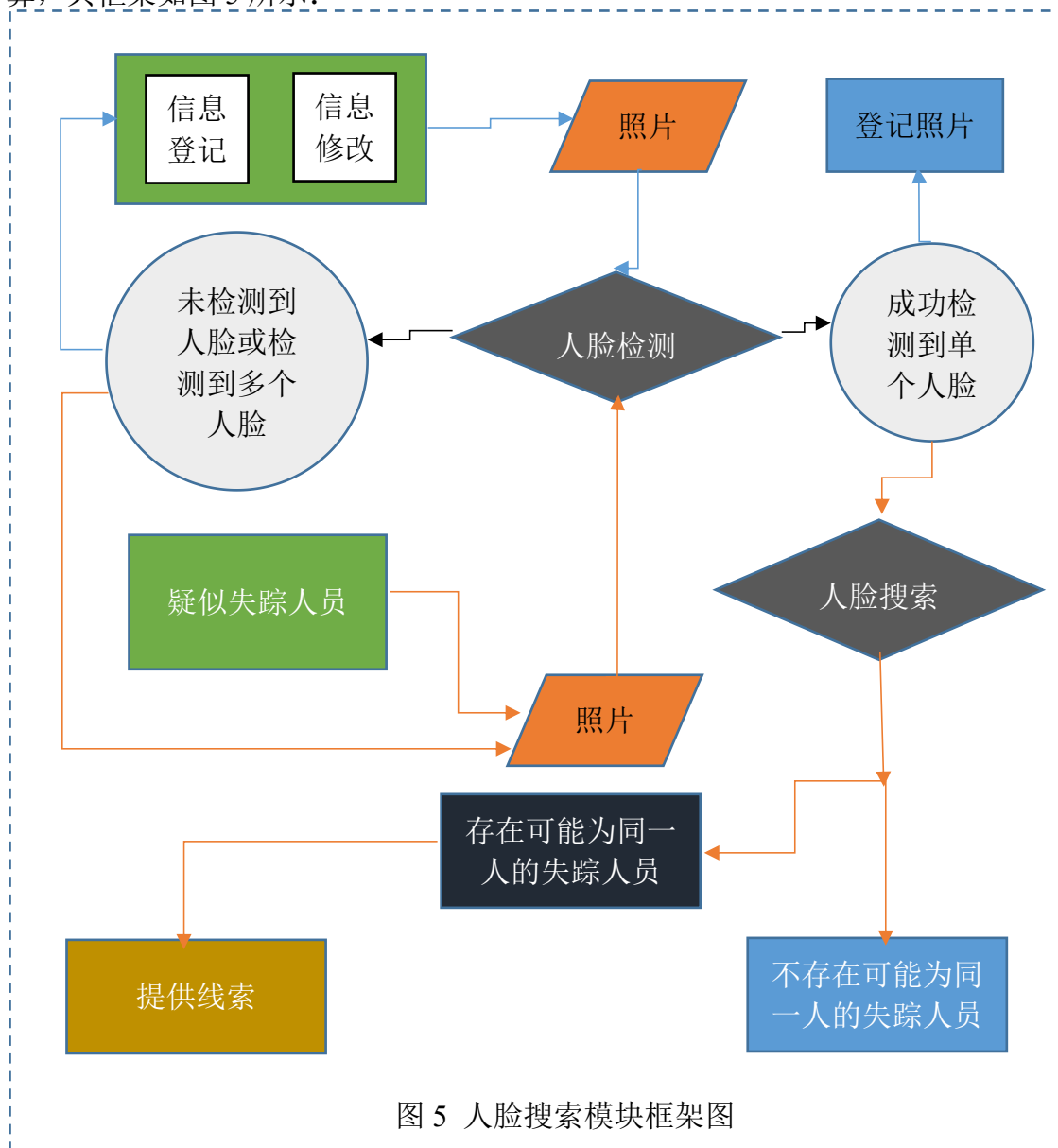
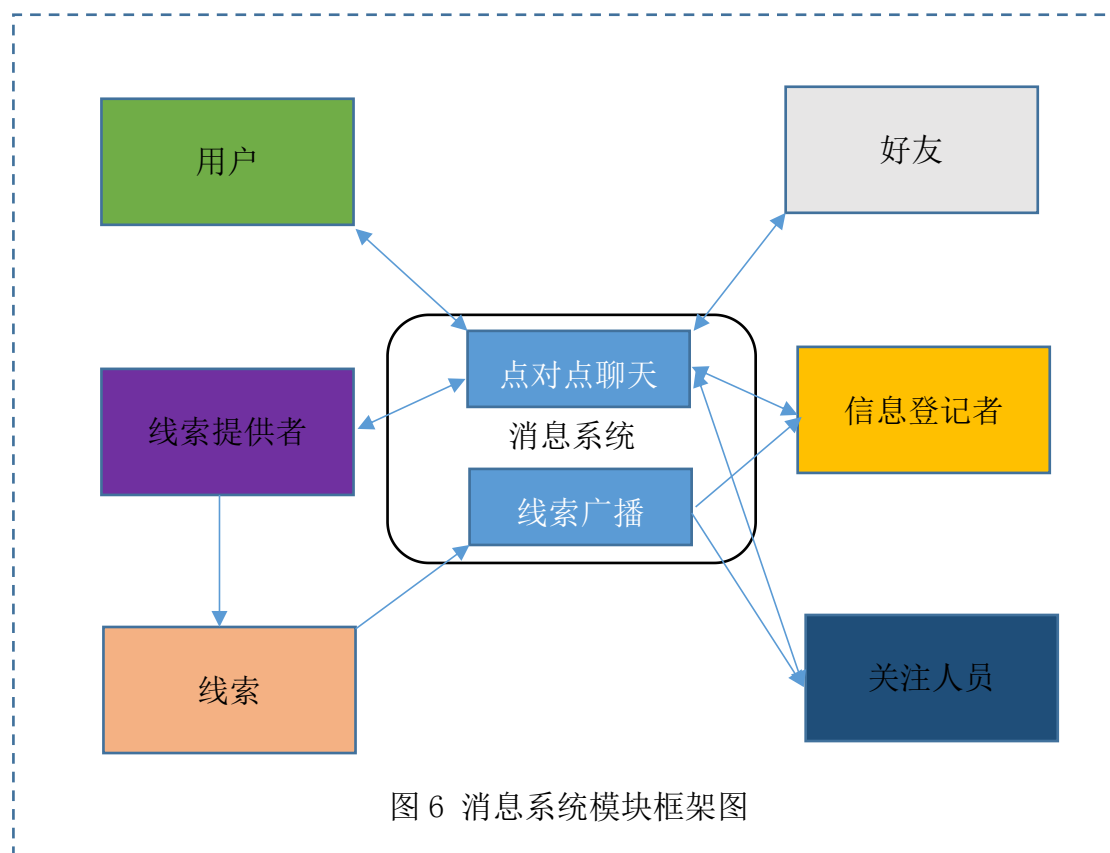


图 5 人脸搜索模块框架图

实时消息推送模块负责在线聊天功能的实现与相关线索的广播，其框架如图 6 所示。



系统的前端架构主要可分为请求处理层和终端展示层。

请求处理层：请求处理层负责处理前端请求，通过与 MapGIS IGServer 相关服务以及业务逻辑层的交互将处理结果返回给终端展示层。

终端展示层：终端展示层负责最终的数据展示以及系统与用户之间的交互。

2.2 功能模块设计

2.2.1 功能模块汇总表

功能模块汇总表如表 1 所示：

表 1 功能模块汇总表



| 寻亲功能子系统 | |
|---------|---|
| 模块名称 | 功能简述 |
| 信息管理模块 | 显示所有失踪人员信息； 按属性进行查询与高级搜索； 对失踪人员进行添加、取消关注； 失踪人员信息登记； 失踪人员信息修改； |
| 地图查询模块 | 矢量图层交互式点查询； 矢量图层交互式线查询； 矢量图层交互式圆查询； 矢量图层交互式拉框； 矢量图层交互式多边形查询； 矢量图层属性查询模块； |
| 地图显示模块 | 图层叠加显示； 地图聚合标注显示； popup 弹出框显示，显示失踪人员详细信息； 地图定位； |
| 人脸搜索模块 | 人脸搜索，相似度匹配； 对匹配相似度较高的失踪人员关注、注册者提供线索； 关注者、注册者与志愿者进行在线聊天； |
| 消息推送模块 | 对志愿者提供的信息进行实时推送； |
| 志愿者模块 | 今日寻亲推送； 寻亲宣传活动发起； |
| 全网搜索模块 | 通过调用 bing 搜索引擎对失踪人员进行全网搜索； |
| 数据分析子系统 | |
| 模块名称 | 功能简述 |
| 风险分析模块 | 易失路段分析，周边情况查看； 区域失踪风险评估； |



| | |
|--------|---|
| 图表分析模块 | 寻亲热力图； 各类统计图表； 地图分析。 |
| 地图显示模块 | 图层叠加显示； 地图聚合标注显示； popup 弹出框显示，显示失踪人员详细信息； 地图要素动画，聚类点影响力范围图； 地图定位； |
| 地图查询模块 | 图层叠加查询； 矢量图层交互式点查询； 矢量图层交互式线查询； |
| 空间分析模块 | 要素缓冲区分析。 |

2.2.2 功能模块关系图

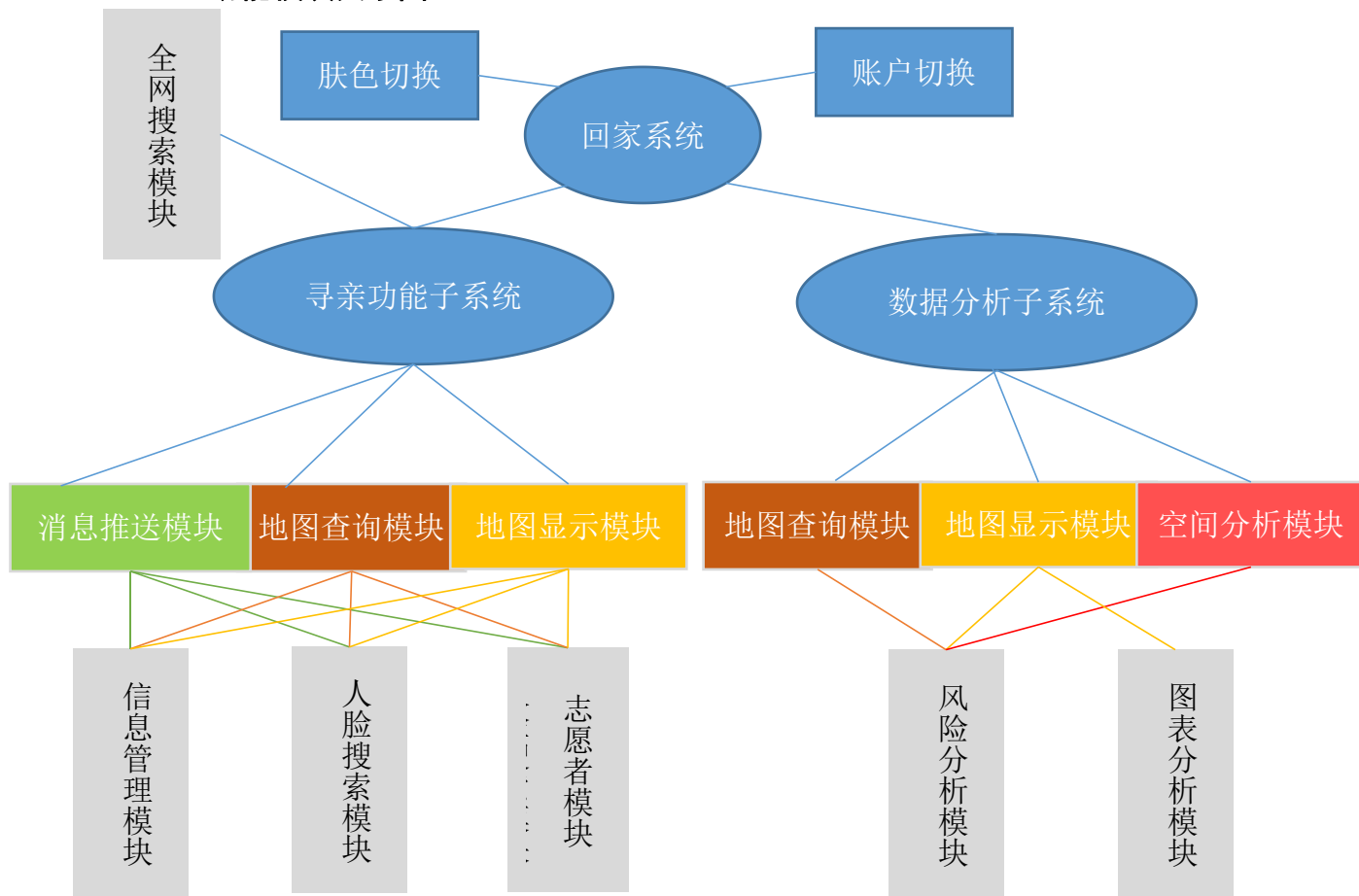


图 7 功能模块关系图



2.2.3 功能模块详述

寻亲功能子系统

全网搜索模块：该模块通过利用 Bing 引擎进行全网搜索，扩大了失踪人员数据来源范围，将结果根据相关度排序展示主页浮动框中，用户通过点击标题可获取详情信息。

地图查询模块：该模块主要实现“回家”——寻亲系统中矢量图层查询的基本功能，其中包括矢量图层交互式点查询、矢量图层交互式线查询、矢量图层交互式圆查询、矢量图层交互式拉框、矢量图层交互式多边形查询、矢量图层属性查询功能，通过调用 zongDyClient.js 等脚本库从 MapGIS IGServer 的空间数据中提取数据进行展示。

地图显示模块：该模块主要实现地图显示功能，展示数据有矢量图层数据、瓦片数据等，展示形式有图层叠加显示、聚合标注形式、Popup 弹出框形式、要素动画形式，通过调用 ol.js、zongDyClient.js 等脚本库实现良好的交互功能，向用户展示相关信息。

信息管理模块：该模块实现系统中信息管理的基本功能，包括显示所有失踪人员信息、按属性进行查询与高级搜索、对失踪人员进行添加、取消关注、失踪人员信息登记、失踪人员信息修改等信息管理操作。

人脸搜索模块：该模块实现了大规模图像信息检索功能，志愿者通过上传含有疑似失踪人员人脸的照片，与服务器中已登记的失踪人员照片进行对比匹配，从而获得相似度最高的 4 张照片以及对应的失踪人员详细信息，供志愿者参考辨别。此模块是驱动“回家”系统的核心引擎；对于相似度大于 80% 的照片提供“一键提供线索”功能，志愿者可以录入疑似失踪人员的详细信息，系统会实时对关注者、注册者进行消息广播，并提供“点对点在线聊天功能”，是连接志愿者与寻亲者的枢纽。



大规模图像信息检索原理介绍：本系统调用 Face++ 接口，利用深度学习框架训练人脸搜索模型，模型具体流程如图 8 所示：

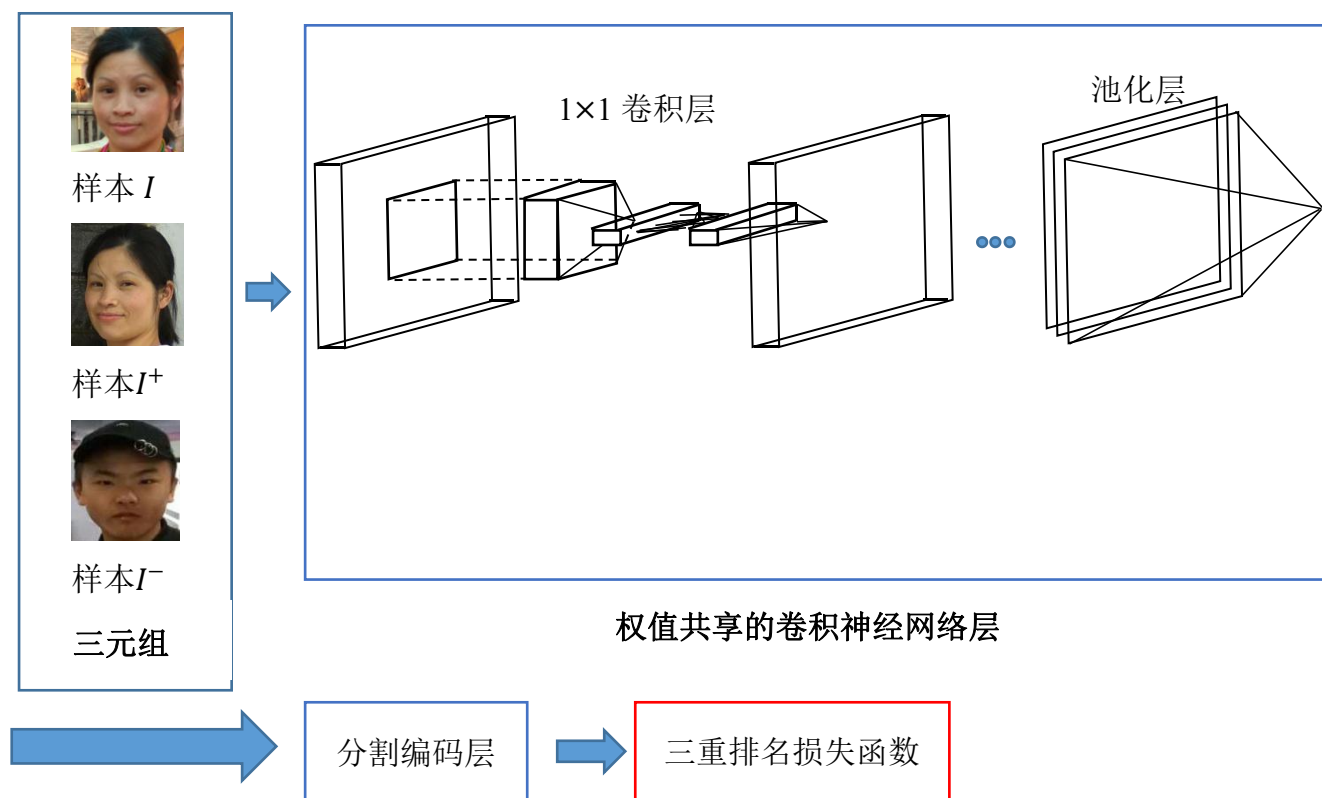


图 8 人脸搜索模型

①获取样本集向量（三元组），如图 8 中 I 、 I^+ 、 I^- ， I 与 I^+ 为相似样本， I 与 I^- 为不相似样本；该三元组样本标签为： I 图片二进制编码与 I^+ 图片二进制编码海明距离小于 I 图片与 I^- 图片距离；

②构建卷积神经网络进行训练，并利用权值共享提高模型效率与准确率；

③对训练后的结果进行分割与编码；

④将编码后的 I 、 I^+ 、 I^- 带入损失函数（式（1））；

⑤不断重复调整模型参数直至模型收敛。

$$\begin{aligned} \ell_{\text{triplet}}(F(I), F(I^+), F(I^-)) \\ = \max(0, \|F(I) - F(I^+)\|_2^2 - \|F(I) - F(I^-)\|_2^2 + 1) \end{aligned}$$



$$s.t. F(I), F(I^+), F(I^-) \in [0,1]^q \quad (1)$$

志愿者模块：志愿者模块实现了“今日寻亲”、“活动发起”两大功能，“今日寻亲”按时间列出最近一周丢失人员信息及地理位置供志愿者查看，通过“活动发起”功能，志愿者可以在地图上绘制路线并进行相关标注，制作自己的活动地图，打印出的地图也可以进行分享。

数据分析子系统：

地图查询模块：该模块在地图查询基础功能上增加了图层缓冲区查询。其流程如下：首先捕获用户交互式点击动作所在坐标，利用坐标组成的多边形形成缓冲区，然后查询缓冲区图层几何信息，最后对失踪点图层进行图层叠加查询，完成缓冲区查询模型。

地图显示模块：地图展示模块用于基本的地图展示，配合其他相关模块进行数据的展示。

空间分析模块：通过调用 MapGIS IGServer 接口完成要素缓冲区分析。

风险分析模块：该模块提供“易失路段分析”、“区域风险分析”两种交互式风险分析功能。“易失路段分析”主要利用“路段失踪风险评估模型”（见 3.1）得出武汉市各路段风险指数，并分级显示于地图上，通过交互点击查询获得选中路段详细信息，通过交互式线查询获得该路段周边走失情况。“区域风险分析”主要利用“区域失踪风险评估模型”（见 3.2），通过图层缓冲区查询，得出缓冲区域内失踪风险指数，并分级显示于地图上。

图表分析模块：该模块结合 E-chart 与失踪人员数据绘制各类分析图，帮助决策人员分析历史与近期人口走失情况，通过分析图表可直观的查看失踪人口的时空分布于年龄分布，如“寻亲热力图”、“近 10 年全国各省份人口失踪情况”、



“失踪人口年龄分布”、“近 20 年男女失踪人数折线图”、“最近 10 天失踪人数统计图”、“影响力范围图”、“各省份男女人数比例图”等。

3. 模型设计

3.1 路段失踪风险评估模型

在易失路段分析模块，项目基于武汉市路网数据与失踪点数据，结合地理学第一定律，利用核密度分析技术，构建了一个路段失踪风险评估模型，对武汉市路段失踪风险指数进行定性定量识别，并通过 MapIGServer 进行可视化展示。

3.1.1 算法原理

核密度的计算方法是基于地理学第一定律，即距离越近的事物关联越紧密，与核心要素越近的位置获取的密度扩张值越大。由于其顾及和体现了空间位置的差异性以及中心强度随距离衰减的特性，该方法适合于城市设施服务影响、交通路段风险评估等连续性地理现象的密度估计。

本项目利核密度法实现走失点影响扩散。核密度的公式定义为：

$$f(x) = \sum_{i=1}^n \frac{1}{h^2} \varphi\left(\frac{s-c_i}{h}\right) \quad (2)$$

式中 $f(x)$ 是位置 s 处的核密度估计函数； h 是路径衰减阈值，即带宽； n 是与位置 s 的路径距离小于或等于 h 的走失点数； φ 函数是空间权重函数，模型选择四次权重方程（式（3））

$$k\left(\frac{s-c_i}{h}\right) = \frac{3}{4} \left(1 - \frac{(s-c_i)^2}{h^2}\right) \quad (3)$$

考虑到均衡不同走失点的分布情况其中带宽 h 由式（4）决定：

$$h = 0.9 \times \min\left(SD, \sqrt{\frac{1}{\ln 2}} \times D_m\right) \times n^{-2} \quad (4)$$

$$SD = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{X})^2}{n} + \frac{\sum_{i=1}^n (y_i - \bar{Y})^2}{n}} \quad (5)$$

（5）中 x_i 和 y_i 是要素 i 的坐标， $\{\bar{X}, \bar{Y}\}$ ，表示输入点的平均中心， n 为要素总数，

（4）式中 SD 为输入点平均中心与所有点的标准距离， D_m 为输入点平均中心与所



有点的距离中值。

核密度估计的主要算法如下：

①定义一个搜索半径 r （由上式（4）、（5）决定），即距离衰减阈值；

②计算每个发生元到衰减阈值 r 内所有相邻路段的距离（点到直线的距离是点到直线的垂线或最近的折点的距离），所有距离小于 r 的相邻路段都应被考虑其中；

③在每个发生元以及它所有相邻路段处，基于所选择的核函数、衰减距离、走失点实体数计算其密度值；

④在与发生元的距离小于阈值以内的每条路段处，合计来自不同发生元的密度值之和，并将总密度（风险指数）分配给该路段，对于其他路段，密度缺省值为零。

最后，为了方便展示与计算，剔除风险度为0的路段并将各路段风险度进行归一化处理至0-10之间（公式（6））。

$$W^* = \frac{W - \min}{\max - \min} \times 10 \quad (6)$$

式（6）中 W 为风险指数

3.1.2 模型构建与可视化

按风险指数将武汉市易失路段分为5类，分类阈值如下：

| 风险指数 | 风险等级 |
|------------------|-------|
| $W < 0.1$ | 可能失踪 |
| $0.1 < W < 0.11$ | 较易失踪 |
| $0.11 < W < 0.3$ | 易失踪 |
| $0.3 < W < 1.0$ | 非常易失踪 |
| $1 < W < 10$ | 极易失踪 |

（表1）风险指数分级表

通过 MapGIS 根据图元属性更改参数，将路段按不同颜色与粗细分级显示并将矢量图层叠加至 OSM 底图上，通过交互式点击查询显示该路段具体信息，点击“周边情况”、输入查询半径、进行线查询即可查看周边走失情况。可视化果

如下图所示:



3.2 区域失踪风险评估模型

在区域失踪风险评估模块, 项目通过交互式缓冲区查询得到区域临近范围内走失点数目, 结合缓冲区面积求得走失密度, 并通过对走失密度进行分级处理。帮助用户对各区域风险定性定量识别, 并通过 MapGIS IGServer 进行可视化展示。

3.2.1 算法原理

失踪风险指数由缓冲区面积与区域内走失点数量得到 (如式 (7))

$$Z = \frac{N}{S} \quad (7)$$

式 (7) 中, Z 为风险指数、 N 为落入缓冲区内的点、 S 为缓冲区面积。其中缓冲区面积计算方法如下, 首先根据式 (8) 将缓冲区边界点经纬度反算为 X, Y 坐标, 用解析法 (式 (9)) 计算出缓冲区面积。

$$X = \frac{uB\pi}{180} - \{c - [0.5 + (F + ME^2) \times E^2] \times NE^2\} \times \sin B \times \cos B \quad (8)$$

$$Y = [1 + (D + KE^2) \times E^2] \times EN \cos B$$

式 (8) 中: $u = 6367588.4969; v = 6399698.902;$

$E = (L - A)\pi \div 180; A$ 为中央子午线经度;

$$S = \cos B^2;$$

$$N = V - [21562.267 - (108.973 - 0.0040S) \times S] \times S$$

$$D = (0.333333 + 0.00123S) \times S - 0.1666667$$

$$F = (0.25 + 0.00252S) \times S - 0.04166$$

$$K = 0.0083 - [0.1667 - (0.1968 + 0.0040S) \times S] \times S$$

$$M = (0.166S - 0.084) \times S$$

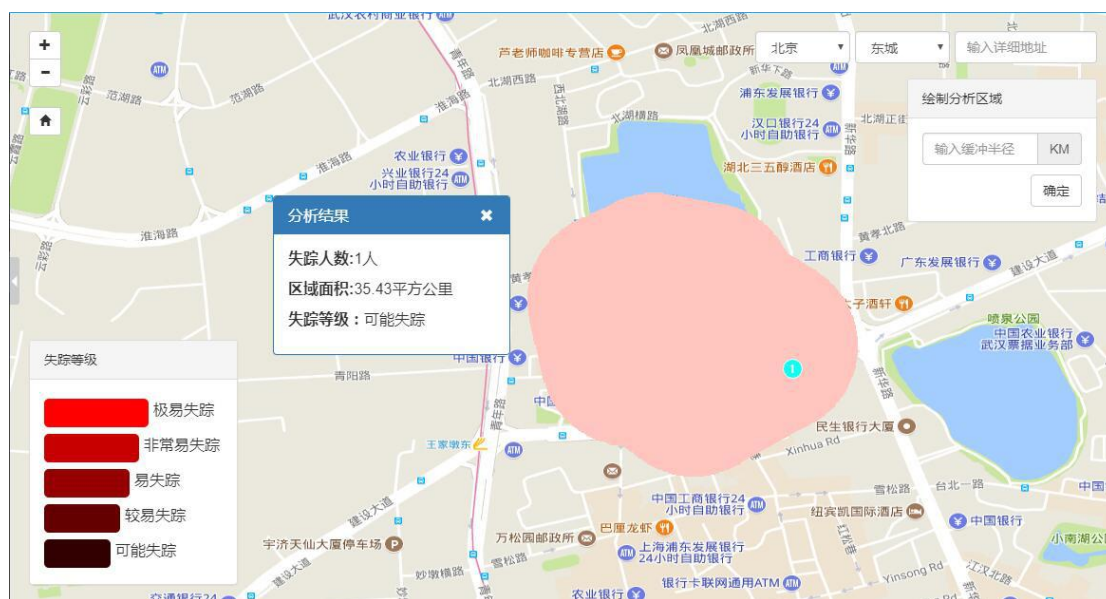
$$\text{面积计算公式: } P = \frac{1}{2} \sum (X_i Y_{i+1} - Y_i X_{i+1}) \quad (9)$$

3.2.2 模型构建与可视化

基于 MapGIS IGServer 平台服务，首先通过用户所绘制图元生成缓冲区、并获取缓冲区几何属性值，通过上式（8）、（9）计算出缓冲区面积，其次通过缓冲区几何属性进行图层叠加查询，获得缓冲区中走失点个数，最终计算该缓冲区的走失点密度。按走失点密度将区域失踪风险分为 5 类，分类阈值如下：

| 走失点密度（人/km ² ） | 风险等级 |
|---------------------------|-------|
| 1 | 可能失踪 |
| 2 | 较易失踪 |
| 3 | 易失踪 |
| 4 | 非常易失踪 |
| 5 | 极易失踪 |

通过图层叠加显示与弹出框形式对分析结果进行可视化分级显示。可视化效果如下图所示：



3.3 走失点聚类模型

在走失点聚类模块，项目基于 3,4000 余条历史失踪人员信息，采用对空间数据聚类十分有效的均值漂移(Mean Shift)聚类算法，对失踪人员经纬度坐标进行聚类，最终得到 197 个类簇，以及每个类簇的失踪人员个数，最后对不同类簇根据失踪人员个数给定相应权值，依据权值绘制失踪影响力范围图。

3.3.1 算法原理

目前已有的研究工作表明，均值漂移(Mean Shift)聚类算法对空间数据的聚类非常有效。均值漂移是一种非参数聚类方法，该聚类算法不需要指定聚类的数量，而且对聚类的形状也没有约束，只需要指定“带宽”，“带宽”代表了观测的尺度。失踪人员数据点含有的经纬度坐标非常适合利用均值漂移聚类算法进行聚类。以下是算法的详细介绍：

给定 n 个 d 维数据点 x_i , $i=1, \dots, n, x_i \in R^d$ ，利用核函数密度估计的方法，得到这 n 个点的概率密度为：

$$f(x) = \frac{1}{nh^d} \sum_{i=1}^n K\left(\frac{x-x_i}{h}\right) \quad (10)$$

其中 $K(x)$ 为核函数， h 为带宽。如果使用高斯核函数等常见的径向对称的核函数，则 $K(x)$ 可以表示成 $K(x) = c \exp(-\|x\|^2/h^2)$ ，其中 c 是一个归一化常数。为了寻找



数据密集的极值点，即求解梯度为 0 的点，令 $\nabla f(x) = 0$ 。

$$\begin{aligned} \nabla f(x) &= \frac{2c}{nh^{d+2}} \sum_{i=1}^n (x_i - x) g\left(\left\|\frac{x - x_i}{h}\right\|^2\right) \\ &= \frac{2c}{nh^{d+2}} \left[\sum_{i=1}^n g\left(\left\|\frac{x - x_i}{h}\right\|^2\right) \right] \left[\frac{\sum_{i=1}^n x_i g\left(\left\|\frac{x - x_i}{h}\right\|^2\right)}{\sum_{i=1}^n g\left(\left\|\frac{x - x_i}{h}\right\|^2\right)} - x \right] \end{aligned} \quad (11)$$

其中 $g(s) = -K'(s)$ 。上式前半部分可以看作基函数为 $g(\|x\|^2)$ 的核函数密度估计在 x 处的取值，后半部分是一个矢量，被称为 mean shift，记为：

$$m_h(x) = \frac{\sum_{i=1}^n x_i g\left(\left\|\frac{x - x_i}{h}\right\|^2\right)}{\sum_{i=1}^n g\left(\left\|\frac{x - x_i}{h}\right\|^2\right)} - x \quad (12)$$

Mean shift 矢量指向概率密度函数增加的方向。因此，均值漂移聚类算法可以使用如下的迭代步骤来得到 $\nabla f(x) = 0$ 的位置。

第 t 步：在位置 x' 处计算 mean shift 矢量 $m_h(x')$

第 $t+1$ 步：跟新 x 的位置： $x^{t+1} = x^t + m_h(x^t)$

重复以上步骤，直至收敛。

3.3.2 模型构建与可视化

为了防止各类簇之间失踪人员数量的差距过大，在构造失踪影响力圆时对每个类簇的失踪人员数量取 \log 。最后得到的失踪影响力圆的半径的计算公式如下：

$$R = \log(num) \cdot c \quad (13)$$

式中， R 表示失踪影响力范围圆的半径， num 表示每个类簇的失踪人员数量， c 表示一个常数，用于调整半径的大小以便于展示。

最后，我们利用 `ol.js` 开发库在地图上添加影响力范围圆用于展示，可以直挂的看到全国各地失踪人员的分布情况与数量对比。可视化效果如下图所示：



4. 数据库设计

4.1 逻辑设计

数据库中的逻辑设计如 E-R 图（图 9）所示：

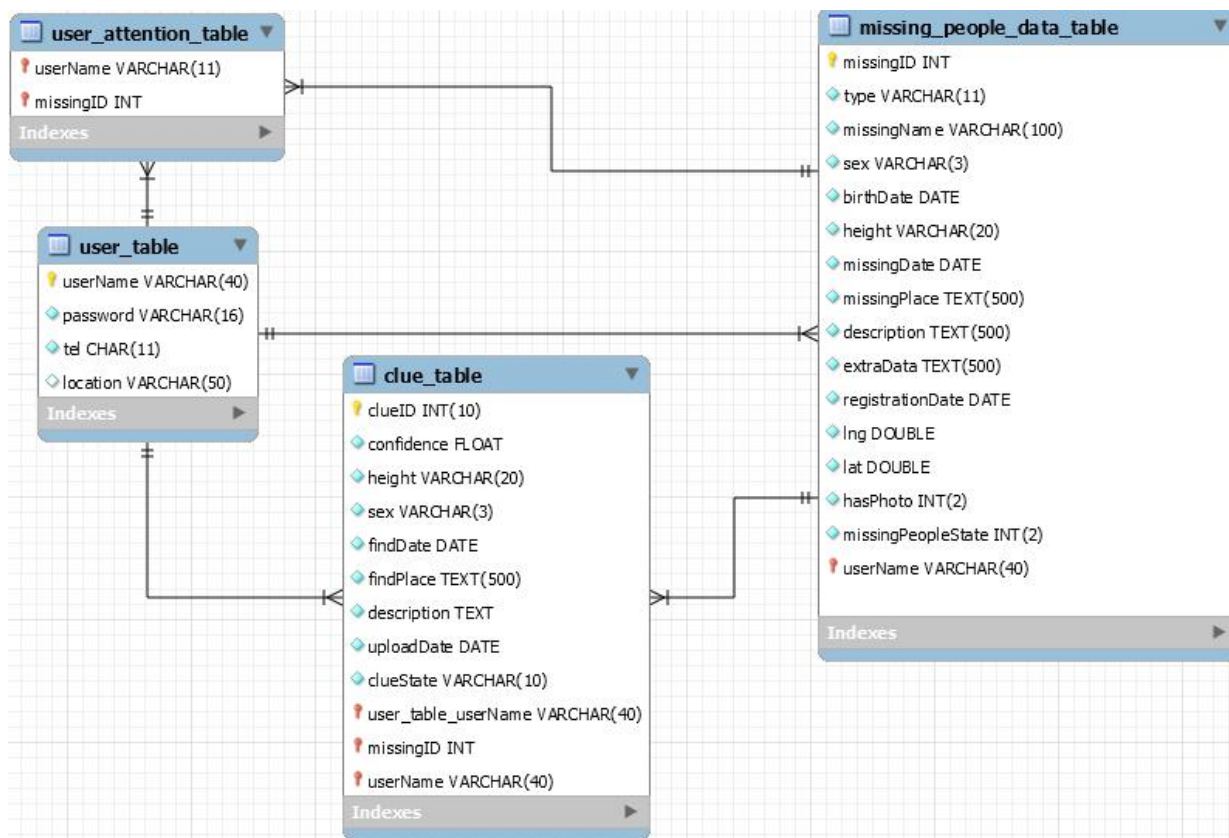


图 9 数据库 E-R 图

4.2 物理设计

4.2.1 表汇总

| 表名 | 功能说明 |
|---------------------------|-------------------|
| user_table | 用于存储用户信息 |
| missing_people_data_table | 用于存储失踪人员信息 |
| user_attention_table | 用于存储用户与其关心的失踪人员关系 |
| clue_table | 用于存储线索信息 |

4.2.2 表[1]:[user_table]表

| 表名 | user_table | | | | | | |
|--------|------------|-------------|-------------|-----------|-----|-----|---------|
| 数据库用户 | root | | | | | | |
| 主键 | userName | | | | | | |
| 其他排序字段 | 无 | | | | | | |
| 索引字段 | userName | | | | | | |
| 序号 | 字段名称 | 数据类型（精度范围） | 允许为空 Y/N | 唯一 Y/N | 区别度 | 默认值 | 约束条件/说明 |
| 1 | userName | VARCHAR(11) | N | Y | 高 | | 主键 |
| 2 | password | VARCHAR(16) | N | N | 中 | | |
| 3 | tel | CHAR(11) | N | Y | 高 | | |
| 4 | location | VARCHAR(50) | N | N | 低 | | |



| | |
|----------|--|
| Mysql 脚本 | <pre>CREATE TABLE IF NOT EXISTS `gohome`.`user_table` (`userName` VARCHAR(40) NOT NULL, `password` VARCHAR(16) NOT NULL, `tel` CHAR(11) NOT NULL, `location` VARCHAR(50) NULL, PRIMARY KEY (`userName`)) ENGINE = InnoDB;</pre> |
|----------|--|

4.2.3 表[2]:[missing_people_data_table]表

| 表名 | | missing_people_data_table | | | | | |
|--------|------------------|---------------------------|-------------|-----------|-------------|---------|-----------------|
| 数据库用户 | | root | | | | | |
| 主键 | | missingID | | | | | |
| 其他排序字段 | | 无 | | | | | |
| 索引字段 | | missingID、userName | | | | | |
| 序号 | 字段名称 | 数据类型（精度范围） | 允许为空 Y/N | 唯一 Y/N | 区 别 度 | 默认 值 | 约束 条件/ 说明 |
| 1 | missingID | INT(10) | N | Y | 高 | | 主键 |
| 2 | type | VARCHAR(11) | N | N | 低 | | |
| 3 | missingName | VARCHAR(100) | N | N | 低 | | |
| 4 | sex | VARCHAR(3) | N | N | 低 | | |
| 5 | birthDate | DATE | N | N | 低 | | |
| 6 | height | VARCHAR(20) | N | N | 低 | | |
| 7 | missingDate | DATE | N | N | 低 | | |
| 8 | missingPlace | TEXT(500) | N | N | 低 | | |
| 9 | description | TEXT(500) | N | N | 中 | | |
| 10 | extraData | TEXT(500) | N | N | 中 | | |
| 11 | registrationDate | DATE | N | N | 低 | | |



| | | | | | | | |
|----------|--------------------|---|---|---|---|-------|----|
| 12 | lng | DOUBLE | N | N | 低 | | |
| 13 | lat | DOUBLE | N | N | 低 | | |
| 14 | hasPhoto | INT(2) | N | N | 低 | | |
| 15 | missingPeopleState | INT(2) | N | N | 低 | 0 | |
| 16 | userName | VARCHAR(40) | N | N | 低 | admin | 外键 |
| MySQL 脚本 | | <pre> CREATE TABLE IF NOT EXISTS `gohome`.`missing_people_data_table` (`missingID` INT UNSIGNED NOT NULL AUTO_INCREMENT, `type` VARCHAR(11) NOT NULL, `missingName` VARCHAR(100) NOT NULL, `sex` VARCHAR(3) NOT NULL, `birthDate` DATE NOT NULL, `height` VARCHAR(20) NOT NULL, `missingDate` DATE NOT NULL, `missingPlace` TEXT(500) NOT NULL, `description` TEXT(500) NOT NULL, `extraData` TEXT(500) NOT NULL, `registrationDate` DATE NOT NULL, `lng` DOUBLE NOT NULL, `lat` DOUBLE NOT NULL, `hasPhoto` INT(2) NOT NULL, `missingPeopleState` INT(2) NOT NULL, `userName` VARCHAR(40) NOT NULL, PRIMARY KEY (`missingID`, `userName`), UNIQUE INDEX `missing_ID_UNIQUE` (`missingID` ASC), INDEX </pre> | | | | | |



| | |
|--|---|
| | <pre> `fk_missing_people_data_table_user_table1_idx` (`userName` ASC), CONSTRAINT `fk_missing_people_data_table_user_table1` FOREIGN KEY (`userName`) REFERENCES `gohome`.`user_table` (`userName`) ON DELETE NO ACTION ON UPDATE NO ACTION) ENGINE = InnoDB;</pre> |
|--|---|

4.2.4 表[3]:[user_attention_table]表

| 表名 | user_attention_table | | | | | | |
|--------|----------------------|-------------|-------------|-----------|-----|-------|-------------|
| 数据库用户 | root | | | | | | |
| 主键 | userName、missingID | | | | | | |
| 其他排序字段 | 无 | | | | | | |
| 索引字段 | missingID、userName | | | | | | |
| 序号 | 字段名称 | 数据类型（精度范围） | 允许为空 Y/N | 唯一 Y/N | 区别度 | 默认值 | 约束条件/ 说明 |
| 1 | missingID | INT(10) | N | N | 中 | | 外键 |
| 2 | userName | VARCHAR(40) | N | N | 中 | admin | 外键 |



| | |
|----------|--|
| Mysql 脚本 | <pre>CREATE TABLE IF NOT EXISTS `gohome`.`user_attention_table` (`userName` VARCHAR(11) NOT NULL, `missingID` INT UNSIGNED NOT NULL, PRIMARY KEY (`userName`, `missingID`), INDEX `fk_uesr_table_has_missing_people_data_table_missing_people_ _idx` (`missingID` ASC), INDEX `fk_uesr_table_has_missing_people_data_table_uesr_table_idx` (`userName` ASC), CONSTRAINT `fk_uesr_table_has_missing_people_data_table_uesr_table` FOREIGN KEY (`userName`) REFERENCES `gohome`.`user_table` (`userName`) ON DELETE NO ACTION ON UPDATE NO ACTION, CONSTRAINT `fk_uesr_table_has_missing_people_data_table_missing_people_ da1` FOREIGN KEY (`missingID`) REFERENCES `gohome`.`missing_people_data_table`</pre> |
| | <pre>(`missingID`) ON DELETE NO ACTION ON UPDATE NO ACTION) ENGINE = InnoDB;</pre> |

4.2.5 表[4]:[clue_table]表

| 表名 | | clue_table | | | | | |
|--------|-------------|--------------------|-------------|-----------|-----|-----|---------|
| 数据库用户 | | root | | | | | |
| 主键 | | clueID | | | | | |
| 其他排序字段 | | 无 | | | | | |
| 索引字段 | | missingID、userName | | | | | |
| 序号 | 字段名称 | 数据类型（精度范围） | 允许为空 Y/N | 唯一 Y/N | 区别度 | 默认值 | 约束条件/说明 |
| 1 | clueID | INT(10) | N | Y | 高 | | 主键 |
| 2 | confidence | FLOAT | N | N | 低 | | |
| 3 | height | VARCHAR(20) | N | N | 低 | | |
| 4 | sex | VARCHAR(3) | N | N | 低 | | |
| 5 | findDate | DATE | N | N | 低 | | |
| 6 | findPlace | TEXT(500) | N | N | 低 | | |
| 7 | description | TEXT | N | N | 中 | | |
| 8 | uploadDate | DATE | N | N | 低 | | |
| 9 | clueState | VARCHAR(10) | N | N | 低 | | |
| 10 | missingID | INT(10) | N | N | 中 | | 外键 |
| 11 | userName | VARCHAR(40) | N | N | 中 | | 外键 |



| | |
|----------|---|
| Mysql 脚本 | <pre>CREATE TABLE IF NOT EXISTS `gohome`.`clue_table` (`clueID` INT(10) UNSIGNED NOT NULL, `confidence` FLOAT UNSIGNED NOT NULL, `height` VARCHAR(20) NOT NULL, `sex` VARCHAR(3) NOT NULL, `findDate` DATE NOT NULL, `findPlace` TEXT(500) NOT NULL, `description` TEXT NOT NULL, `uploadDate` DATE NOT NULL, `clueState` VARCHAR(10) NOT NULL, `missingID` INT UNSIGNED NOT NULL, `userName` VARCHAR(40) NOT NULL, PRIMARY KEY (`clueID`, `missingID`, `userName`), INDEX `fk_clue_table_missing_people_data_table1_idx` (`missingID` ASC, `userName` ASC), CONSTRAINT `fk_clue_table_missing_people_data_table1` FOREIGN KEY (`missingID`, `userName`) REFERENCES `gohome`.`missing_people_data_table` (`missingID`, `userName`) ON DELETE NO ACTION ON UPDATE NO ACTION) ENGINE = InnoDB;</pre> |
|----------|---|



5. 系统亮点与技术特色

5.1 系统亮点

5.1.1 人脸搜索的运用

在寻人过程中，失踪人员的照片往往能起到决定性作用，可以帮助用户精准锁定疑似失踪人员，从而提供重要线索。项目提供的人脸搜索功能使得用户能够上传疑似失踪人员的照片进行搜索，系统会返回已登记的失踪人员中相似度最大的 4 个人脸，并提供相似度。当所上传的疑似失踪人员照片与已登记的失踪人员可能为同一人时(相似度超过 80%)，用户可提供疑似失踪人员的相关线索。

系统利用最前沿的深度学习技术，解决了以往棘手的疑似失踪人员身份确认的问题。

5.1.2 建模思维的运用

数学模型与算法具有很强的科学性和正确性，利用合理的数学模型结合相应的算法，能够揭示数据背后的规律与奥秘。本系统结合核密度分析，缓冲区分析以及均值漂移聚类算法，建立了路段失踪风险评估模型、区域失踪风险评估模型和走失点聚类模型。我们利用 MapGIS IGServer 提供的强大的服务发布功能和可视化功能，将建立的模型用于实际路段和区域的失踪风险分析，使得决策者可以直观的了解某一区域或路段的失踪情况，并查看已失踪人员的详细信息。利用走失点聚类模型生成的失踪影响力范围图可以直观的了解全国各地的失踪人口情况。

5.1.2 数据分析功能的添加

传统的寻亲网站往往只有简单的信息登记与展示，却忽视了这些大量失踪人员信息背后蕴含的规律。我们通过对这些大量含地理信息数据的挖掘，利用 MapGIS IGServer 与 echart 图表进行分析与展示，可以宏观的查看失踪人员信息统计分析情况，为决策者提供一定启发。



5.2 技术特色

5.2.1 MapGIS Web 端插件式开发

为实现系统不同功能之间的解耦，本项目采用 MapGis 提供的纵生式开发模式，使用纵生式开发模式开发软件应用，可以同步开发，也可以异步开发、异步上线，所开发出来的软件应用具备纵生、飘移、聚合、重构 4 大云特性，能极大提高代码复用率，延长软件生命周期，降低生产成本，提高生产效率。

本系统共使用了 5 个功能插件，分别是菜单插件、Popup 展示插件、热力图插件、易失路段分析插件以及区域失踪风险分析插件。其中，菜单插件主要用于主页菜单的显示；Popup 展示插件主要用于聚合失踪人员聚合标注以及 Popup 展示框的实现；热力图插件主要用于失踪人员热力图的展示；易失路段分析主要用于实现武汉市相应路段的失踪风险分析功能；区域失踪风险分析插件主要用于分析在地图上交互式绘制形成的区域指定半径内的失踪人员以及该区域的失踪风险。

5.2.2 实时通信系统

寻亲系统需要解决的一大技术问题就是最新线索的广播，即将志愿者提供的最新线索及时推送给信息的登记者和关注者，在实时通信方面 WebSocket 是最理想的传输方式，除了能高效使用服务器内存，低延迟，还能实现客户端和服务端的全双工通信。我们利用 Tomcat 提供的 WebSocket API 实现失踪人员线索的及时广播和用户之间的点对点交流。

5.2.3 爬虫技术

本系统数据来源于“宝贝回家网”，利用正则表达式对网站数据进行提取。数据可以每日跟新，但不会对网站造成任何的安全隐患。



6. 界面设计

6.1 设计原则

在界面设计方面我们遵循易用性原则、规范性原则、帮助设施原则以及美观与协调性原则。

(1)易用性原则：本系统所采用的按钮图标直观清晰，区分度高，菜单名称用词准确表达清晰。多按钮使用展开方式，防止不同功能按钮之间相互影响。在大多数情况下，用户能够不参照用户手册就能了解界面功能并进行正确的相关操作。

(2)规范性原则：本系统界面包含菜单栏、导航栏、操作区等部分，具有包含关系的菜单进行分级展示，导航栏可实时查看登录状态，操作区界面充足操作流畅。

(3)帮助设施原则：本系统提供了详尽而可靠的帮助文档，在用户使用产生迷惑时可以很容易从帮助文档中寻求解决方法。帮助文档中的性能介绍与说明要与系统的实际操作完全一致。

(4)美观与协调性原则：界面分布合理，色彩搭配赏心悦目。系统界面采用 CSS3 和 Html5 编写，简洁大方。首先，菜单栏可进行隐藏，扩大操作界面；操作界面整体以地图为主体，相应的交互式输入输出均以弹出框的形式展现，交互功能多样而又不影响用户对地图的查看与操作。其次，人脸搜索、全网搜索、区域失踪风险分析等耗时操作均展示加载动画，不至于让用户误以为系统停止响应。

6.2 界面布局

界面主体部分由地图占据，可进行地图相关操作，界面左侧为多级功能菜单



栏,可选择不同功能模块进行操作,切换按钮可以从失踪人员照片展示视图切换为操作视图进行一系列地图相关操作。界面整体布局如图 10 所示:

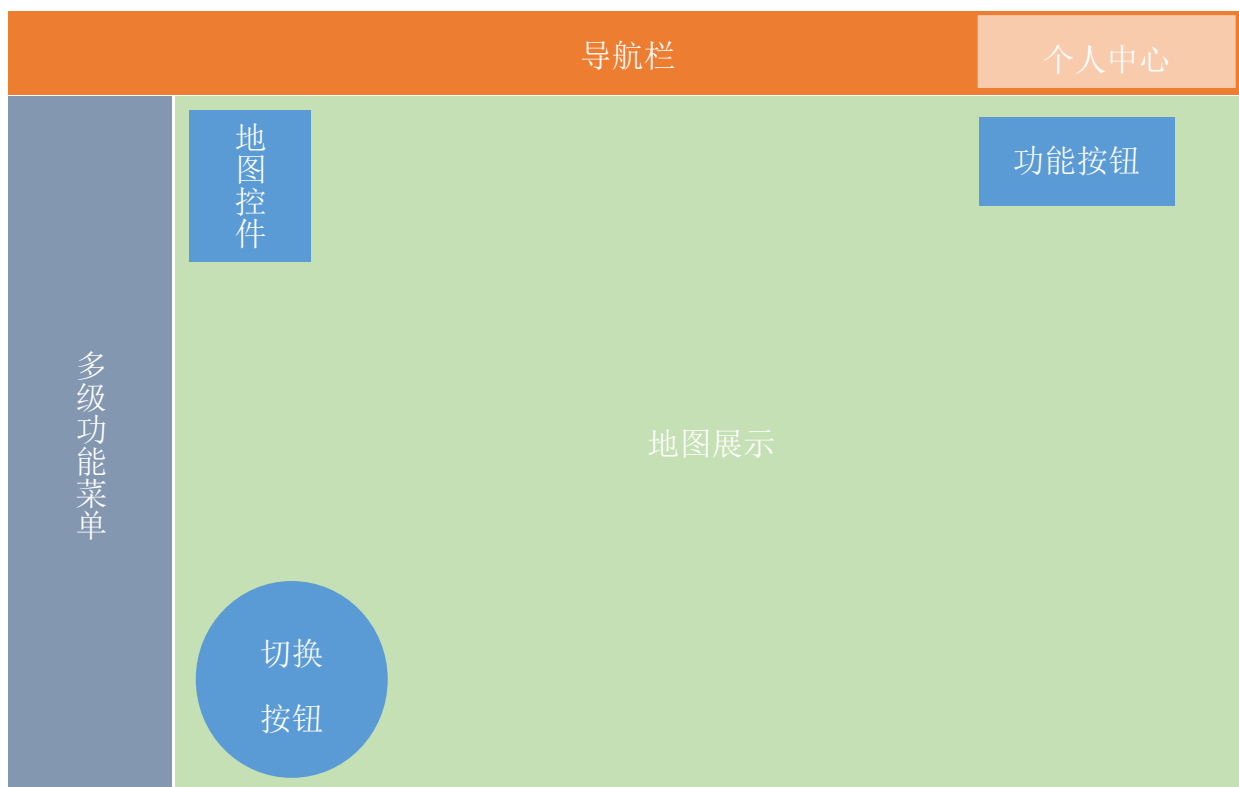
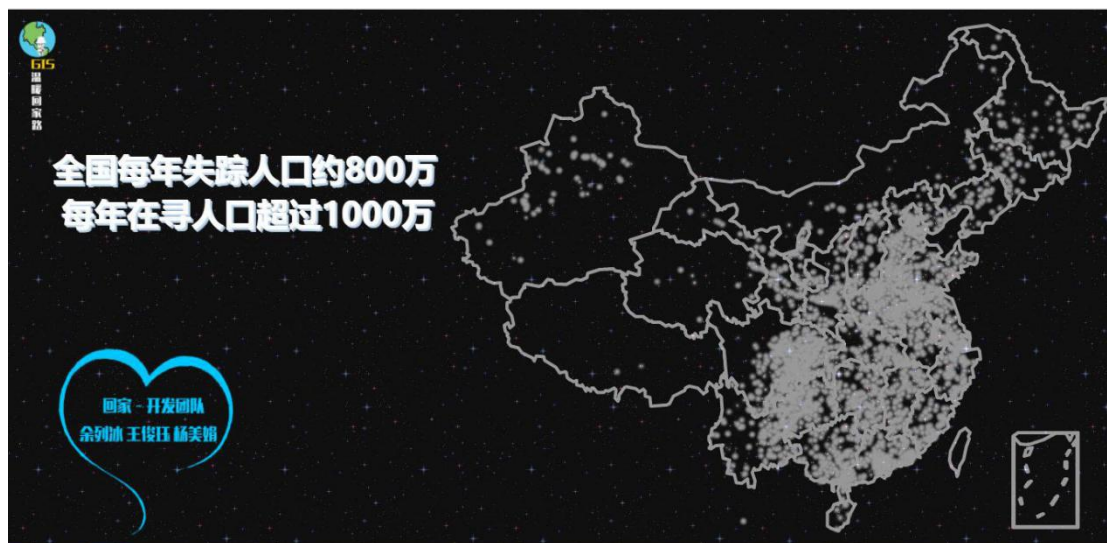


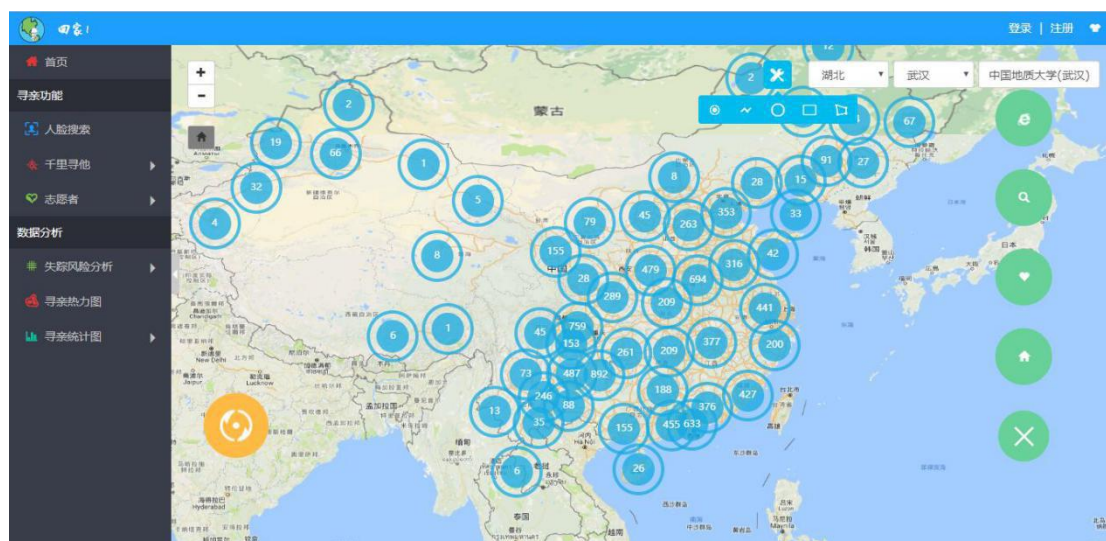
图 10 界面布局设计

6.3 界面展示

欢迎页面



主页面



7.项目总结

“回家”——寻亲系统是一个颇具社会意义的系统。技术上，它巧妙地结合了 MapGIS 所提供的强大的 GIS 功能与深度学习所带来的人脸图像识别和大规模图像检索功能，极大地提高了寻人效率与精度；社会意义上，本系统能够联合大众的力量，发挥互联网的优势，帮助寻找失踪人员，于个人、于社会，都是非常有意义的。

在项目开发过程中我们将重心放在失踪人员数据的分析与可视化，利用 MapGIS 平台提供的存储、查询以及强大的空间数据分析功能实现所需功能。同时，我们在仔细对比当前各大寻亲网站之后，总结不足，加入最前沿的人脸识别与大规模图像检索技术，使得寻亲过程更加高效准确。

整个项目以 MapGIS 为核心，展开各类空间数据的操作与分析功能。以人脸搜索为桥梁，增加人脸关键信息的检索功能，弥补传统寻亲网站的不足。