

学 号：_____

密 级：_____

合肥工业大学

Hefei University of Technology

本科毕业设计（论文）

UNDERGRADUATE THESIS



类 型：_____ 毕业设计

题 目：膀胱镜 AI 辅助诊断系统的设计与实现

专业名称：_____

入校年份：_____

学生姓名：_____

指导教师：_____

学院名称：_____

完成时间：_____ 2023 年 5 月

合 肥 工 业 大 学

本科毕业设计（论文）

膀胱镜 AI 辅助诊断系统的设计与实现

学生姓名：_____

学生学号：_____

指导教师：_____

专业名称：_____

学院名称：_____

2023 年 5 月

A Dissertation Submitted for the Degree of Bachelor

**Design and implementation of cystoscopy
AI-assisted diagnosis system**

By

Hefei University of Technology

Hefei, Anhui, P.R.China

May, 2023

毕业设计（论文）独创性声明

本人郑重声明：所呈交的毕业设计（论文）是本人在指导教师指导下进行独立研究工作所取得的成果。据我所知，除了文中特别加以标注和致谢的内容外，设计（论文）中不包含其他人已经发表或撰写过的研究成果，也不包含为获得合肥工业大学或其他教育机构的学位或证书而使用过的材料。对本文成果做出贡献的个人和集体，本人已在设计（论文）中作了明确的说明，并表示谢意。

毕业设计（论文）中表达的观点纯属作者本人观点，与合肥工业大学无关。

毕业设计（论文）作者签名：

签名日期：2023 年 5 月 17 日

毕业设计（论文）版权使用授权书

本学位论文作者完全了解合肥工业大学有关保留、使用毕业设计（论文）的规定，即：除保密期内的涉密设计（论文）外，学校有权保存并向国家有关部门或机构送交设计（论文）的复印件和电子光盘，允许设计（论文）被查阅或借阅。本人授权合肥工业大学可以将本毕业设计（论文）的全部或部分内容编入有关数据库，允许采用影印、缩印或扫描等复制手段保存、汇编毕业设计（论文）。

（保密的毕业设计（论文）在解密后适用本授权书）

学位论文作者签名：

指导教师签名：

签名日期：2023 年 5 月 17 日

签名日期：2023 年 5 月 26 日

摘要

膀胱癌是国内外泌尿生殖系最常见的恶性肿瘤之一，膀胱癌的早期诊断对患者的治疗和康复起着重要的作用。膀胱镜是膀胱癌早筛最常用的方式，但膀胱镜的检测结果很大程度上依赖于镜检医生的临床经验和主观意见，而且膀胱癌的发病率高、患者较多，诊断医师的阅片压力大，漏诊、误诊时有发生，诊断的准确率和早诊率较低。目前亟需引入 AI 技术辅助医生进行膀胱镜诊断，从而降低医生的阅片压力、提高膀胱镜诊断的准确率并推进膀胱镜领域的产业升级。近年来，随着深度学习的深入发展以及目标检测算法识别率的提高，这使得设计膀胱镜 AI 辅助诊断系统成为可能。

本文针对深度学习、目标检测、医学图像处理领域进行研究，并基于深度学习的目标检测网络搭建了膀胱镜 AI 辅助诊断系统。本文的主要工作有：

首先是目标检测模型部分，本论文对深度学习模型进行了调研，同时结合需求将 YOLO V5 作为系统的目标检测模型。本文联合合作医院进行膀胱镜数据集的标注，最终基于标注的膀胱镜数据集训练出目标检测模型。

其次，本文通过场景分析、病例报告分析、网络调研进行需求分析，并基于需求分析的结果进行膀胱镜 AI 辅助诊断系统的详细设计、开发、测试、部署。

综上所述，本文融合了深度学习和软件工程技术，开发并实现了膀胱镜 AI 辅助诊断系统，从而辅助医生诊断、减轻医生的阅片压力。

关键词：膀胱癌；深度学习；目标检测；计算机辅助诊断；B/S 架构

ABSTRACT

Bladder cancer is one of the most common malignant tumors of the genitourinary system at home and abroad. Early diagnosis of bladder cancer plays an important role in the treatment and rehabilitation of patients. Cystoscopy is the most commonly used method for early screening of bladder cancer, but the detection results of cystoscopy largely depend on the clinical experience and subjective opinions of microscopic doctors, and the incidence of bladder cancer is high and there are many patients. Tablet pressure is high, missed diagnosis and misdiagnosis occur from time to time, and the diagnostic accuracy and early diagnosis rate are low. At present, it is urgent to introduce AI technology to assist doctors in cystoscopy diagnosis, so as to reduce the pressure on doctors to read films, improve the accuracy of cystoscopy diagnosis, and promote the industrial upgrading of cystoscopy. In recent years, with the in-depth development of deep learning and the improvement of the recognition rate of target detection algorithms, it has become possible to design an AI-assisted diagnostic system for cystoscopy.

This dissertation paper conducts research in the fields of deep learning, target detection, and medical image processing, and builds a cystoscope AI-assisted diagnosis system based on the deep learning target detection network. The main work of this paper is as follows:

The first is the part of the target detection model. In this paper, the deep learning model is investigated, and YOLO V5 is used as the target detection model of the system in combination with the requirements. This paper cooperates with cooperative hospitals to annotate the cystoscopy dataset, and finally trains a target detection model based on the annotated cystoscopy dataset.

Secondly, this dissertation conducts demand analysis through scenario analysis, case report analysis, and network research, and based on the results of the demand analysis, the detailed design, development, testing, and deployment of the cystoscopy AI-assisted diagnosis system are carried out.

To sum up, this dissertation integrates deep learning and software engineering technology, develops and implements an AI-assisted diagnostic system for cystoscopy, so as to assist doctors in diagnosis and reduce the pressure on doctors in film reading.

KEYWORDS: Bladder Cancer; Deep Learning; Object Detection; Computer-Aided Diagnosis; B/S Architecture.

目录

1 绪论.....	1
1.1 研究背景.....	1
1.2 国内外研究现状.....	2
1.3 本文的主要工作.....	3
1.4 本文的组织结构.....	3
2 系统相关技术介绍和开发环境.....	5
2.1 系统相关技术介绍.....	5
2.2 技术选型.....	5
2.2.1 Spring Boot.....	5
2.2.2 Vue.js.....	6
2.2.3 Pytorch + Flask.....	6
2.3 硬件平台.....	7
2.4 开发工具.....	7
2.5 本章小结.....	7
3 需求分析及概要设计.....	8
3.1 需求概览.....	8
3.2 登录模块.....	11
3.2.1 功能描述.....	11
3.2.2 业务流程图.....	11
3.3 病例模块.....	12
3.3.1 功能描述.....	12
3.3.2 业务流程图.....	12
3.4 部门模块.....	13
3.4.1 功能描述.....	13
3.4.2 业务流程图.....	13
3.5 数据可视化模块.....	14
3.5.1 功能描述.....	14
3.5.2 业务流程图.....	14
3.6 本章小结.....	15

4 系统设计与实现.....	16
4.1 系统框架设计.....	16
4.2 数据库设计.....	16
4.2.1 数据库概念结构设计.....	16
4.2.2 数据库逻辑结构设计.....	17
4.2.3 数据字典设计.....	18
4.3 前端页面设计.....	20
4.4 后端设计.....	21
4.4.1 登录模块.....	21
4.4.2 数据可视化模块.....	23
4.4.3 病例模块.....	25
4.4.4 部门管理模块.....	29
4.5 算法模块设计.....	31
4.6 成果展示.....	33
4.7 关键问题分析及解决方案.....	45
4.7.1 系统安全性问题.....	45
4.7.2 数据集问题.....	46
4.8 本章小结.....	46
5 总结与展望.....	47
5.1 论文工作总结.....	47
5.2 未来研究展望.....	47
参考文献.....	48
致谢.....	50

插图清单

图 3.1 膀胱镜检查报告单.....	8
图 3.2 登录模块业务流程图.....	11
图 3.3 病例模块流程图.....	12
图 3.4 部门模块流程图.....	13
图 3.5 数据可视化模块流程图.....	14
图 4.1 系统框架图.....	16
图 4.2 数据库的概念数据模型.....	17
图 4.3 数据库的概念数据模型.....	17
图 4.4 前端页面设计概览.....	21
图 4.5 Faster r-cnn 模型图 ^[29]	31
图 4.6 YOLO V1 模型图 ^[30]	31
图 4.7 合作医院标注的膀胱镜数据集.....	32
图 4.8 YOLO V5 模型的训练过程.....	32
图 4.9 登录功能展示.....	33
图 4.10 医生端首页功能展示.....	34
图 4.11 医生端病例诊断新增功能展示.....	34
图 4.12 医生端病理诊断上传视频功能展示.....	35
图 4.13 医生端病理诊断上传视频进度条功能展示.....	35
图 4.14 医生端新增视频后的功能展示.....	36
图 4.15 医生端人工诊断功能展示.....	36
图 4.16 医生端截取疑似病变区域功能展示.....	37
图 4.17 医生端疑似病变区域列表功能展示.....	37
图 4.18 医生端疑似病变区域个数列表功能展示.....	38
图 4.19 医生端膀胱镜检查报告单功能展示.....	38
图 4.20 医生端填写膀胱镜检查报告单功能展示.....	39
图 4.21 医生端病例管理功能展示.....	39
图 4.22 医生端病例详情页面 1.....	40
图 4.23 医生端病例详情页面 2.....	40

图 4.24 医生端导出膀胱镜检查报告单功能展示.....	41
图 4.25 医生端个人信息修改功能展示.....	41
图 4.26 管理员端首页功能展示.....	42
图 4.27 管理员端可视化信息统计功能展示.....	42
图 4.28 管理员端病例管理功能展示.....	43
图 4.29 管理员端人员管理功能展示 1.....	43
图 4.30 管理员端人员管理功能展示 2.....	44
图 4.31 管理员端医院管理功能展示.....	44
图 4.32 管理员端个人信息管理功能展示.....	45
图 4.33 Sa-Token 的 start 趋势	45

表格清单

表 3.1 需求分析表.....	10
表 3.2 登录模块功能描述表.....	11
表 3.3 病例模块功能描述表.....	12
表 3.4 部门模块功能描述表.....	13
表 3.5 数据可视化模块功能描述表.....	14
表 4.1 用户信息表数据字典.....	18
表 4.2 医院信息表数据字典.....	18
表 4.3 地域信息表数据字典.....	18
表 4.4 病例信息表数据字典.....	19
表 4.5 视频信息表数据字典.....	19
表 4.6 图片信息表数据字典.....	20
表 4.7 标注信息表数据字典.....	20
表 4.8 接口 1 描述.....	21
表 4.9 接口 2 描述.....	22
表 4.10 接口 16 描述.....	22
表 4.11 接口 17 描述.....	23
表 4.12 接口 3 描述.....	23
表 4.13 接口 4 描述.....	24
表 4.14 接口 24 描述.....	24
表 4.15 接口 5 描述.....	25
表 4.16 接口 11 描述.....	25
表 4.17 接口 9 描述.....	26
表 4.18 接口 12 描述.....	26
表 4.19 接口 7 描述.....	27
表 4.20 接口 15 描述.....	27
表 4.21 接口 21 描述.....	28
表 4.22 接口 32 描述.....	28
表 4.23 接口 10 描述.....	29

表 4.24 接口 25 描述.....	29
表 4.25 接口 22 描述.....	30
表 4.26 接口 30 描述.....	30

1 绪论

1.1 研究背景

2019 年国家肿瘤临床医学研究中心发布的中国恶性肿瘤流行情况显示^[1], 男性每年新发膀胱癌病例大约为 6.2 万、发病率约为 8.83/10 万人、中标率约为 5.79/10 万人, 并且膀胱癌的发病率和死亡率呈现逐年上升的态势。膀胱癌是中国男性泌尿生殖系恶性肿瘤发病率第 1 位的癌症^[2], 也是中国恶性肿瘤发病率第 8 位的癌症。

膀胱癌是国内外泌尿生殖系最常见的恶性肿瘤之一, 膀胱癌的早期诊断对患者的治疗和康复起着重要的作用。目前来看, 膀胱镜检查在膀胱癌的诊断上是最为可靠的方式^[3], 同时膀胱镜在膀胱癌的治疗、手术、复查随访方面也起着关键性的作用。但是膀胱镜的早诊率较低, 主要原因有三个: 一是传统的膀胱镜检查方式本身有着特异性、敏感性较低的问题^[3], 膀胱镜的检查方式亟待需要其他高新技术的融合来优化传统的镜检方式; 二是膀胱镜的诊断结果很大程度上依赖于镜检医生的临床经验和主观意见; 三是膀胱癌发病率高, 膀胱癌的患者较多, 镜检医生的判图、分析的工作量极大, 考虑到疲劳、精力、时间等限制因素, 医生错判、误判时有发生。

1943 年 McCulloch 等人^[4]提出了 AI 的雏形即“神经网络”的概念, 经过数十年的发展后, 人类首次提出了(Artificial Intelligence)的概念, 自此之后的几十年里, 人工智能(AI)在不断的发展, 从 1997 年 IBM“深蓝”战胜国际象棋冠军, 到 2010 年 Hinton 用深度学习模型 AlexNet^[5]以 57.1%的准确率碾压第二名的分类性能, 到 2016 年基于深度学习的 AlphaGo 以 4:1 的战绩打败了职业棋手, 再到如今引爆全球的聊天机器人 chatGPT, AI 已经取得惊人的成就。

随着互联网、5G 等高新技术的发展, 全国乃至全球的数据量呈现着井喷式的态势增长。人工智能依托于大数据, 目前在计算机视觉、数据挖掘、自然语言处理等方面已经展现了其惊人的应用潜力, 解决了很多传统方法无法完成的难题。医疗行业因受其原有技术方法的限制, 现亟需 AI 的赋能来完成医疗行业的信息化、科技化的产业升级。目前, AI+医疗的产业模式正在迅速兴起, 国内外的公司都在通过 AI 赋能医疗进行产业升级, 例如百度医疗大脑^[6]; AliveCor 公司基

于人工智能辅助心房颤动早期发现等。由此可见 AI 已经在医疗行业的方方面面得到了广泛应用。

然而, AI 在膀胱镜的早期诊断中却少有应用, 同时膀胱镜由于其本身检测方式的限制, 使得膀胱镜诊断的人力资源成本依旧昂贵, 诊断医师的阅片压力极大, 漏诊、误诊时有发生, 诊断的准确率和早诊率依旧很低。因此, 目前亟需引入 AI 技术来辅助医生来进行膀胱镜诊断来降低医生的阅片压力并提高膀胱镜的诊断的准确率, 从而推进膀胱镜领域的产业升级。

1.2 国内外研究现状

近年来, 随着深度学习的深入发展以及目标检测算法识别率的提高, 这使得设计膀胱镜 AI 辅助诊断系统成为可能。由于膀胱镜数据集的匮乏以及数据标注成本的昂贵, 目前国内外的公司还在对内窥镜(膀胱镜是内窥镜的一种)的 AI 辅助系统进行探索, 随着国内外的研究深入, 相关的内窥镜系统也相继问世并逐渐得到应用。

2010 年的 ImageNet 比赛中, Krizhevsky 等人^[5]首次将深度学习模型应用到图像处理中, 并以绝对优势的准确率领先第二名。自此引发了学术界、工业界对深度学习图像处理的热潮。

Nairveen Ali 等人^[8]的研究结果证明了基于人工智能的分类模型在基于 BL 膀胱镜图像的膀胱癌诊断中的潜力。Nairveen Ali^[8]等人同时也表明通过此项研究可以建立一个自动 blue light (BL)膀胱镜平台, 帮助医生识别和分类潜在的感兴趣的病变。将此类平台应用于临床常规旨在协助外科医生并帮助癌症诊断。

Hashemi 等人^[9]应用两个著名的卷积神经网络 (Convolutional Neural Networks) 和多层神经网络对膀胱膀胱镜检查图像进行分类并提出了一种动态改变收敛速度的算法以便改进多层神经网络, 降低误差, 提高了深度学习模型的收敛速度。

Du 等人^[10]构建了一个基于 Caffe 框架的 CNN 模型, 该模型可以在膀胱镜检查下区分膀胱癌和正常膀胱粘膜, 还通过 EasyDL 构建模型, 实现了更高的精度。同时还在手机上部署了 EasyDL 模型, 可以通过手机的力量轻松识别膀胱癌图像。

2019 年中国武汉楚精灵医疗科技有限公司开发出了内镜精灵^[11], 该产品用于对消化系统内镜的可视化影像进行即时检测, 并指导医生检测操作和协助医师治疗。

上述研究表明深度学习技术已逐渐成为解决内窥镜问题的重要工具，同时通过人工智能技术可以有效减轻医生的诊断压力、提高诊断的准确率并加快医院科室的智能化升级的进程，故膀胱镜 AI 辅助诊断系统是符合当今创新技术和 AI+ 医疗领域的前沿方向。

1.3 本文的主要工作

本文的主要工作是研究膀胱镜的诊断场景、膀胱镜图像处理领域的目标检测模型以及软件的设计与开发，并基于合作医院团队标注的数据集开发膀胱镜 AI 辅助诊断系统，从而辅助医生诊断，减轻医生的阅片压力，具体的工作如下：

- 调研膀胱镜与人工智能结合的研究现状，对医学图像处理尤其是膀胱镜图像处理进行全面的了解，并对当今公司的内窥镜(膀胱镜是内窥镜的一种)与 AI 结合的商业化产品进行了解与学习。
- 对合作医院进行调研，观看膀胱镜的检测视频、阅览脱敏后的膀胱镜检查报告单，最后根据上述调研的信息进行膀胱镜 AI 辅助诊断的需求分析。
- 联合合作医院泌尿外科医生标注膀胱镜图像的数据集。研究当前主流的目标检测算法即 Faster r-cnn 算法^[29]系列和 YOLO 算法^[30]系列，并在 YOLO V5 的算法模型上进行训练和测试，并将 YOLO V5 算法模型应用到膀胱镜 AI 辅助诊断系统中。
- 在合作医院的指导下，我完成了膀胱镜 AI 辅助诊断系统的设计与实现，此系统是基于 B/S 架构进行设计和实现的，前端通过 Vue 搭建，后端基于 Spring Boot、MyBatis、Sa-Token 框架主要搭建的，其次通过 Flask+Pytorch 搭建人工智能微服务接口，数据库采用 MySQL 数据库。我完成了系统的需求分析、概要设计、详细设计、前后端的搭建与联调、系统测试的软件开发全流程。

1.4 本文的组织结构

本文的组织结构如下：

第一章绪论。本章主要介绍膀胱镜图像处理的课题背景和研究的意义以及 AI+医疗的未来趋势，并深入调研了国内外膀胱镜图像处理结合研究现状。

第二章系统开发环境。本章主要说明系统开发的硬件平台、所用到的开发工具及其开发的框架。本项目通过主流、稳定的 Vue 框架进行前端的开发，用低耦

合、高内聚的 **Spring Boot** 作为后端的开发框架，用 **Pytorch** 框架进行目标检测模型的搭建、训练和预测并最终通过 **Flask** 将目标检测模型作为微服务整合到膀胱镜 **AI** 辅助诊断系统中。

第三章需求分析及概要设计。本章主要说明此系统的应用场景以及客户的需求，根据前期的准备和调研进行需求分析。基于分析好的需求，我对系统进行概要设计，将膀胱镜 **AI** 辅助诊断系统划分对应的模块，并熟悉各个模块的功能和业务流程。

第四章系统设计与实现。本章主要基于前期的需求分析和概要设计对系统进行详细设计，系统的详细设计包括系统功能模块设计、数据库设计、页面 **UI** 设计、算法设计、后端设计以及所遇到的关键问题和对应的解决方案。同时也会详细地展示系统的功能。

第五章总结与展望。本章主要对系统的开发、论文的撰写进行相应的总结，并针对未来的膀胱镜图像处理的发展进行了分析与畅想。

2 系统相关技术介绍和开发环境

2.1 系统相关技术介绍

本系统是基于 B/S 架构进行开发的，涉及到前端、后端、数据库技术，同时本系统也是基于人工智能目标检测算法进行开发的，涉及到深度学习、目标检测的技术。

前端的作用是搭建网页的页面，用户通过前端搭建的页面进行交互，前端调用后端的逻辑接口并将后端返回的数据渲染到系统的页面上，以此来完成用户与系统的交互。

数据库是用来存储系统数据的系统，数据库分为非关系型数据库，例如 MySQL^[13]、Oracle^[14]和关系型数据库 Neo4j^[15]图数据库、redis^[16]。本系统采用的是关系型数据库 MySQL，针对需求设计出系统所用到的表以及各个表之间的关系。

后端的作用是处理业务逻辑和数据库存储的数据，并会根据输入的数据对数据库进行增、删、改、查的操作，进行相应的处理后将数据返回给前端或者用户。

深度学习是一类机器学习^[12]，它在非结构化数据上表现得更好。深度学习技术优于当前的机器学习技术。它使计算模型能够从多个级别的数据中逐步学习特征。随着可用数据量的增加以及提供强大计算机的硬件的进步，深度学习的普及得到了放大。深度学习在图像处理、视频处理、语音处理和音频处理方面带来了全新的突破。同时深度学习为传统行业医疗、化学、工业等领域赋能，对传统行业进行信息化、智能化的升级。

目标检测是深度学习领域中的一个常见应用问题^[17]，目标检测技术主要是在图像中准确地检测到特定目标类别的对象，并对每个检测到的对象进行相应回归框和类别的标注。在过去的数十年内，科研界和工业界对其进行了深入地研究和探讨。本系统读入膀胱镜影像视频，并通过人工智能目标检测模型标记出视频中的疑似病变区域，从而辅助医生诊断。

2.2 技术选型

2.2.1 Spring Boot

Spring Boot^[18]是一个当今流行的后端框架，它是基于 Spring 框架衍生出来的，相比于 Spring^[19]框架，Spring Boot 免去了许多重复繁琐的配置、XML 文件的编写以及插件的集成；Spring Boot 还有自动配置的功能，通过此功能可以自动

将类库引入到项目中，简化了 Spring 的配置。Spring Boot 有着约定大于配置的特性，正如官网所写的 Just Run，开发人员只需遵循 Spring Boot 的规则便能快速的搭建出一个后端项目。Spring Boot 的最大特点便是快速启动，Spring Boot 集成了 Undertow、Tomcat^[20]、Jetty 三种嵌入式容器。相比于传统的 java 后端程序，Spring Boot 可以快速的启动项目。Spring Boot 也内部集成了很多的插件，开发者可以直接使用，从而提高开发的效率。同时，得益于 Spring Boot 的 IOC 和 AOP 思想，java 项目变得更加的高内聚低耦合，从而提高项目的鲁棒性和可维护性。

使用 Spring Boot 项目只需如下步骤，在 Idea 中通过 Spring Initializr 创建 Spring Boot 项目；在 maven 中引入相关的依赖，将第三方类库引入到项目中；编写相应的业务逻辑代码；最终将项目部署到服务器上输入相关命令，启动应用程序。

Spring Boot 是开源优秀的 java 后端开发框架，基于 IOC、AOP 的思想以及自动配置和快速启动的特性，使得开发者不用管繁琐的配置，可以专注于业务逻辑代码的编写并快速地搭建项目。Spring Boot 目前已经被国内外的各大厂商进行应用，已经成为 java 开发的主流框架。

2.2.2 Vue.js

Vue.js^[21]是 2014 年 Evan You 发布的渐进性式 JavaScript^[22]框架，用来构建交互式的用户界面，同时也符合前后端分离的开发思想。Vue.js 是轻量级架构，整个文件大小仅仅只有 20kb 左右，同时 Vue.js 采用虚拟 DOM 来提高性能，这导致 Vue.js 不仅有着轻量级的体量还有着非常优秀的性能。Vue.js 有着一套响应式数据绑定系统，它可以在相应的数据变化时进行更新。Vue.js 将一个页面拆分成不同的组件，并且不同的组件也可以进行组合，从而降低项目的复杂度，便于开发者进行开发。Vue.js 是一个优秀的开源前端框架，使得前端开发者可以快速的搭建构建高质量的应用程序。

2.2.3 Pytorch + Flask

FaceBook 于 2016 年推出了一个开源的 Deep Learning 框架 Pytorch^[23]，用于科学计算和人工智能研究，目前 Pytorch 已经被广泛的应用到自然语言处理、计算机视觉等深度学习的领域。相比于 Caffe^[24]、TensorFlow^[25]、Keras^[26]、MXNet^[27]深度学习框架相比，Pytorch 是一个简单、易用的框架，它可以快速的完成神经网络的搭建神经网络。目前国内外的科研领域的学者大部分用 Pytorch 来构建深度学习模型，同时 Pytorch 因其部署的优化、模型的压缩，也深受工业界的喜爱。

Flask^[28]框架是一个基于 Python 的微型 Web 应用框架，Flask 包含一些工具类和第三方库，可以帮助开发者快速地构建、部署 Python 网页应用程序。

2.3 硬件平台

开发用的电脑配置如下：

- CPU: i5-9300H
- 显卡: NVIDIA GeForce GTX 1660ti
- 内存: 8G

训练目标检测模型的机器配置如下：

- CPU: AMD Ryzen Threadripper 3960X 24-Core Processor
- 显卡: NVIDIA GeForce RTX 3090
- 内存: 128G

此项目采用腾讯云服务器，在服务器上分别部署前端、后端、数据库、人工智能微服务。

2.4 开发工具

Spring Boot: 采用 IDEA 来搭建后端项目。

Vue.js: 采用 WebStorm 来搭建前端的项目。

Pytorch + Flask: 采用 Pycharm 来搭建人工智能微服务项目。

2.5 本章小结

本章对项目所涉及到的前后端、深度学习、目标检测等技术进行了相关介绍，并说明了项目的技术选型，同时还介绍了项目的硬件平台以及开发用到的开发工具。

3 需求分析及概要设计

3.1 需求概览

为更好地挖掘客户的需求，我对合作医院的膀胱镜应用场景进行了调研，下图是调研的膀胱镜检查报告单：

某某医院

膀胱镜检查报告单

检查号: 20200205		仪器型号:	
姓名: 李某某	性别: 男 年龄: 58岁 科别:	申请医师:	
病历号:	住院号:	病区:	床位号:
临床诊断:		麻醉用药:	



检查所见:

直视下进镜，见膀胱颈口肿瘤覆盖，基底位于颈口11-12点，膀胱三角区、左右侧壁、后壁多发占位，双侧输尿管开口未查及，取组织三块送病理检查。



检查提示:

膀胱肿瘤

医生建议:

送病理检查，手术治疗

仅供临床参考，不作证明材料！

检查医师: admin

检查日期: 2020-02-24

图 3.1 膀胱镜检查报告单

根据网络调研和场景调研，我详细地分析了本系统的需求。本系统主要分两个角色，分别是医生端和管理员端。

医生端的功能如下：

- 基于深度学习的目标检测模型辅助医生诊断
- 对病例信息进行查阅和管理
- 查询可视化的病例统计信息
- 管理个人信息

管理员端的功能如下：

- 查询安徽省地图的统计信息
- 查询可视化的病例、人员、医院的统计信息
- 对人员进行查询和管理
- 对医院信息进行查询和管理
- 对病例信息进行查阅和管理

本系统主要是基于深度学习技术辅助医生进行膀胱镜检查，从而减轻医生的诊断和阅片压力，从而提高早诊率，降低误诊率和错诊率。**本系统主要有以下的功能：**

- 医生上传一个或多个膀胱镜影像视频，系统通过基于深度学习的目标检测模型检测视频中的癌变区域，并标注出影像中的疑似病变区域从而辅助医生诊断。
- 医生根据分析的膀胱镜影像视频给出检查提示和建议，并最终生成膀胱镜检查报告单持久化保存到专属的数据库中。
- 医生和管理员可以查看、管理膀胱镜的病例信息，并且系统支持将病例导出 pdf，方便用户的查看。
- 系统向医生、管理员可视化的展示病例的统计信息，并且向管理员展示安徽省地图的病例信息和人员信息。
- 管理员可以对系统的人员信息进行查询、修改、新增操作，并且可以对系统的医院信息进行查询、修改、新增操作。

根据上述的需求分析，我针对本系统制作了需求分析表：

表 3.1 需求分析表

功能类别	子模块	子功能	功能描述
登录功能	用户账号	无	用户账号便于管理
	用户密码	无	前端加密不用后端加密
	验证码校验	无	验证是否为人工操作
	注册	无	注册不用放到登录里，注册放到系统组织管理员里面进行注册
	退出登录	无	用户可以退出功能
病例模块	膀胱镜病例 视频分析	AI 分析	根据人工智能算法实时分析膀胱镜视频
		医生诊断	医生根据人工智能的辅助信息进行诊断
		生成报告	医生根据分析信息生成报告单
	病例管理	新增	医生新建一个病例
		查询	根据不同的需求对病例进行条件查询
		导出	将病例导出为 pdf
部门管理功能	系统人员管理	查询	系统人员信息查询
		修改	系统人员信息修改
		新增	新增系统人员
	医院管理	查询	医院信息查询
		修改	医院信息修改
		新增	新增医院
数据可视化功能	统计	无	根据医院、病例进行数据统计分析
	折线图展示	无	以折线图形式展示数据
	地图展示	无	以地图形式展示相关数据

3.2 登录模块

3.2.1 功能描述

表 3.2 登录模块功能描述表

需求编号	1	需求名称	登录模块需求
优先级	2	需求描述	用户通过登录模块进行登录
参与者	医生、管理员		
过程描述	医生或管理员输入账号和密码，密码进行加密后传输，再通过验证码验证，保证是人为的操作后，验证码校验成功后密码和数据库进行匹配。同时将医生或管理员的信息存储到服务器的内存中，并将用户的 token 凭证返回给前端，前端保存 token 凭证。医生或管理员在请求后端接口时，需要将 token 凭证带上，后端根据 token 凭证判断是否拦截请求。后端通过 token 值进行权限控制 and 安全管理，使得系统的登录和整体框架变得更加的鲁棒和安全。		
业务规则	医生或管理员必须通过密码和验证码的双重验证才能进行登录，验证码可以有效地防止恶意攻击。 为使系统变得更加的鲁棒，医生或管理员即使通过更改浏览器的 url 地址也不能进行不同页面的跳转。医生或管理员在点击登录按钮时，系统会向用户反馈登陆的状态：登录成功、验证码错误、密码错误、没有此用户名等提示信息。		

3.2.2 业务流程图

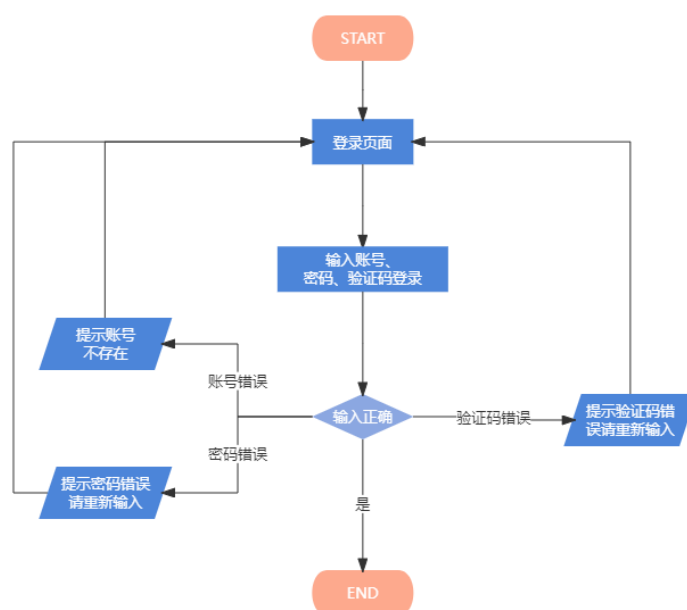


图 3.2 登录模块业务流程图

3.3 病例模块

3.3.1 功能描述

表 3.3 病例模块功能描述表

需求编号	2	需求名称	病例模块需求
优先级	1	需求描述	用户新增、管理、查询病例
参与者	医生、管理员		
过程描述	<p>医生填写病例信息并新建一个病例，之后医生上传一个或多个膀胱镜影像视频进行诊断，医生可以进行人工诊断或 AI 诊断。医生在人工诊断中可以截取医生认为的疑似病变区域。</p> <p>医生在人工智能辅诊中可以通过目标检测模型自动的标注出可疑的疑似病变区域，辅助医生进行诊断。</p> <p>医生根据截取的疑似病变区域进行诊断，并撰写检查所见、检查提示、建议，最终将病例持久化的保存在特定的数据库中。</p> <p>医生和管理员可以根据相应的查询条件对病例进行检索，并可以导出病例的 pdf。</p>		
业务规则	<p>医生需要填写完整的病例信息才可以进行之后新增病例、上传视频的操作。</p> <p>考虑到患者的隐私性，医生只能查看自己诊断的病例，无权查看其它医生诊断的病例。</p> <p>管理员可以查看所有的病例，但没有权限进行新增病例和人工智能辅助诊断。</p>		

3.3.2 业务流程图

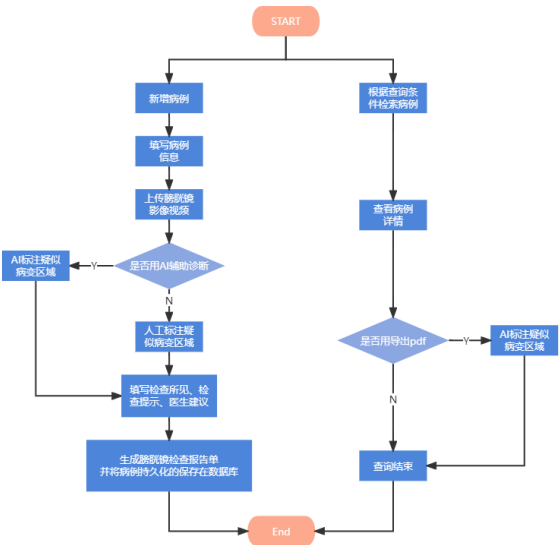


图 3.3 病例模块流程图

3.4 部门模块

3.4.1 功能描述

表 3.4 部门模块功能描述表

需求编号	3	需求名称	部门模块需求
优先级	3	需求描述	管理员对人员、医院信息进行管理
参与者	管理员		
过程描述	管理员对系统人员的账号、姓名、年龄、所在的医院、工作状态、权限等查询条件进行检索。 管理员对系统人员的信息进行修改。 管理员对系统人员的进行新增。 管理员对医院的名称、所在的城市和县城等查询条件进行信息检索。 管理员对医院的信息进行修改。 管理员对医院的信息进行新增。		
业务规则	只有管理员端有权限对人员、医院信息进行管理，而医生端没有权限。		

3.4.2 业务流程图

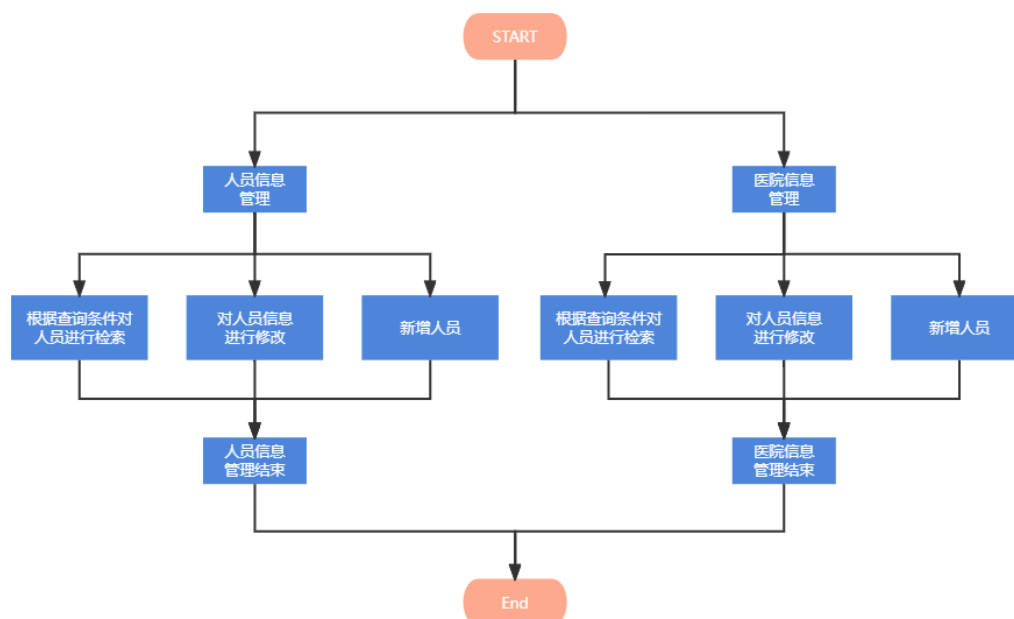


图 3.4 部门模块流程图

3.5 数据可视化模块

3.5.1 功能描述

表 3.5 数据可视化模块功能描述表

需求编号	4	需求名称	数据可视化模块需求
优先级	3	需求描述	医生和管理员通过可视化数据信息来快速了解系统的数据概览
参与者	医生和管理员		
过程描述	<p>医生通过可视化模块了解到医生所诊断的总病例数、人工诊断病例数、AI 辅诊病例数。</p> <p>医生\管理员可以了解到前六个月中每个月自己诊断的总病例数、人工诊断病例数、AI 辅诊病例数。</p> <p>管理员可以查看安徽省地图中的各个区域所对应的病例数量。</p> <p>管理员可以了解整个系统的总病例数、总医生数、总医院数。</p>		
业务规则	<p>管理员可以查看安徽省的病例数量统计概览，而医生没有权限查看这类信息。</p> <p>医生的可视化模块只会统计自己诊断的病例并不会统计其他医生诊断的病例。</p>		

3.5.2 业务流程图

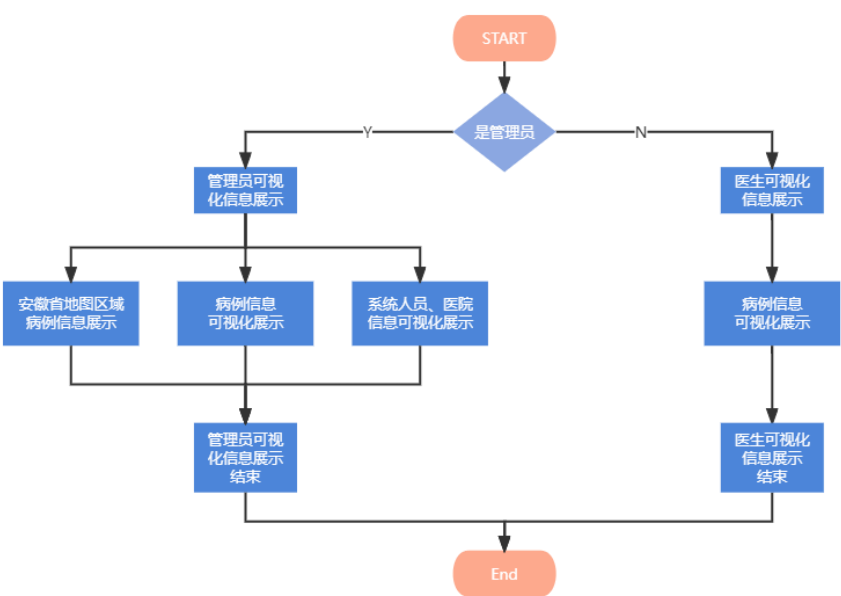


图 3.5 数据可视化模块流程图

3.6 本章小结

本章首先讲述了系统的网络调研和场景调研,并基于调研的信息介绍了系统的角色和需求概览表。基于需求概览表,对本系统进行了概要设计,本系统主要分为4个模块,分别是登录模块、病例模块、部门模块、数据可视化模块,最终对这四个模块的功能和业务流程图进行了详细的描述。

4 系统设计与实现

4.1 系统框架设计

基于上述对系统的需求分析和概要设计，我们将整个系统分为登录模块、病例模块、部门模块、数据可视化模块，并对每个模块的主要功能点进行了分析和描述：

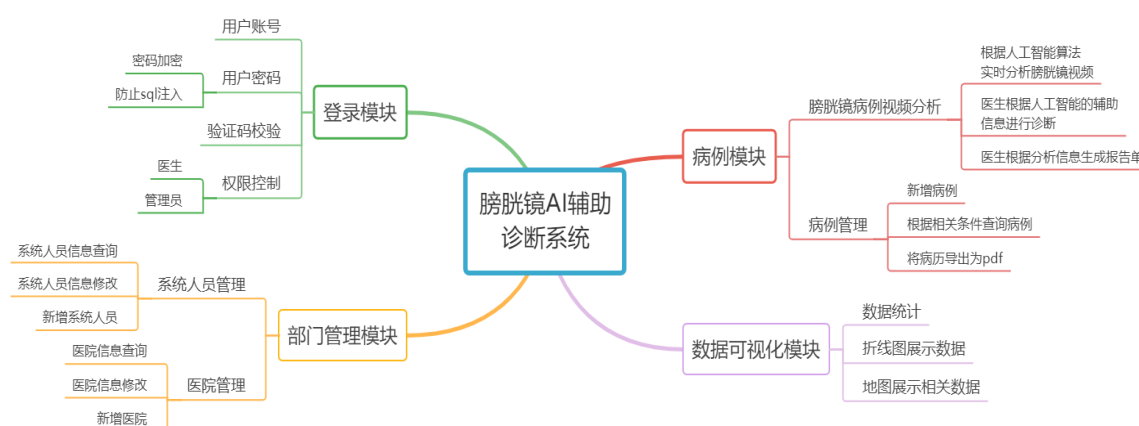


图 4.1 系统框架图

4.2 数据库设计

4.2.1 数据库概念结构设计

本系统主要存储用户的信息，病例的信息、医院的信息、地域的信息、同时病例中包含一个或多个视频信息，所以还要存储视频信息。一个视频包含一个或多个疑似病变图片的信息，所以还要存储图片信息。一个图片信息包含一个或多个标注框信息，故需要存储标注框信息。

综上，数据库总共需要用户信息表、医院信息表、地域信息表、病例信息表、视频信息表、图片信息表、标注信息表。以下是数据库的概念数据模型：

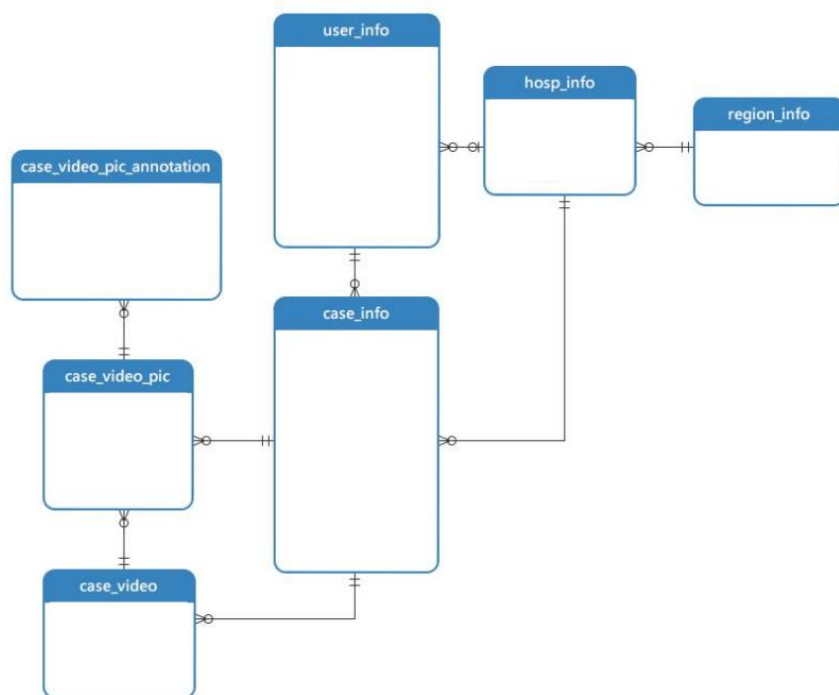


图 4.2 数据库的概念数据模型

4.2.2 数据库逻辑结构设计

根据上述数据库的概念数据模型进行完善，并填写出每个实体对应的属性，数据库的逻辑结构模型如下：

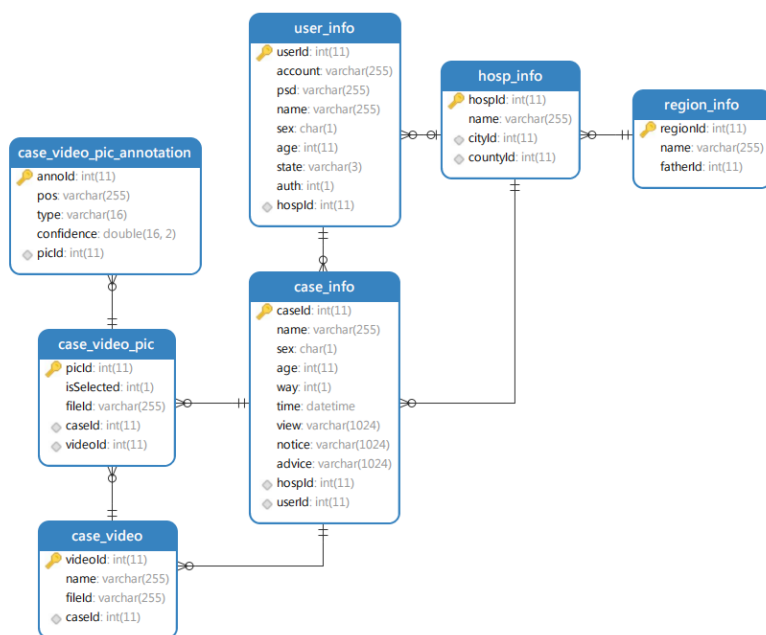


图 4.3 数据库的概念数据模型

4.2.3 数据字典设计

(1) 用户信息表数据字典(user_info 表)

表 4.1 用户信息表数据字典

PK/FK	字段	类型	描述
PK	userId	int(11)	用户 id
	account	varchar(128)	用户账号
	psd	varchar(64)	用户密码
	sex	char(1)	用户性别
	age	Int(3)	用户年龄
	state	varchar(6)	用户在职状态
	auth	int(1)	用户权限
FK	hospId	int(11)	医院 id

(2) 医院信息表数据字典(hosp_info 表)

表 4.2 医院信息表数据字典

PK/FK	字段	类型	描述
PK	hospId	int(11)	医院 id
	name	varchar(128)	医院名称
FK	cityId	int(11)	医院所在城市的 Id
FK	countyId	int(11)	医院所在的县/区 Id

(3) 地域信息表数据字典(region_info 表)

表 4.3 地域信息表数据字典

PK/FK	字段	类型	描述
PK	regionId	Int(11)	地区 id
	name	varchar(16)	地区名称
	fatherId	int(11)	县/区属于市，所以它的 fatherId 是市对应的 regionId(外键与该表的 regionId 相连)

(4) 病例信息表数据字典(case_info 表)

表 4.4 病例信息表数据字典

PK/FK	字段	类型	描述
PK	caseId	int(11)	病例 Id
	name	varchar(32)	患者姓名
	sex	char(1)	性别
	age	age(3)	年龄
	way	int(1)	诊断方式: 0 为人工诊断, 1 为 AI 辅诊, 默认为 0
	time	date	检查日期
	view	varchar(1024)	检查所见
	notice	varchar(1024)	检查提示
	advice	varchar(1024)	医生建议
FK	userId	int(11)	检查的医生 id(外键与 user_info 的 userId 相连)
FK	hospId	int(11)	医院 id(外键与 hosp_info 的 hospId 相连)

(5) 视频信息表数据字典(case_video 表)

表 4.5 视频信息表数据字典

PK/FK	字段	类型	描述
PK	videoId	int(11)	视频 id
	name	varchar(255)	视频名称
	fileId	varchar(255)	云点播 id
	url	varchar(255)	视频的 url 路径
FK	caseId	int(11)	视频对应的病例
	state	int(1)	视频状态

(6) 图片信息表数据字典(case_video_pic 表)

表 4.6 图片信息表数据字典

PK/FK	字段	类型	描述
PK	picId	int(11)	病例图片 id
	isSelected	int(1)	图片是否在报告单中, 0 代表不在, 1 代表在, 默认为 0
	fileId	varchar(255)	云点播 id
	url	varchar(512)	图片 url 路径
FK	videoId	int(11)	图片对应的视频(外键与 case_video 的 videoId 相连, 一个视频有多张图片)
FK	caseId	int(11)	图片对应的病例(外键与 case_info 的 caseId 相连, 一个病例有多个图片)

(7) 标注信息表数据字典(case_video 表)

表 4.7 标注信息表数据字典

PK/FK	字段	类型	描述
PK	annoId	int(11)	图片标注 id
	pos	varchar(128)	位置信息(x,y,w,h)
	type	varchar(16)	类别
	confidence	varchar(8)	置信度
FK	picId	int(11)	病例图片

4.3 前端页面设计

通过之前的需求分析, 我针对本系统共设计了 20 多个页面。并且我遵从用户友好性、简洁等软件工程的设计理念进行页面设计。下图是系统前端页面设计的概览图:

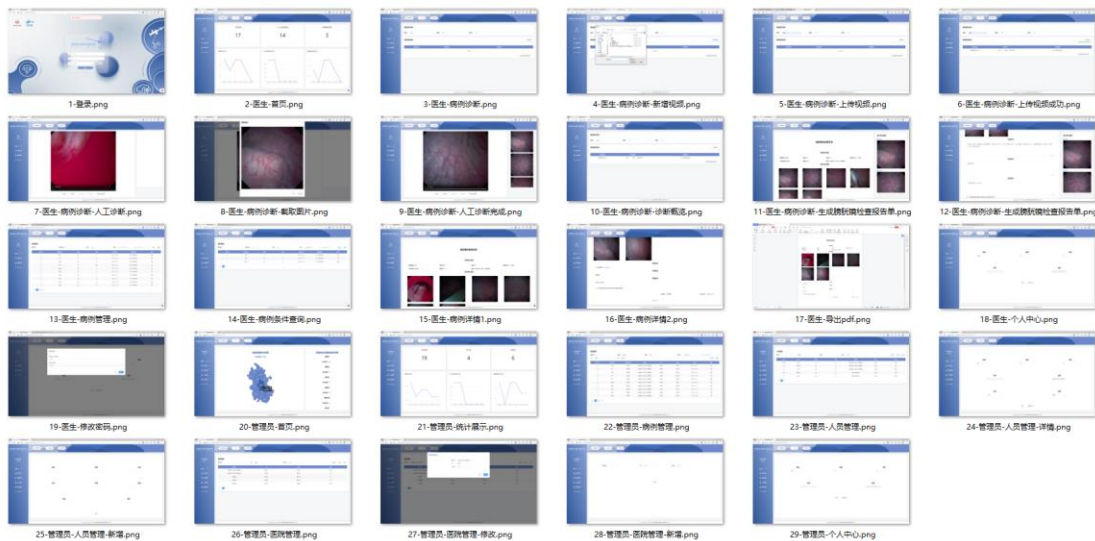


图 4.4 前端页面设计概览

4.4 后端设计

本文对后端划分成登录模块、数据可视化模块、病例模块、部门管理模块，并基于公司的规范撰写后端接口文档。后端总共设计了 30 多个接口，但受限于篇幅，在本节仅展示重要的后端接口设计。

4.4.1 登录模块

表 4.8 接口 1 描述

接口序号	1		
接口地址	http://服务器地址:8088/userInfo/login		
方法名称	login		
参数列表	参数名称	类型	备注
	account	String	用户名
	psd	String	密码
接口内容说明	根据用户名和 psd 在 user_info 表查找，如果用户名和 psd 都对，则保存 userId、account、auth、token，返回登录状态、token 信息、userId、name、权限信息(返回值为 JSONObject 对象)		
调用者说明	前端登录按钮触发的方法 前端需要保存后端传过来的信息		
其他说明	无		

表 4.9 接口 2 描述

接口序号	2		
接口地址	http://服务器地址:8088/userInfo/getInfo		
方法名称	getInfo		
参数列表	参数名称	类型	备注
	无	无	无
接口内容说明	前端传 token，后端根据 token 找到后端保存的 userId，将 user_info、hosp_info 连接，通过 userId==userId 条件，返回 user_info 以及医院名称的信息		
调用者说明	前端图 8 初始化方法		
其他说明	无		

表 4.10 接口 16 描述

接口序号	16		
接口地址	http://服务器地址:8088/userInfo/updatePsd		
方法名称	updatePsd		
参数列表	参数名称	类型	备注
	psd	String	用户密码
	newPsd	String	用户新密码
接口内容说明	前端传 token，后端根据 token 找到后端保存的 userId，在 user_info 表中，如果 userId 和 psd 符合，则更新 psd，返回为 true 否则返回 false		
调用者说明	前端图 8 修改密码调用的方法		
其他说明	无		

表 4.11 接口 17 描述

接口序号	17		
接口地址	http://服务器地址:8088/userInfo/updateUserInfo		
方法名称	updateUserInfo		
参数列表	参数名称	类型	备注
	userInfo	UserInfo	UserInfo 类
接口内容说明	在 user_info 表中，根据 userId 修改对应的信息，返回更新成功与否		
调用者说明	前端图 8 保存调用的方法，账号不能更改		
其他说明	无		

4.4.2 数据可视化模块

表 4.12 接口 3 描述

接口序号	3		
接口地址	http://服务器地址:8088/caseInfo/numberData		
方法名称	numberData		
参数列表	参数名称	类型	备注
	无	无	无
接口内容说明	<p>前端传 token，后端根据 token 找到后端保存的 userId，</p> <p>在 case_info 表的 userId==查询的 userId 得到 case_info 表的总病例数，在 case_info 表的 userId==查询的 userId and way==0 得到 case_info 表的人工诊断数量，将得到的总病例数-人工诊断数量得到 AI 辅诊病例数量，</p> <p>返回总病例数、人工诊断数、AI 辅诊病例数；</p>		
调用者说明	<p>前端图 2 页面初始化调用的方法，</p> <p>前端图 10 页面初始化调用的方法</p>		
其他说明	无		

表 4.13 接口 4 描述

接口序号	4		
接口地址	http://服务器地址:8088/caseInfo/monthData		
方法名称	monthData		
参数列表	参数名称	类型	备注
	无	无	无
接口内容说明	<p>医生端：前端传 token，后端根据 token 找到后端保存的 userId，依次查询前六个月的总病例数、人工诊断数、AI 辅诊病例数，在 case_info 表的 userId==查询的 userId 得到 case_info 表的总病例数，在 case_info 表的 userId==查询的 userId and way==0 得到 case_info 表的人工诊断数量，将得到的总病例数-人工诊断数量得到 AI 辅诊病例数量，返回前六个月的总病例数、人工诊断数、AI 辅诊病例数；管理员端：前端传 token，后端根据 token 找到后端保存的 userId，如果 auth==1，将 userId="", 依次查询前六个月的总病例数、人工诊断数、AI 辅诊病例数，在 case_info 表得到 case_info 表的总病例数，在 case_info 表的 way==0 得到 case_info 表的人工诊断数量，将得到的总病例数-人工诊断数量得到 AI 辅诊病例数量，返回前六个月的总病例数、人工诊断数、AI 辅诊病例数</p>		
调用者说明	<p>前端图 2 页面初始化调用的方法</p> <p>前端图 10 页面初始化调用的方法</p>		
其他说明	无		

表 4.14 接口 24 描述

接口序号	24		
接口地址	http://服务器地址:8088/userInfo/resetPsd		
方法名称	resetPsd		
参数列表	参数名称	类型	备注
	userId	String	用户 id
接口内容说明	根据 userId 将密码重置为 1234，返回重置成功与否		
调用者说明	前端图 13 的重置密码调用的方法		
其他说明	无		

4.4.3 病例模块

表 4.15 接口 5 描述

接口序号	5		
接口地址	http://服务器地址:8088/caseInfo/addCase		
方法名称	addCase		
参数列表	参数名称	类型	备注
	caseInfo	CaseInfo	CaseInfo 类文字信息
接口内容说明	根据前端传来的病例基本信息，在 case_info 表里面新建一个病例信息（病例信息中的医生 id 通过 sa-token 获取，医院 id 通过在 user_info 表里面获得，way 字段设置为 0，将当前时间作为），并返回给前端一个 caseId		
调用者说明	前端图 3 页面的新建病例按钮，前端需要将该 caseId 进行全局保存，caseId 默认值为 0		
其他说明	无		

表 4.16 接口 11 描述

接口序号	11		
接口地址	http://服务器地址:8088/caseVideo/initialCaseVideo		
方法名称	initialCaseVideo		
参数列表	参数名称	类型	备注
	caseId	String	接口 5 返回的 caseId
接口内容说明	在 case_video 表中，根据 caseId==caseId，返回 case_video 的所有信息以及在 case_video_pic 表中，根据 videoId== videoId，返回图片的个数；		
调用者说明	前端图 3 初始化的方法，如果 caseId 不为-1 调用该方法 根据 state 视频状态；state=0 代表该视频未被 ai 分析过；state=1 代表该视频正在被 ai 分析；state=2 代表已经被 ai 分析过了		
其他说明	无		

表 4.17 接口 9 描述

接口序号	9		
接口地址	http://服务器地址:8088/caseVideoPic/addPic		
方法名称	addPic		
参数列表	参数名称	类型	备注
	caseVideoPic	CaseVideoPic 类	CaseVideoPic 类
接口内容说明	根据传进来的参数，在 case_video_pic 表里面新增数据，返回是否新增成功		
调用者说明	前端图 4 页面的截图按钮调用的方法，同时将云点播保存的图片保存到右边的图像列表中，右边的图像可以放大查看		
其他说明	无		

表 4.18 接口 12 描述

接口序号	12		
接口地址	http://服务器地址:8088/caseVideoPic/getPicByCaseId		
方法名称	getPicByCaseId		
参数列表	参数名称	类型	备注
	caseId	String	接口 5 返回的 caseId 即前端全局变量 caseId
	isSelected	int	是否在病例中的图片，1 代表在，0 代表不在，如果 isSelected 传的值 == "" 代表不做为查询条 1 件
接口内容说明	在 case_video_pic 表中，根据 caseId==caseId and isSelected == isSelected 返回 case_video_pic 的 fileId、picId、url 列表		
调用者说明	前端图 5 右边的初始化的方法； 图 7 的病例详情		
其他说明	无		

表 4.19 接口 7 描述

接口序号	7		
接口地址	http://服务器地址:8088/objectDetection		
方法名称	objectDetection		
参数列表	参数名称	类型	备注
	video	视频流	图 4 实时播放的视频
接口内容说明	根据视频流的当前帧数，返回位置信息、类别、置信度		
调用者说明	前端图 3 页面上传视频按钮调用的方法		
其他说明	无		

表 4.20 接口 15 描述

接口序号	15		
接口地址	http://服务器地址:8088/caseInfo/getCaseInfo		
方法名称	getCaseInfo		
参数列表	参数名称	类型	备注
	caseId	String	病例号(可能不传)
	name	String	名字(可能不传)
	sex	String	性别(可能不传)
	startTime	String	开始时间(可能不传)
	endTime	String	结束时间(可能不传)
	page	String	页面
	size	String	一面展示的数据
接口内容说明	在 case_info 表中，根据上述的条件， 返回 case_info 所有信息的病例列表和病例视频中含有图片个数		
调用者说明	前端图 6 初始化、查询、重置调用的方法		
其他说明	无		

表 4.21 接口 21 描述

接口序号	21		
接口地址	http://服务器地址:8088/caseInfo/getCaseAndHospInfo		
方法名称	getCaseAndHospInfo		
参数列表	参数名称	类型	备注
	caseId	String	病例号(可能不传)
	userName	String	病例医生的名字
	hospName	String	医院的名字(可能不传)
	cityId	int	医院所在的城市 id
	countyId	int	医院所在的县 id
	startTime	String	开始时间(可能不传)
	endTime	String	结束时间(可能不传)
接口内容说明	将 case_info、user_info、hosp_info、region_info 连接，根据上述的条件，返回表格中所需的病例列表		
调用者说明	前端图 11 的初始化方法、查询方法、重置方法。注：前端需要传城市和县的 id		
其他说明	无		

表 4.22 接口 32 描述

接口序号	32		
接口地址	http://服务器地址:8088/caseVideoPic/updateSelectPic		
方法名称	updateCaseInfo		
参数列表	参数名称	类型	备注
	picIds	List	图 5 中疑似病变列表
接口内容说明	在 case_video_pic 表中，对 picIds 列表中根据 picId 将对应的 isSelected 赋值为 1		
调用者说明	前端图 5 保存按钮的调用方法，前端的全局变量 caseId 置为-1		
其他说明	无		

表 4.23 接口 10 描述

接口序号	10		
接口地址	http://服务器地址:8088/caseInfo/initialCaseInfo		
方法名称	initialCaseInfo		
参数列表	参数名称	类型	备注
	caseId	String	接口 5 返回的 caseId
接口内容说明	在 case_info 表中根据 caseId==caseId, 返回 case_info 表的所有信息(case_info 表的所有信息+医院名称和医生名称)		
调用者说明	前端图 3 初始化的方法, 如果 caseId > 0 调用该方法; 前端图 5 的初始化的方法, 前端图 7 的初始化的方法		
其他说明	无		

4.4.4 部门管理模块

表 4.24 接口 25 描述

接口序号	25		
接口地址	http://服务器地址:8088/hospInfo/getHospInfo		
方法名称	getHospInfo		
参数列表	参数名称	类型	备注
	name	String	医院名字
	cityId	int	医院所在城市
	countryId	int	医院所在的县
	page	String	页面
	size	String	一面展示的数据
接口内容说明	在 hosp_info 表和 region_info 中, 根据传进来的参数, 返回医院信息列表		
调用者说明	前端图 15 的初始化、搜索、重置方法		
其他说明	无		

表 4.25 接口 22 描述

接口序号	22		
接口地址	http://服务器地址:8088/userInfo/getUserInfo		
方法名称	getUserInfo		
参数列表	参数名称	类型	备注
	userId	String	用户 id
	account	String	用户账号
	name	String	姓名
	hospName	String	医院名称
	state	String	医生状态, 在职 or 不在职
	page	String	页面
	size	String	一面展示的数据
接口内容说明	将 user_info、hosp_info 连接, 根据上述的条件, 返回用户 id、账号、姓名、性别、年龄、医院、状态列表		
调用者说明	前端图 12 的初始化方法、查询方法、重置方法		
其他说明	无		

表 4.26 接口 30 描述

接口序号	30		
接口地址	http://服务器地址:8088/hospInfo/getAllHospInfo		
方法名称	getAllHospInfo		
参数列表	参数名称	类型	备注
	无	无	无
接口内容说明	在 hosp_info 表中, 返回所有信息列表		
调用者说明	前端图 14 医院下拉框信息		
其他说明	无		

4.5 算法模块设计

通过研究，目标检测主流的方法分为两类，一个是以精度高、检测速度慢的 two stage 方法，代表模型是 Faster r-cnn^[29]，另一个是以 YOLO^[30]系列为首的速度快精确度稍低的 one stage 方法。接下来对 Faster r-cnn 和 YOLO 系列的深度学习模型进行相关介绍。

Faster r-cnn 模型主要分为两个部分：RPN 和检测网络。RPN 是一种特殊的卷积神经网络，它负责生成候选区域。检测网络是 Faster r-cnn，它是一个全卷积网络，将 RPN 产生的候选区域通过 RoI Pooling 层转换为固定大小的 Feature Maps，然后将 Feature Maps 输入到全卷积网络中进行分类和回归，最终得到检测的结果。

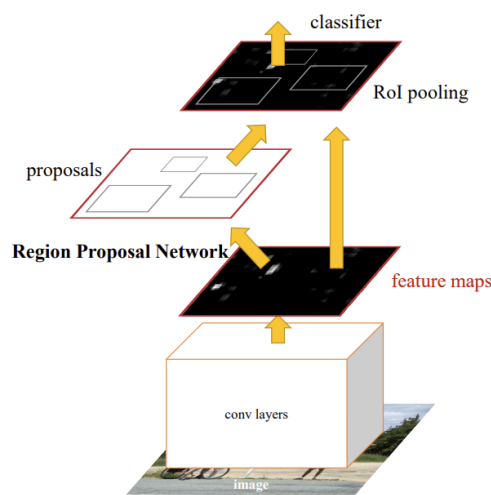


图 4.5 Faster r-cnn 模型图^[29]

YOLO V1 网络模型主要包含以下部分：特征提取网络和检测网络。YOLO V1 将输入图像分成网格，每个网格预测 B 个边界框，其中每个边界框也有得分。最终，YOLO 通过得分和分类得分进行加权，最终进行检测。

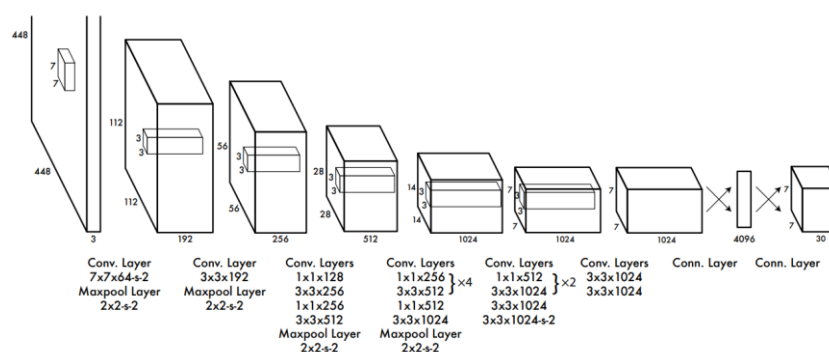


图 4.6 YOLO V1 模型图^[30]

本系统对算法的检测速度有着一定的要求，而检测的准确度没有太高的要求，故本系统将选择 YOLO V5 模型作为检测膀胱癌图像的方法。

如图是合作医院标注的膀胱镜数据集。

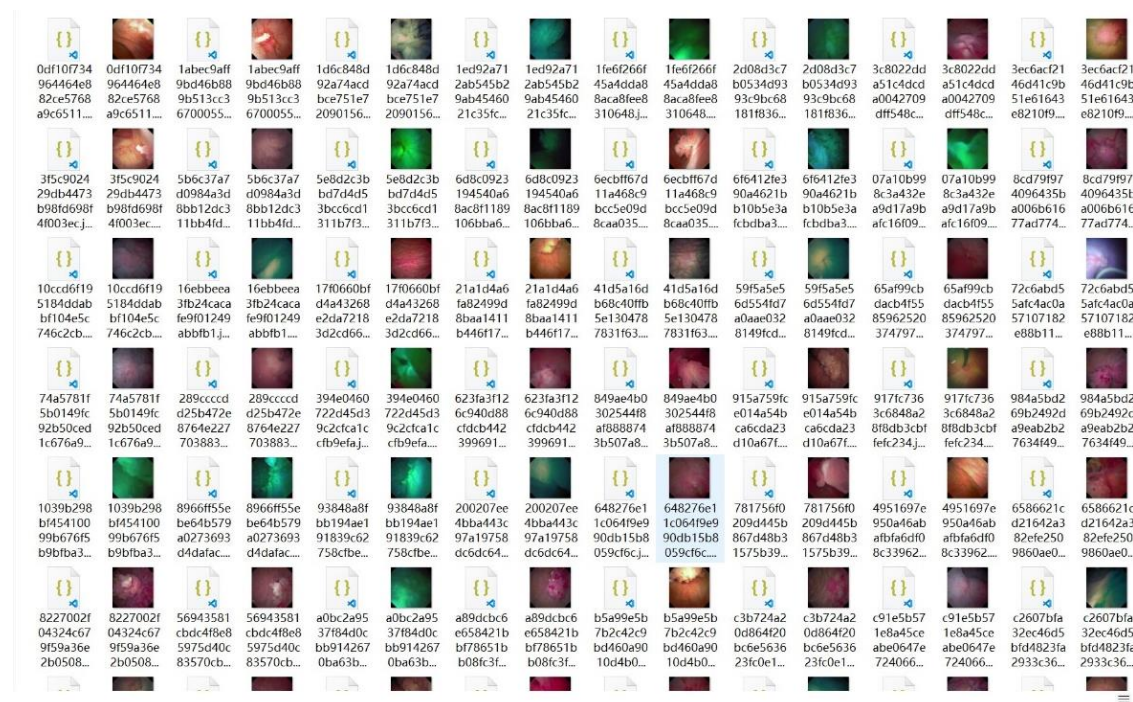


图 4.7 合作医院标注的膀胱镜数据集

最终基于合作医院标注的膀胱镜数据集训练出目标检测算法模型。下图是 YOLO V5 模型的训练过程。

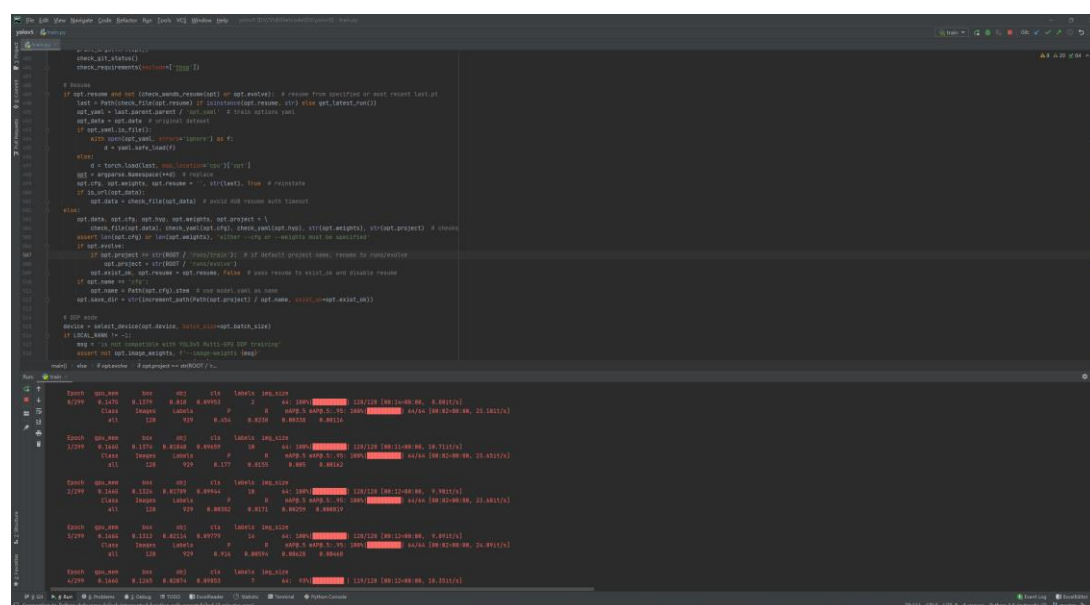


图 4.8 YOLO V5 模型的训练过程

4.6 成果展示

概览：本系统主要分为医生和管理员两个角色。医生主要有病例诊断、AI 辅助分析、病例管理、数据可视化信息展示功能。管理员主要有病例管理、数据可视化信息展示、医院管理、人员管理等功能。

代码量：本系统前端主要用 Vue 、Element-ui、Vuex、axios、echarts 框架来共同搭建页面，并且用 WebStorm 开发工具来编写前端代码；后端主要用 Spring Boot 、MyBatis 来共同搭建系统后端的业务逻辑，通过 Sa-Token 框架来进行登录管理和权限的控制管理，并用 IDEA 开发工具编写后端的代码；人工智能微服务主要用 Flask 、Pytorch 来共同搭建微服务接口，并用 PyCharm 开发工具来编写人工智能微服务的代码。膀胱镜 AI 辅助诊断系统的 3 个模块的总代码量约为 1.3 万行，以下是各个模块的代码量：

- 前端(Vue + Element-ui + Vuex + axios + echarts): 7000 行
- 后端(Spring Boot + MyBatis + Sa-Token): 4000 行
- 人工智能微服务 (Flask + Pytorch): 2000 行

登录功能展示：用户输入账号、密码、验证码进行登录，如果条件错误，则会弹出提示信息来提醒用户。

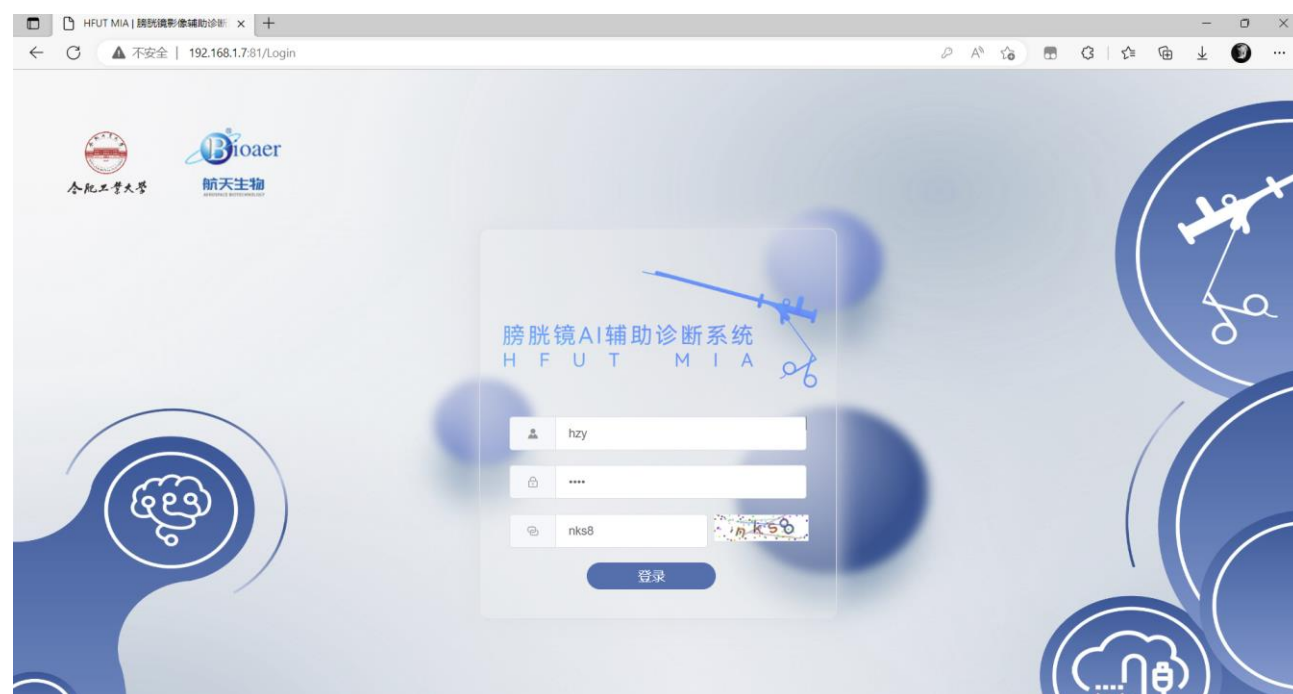


图 4.9 登录功能展示

医生端首页功能展示：医生端登录成功后，首先进入首页，首页主要统计了总病例数、人工诊断病例数、AI 辅助病例数以及一些折现统计图。

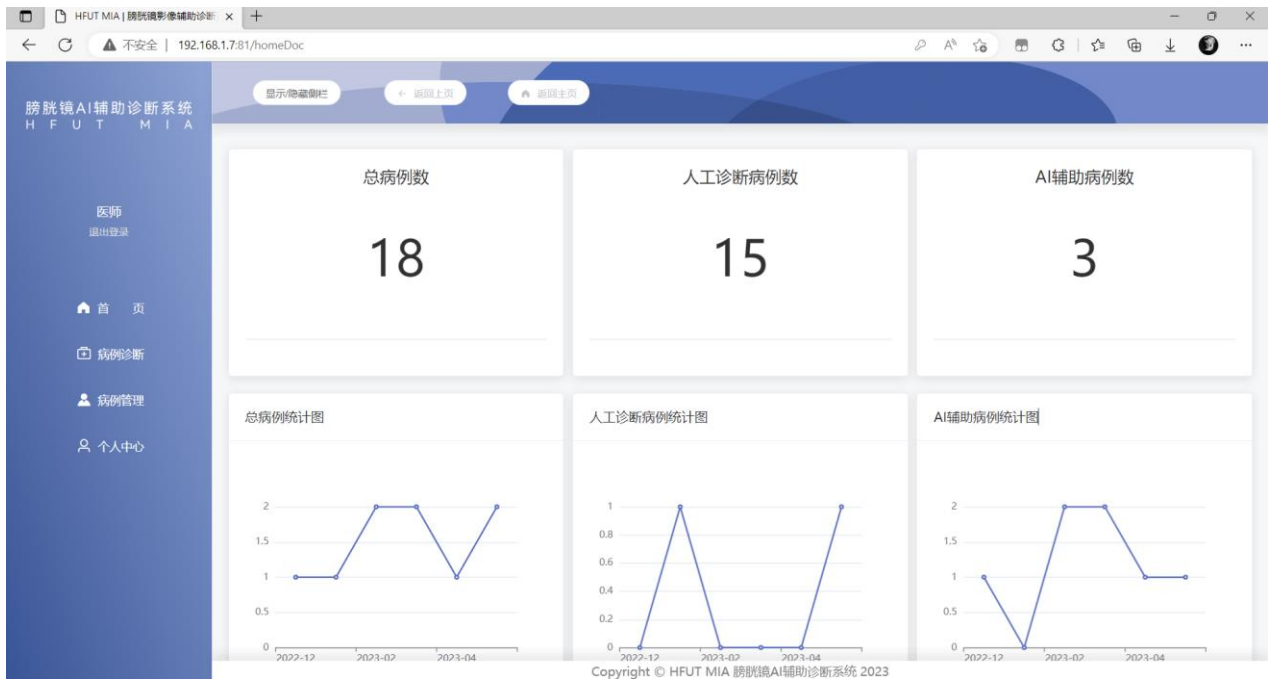


图 4.10 医生端首页功能展示

医生端病例诊断功能展示：医生可以填写病人的信息来新增病例。

The screenshot displays the '病例诊断' (Case Diagnosis) form. It features a '病例基本信息' (Case Basic Information) section with fields for '姓名' (Name) set to '罗文', '性别' (Gender) set to '男', and '年龄' (Age) set to '66'. Below this is a '病例视频列表' (Case Video List) section with a '新增视频' (Add Video) button and a table with columns '视频名称' (Video Name), '视频诊断' (Video Diagnosis), and '诊断概览' (Diagnosis Overview). The table currently shows '暂无数据' (No data). At the bottom right, there is a button labeled '生成膀胱镜检查报告单' (Generate Bladder Endoscopy Report). The footer includes the copyright notice 'Copyright © HFUT MIA 膀胱镜AI辅助诊断系统 2023' and a small logo.

图 4.11 医生端病例诊断新增功能展示

填写信息后，医生进行新增视频

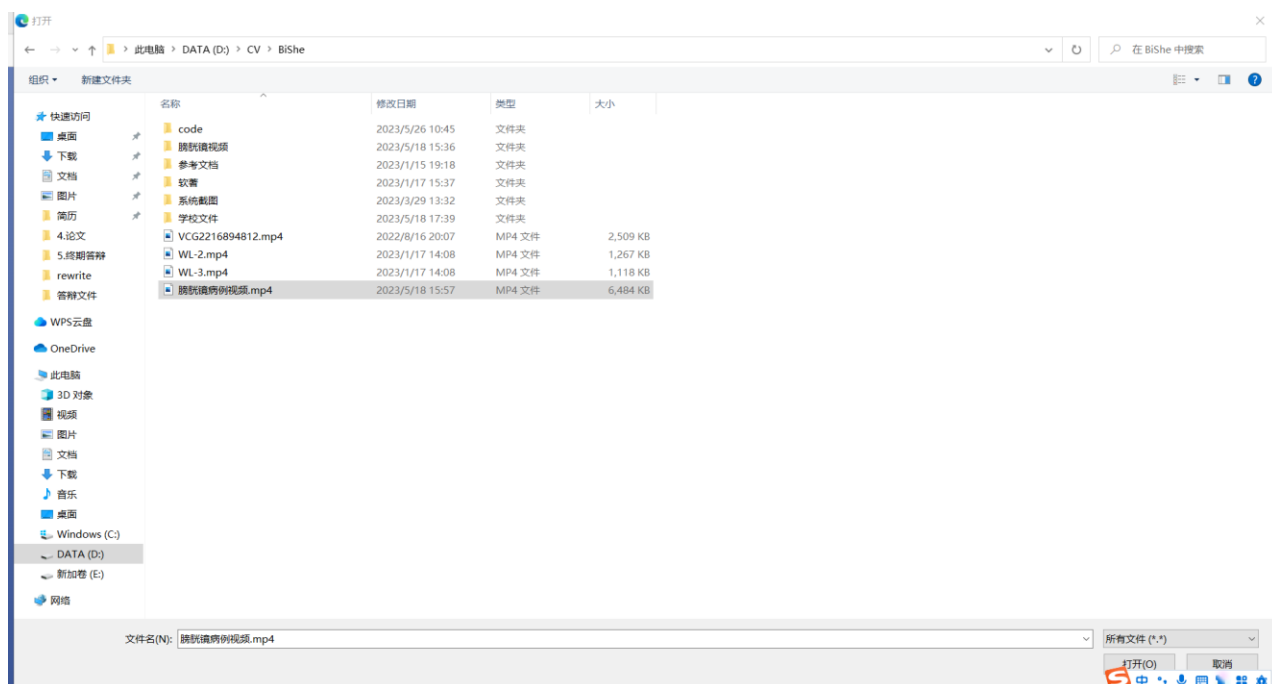


图 4.12 医生端病理诊断上传视频功能展示

上传视频时会出现上传的进度条



图 4.13 医生端病理诊断上传视频进度条功能展示

医生上传视频成功后，会出现视频列表，医生可以选择人工诊断或者 AI 辅诊，点击 AI 辅诊，系统会自动标注疑似病变区域的图片。



图 4.14 医生端新增视频后的功能展示

医生端点击人工诊断的按钮，会出现以下页面

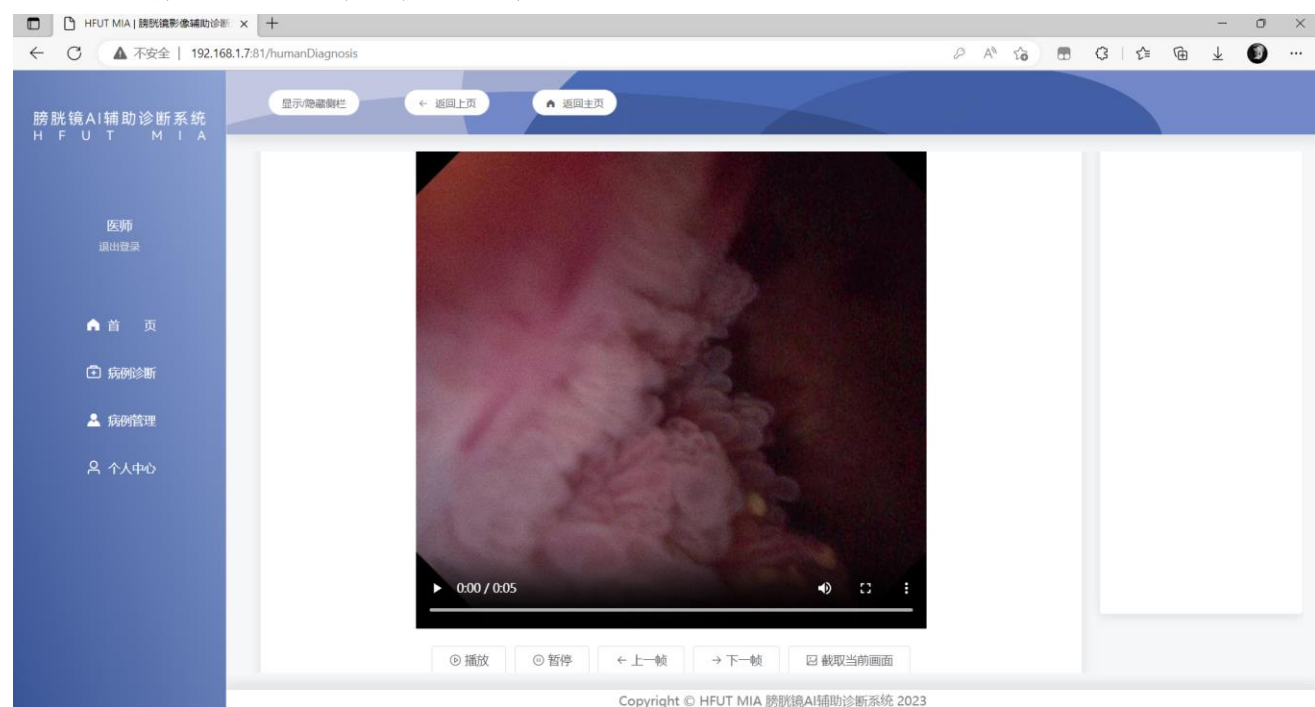


图 4.15 医生端人工诊断功能展示

医生如果看到了疑似病变区域，可以点击截取图片按钮将疑似病变区域持久化的存放到系统中。

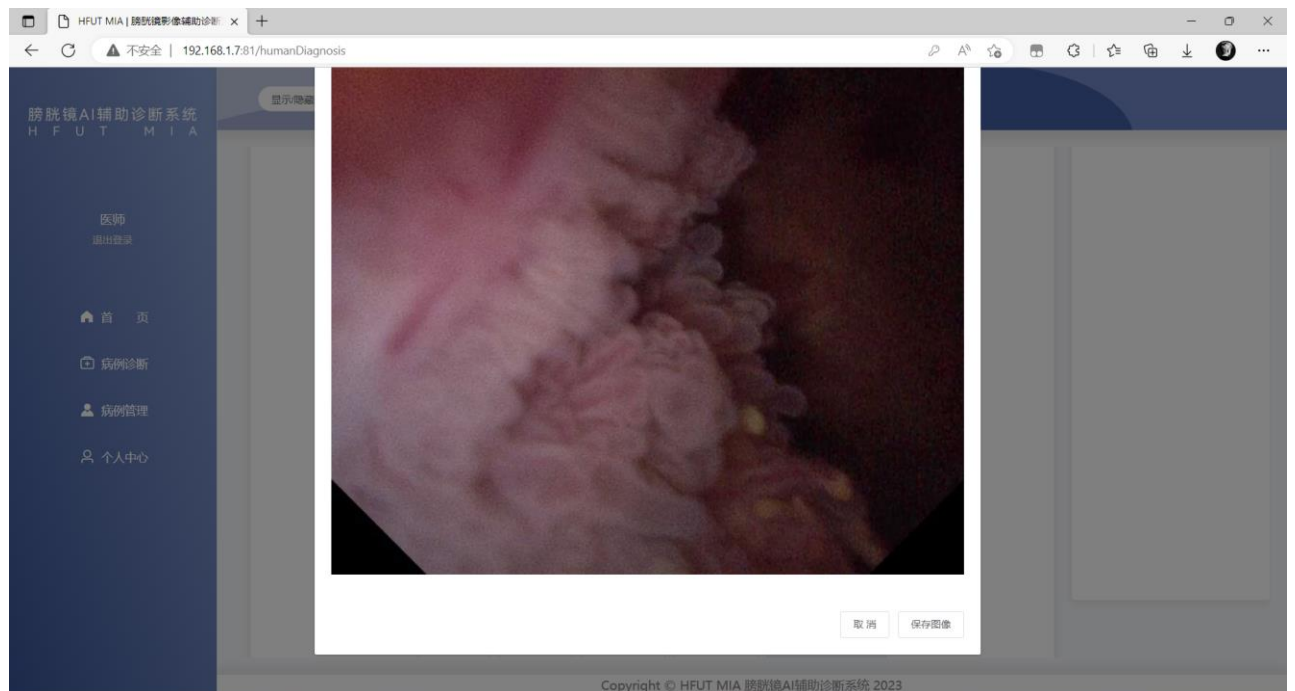


图 4.16 医生端截取疑似病变区域功能展示

医生截取的疑似病变区域会放在右边的区域进行展示

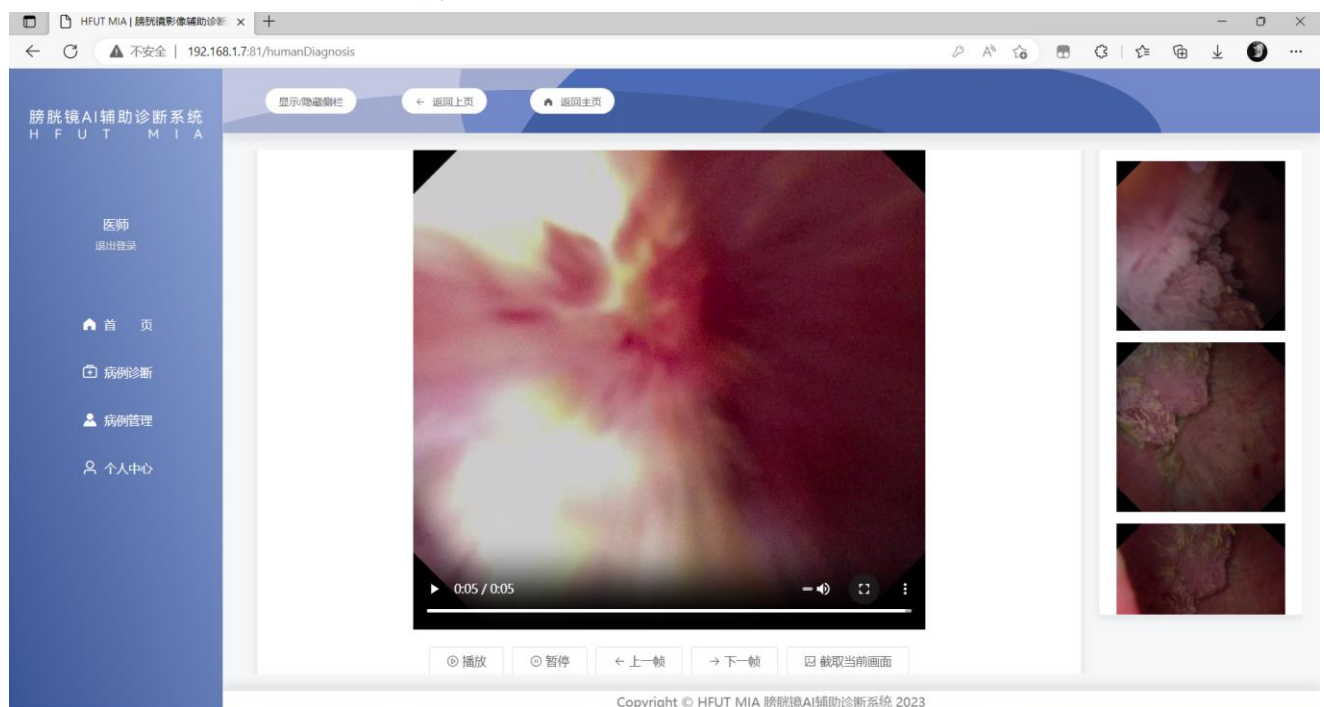


图 4.17 医生端疑似病变区域列表功能展示

医生进行人工诊断后或者 AI 辅诊后，系统会提示该视频中存在的疑似病变区域的个数。



图 4.18 医生端疑似病变区域个数列表功能展示

医生在进行人工诊断或 AI 辅诊后，点击生成膀胱镜检查报告单按钮，进入膀胱镜检查报告单的页面，医生选择右侧的图像移入膀胱镜检查报告单中。



图 4.19 医生端膀胱镜检查报告单功能展示

医生基于疑似病灶图像填写检查所见、检查提示、医师建议。



图 4.20 医生端填写膀胱镜检查报告单功能展示

本系统支持医生端查询自己所诊断的病例，并且可以通过病历号、患者姓名、性别、病例诊断时间等条件进行查询。

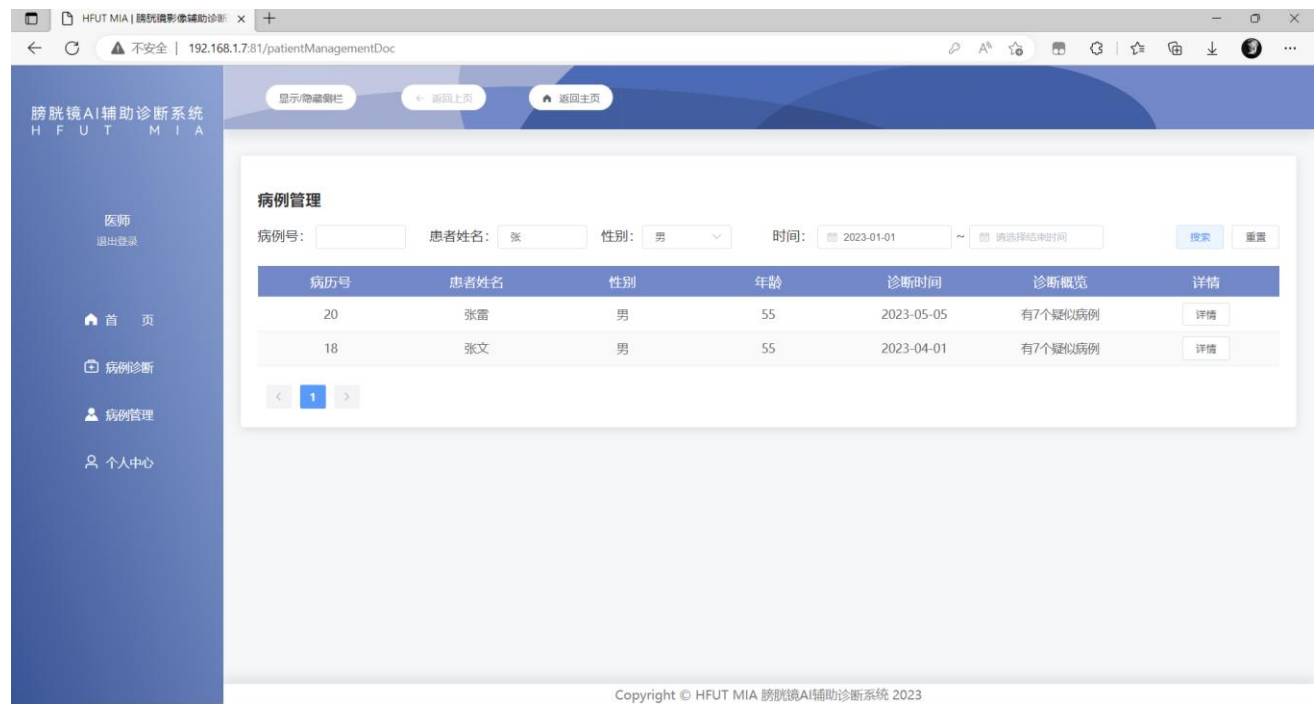


图 4.21 医生端病例管理功能展示

医生可以查看病例的详情信息。



图 4.22 医生端病例详情页面 1



图 4.23 医生端病例详情页面 2

同时医生可以导出膀胱镜检查报告单的 pdf 文件



图 4.24 医生端导出膀胱镜检查报告单功能展示

医生可以对自己的信息进行修改。



图 4.25 医生端个人信息修改功能展示

管理员登录成功后，首先进入首页，首页呈现的是安徽省病例分布图和安徽省各市病例情况列表



图 4.26 管理员端首页功能展示

管理员可以查询系统的可视化信息统计展示，快速了解系统的病例数、医院数、用户数以及前六个月的诊断概览。

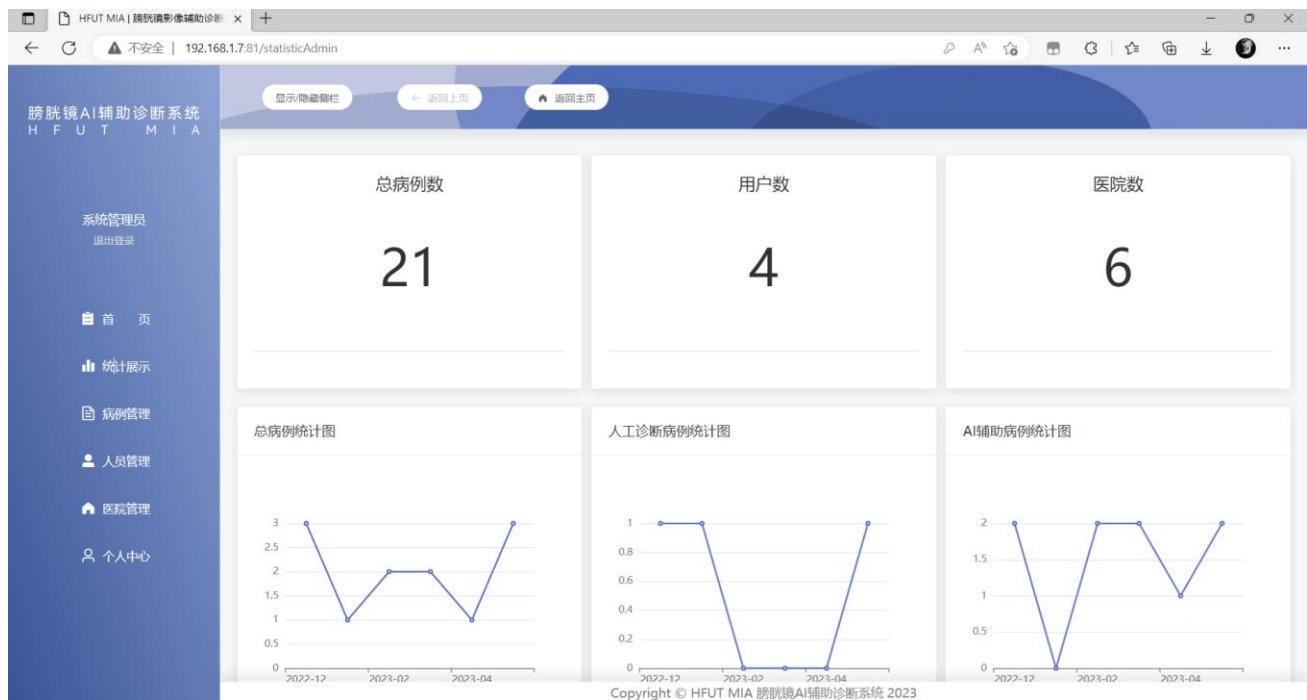


图 4.27 管理员端可视化信息统计功能展示

管理员对系统的病例进行管理。并且可以通过病历号、医师名字、医院名、时间、市和县条件进行查询。



图 4.28 管理员端病例管理功能展示

管理员对人员的信息进行管理。并且可以通过账号、姓名、医院名、状态条件进行查询。

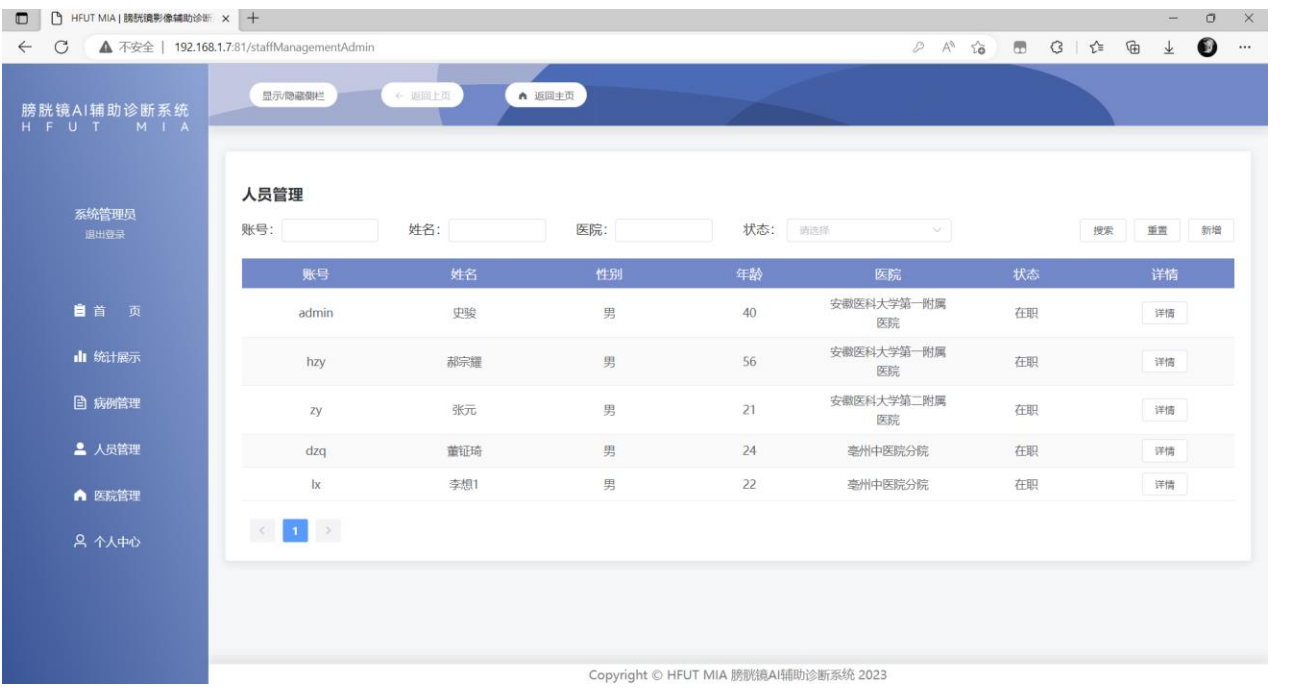


图 4.29 管理员端人员管理功能展示 1

管理员可以查看、修改系统人员的详情信息。



图 4.30 管理员端人员管理功能展示 2

管理员对医院的信息进行管理。并且可以通过账号、姓名、医院名、状态条件进行查询。

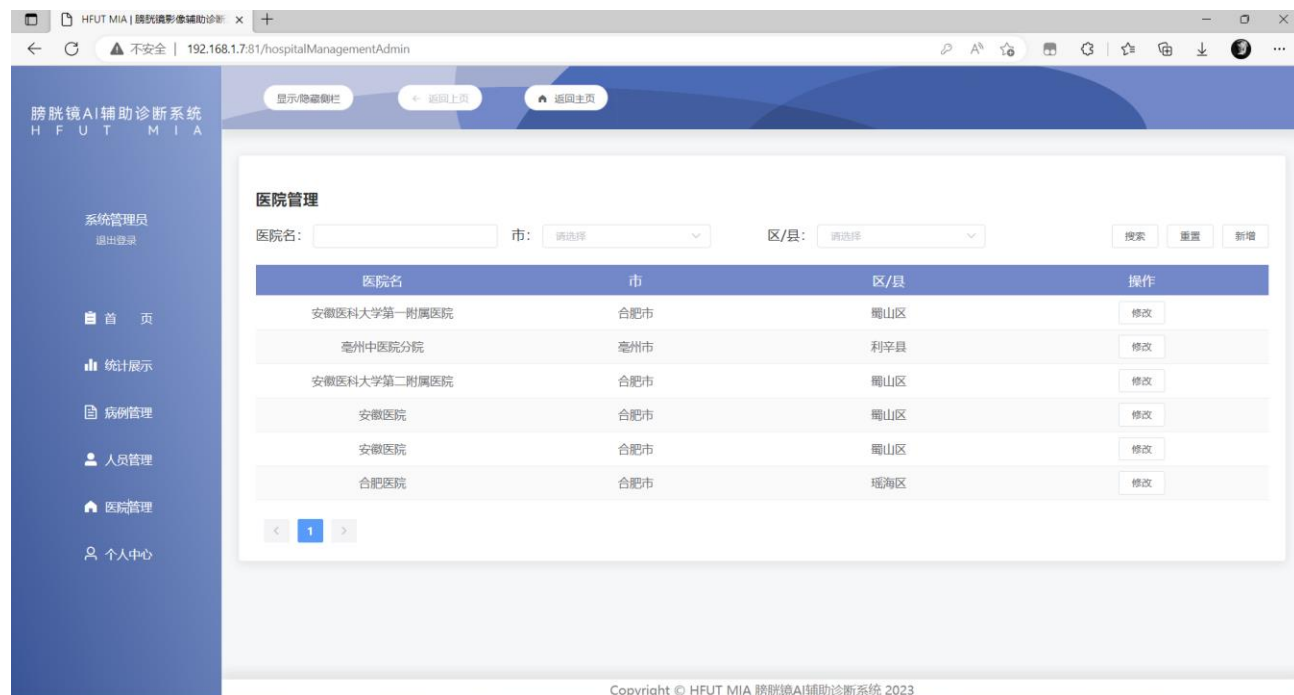


图 4.31 管理员端医院管理功能展示



图 4.32 管理员端个人信息管理功能展示

4.7 关键问题分析及解决方案

在开发系统时，我主要遇到了系统安全性的问题和用来训练深度学习模型的数据集少的问题。

4.7.1 系统安全性问题

在系统开发初期，本系统的鉴权机制和拦截机制较为简单，系统的安全性较低，仅仅通过前端的路由跳转来进行拦截和权限控制，后端的接口是完全暴露在网络中的，整个系统的安全性非常没有保障。

为增加数据的安全性以及后端接口的鉴权控制，我在网络上进行了权限框架的调研，我发现了热门、API 简洁的轻量级权限框架 Sa-Token。

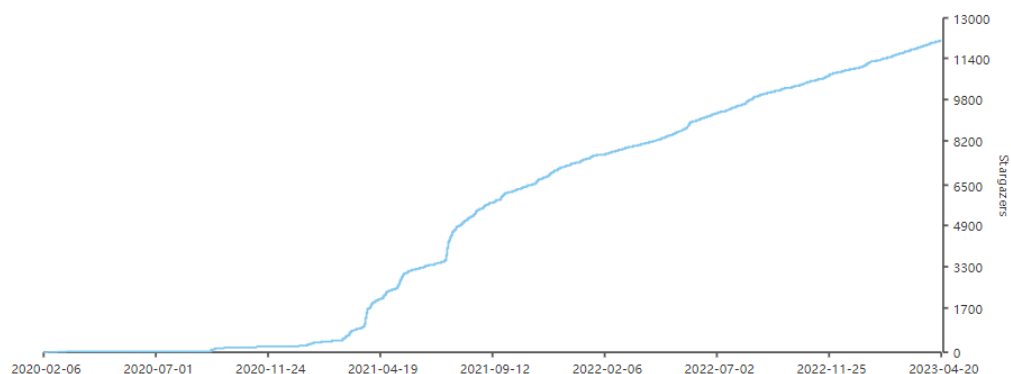


图 4.33 Sa-Token 的 start 趋势

本系统的后端拟采用 Sa-Token 权限控制框架，对访问系统的请求进行拦截，并根据携带的 token 凭证进行权限控制和拦截，同时前端将采用路由守卫，来防止用户的非法路由跳转。

通过 Sa-Token 和路由守卫的安全防护机制来共同保护系统，以此来提高系统的安全性和鲁棒性。

4.7.2 数据集问题

深度学习模型的准确度取决于算法的模型和数据集的质量，但医学数据集的标注是及其昂贵和耗时的，目前社会上没有高质量的膀胱癌数据集，这使得项目的开发陷入困境。

通过网上的调研，我找到了开源的内窥镜数据集，我们基于这个公共的数据集进行了预训练。同时本项目所在的实验室联合合作医院的团队进行数据集的标注。最终通过标注的膀胱镜数据集在预训练的模型上进行训练并得到目标检测的深度学习模型。

4.8 本章小结

本章主要介绍了膀胱镜 AI 辅助诊断系统的设计与实现，并着重介绍了系统框架设计、数据库的概念结构、逻辑结构、数据字典设计、前端页面 UI 设计、后端接口设计、算法模块设计，最终通过图文的方式演示了整个系统。

5 总结与展望

5.1 论文工作总结

本论文主要做了两项工作：一是基于合作医院标注的膀胱镜数据集训练出目标检测模型；二是基于需求分析、场景调研开发了膀胱镜 AI 辅助诊断系统。

首先是目标检测模型部分，本论文对目标检测模型进行了调研，我了解到当前基于深度学习的目标检测模型方法总共分为两类，分别是以 Faster r-cnn 为首的精度高、检测速度慢的 two stage 方法，另一个是以 YOLO 系列为首的速度快精确度稍低的 one stage 方法。本系统对于膀胱癌检测的时间要求较高而对检测的准确率的要求不高，故本论文选择 YOLO V5 目标检测模型作为我们的系统模型。同时本系统也联合了合作医院进行膀胱镜数据集的标注。最终基于标注的膀胱镜数据集训练出目标检测模型。

第二项工作是膀胱镜 AI 辅助诊断系统的实现工作，本论文通过场景分析、病例报告分析、网络调研进行需求分析，并基于需求分析进行详细设计、开发、测试并最终部署上线膀胱镜 AI 辅助诊断系统。通过此系统可以辅助医生进行膀胱镜诊断,提高膀胱癌阅片的准确性和效率。

5.2 未来研究展望

在做完系统的设计、开发、算法研究后，我对未来的研究展望如下：

- 1.阅读计算机视觉、目标检测、医学图像处理相关方面的论文，对模型的结构进行进一步的优化，例如可以将模型的特征提取阶段引入注意力机制来让模型更好地提取膀胱癌的图像特征。

- 2.一个好的模型不仅取决于模型的结构还取决于训练数据集的质量。我将继续联合合作医院的团队继续标注高质量的膀胱镜数据集，从而提升模型检测的准确性。

我也希望自己能够在医学图像处理、计算机视觉的领域中继续深入地学习，并为推动中国人工智能、医学图像处理领域的发展而努力。

参考文献

- [1] 郑荣寿,孙可欣,张思维等.2015 年中国恶性肿瘤流行情况分析[J].中华肿瘤杂志,2019,41(01):19-28.
- [2] 李靖,王斌,张震等.膀胱灌注流量及膀胱充盈程度在膀胱热灌注化疗中的应用[J].实用医学杂志,2020,36(15):2167-2169.
- [3] 韦荣超,吴承耀,张振声等.膀胱镜检查在膀胱癌诊断的研究进展[J].第二军医大学学报,2012,33(11):1257-1259.
- [4] McCulloch W S, Pitts W. A logical calculus of the ideas immanent in nervous activity[J]. The bulletin of mathematical biophysics, 1943, 5: 115-133.
- [5] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[J]. Communications of the ACM, 2017, 60(6): 84-90.
- [6] 许亚岚. 医疗+ AI 就医新体验[J]. 经济, 2017 (17): 56-59.
- [7] 林松. 基于深度学习的内窥镜图像微小出血病灶检测技术研究[D].华中科技大学,2019.DOI:10.27157/d.cnki.ghzku.2019.001563.
- [8] Ali N, Bolenz C, Todenhöfer T, et al. Deep learning-based classification of blue light cystoscopy imaging during transurethral resection of bladder tumors[J]. Scientific reports, 2021, 11(1): 11629.
- [9] Hashemi S M R, Hassanpour H, Kozegar E, et al. Cystoscopy image classification using deep convolutional neural networks[J]. International Journal of Nonlinear Analysis and Applications, 2019, 10(1): 193-215.
- [10] Du Y, Yang R, Chen Z, et al. A deep learning network - assisted bladder tumour recognition under cystoscopy based on Caffe deep learning framework and EasyDL platform[J]. The International Journal of Medical Robotics and Computer Assisted Surgery, 2021, 17(1): 1-8.
- [11] 王雪梅, 吴练练, 于红刚. 内镜精灵结肠镜检查对结肠息肉检出率的影响[J]. 中华消化内镜杂志, 2020, 37(11):816-820.
- [12] Mathew A, Amudha P, Sivakumari S. Deep learning techniques: an overview[J]. Advanced Machine Learning Technologies and Applications: Proceedings of AMLTA 2020, 2021: 599-608.
- [13] DuBois P. MySQL[M]. Pearson Education, 2008.
- [14] Boicea A, Radulescu F, Agapin L I. MongoDB vs Oracle--database comparison[C]//2012 third international conference on emerging intelligent data and web technologies. IEEE, 2012: 330-335.
- [15] Vukotic A, Watt N, Abedrabbo T, et al. Neo4j in action[M]. Shelter Island: Manning, 2015.

-
- [16] Carlson J. Redis in action[M]. Simon and Schuster, 2013.
 - [17] Wu X, Sahoo D, Hoi S C H. Recent advances in deep learning for object detection[J]. Neurocomputing, 2020, 396: 39-64.
 - [18] Walls C. Spring Boot in action[M]. Simon and Schuster, 2015.
 - [19] Walls C. Spring in action[M]. Simon and Schuster, 2022.
 - [20] Vukotic A, Goodwill J. Apache tomcat 7[M]. New York: Apress, 2011.
 - [21] Filipova O. Learning Vue. js 2[M]. Packt Publishing Ltd, 2016.
 - [22] Wirfs-Brock A, Eich B. JavaScript: the first 20 years[J]. Proceedings of the ACM on Programming Languages, 2020, 4(HOPL): 1-189.
 - [23] Ketkar N, Moolayil J, Ketkar N, et al. Introduction to pytorch[J]. Deep Learning with Python: Learn Best Practices of Deep Learning Models with PyTorch, 2021: 27-91.
 - [24] Jia Y, Shelhamer E, Donahue J, et al. Caffe: Convolutional architecture for fast feature embedding[C]//Proceedings of the 22nd ACM international conference on Multimedia. 2014: 675-678.
 - [25] Pang B, Nijkamp E, Wu Y N. Deep learning with tensorflow: A review[J]. Journal of Educational and Behavioral Statistics, 2020, 45(2): 227-248.
 - [26] Gulli A, Pal S. Deep learning with Keras[M]. Packt Publishing Ltd, 2017.
 - [27] Chen T, Li M, Li Y, et al. Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems[J]. arXiv preprint arXiv:1512.01274, 2015.
 - [28] Ghimire D. Comparative study on Python web frameworks: Flask and Django[J]. 2020.
 - [29] Ren S, He K, Girshick R, et al. Faster r-cnn: Towards real-time object detection with region proposal networks[J]. Advances in neural information processing systems, 2015, 28.
 - [30] Redmon J, Divvala S, Girshick R, et al. You only look once: Unified, real-time object detection[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 779-788.

致谢

不经意间，我也到了该毕业的年纪了，大学时光匆匆划过，仿佛昨天我还是大一新生，坐在课堂上听着老师的敦敦教诲，敲着一知半解的代码。回首大学四年，这一路走的并不容易，但好在有我的父母、老师、朋友在一直地鼓励我和帮助我。在此，我想向他们表示感谢。

首先，我想感谢我的父母，感谢父母把还是牙牙学语的我教会成为一个上进、热爱生活的青年，也感谢父母一直以来对我的支持和鼓励。

我非常感谢我的指导老师xxxx老师。xxxxx老师是我大学道路上的灯塔，不仅在毕设论文上给了我很大的帮助，还在学业、竞赛、开发、科研上给予了我很大的指导，带着我做项目、申请专利而且鼓励我、引导我做科研实验、对模型进行优化并最终在医学图像处理领域的顶会 ISBI 上发表论文。同时，我也非常感谢我的朋友，在我生活中遇到困难时，给予我鼓励和帮助。

同时希望自己能够做到 stay hungry stay foolish，也希望自己能够保持对科研的热爱。

2023 年 5 月 17 日