



# 南京大學

## 研究生畢業論文

(申請工程碩士學位)

論文題目 基于变异分析的医疗图像扩增系统设计与实现

作者姓名 王黛薇

学科、专业名称 工程硕士（软件工程领域）

研究方向 软件工程

指导教师 陈振宇 教授，房春荣 助理研究员

2021年5月19日

学 号 : MF1932167  
论文答辩日期 : 2021 年 5 月 20 日  
指 导 教 师 : ( 签 字 )



# **The Design and Implementation of Medical Image Augmentation System Based on Mutation Analysis**

By

**Daiwei Wang**

Supervised by

Professor **Zhenyu Chen**

Assistant Professor **Chunrong Fang**

A Thesis

Submitted to the Software Institute

and the Graduate School

of Nanjing University

in Partial Fulfillment of the Requirements

for the Degree of

**Master of Engineering**

Software Institute

May 2021



# 南京大学研究生毕业论文中文摘要首页用纸

毕业论文题目：基于变异分析的医疗图像扩增系统设计与实现

工程硕士（软件工程领域） 专业 2019 级硕士生姓名：王黛薇

指导教师（姓名、职称）：陈振宇 教授，房春荣 助理研究员

## 摘 要

伴随着医疗图像诊断日益增长的准确性和效率要求，以及硬件算力和算法模型的飞速发展，人工智能技术在医疗图像领域有越来越广泛的应用场景。数据集的数量和质量是影响智能医疗诊断模型性能的关键因素，数量不足且质量欠佳的数据集会对模型的训练和测试结果产生负面影响，进而影响深度学习模型的效果，尤其是在医疗等安全攸关的领域，可能会导致严重的事故和损失，其对于数据扩增的需求更为迫切。但由于医疗图像标准化和病人隐私限制等原因，数据的获取和分享一直受阻。因此，如何有效地产生大量、可靠的医疗图像数据集成为了亟待解决的问题。

本文实现了医疗图像扩增系统，主要提供对于医疗图像的智能扩增方案和多维质量评估。由于医疗图像具有整体相似性和局部多样性特征，系统结合病灶征象和变异分析设计实现了基于图像特征变异和领域语义变异的医疗图像扩增技术，以挖掘其潜在信息，生成丰富且具有多样性的医疗图像扩增数据集。同时提出图像扩增质量多维评估体系，从多个维度采用评估指标对系统扩增质量进行分析和评估，并利用 Echarts 库生成可视化结果，包括数据集整体质量评估分数多维雷达图和图像质量评估分数分布图。其次，系统基于 Spring Boot 框架为用户提供上传管理数据集和执行控制扩增任务的交互接口，同时可通过 Docker 调用图像扩增服务和质量评估服务。此外，向下制定统一的接口标准，以便于实现更多微服务的扩展与兼容。

目前该系统实现了 21 种扩增技术，并面向用户提供差异化的自定义扩增方案。基于 DeepLesion 数据集和 ResNet-50 图像分类模型在数据量、方法设置、扩增效果和扩增质量四个维度进行实验，结果表明通过医疗图像扩增系统产生的数据集一方面可以缓解由于收集困难带来的医疗图像数据量不足的问题，同时可在一定程度上保证扩增数据集的质量。另一方面，则能够提升医疗影像辅助诊断模型的效果，所有实验中模型的准确率平均有约 5% 的提升。

关键词：图像扩增，变异分析，病灶识别，多维评估



## 南京大学研究生毕业论文英文摘要首页用纸

THESIS: The Design and Implementation of Medical Image Augmentation System Based on Mutation Analysis

SPECIALIZATION: Software Engineering

POSTGRADUATE: Daiwei Wang

MENTOR: Professor **Zhenyu Chen**

Assistant Professor **Chunrong Fang**

### **Abstract**

With the increasing accuracy and efficiency requirements of medical image diagnosis, as well as the rapid development of hardware computing power and algorithm model, artificial intelligence technology has been applied more and more widely in the field of medical image. Meanwhile, the number and quality of datasets are the key factors that affect the performance of the intelligent medical diagnostic model. Insufficient and poor quality datasets can negatively impact the training and testing results of the model, which affect the effectiveness of the deep learning model, especially in security-critical areas such as medical field. It may cause serious accidents and losses, so the demand for data augmentation is even more compelling. However, due to data standardization and privacy restriction, the acquisition and sharing of medical image data have been hindered. Therefore, how to effectively generate a large number of reliable medical image dataset has been an urgent problem to be solved.

This thesis implements the medical image augmentation system, which mainly provides the intelligent augmentation solutions and multidimensional quality evaluation for medical image. Because of the overall similarity and partial diversity of medical images, the system designs and implements medical image augmentation technology based on image feature mutation and domain semantic mutation by combining lesion signs and mutation analysis, so as to obtain its potential information and generate rich and diverse medical image datasets. At the same time, we propose a multi-dimensional evaluation system of image augmentation quality. The evaluation indexes are used to analyze and evaluate the augmentation effect of the system from multiple dimensions,

and the visualization results are generated by using the Echarts library, including the multi-dimensional radar map of the overall quality evaluation score and the distribution map of the image quality evaluation score. Based on the Spring Boot framework, the system provides users with an interactive interface to manage datasets and augmentation tasks, while calling image augmentation service and quality evaluation service through Docker. In addition, unified interface standards are developed downward to facilitate the expansion and compatibility of more microservices.

At present, the system implements 21 augmentation methods and provides users with differentiated custom augmentation schemes. Based on the DeepLesion dataset and ResNet-50 image classification model, experiments are carried out in the four dimensions of data volume, method setting, augmentation effect and augmentation quality. The results indicate that the dataset generated by the intelligent medical image augmentation system can alleviate the difficulty of data collection on the one hand, and the quality of the augmented dataset can be guaranteed to a certain extent. On the other hand, it can improve the effect of the medical image assisted diagnostic model. In all experiments, the accuracy of the model has an average improvement of about 5%.

**Keywords:** Image Augmentation, Mutation Analysis, Lesion Recognition, Multidimensional Evaluation

# 目录

表 目 录 .....	ix
图 目 录 .....	xii
<b>第一章 引言</b> .....	<b>1</b>
1.1 研究背景与意义 .....	1
1.2 国内外研究现状 .....	2
1.2.1 图像扩增技术研究现状 .....	2
1.2.2 图像扩增工具研究现状 .....	4
1.2.3 病灶识别技术研究现状 .....	5
1.3 本文主要工作 .....	5
1.4 本文组织结构 .....	6
<b>第二章 相关技术综述</b> .....	<b>9</b>
2.1 医疗领域相关概念 .....	9
2.1.1 医疗图像 .....	9
2.1.2 病灶 .....	10
2.1.3 征象 .....	10
2.2 前端相关技术 .....	11
2.2.1 Element UI 框架 .....	11
2.2.2 Echarts 库 .....	11
2.2.3 Vue 生态系统 .....	11
2.3 后端相关技术 .....	13
2.3.1 Spring Boot 框架 .....	13
2.3.2 Docker .....	13
2.4 本章小结 .....	14

<b>第三章 需求分析与概要设计</b> .....	<b>15</b>
3.1 系统整体概述 .....	15
3.2 需求分析 .....	16
3.2.1 功能性需求 .....	16
3.2.2 非功能性需求 .....	18
3.2.3 系统用例图 .....	18
3.2.4 系统用例描述 .....	18
3.3 系统概要设计 .....	23
3.3.1 系统架构设计 .....	23
3.3.2 系统 4+1 视图 .....	25
3.4 扩增任务执行控制模块 .....	29
3.4.1 架构设计 .....	29
3.4.2 流程设计 .....	30
3.5 医疗图像变异扩增模块 .....	31
3.5.1 架构设计 .....	31
3.5.2 流程设计 .....	32
3.6 扩增质量多维评估模块 .....	33
3.6.1 架构设计 .....	33
3.6.2 流程设计 .....	34
3.6.3 评估指标设计 .....	35
3.7 本章小结 .....	38
<b>第四章 系统详细设计与具体实现</b> .....	<b>39</b>
4.1 扩增任务执行控制模块 .....	39
4.2 医疗图像变异扩增模块 .....	44
4.2.1 医疗图像病灶识别子模块 .....	48
4.2.2 图像特征变异扩增子模块 .....	50
4.2.3 领域语义变异扩增子模块 .....	55
4.3 扩增质量多维评估模块 .....	58
4.4 本章小结 .....	60

<b>第五章 系统测试与实验分析</b> .....	<b>63</b>
5.1 系统测试 .....	63
5.1.1 测试目标及测试环境 .....	63
5.1.2 功能测试 .....	63
5.2 实验分析 .....	66
5.2.1 实验对象 .....	67
5.2.2 实验设计及结果分析 .....	68
5.2.3 案例分析 .....	71
5.3 本章小结 .....	72
<b>第六章 总结与展望</b> .....	<b>73</b>
6.1 总结 .....	73
6.2 展望 .....	74
<b>参考文献</b> .....	<b>75</b>
<b>简历与科研成果</b> .....	<b>81</b>
<b>致谢</b> .....	<b>83</b>



## 表 目 录

3.1	系统功能性需求列表	17
3.2	系统非功能性需求列表	17
3.3	系统用例列表	19
3.4	上传数据集用例描述	19
3.5	执行扩增任务用例描述	20
3.6	查看扩增任务信息用例描述	21
3.7	查看扩增任务报告用例描述	22
3.8	下载扩增数据集用例描述	22
3.9	查看评估结果用例描述	23
4.1	扩增任务执行控制模块主要类	40
4.2	Job 类主要字段列表	42
4.3	Worker 类主要字段列表	42
4.4	DataSet 类主要字段列表	43
4.5	Pipeline 类属性列表	46
4.6	图像特征变异扩增方法列表	51
4.7	领域语义变异扩增方法列表	56
5.1	测试环境参数列表	63
5.2	上传数据集测试用例	64
5.3	查看数据集列表测试用例	65
5.4	查看扩增结果相关测试用例	65
5.5	执行扩增任务测试用例	66
5.6	查看扩增任务相关测试用例	67
5.7	扩增数量提升实验结果	68
5.8	扩增方法效果实验结果	69
5.9	扩增工具效果对比实验结果	70
5.10	扩增工具质量评估实验结果	70



## 插图

3.1	系统整体模块图	15
3.2	系统用例图	18
3.3	系统架构设计图	24
3.4	系统逻辑视图	25
3.5	系统开发视图	26
3.6	系统进程视图	27
3.7	系统物理视图	28
3.8	扩增任务执行控制模块架构图	29
3.9	扩增任务执行控制模块流程图	30
3.10	医疗图像变异扩增模块架构图	31
3.11	医疗图像变异扩增模块流程图	32
3.12	扩增质量多维评估架构图	33
3.13	扩增质量多维评估流程图	34
4.1	扩增任务执行控制模块时序图	39
4.2	扩增任务执行控制模块类图	41
4.3	执行扩增任务核心代码	43
4.4	获取扩增结果核心代码	44
4.5	医疗图像变异扩增模块时序图	45
4.6	医疗图像变异扩增模块类图	46
4.7	扩增工具调用核心代码	47
4.8	方法配置文件示例代码	48
4.9	PANet 模型算法结构图	49
4.10	图像特征变异扩增方法示例	52
4.11	领域语义变异扩增方法示例	56
4.12	病灶复制方法代码	57
4.13	扩增质量多维评估模块时序图	59

4.14 扩增质量多维评估模块类图.....	59
4.15 扩增质量多维评估模块页面.....	60
5.1 DeepLesion 数据集示例图（左：肺部；右：肾脏） .....	68
5.2 新建扩增任务——完善任务信息页面 .....	71
5.3 新建扩增任务——添加扩增方法页面 .....	72
5.4 查看扩增任务报告页面 .....	72

## 第一章 引言

### 1.1 研究背景与意义

近年来，由于社会需求和技术发展现状，医疗领域对医疗影像诊断的检出率和准确性提出了更高的要求。由于我国人口总量不断增加，同时伴随着人口老龄化趋势加速，导致社会医疗需求不断增大。此外，人均 GDP 的提高带来了消费升级，居民的生活水平提高，使得人们对于优质便捷的医疗服务的需求也日益增长。据统计，中国的医疗影像数据年增长率已达到 30% 左右，但放射科医生的年增长率仅为 4.1%，医师阅片的负担大大提高，因此，仅依靠传统医疗手段无法满足患者对诊断效率和准确率日益增长的要求。医疗领域现状以及飞速发展的硬件算力和算法模型，促使了人工智能技术在医疗影像领域的应用和落地。

“人工智能 + 医疗影像”主要应用于医学影像诊断。医疗影像辅助诊断结合人工智能技术与医疗影像数据在医疗影像辅助诊断设备上协助医生完成快速读片、诊断等繁重的工作，减轻其工作负担，同时减少实际诊断中的漏检误检。具体来说，医学影像辅助诊断通过深度学习技术识别、分析和处理非结构化的影像数据，从中提取有效特征，然后将医疗影像数据和诊断先验知识作为训练集输入深度学习模型进行训练，通过不断训练、调参和验证对医疗影像数据进行智能推理，最终输出准确有效的病症诊断结果。对于智能诊断而言，算法是智能诊断的技术核心，而医疗影像则是智能诊断的基础。由于医疗诊断分析任务需要较高的准确率和效率，因此需要大量数据进行模型训练。但病人的隐私、公司间的利益冲突以及电子病历的缺乏等问题，阻碍着医疗领域的信息和数据共享，使得医疗影像数据的收集尤为困难。此外，由于医疗领域的专业性较强，需要大量专业人力和时间成本对医学图像进行标注，尤其是在安全攸关的医疗领域，需要通过标注数据集的大量训练才能达到较高的准确率和精确度，但这是一个非常消耗时间和人力资源的过程。与传统软件不同，人工智能系统的决策逻辑是从训练样本中学习得到的，因此一旦数据量不足且数据质量欠佳，缺少有效特征，就会对智能诊断模型造成过拟合、无法收敛、容易陷入局部最优值等问题，这些模型问题可能会使医疗诊断发生错误，甚至造成严重的医疗事故。

目前一种解决数据量不足的主流方法是让有限的图像数据通过特定技术产生更多的等价数据，即图像扩增技术。研究表明，利用常规的图像扩增技术可以在一定程度上增加医疗图像的数量。但由于医疗领域图像的特殊性，由通用

的扩增技术生成新图像的质量并不能得到保证。同时，与自然图像不同，由于人体组织本身相似度较高，但在相似背景组织中的细微变化又有可能代表着某种病变，因此医学图像同时具有整体相似性和局部多样性的特点，而常规扩增技术无法保证上述特征在新图像中的表现。另外，由于疾病的复杂性和多样性，不同疾病的病理及其相应的影像学表现也非常复杂，通用的图像扩增技术在医疗领域并不完全适用。

因此，本文实现了医疗图像扩增系统，并提出了基于病灶征象和变异分析的医疗图像扩增技术。通过从原始医疗图像中提取病灶区域以及征象表现，结合医学图像特征，分别从图像特征变异和领域语义变异两方面研究扩增技术，并通过多种评估维度评价扩增质量。生成的扩增数据集一方面可以缓解由于收集困难带来的医疗图像数据量不足的问题，另一方面则可以分别从评估和优化两个方面提升医疗影像辅助诊断模型的效果。在模型评估层面，将扩增后得到的医疗数据集作为测试数据集，通过对模型进行大量数据的测试，从多病理类型、多阶段表现等方面对医疗诊断模型的性能进行检测，同时对其鲁棒性进行评估。对于低质量畸变的影像样本，检测深度学习模型能否准确地拒绝畸变样本，提升鲁棒性。在模型优化层面，通过医疗图像数据扩增技术扩展训练数据集，并根据图像质量从而提高模型的精度和泛化能力。

## 1.2 国内外研究现状

人工智能技术的不断研究创新同时带动了相关技术在医疗领域的应用和快速发展。其中，医疗影像作为诊疗数据的重要组成部分，包括超声、放射、病理、内镜等影像表现形式。随着深度学习技术的发展，人工智能在医学影像的诸多领域实现了突破。

### 1.2.1 图像扩增技术研究现状

在扩增策略设计方面，Perez 等人 [1] 在皮肤黑色素瘤病变分类模型上对 13 种数据扩增方案及组合进行了实验，结果表明图像扩增在模型训练和测试中占据重要部分，同时与直接利用新图像相比对模型有更高的性能提升。Luis 等人 [2] 针对图像分类问题，比较分别采用传统数据扩增方法和 GAN [3] 完成数据扩增的效果。其实验结果表明，GAN 和神经网络扩增方式的表现并不比传统的扩增效果好很多，且消耗的计算时间几乎是传统方法的 3 倍以上。DeVries 和 Taylor [4] 则提出了不在输入空间中而是在学习的特征空间中进行数据扩增处理，即在特征空间通过不区分特定领域的内插、外推和加噪声方法进行统一扩增。这种方法虽然能够排除具体数据集中由不同分布产生的干扰，应用于各种不同领域数

数据集，但不具备针对医疗领域的特化操作，且实现也远不如直接对输入端进行扩增方便。相似地，Lim 等人 [5] 也针对数据集的潜在分布进行扩增。通过在数据集的潜在空间上施加一个分布，有选择的对潜在向量进行采样和解码生成合成样本。而 Cubuk 等人 [6] 不只限于一种单独的扩增策略，而是通过在包含多个数据扩增策略的集合中进行搜索，基于目标数据集对特定策略的效果进行评估。该研究通过对策略空间进行离散搜索来寻找最佳数据扩增策略，并将其应用于目标数据集，以提高模型的性能及准确率。对于只能获取到极少标记数据的情况，Zhang 等人 [7] 提出了一种深度对抗数据扩增技术来解决其分类问题，即将数据扩增过程整合到生成对抗网络中，并为其中的鉴别器提出一种新的损失函数以达到数据扩增的目标。

在扩增方法设计方面，Zhong 等人 [8] 设计了随机擦除方法，即随机选择图像中的矩形区域，通过产生随机像素值来覆盖其原始像素。该方法通过对图像局部进行不同程度的擦除，一方面可以减少模型的过拟合风险，另一方面则能提高模型的鲁棒性，本系统也将该方法引入图像特征变异和领域语义变异模块。此外，Inoue [9] 提出 SamplePairing 技术，即通过对应像素取平均值的方式将训练集中的任意两张图片进行叠加，来合成一个新的样本，以达到扩充训练集大小的目的。这种方法对类似医疗图像等训练数据量有限的任务非常有效。Zhang 等人 [10] 提出了 mixup，即通过线性插值构建训练数据的领域值，以遵循邻域风险最小化原则。实验结果表明该方法提高了模型的泛化能力和面向对抗样本的鲁棒性，同时能够降低模型对已损坏标签样本的记忆，并增强训练 GAN 时的稳定性。

由于医疗领域图像与普通图像的差异，对医疗图像的扩增策略和方法研究都具有一定特殊性。在针对医疗图像的数据扩增策略研究中，有监督数据扩增方法基于原始数据采用预设的数据变换规则进行扩增。Ornek 等人 [11] 通过使用新生儿热图像对旋转、镜像、缩放、剪切、颜色变化、锐化、模糊等传统数据扩增方法进行了比较，所得结果表明，简单、低成本的基本数据扩增方法可以提高分类网络性能。Hussain 等人 [12] 则基于乳腺肿块图像和非肿块图像分类任务对多种扩增策略进行了比较，实验表明，扩增训练集保留原始医学图像属性的程度决定了模型的性能，即对原始医疗图像的改变越小，应用于模型时的性能表现越好。而对于无监督数据扩增方法，Zhao 等人 [13] 提出了一种通过从图像中学习转换模型并基于该转换和原始标注样本生成新样本的图像扩增方法。转换模型中包括对空间变形和强度变换特征的学习，能够模拟组合医疗图像中非线性变形、成像强度等复杂的变化。Shin 等人 [14] 基于两个公开的脑磁共振图

像数据集 ADNI<sup>1</sup>和 BRATS<sup>2</sup>训练生成对抗网络来合成具有脑肿瘤的患病磁共振图像，并展示了该方法平衡数据集和作为匿名化工具的优势。Maayan 等人 [15] 也引入了生成对抗网络，基于肝脏 CT 图像数据集来合成高质量的肝脏病变 ROI 图像，通过添加合成扩增的数据，可以在基础扩增数据基础上提高模型性能。

由上述技术研究现状可以看出，目前在有监督和无监督扩增方面均有较深入的方法和实验，但对于需要在系统中实时使用的技术而言，生成对抗网络等无监督图像扩增方法耗费时间长，对于网络的训练仍需要大量数据，并且由于增加模型输出图像大小可能会导致训练不稳定和不收敛，难以从中获取高分辨率输出，因此不太具有实用性。

### 1.2.2 图像扩增工具研究现状

目前主流的图像扩增工具包括 `imgaug`、`Albumentations` 和 `Augmentor` 等工具，通过集成多种扩增方法对特定应用场景提供解决方案。这些图像扩增工具均面向通用领域，并不提供针对医疗图像的特定扩增方法。在本文第五章的实验分析部分也将对比本系统和上述主流图像扩增工具在医疗图像数据集上的表现。

`imgaug`<sup>3</sup>是基于 `python` 的图像扩增库，它通过输入图片产生新图片的集合，并实现了多种不同的方法通过随机顺序组合策略来改变图片以达到数据扩增的目的，同时支持关键点和 `Bounding box` 同步变换，能够应用于图像分类、图像分割、关键点定位、目标检测等任务。在实际使用中，通过 `Sequential` 创建扩增实例，并在参数中定义扩增方法序列，最后应用 `augment_image` 方法实现扩增流程。

`Albumentations`[16] 是一个基于 `OpenCV` 的快速训练数据增强库，拥有简单且强大的可以用于分类、分割、检测等任务的接口，易于定制且易扩展。具有较快的处理速度，并且支持 `Pytorch`。`Albumentations` 可以对图像进行逐像素的转换和基于空间的转换操作，包括模糊、下采样、高斯造点、动态模糊、`RGB` 转换、随机雾化、裁剪、翻转、随机裁剪等。其中可通过 `Compose` 类对扩增方法进行组合，也可以采用 `OneOf` 类对一组操作进行随机选择。

`Augmentor`[17] 是管道化渲染的图像增强库，可提供用于生成机器学习问题的图像数据的操作。其中包含 `Rotate`、`Crop` 等用于图像扩增的类，并在类中实现具体的扩增方法。数据增强通常是一个多阶段过程，因此 `Augmentor` 采用基

<sup>1</sup>[www.adni-info.org](http://www.adni-info.org)

<sup>2</sup>[braintumorsegmentation.org](http://braintumorsegmentation.org)

<sup>3</sup><https://imgaug.readthedocs.io/en/latest/>

于管道的处理方法，通过依次添加操作形成最终的操作管道。首先将图像送入管道中，管道中的操作再依次按照设定的概率随机地作用到图片上，最后生成新的图片并进行保存。

### 1.2.3 病灶识别技术研究现状

对病灶的识别是诊断中的重要环节，同时也是智能医疗影像的重要研究方向。结合人工智能技术对病灶进行识别定位，可以提高智能医疗诊断系统的效率和准确性，同时减轻医生的阅片负担，降低其工作量。Lo 等人 [18] 在 1995 年提出了最早的病灶识别模型，即通过 CNN 对 X 线图像中的肺结节进行检测。此后大多数的病灶识别模型都是首先采用 CNN 对像素进行分类，再通过对分类结果进行处理来获取病灶区域。如 Barbu 等人 [19] 通过改变 CNN 网络中的损失函数进行实验，以提升模型对结节的检测效果。同时 Roth 等人 [20] 也提出一种将提取出的器官内部特征和边界特征通过随机森林进行空间聚集整合的整体学习方法，并将其应用于具有高度解剖变异性的胰腺图像。此外，Varghese 等人 [21] 提出采用生成对抗网络从 MRI 中检测脑部病变的半监督技术，其中在生成器网络中建模正常病理图像块的分布，并在判别器网络中计算测试图像块的后验分布，并通过二值化后验概率图来完成病变区域的描绘。在此基础上，Chen 等 [22] 提出了对抗自编码器模型，实现了更广泛病理图像的病灶检测。

本系统中基于 PANet[23] 实现病灶区域识别功能。PANet 路径聚合网络可以增强在医疗图像中重要的低维特征定位信息流，建立底层特征和高层特征之间的信息路径，从而增强整个特征层次架构。此外，对各个维度提取出的特征进行融合，让每个特征层次的有用信息直接传播到后续的候选区域子网络，以提高模型对病灶位置识别的准确率。

## 1.3 本文主要工作

本文主要基于医疗图像的特殊性，设计实现了医疗图像扩增系统，并提出了结合病灶征象和变异分析的扩增技术，分别从图像特征变异和领域语义变异两个方面设计实现了多种扩增技术，同时采用多维评估指标对扩增质量进行评价。工作主要体现在以下几个方面：

(1) 设计并提出了基于病灶征象的扩增关联规则。首先从医疗图像中识别病灶区域，通过在图像中对识别出的病灶进行多样化处理，结合医疗领域先验知识从同一个体的病情发展阶段和不同个体的病情表现形式两个维度对肺部医疗图像研究扩增规则，以模拟现实中病人患病时可能产生的影像学表现。

(2) 设计并实现了扩增任务执行控制模块。主要负责数据集和扩增任务的创建管理以及扩增任务执行控制的业务流程。通过前后端交互调用，完成前端渲染与扩增服务和评估服务之间的业务逻辑和数据传输，并作为系统的控制枢纽为不同的微服务提供统一的接口规范。

(3) 设计并实现了医疗图像变异扩增模块。主要集成病灶区域子模块、图像特征变异扩增子模块和领域语义变异扩增子模块，通过创建 **Docker** 容器完成扩增任务的实际执行流程，并根据传输配置参数调用具体的扩增方法，最后回传任务状态信息并存储扩增结果。

(3) 设计并实现了医疗图像病灶识别子模块。从医疗图像中提取病灶区域定位等特征。由于医疗图像的特殊性，病灶细节和特征更多依赖于图像纹理、像素值等低层特征，因此对于病灶的识别定位需要更多地考虑图像低层特征，但同时也需要结合高层特征中的轮廓、类别等信息。

(4) 设计并实现了图像特征变异扩增子模块。通过对图像固有属性特征进行变异分析，基于医疗图像的整体相似性设计扩增技术并进行实现。通过该模块，可以产生具有整体变化特征的扩增图像，并在较大程度上保留了医疗图像的完整度。

(5) 设计并实现了领域语义变异扩增子模块。基于医疗领域的专业先验知识，对病灶识别子模块中提取的病灶区域特征进行变异分析，并结合基于病灶征象的扩增关联规则对多种病灶征象进行组合、变化等处理，获得具有局部发展多样性和整体表现多样性的扩增图像。

(6) 设计并实现了扩增质量多维评估模块。通过多个维度的评估指标对扩增产生的数据集进行质量评估，并采用 **Echarts** 库从整体表现和具体分布两方面形成可视化评估报告。

## 1.4 本文组织结构

本文的组织结构如下：

第一章引言，主要介绍了社会发展带来了人工智能在医疗影像应用的发展趋势，由此产生了对医疗图像数据集的数量和质量需求，引出医疗图像扩增系统的研发目的以及意义。同时介绍了图像扩增技术及工具的国内外研究现状。

第二章相关技术介绍，主要介绍医疗专业领域的相关技术术语，以及系统开发中使用的相关技术和算法，包括医疗图像、病灶、前后端框架、**Docker** 等。

第三章需求分析与概要设计，主要涉及到系统需求设计、整体架构设计以及各个功能模块的初步设计。其中需求设计包括功能性需求、非功能性需求以

及对应功能需求的用例设计，整体架构设计包括系统总体架构图以及 4+1 视图，对各个功能模块的初步设计则对应每个模块的架构图和流程图。

第四章系统详细设计与具体实现，主要在系统需求分析与概要设计的基础上，对扩增任务执行控制模块、医疗图像变异扩增模块和扩增质量多维评估模块及医疗图像变异扩增模块中各个子模块的内部架构及业务逻辑进行详细阐述。

第五章系统测试与实验分析，主要根据系统功能需求设计功能测试用例，并在测试环境上对各个功能点进行测试，同时通过多个控制变量实验对系统效果进行验证和评估，以测试系统的正确性、有效性和健壮性。

第六章总结与展望，主要对整个系统进行总结，并根据研究和开发中遇到的问题提出可能的改进方向。



## 第二章 相关技术综述

### 2.1 医疗领域相关概念

#### 2.1.1 医疗图像

从表现方式来说，医学图像是解剖学区域的内部结构或功能的表示。它是像素阵列的形式，是由采样或重建过程产生的离散表示，其将数值映射到空间的位置，用于描述解剖结构或功能的细节表达。医疗图像作为医疗大数据中举足轻重的一部分，在疾病诊断、治疗方案的制定和疾病预后中具有重要参考价值 [24]，通过挖掘其包含的大量信息可以得到反映人类健康和疾病的信息。

从采集方式来说，医疗图像是指为了医疗或医学研究，对人体或人体某部分，以非侵入方式取得组织内部结构和功能状态的技术与处理过程。医疗图像起源于 19 世纪，德国物理学家伦琴发现 X 射线后将其应用于医疗图像采集研究。在此之后医疗影像产业飞速发展，陆续出现了 X 射线成像、核磁共振成像、超声波成像、核医学成像等技术。目前医学影像设备的市场规模约占我国整个医疗器械行业的 16%，已经成为我国医疗器械市场规模最大的子行业。同时在今后的医学检测手段中，医学影像数据占据了 90% 的医疗信息，是疾病筛查和诊治最主要的信息来源，也是辅助临床疾病诊疗的重要手段<sup>1</sup>。目前医学影像的设备主要包括数字 X 线摄影 (DR)、计算机断层扫描 (CT)、核磁共振 (MRI) 和 PET、PET-CT、PET-MR 等，以及小型设备超声和内镜等。

数字 X 线摄影 (DR) 是将透过人体的 X 射线信息通过影像增强管进行传输，再由视频摄像机采集后转换为数字信号。由于人体中的不同器官组织在体积和密度上存在差异，对 X 射线通过时有不同强度的吸收，因此最终将依据通过的 X 射线量形成像素值间存在差异的灰度图像。但传统的 DR 只能在二维成像，无法提供关于人体器官组织或其他部位内部结构的具体信息，因此也无法提供被检测部位的三维结构信息。同时在 X 射线检测时会产生大量散射，也会丢失部分成像信息。此外，人体中一些不同的组织在一定程度上对 X 射线拥有近似吸收能力，因此无法通过 DR 扫描对这些不同的器官组织进行区分 [25]。

与 X 线检查不同，计算机断层扫描 (CT) 使用从不同角度进行的许多 X 射线测量值的计算机处理组合来生成特定扫描区域的横截面图像，使用户无需切割即可看到物体内部。照射一次 CT，相当于在同一部位不同层次进行 X 射线扫

---

<sup>1</sup><https://www.hsmap.com/detail/1/393>

描，因为人体各个组织的密度存在差异，当 X 射线经过时对其吸收率不同，再经计算机处理获得密度差别图像。通过 CT 可以获取普通 X 射线技术无法看到的部位，从而发现微小的病变。一方面，CT 提高了对人体在维度和细粒度方面的检查区间，另一方面也提高了对于病症诊断的检出率和准确率。但由于其成像原理，CT 无法分辨内部密度差异较小的软组织，只适用于对密度差异较大的器官组织进行扫描。

核磁共振 (MRI) 是诊断和评估疾病的重要工具 [26]。其基本成像原理是将人体置于特殊的磁场中，用无线电射频脉冲激发人体内氢原子核，引起氢原子核共振，并吸收能量 [27]。在停止射频脉冲后，通过采集人体内水分、有机分子中大量存在的氢核产生磁共振现象后放出的射频信号对人体组织进行成像。由于不同组织的氢核含量不同，其放出的射频信号有高低之分，即通过信号密度进行成像。MRI 除了能够产生与 CT 图像类似的横截面，它还能从矢状面、冠状面以及各种斜面对人体组织进行刻画，同时在成像过程中不会有伪影的出现。但其在氢核缺乏或含量较少的组织如骨骼、钙化部位等不能成像，并且其空间分辨率低于 CT，因此，CT 和 MRI 是相辅相成的两种成像技术，在实际应用中医生通常需要结合两者结果对疾病进行更准确的诊断。

### 2.1.2 病灶

病灶是指人体发生病变的部位，即包含病原微生物的局部病变组织。病灶的范围较广，主要包括良性病灶和恶性病灶两种类型，比如癌细胞就属于恶性病灶。病灶除了对病变的部位造成伤害，还可能导致其它器官发生病变，这在医疗领域被称为病灶感染。由于病灶里包含病原体微生物，当身体免疫力下降时，就可能感染周围的健康组织。其主要形式是慢性炎症，既可以是静止的，也可以是活动的。病灶可以出现在身体的任何部位，并且可由人体中任何组织器官的病变产生，这些疾病的病灶是具有活动性的，能够侵犯其它原本健康的组织。要治疗疾病，必须要确定病灶并将其完全去除才可以治愈。

### 2.1.3 征象

征象表示在病理或身体检查中发现可用于判断疾病发展状况的迹象及指标，其在影像学中则表现为图像中某些能够提供患病信息的区域特征，即病灶在影像学上的表现特征。如肺结节的征象包括肺结节形态学特征中的圆形肿块征、分叶征、棘状突起、结节征、空泡征和空洞征，病灶密度特征中的纯磨玻璃结节、混合磨玻璃结节和实性结节，以及病灶表面特征中的毛刺、分叶和晕征，这些征象都属于医疗图像上的肺结节病灶表现特征。征象能够为医生提供关于病灶的

类型等细节信息，有助于病症诊断。

## 2.2 前端相关技术

### 2.2.1 Element UI 框架

Element-UI 框架是一款以 Vue 2.0 为基础的前端开发组件库。饿了么前端团队在设计和开发其中的组件时，遵循了一致性、反馈性、效率性和可控性的设计原则，使组件间的结构和表现以及组件与现实生活中的流程、逻辑保持一致，具有简洁、清晰、明确、美观的操作流程、语言表达和设计风格，让用户在操作后可以清晰地获取到组件变化的反馈，能够快速理解并自由决策。Element-UI 具有丰富的组件类型，包含基础组件、表单类、数据类、提示类、导航类和其它类共 56 个组件。此外，它还支持自定义主题，可以在线修改全局和组件颜色、样式等属性，并进行实时预览，同时还可以将自定义的样式进行打包以供后续下载。除了组件，Element-UI 还在开发、测试、构建、部署以及持续集成等工程化流程方面制定了一套完整的解决方案，使得前端能够便捷快速地进行整体框架和工程流程的构建。

### 2.2.2 Echarts 库

Echarts (Enterprise Charts) 是基于 JavaScript 的图表可视化库，支持 Web 端和移动端开发，并针对移动端进行了细致的优化，同时可兼容 IE、Chrome、Safari 等大部分浏览器。Echarts 通过增量渲染技术，能够实现千万级的商业数据图表。其底层依赖于轻量级的 Canvas 类库 ZRender，支持包括折线图、雷达图、饼图、关系图、漏斗图等多种图表类型，并且可以进行图表间的任意组合，为用户提供多样的个性化定制图表方案。同时提供了对图表进行数据筛选、细节展示等深度交互功能，实现了基于数据变化的动态图表，大大提高了用户体验，使用户能够对数据进行挖掘和整合。

### 2.2.3 Vue 生态系统

Vue 是一套构建 UI 的渐进式 JavaScript 框架，并且支持自底向上逐层应用。一方面，Vue 可以便捷地完成 Spring MVC 框架中视图层与模型层间的信息交互，并通过组件将数据在视图层进行显示。另一方面，Vue 可以将第三方库应用于现有项目中，从而构造包含复杂逻辑和多种组件的应用。Vue 实现了组件化应用构建，通过小型、独立、可复用的组件实现大型应用程序的构建和管理，包含跨组件数据流、自定义事件通信以及构建工具集成等功能。同时，Vue 使用基于 HTML 的模板语法，支持对模型底层数据和 DOM 节点之间的绑定，捕捉模型变

化并自动寻找最优策略应用于 DOM 操作，同时重新对视图层进行渲染。Vue 包括 Vue CLI、Vue Router、Vuex 和 Vue Loader 等核心插件，并构成了 Vue 框架的整个生态系统。

Vue CLI 是基于 Vue.js 进行快速开发的命令行界面 (Command-Line Interface, 脚手架)。Vue CLI 完全围绕插件而构建，具有灵活性、可扩展、可配置、可升级的特点，支持快速原型开发，并通过简化配置流程实现开发环境高效搭建。对于大型应用的开发，Vue CLI 可以对项目进行可视化创建和管理，协助用户完成高效的项目开发、部署、测试等基础工作。Vue CLI 包含 CLI、CLI Service 和 CLI 插件三个主要组件，其中 CLI 通过 vue 命令完成新项目构建、原型开发以及对项目的可视化管理；CLI Service 作为 CLI 的附属依赖项服务，可以实现项目开发、打包，以及检查内部 webpack 项目的配置等功能；而 CLI 插件为项目提供额外的功能，即项目可在 CLI 插件中选择可选功能。

Vue Router 是与 Vue.js 深度集成的官方路由管理器，适用于构建单页应用。单页应用指单一页面应用程序，当由于数据更新或用户操作需要重新加载页面时，单页应用只会对页面中的特定内容进行更新，而不会对整个页面进行重绘，极大地减少了资源负载。Vue 的单页应用通过切换路由对页面访问路径进行重设，从而实现页面和组件的切换和跳转。其中包含 hash 和 history 两种路由方式。hash 模式中用 URL 中的 hash 值作为路由，当 hash 值变化时，可由 hashchange 监听探测到这一事件，并对页面中指定位置的内容进行更新，同时由于 http 请求中不包含 hash 值内容，因此其值变化不会导致页面重新加载，满足单页应用的性质。history 模式则是基于 HTML5 历史记录 API 和服务器配置，通过 pushState()、replaceState() 等方法实现在不重新加载页面前提下的 URL 跳转。

Vuex 是基于 Vue.js 的组件状态管理模式，通过 store 将项目中所有组件的状态集中进行存储管理，并制定规则保证视图的变化不会影响到组件状态，且组件状态不会发生异常改变。当 store 中的状态发生改变时，基于细粒度数据响应机制，对应组件也会相应地进行高效更新，其中状态只能通过显式地提交 mutation 来改变，便于对组件状态进行获取和跟踪。

Vue Loader 是一个 webpack 的加载器，能够将单文件组件格式的 Vue 文件解析并转换为 JavaScript 模块。Vue 单文件组件采用类 HTML 语法，包含 <template>、<script>、<style> 三个顶级语言块和可选的自定义块。Vue Loader 通过对 Vue 文件中的语言块进行提取和解析，调用相应的 loader 处理，组装成一个 ES Module，最后导出为 Vue 组件对象。

## 2.3 后端相关技术

### 2.3.1 Spring Boot 框架

Spring Boot 框架是基于 Spring 设计的开源轻量级框架，为微服务体系开发中的架构问题提供了一系列简化、便捷的解决方案，内置 Tomcat 容器，可对 Spring 应用进行快速搭建、配置和开发，同时基于约定大于配置的策略，对 Spring 中的繁琐配置流程进行了优化，对第三方框架进行快速整合，减少甚至不需要配置文件，提高了项目的开发效率。此外，通过在 `pow.xml` 文件中添加相关依赖包并采用注解替代配置文件，达到开箱即用的策略。Spring Boot 支持关系数据库和非关系数据库，拥有简洁的安全策略集成，通过调用 Actuator 组件可对程序运行状态进行监控，并且可以避免大量 Maven 导入和版本冲突，提供了快速使用 Spring 框架的方式。

### 2.3.2 Docker

Docker 是一个开源的、基于 LXC (LinuX Container) 的高级容器管理引擎，以容器为基础单位提供面向微服务的轻量级虚拟化解决方案。通过 Docker 可以为任意应用创建一个轻量级的、可移植的、标准化的、自管理的容器，同时支持跨平台的开发、测试和运行。容器具备自包含的能力，通过 Docker 将底层环境打平，用户可以将一个容器镜像运行在任何操作系统的宿主机上，也就是“Build once, Run anywhere”。

Docker 在操作系统层面进行构建，利用其容器化技术，不依赖于独立的操作系统运行，突破了宿主机的磁盘 IO 性能，提高了底层资源使用率。它是应用程序与操作系统之间的隔离层，容器只关心中间件以上的应用层，而 Docker 负责处理中间件与操作系统之间的问题。此外，还引入了类似于源码管理的机制，将容器的快照历史版本进行记录，便于容器切换。同时，Docker 可以通过一个可编辑的 Dockerfile 文件来构建整个容器，应用程序开发人员通过发布 Dockerfile 来定义应用程序的环境和依赖，便于持续交付。Docker 可以为开发、测试等环节提供一次性环境，如本地测试他人的软件、持续集成时提供单元测试和构建的环境等。此外，由于 Docker 容器可以随时执行和停止，并且适合动态扩容，便于提供多样化且动态定制的云服务和微服务架构。

Docker 整体采用 C/S 架构，通过 Daemon 对 Docker 容器进行构建和运行，其中 Docker 客户端使用 RESTAPI 与 Daemon 进行通信，并且同时支持本地和远程通信。通过 Docker Client 将需要执行的 Docker 命令发送给 Docker 运行节点上的 Docker Daemon，再将用户的请求进行分解执行。

Docker 基于一个基础镜像，将需要进行安装的依赖和程序根据 Dockerfile 文件逐层叠加到该镜像中，以完成镜像构建。镜像的每一层都有一个独立的 id，Docker 使用 Union FS 对这些镜像进行分层和合层记录。当使用同一个基础镜像时，利用 Docker 镜像分层功能调用已有的镜像层并拉取没有的镜像层，减少重复镜像层拉取，达到资源使用最大化。当容器运行时，Docker 会在容器镜像的最上层生成一个 Container 层，并将该镜像完整地加载到内存中运行。在 Container 层中，对于任意容器的修改不会影响到底层镜像，而当容器消亡时，相应的修改也将一并删除。

## 2.4 本章小结

本章分别从医疗领域相关概念、前端技术和后端技术三方面对系统采用的技术栈进行了介绍。系统基于医疗图像、病灶和征象的原理、特征等先验知识设计了图像特征变异技术和领域语义变异技术，并采用 Docker 对底层医疗图像扩增工具进行封装打包。服务器端基于 Spring Boot 框架进行搭建和开发，在其中创建 Docker 容器并通过调用 Shell 文件执行图像扩增任务。前端则基于 Vue 生态和 Element UI 框架进行构建，以完成页面的渲染和显示。

## 第三章 需求分析与概要设计

### 3.1 系统整体概述

该系统主要针对医疗辅助诊断相关场景，通过基于病灶征象和变异分析的医疗图像扩增技术，由 AI 数据扩增平台调用医疗图像扩增服务，从图像特征变异和领域语义变异两方面对指定数据集进行扩增，并根据多维评估指标对扩增数据集进行全面的质量评估，以产生高质且足量的扩增图像数据集，以改善和评估模型的性能和鲁棒性。

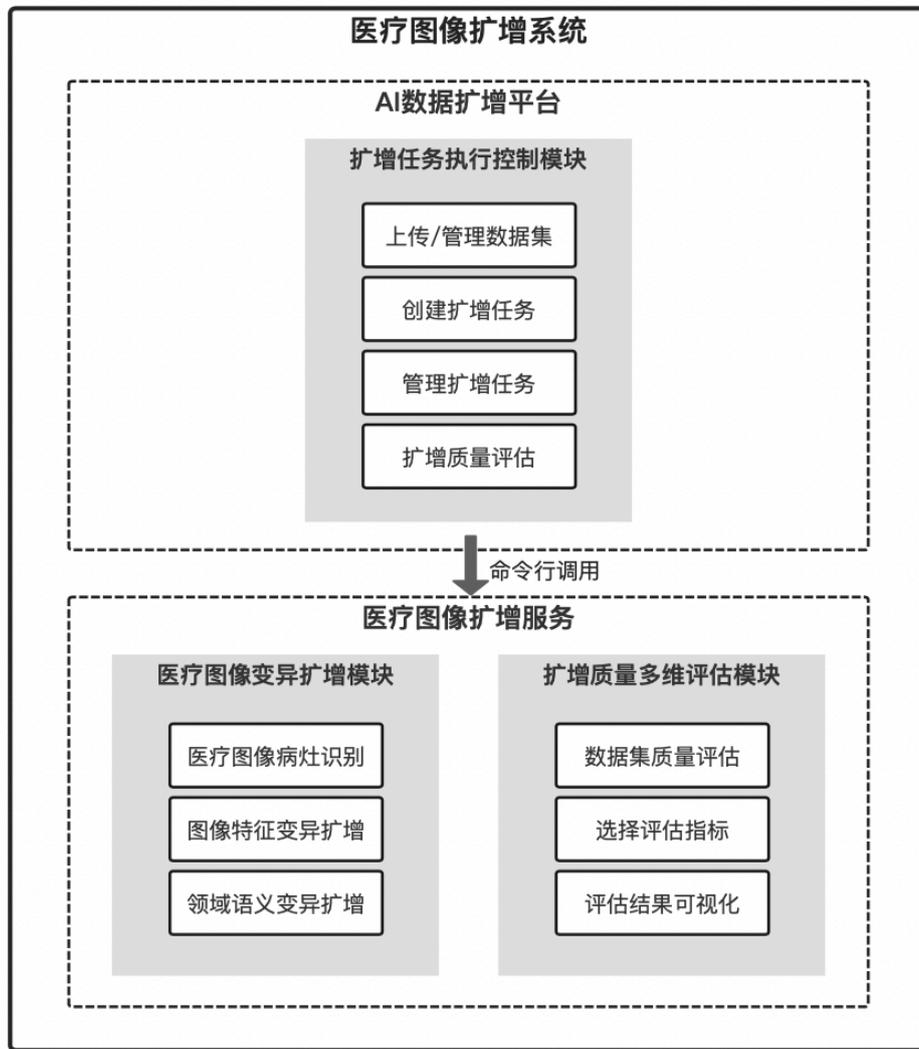


图 3.1: 系统整体模块图

系统整体模块图如图 3.1 所示。从功能上看，扩增任务执行控制模块、医疗图像变异扩增模块和扩增质量多维评估模块分别实现了对数据集的上传和管理、对扩增任务的创建和管理、医疗图像病灶识别、图像特征变异扩增、领域语义变异扩增、扩增数据集的质量评估以及评估结果可视化等功能。其中医疗图像变异扩增模块和扩增质量多维评估模块构成了医疗图像扩增服务体系，通过扩增任务执行控制模块所在的 AI 数据扩增平台对该服务进行 Docker 命令行调用来实现医疗图像数据集的扩增和评估。此外 AI 数据扩增平台还面向所有微服务制定了统一的接口标准，以便于更多附加服务的扩展和接入，例如文本扩增、音频扩增、模型评估等应用。

从业务流程上看，用户可在 AI 数据扩增平台上传多个原始数据集，并由平台保存到数据库中。用户可基于原始数据集创建扩增任务，并自定义扩增方法及参数，当任务创建完成后，平台用命令行调用医疗图像扩增服务中的变异扩增模块对原始数据集进行扩增处理，并将扩增得到的数据集传输回平台，用户可在该扩增任务的报告中查看任务的配置信息、运行状态，并且可以将扩增数据集下载到本地。此外，用户还可选择扩增任务和评估指标，对扩增数据集的质量进行评估，并生成可视化评估结果。

## 3.2 需求分析

在需求分析阶段，需要根据用户要求和系统整体需求，明确需要完成的工作和达到的目标，以指导系统的概要设计和整体架构。下面从功能性和非功能性需求以及系统用例三个方面进行系统需求分析。

### 3.2.1 功能性需求

根据功能模块划分，系统粗粒度的功能可分为扩增任务执行控制、医疗图像变异扩增以及扩增质量多维评估。扩增任务执行控制模块通过对医疗图像变异扩增模块和扩增质量多维评估模块进行调用，以实现图像扩增和质量评估的各个功能点。根据业务流程，又可以将系统功能进行进一步细分。

系统功能性需求列表如表 3.1 所示。用户首先可以将原始数据集上传到系统，并基于该数据集创建扩增任务，同时完善任务相关信息，设置扩增方法，以启动扩增任务流程，用户可以在扩增任务列表随时查看各个任务的进展情况，以及扩增任务的相关配置信息和状态信息。当扩增完成后，用户可将扩增完成的数据集下载到本地，以查看其扩增情况。除此之外，用户还可基于特定扩增任务选择数据评估指标，多维度评估图像扩增的效果。

表 3.1: 系统功能性需求列表

需求编号	需求名称	需求描述
R1	上传数据集	用户可以在上传数据集页面, 选择需要上传的数据集压缩包或文件链接, 点击上传按钮进行上传
R2	查看数据集列表	用户可以在管理数据集页面查看自己上传的所有数据集列表和其数量、大小等信息
R3	执行扩增任务	用户可以在新建扩增任务页面, 选择待扩增数据集, 添加扩增任务信息, 并设置扩增方法, 然后点击创建任务按钮来执行扩增任务
R4	选择待扩增数据集	用户可以在新建扩增任务页面, 选择需要进行扩增的原始数据集
R5	添加扩增任务信息	用户可以在新建扩增任务页面, 填写扩增任务名称、备注和输出文件数量
R6	设置扩增方法	用户可以在新建扩增任务页面, 自定义添加扩增方法并设置相应方法参数
R7	查看扩增任务列表	用户可以在管理扩增任务页面查看自己创建的所有扩增任务列表和当前任务状态
R8	查看扩增任务信息	用户可以在管理扩增任务页面, 点击详情按钮, 查看当前扩增任务的名称、数据集、扩增方法等详细配置信息
R9	拷贝扩增任务	用户可以管理扩增任务页面, 点击扩增任务对应的拷贝按钮, 以相同的配置信息再次执行该任务
R10	查看扩增结果报告	用户可以在管理扩增任务页面, 点击报告按钮, 查看扩增任务运行结果, 包括数据集、任务配置和运行的基本信息
R11	下载扩增数据集	用户可以在扩增任务报告页面的结果下载位置将扩增后的数据集下载到本地
R12	设置评估参数	用户可以在扩增多维评估页面选择待评估任务和评估指标等参数
R13	查看评估结果	用户可以在扩增多维评估页面查看可视化评估结果报告, 包括雷达图、分布图等

表 3.2: 系统非功能性需求列表

类别	需求描述
性能	在正常网络状态下, 系统应控制除执行扩增任务外的页面跳转、简单查询等普通操作的响应时间在 500ms 以内, 扩增任务的成功率达到 99%。
安全性	系统需要保证其访问和数据安全, 通过权限控制、数据加密等手段保证数据在采集、传输、处理过程中不被偷窥、窃取、篡改。
可维护性	系统应当采用低耦合模块化构建, 并用日志记录系统运行状态和缺陷信息, 能够及时对故障进行准确定位和排除。
可扩展性	系统需要针对不同的应用接口设计统一的标准, 以便于功能的添加和扩展, 同时, 类似组件应当进行统一设计, 便于复用。
可靠性	系统需要保证在发生内部或外部故障时, 不影响用户的行为操作及数据。例如断网时, 需要对数据做好本地保存, 系统进行更新时也需要准备回滚方案。
易用性	系统应当遵循以用户体验为中心的界面设计原则, 文字标识简洁清晰, 并为用户提供适当的信息提示和引导。

### 3.2.2 非功能性需求

医疗图像扩增系统的整体目标是以现有的医疗数据集为基础，使用基于病灶征象和变异分析的扩增技术，通过不同的扩增方式，产生可用于深度学习模型优化和评估的医疗图像数据集。同时，在 AI 数据扩增平台的使用中，需要满足多个方面的非功能性需求，如以提供持续稳定且高效的服务。非功能性需求列表如表 3.2 所示。

### 3.2.3 系统用例图

系统用例图如图 3.2 所示，描述了用户在该系统中的所有用例。其中用户可以上传数据集，执行、拷贝和删除扩增任务，以及查看扩增任务的详情、评估和结果报告，并进行扩增数据集的下载。其中执行扩增任务包括添加扩增任务信息、选择待扩增数据集和设置扩增方法及参数。

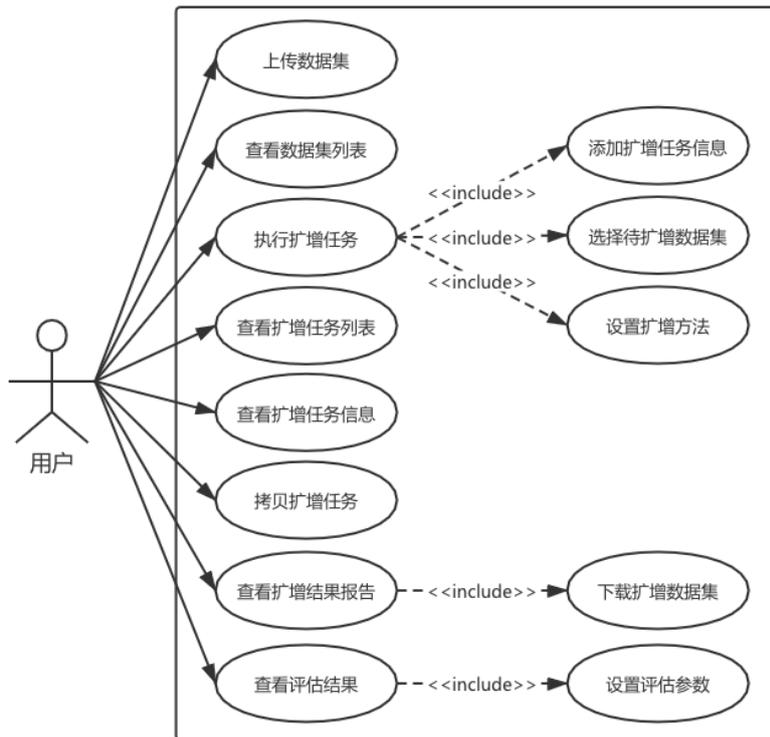


图 3.2: 系统用例图

### 3.2.4 系统用例描述

下面对上述系统用例图中主要展示用例进行详细阐述，将会介绍各个用例详细描述，包括正常流程、扩展流程、触发条件、前置条件和后置条件等信息，涉及用例跟功能需求的对应关系可如表 3.3 所示。其中，囿于论文篇幅和突出重

点需要，对 UC2 查看数据集列表、UC4 查看扩增任务列表和 UC6 拷贝扩增任务的用例描述并未详细给出。

表 3.3: 系统用例列表

用例编号	用例名称	功能需求编号
UC1	上传数据集	R1
UC2	查看数据集列表	R2
UC3	执行扩增任务	R3、R4、R5、R6
UC4	查看扩增任务列表	R7
UC5	查看扩增任务信息	R8
UC6	拷贝扩增任务	R9
UC7	查看扩增结果报告	R10
UC8	下载扩增数据集	R11
UC9	查看评估结果	R12、R13

表 3.4: 上传数据集用例描述

ID	UC1
用例描述	上传数据集
参与者	用户
触发条件	用户点击菜单项进入上传数据集页面
前置条件	用户需已被识别和授权
后置条件	用户能在管理数据集页面检索到该数据集，且显示的数据集信息与真实信息相符
优先级	高
正常流程	<ol style="list-style-type: none"> <li>1. 用户选择数据集类别，并选择通过文件或链接进行数据集上传；</li> <li>2. 系统对应显示文件方式上传或链接方式上传；</li> <li>3. 用户通过选择或拖拽本地文件压缩包完成文件方式上传，或输入数据集名称与数据集链接完成链接方式上传；</li> <li>4. 系统显示数据集上传成功。</li> </ol>
扩展流程	<ol style="list-style-type: none"> <li>3a. 用户上传的数据集不是 zip 类型；                             <ol style="list-style-type: none"> <li>3. 系统提示仅支持上传扩展名为.zip 的数据集压缩包</li> </ol> </li> <li>3b. 用户上传的数据集超过 500MB；                             <ol style="list-style-type: none"> <li>3. 系统提示仅支持上传小于 500MB 的文件</li> </ol> </li> </ol>
特殊需求	系统需支持多个数据集同时上传

UC1 上传数据集用例描述如表 3.4 所示。当用户完成注册和登录流程后，可以从菜单栏的上传数据集选项卡直接进入上传数据集页面，选择需要上传的数据集类别和上传方式，上传方式包括文件方式上传和链接方式上传。若用户选择文件方式上传，可通过点击选择本地文件或拖拽本地文件进行上传，若用户选择链接方式上传，则需要输入数据集名称和数据集链接进行上传。此外，系统对

于上传的数据集大小和类型等也有一些限制。目前系统只支持 zip 类型文件，数据集大小限制在 500MB 以内，并且对于图像数据集，支持 png、bmp、gif、jpeg、jpg 和 tif 的图片格式。

表 3.5: 执行扩增任务用例描述

ID	UC3
用例描述	执行扩增任务
参与者	用户
触发条件	1. 用户点击菜单项进入新建扩增任务页面 2. 用户在管理数据集页面，点击特定数据集对应的新建按钮进入新建扩增任务页面
前置条件	1. 用户需已被识别和授权； 2. 用户需有上传完成的数据集。
后置条件	用户能在管理扩增任务页面检索到该扩增任务，且扩增任务执行状态正常
优先级	高
正常流程	1. 用户通过下拉选择数据集类别； 2. 系统显示筛选后的数据集列表； 3. 用户通过下拉选择待扩增数据集； 4. 系统显示数据集中文件数，并根据数据集名称自动填写扩增任务名称 5. 用户可修改扩增任务名称，并填写任务描述； 6. 用户通过下拉框选择或自定义输入需要扩增输出的文件数量； 7. 用户在添加扩增方法表单中选择多种通用规则扩增方法和医疗领域扩增方法； 8. 系统在设置方法参数表单中显示用户选择的扩增方法所需要设置的参数列表； 9. 用户根据扩增需求填写或选择各个方法的扩增参数，并可查看其取值范围； 10. 用户点击创建任务； 11. 系统显示扩增任务创建成功，并跳转到管理扩增任务界面，在任务列表中显示当前任务的基本信息和执行状态。
扩展流程	6a. 用户输入的扩增倍数超过 20； 6. 系统显示最多只输出 20 倍张图片 9a. 用户在设置方法参数表单中点击特定扩增方法对应的帮助按钮； 9. 系统跳转到使用帮助页面，介绍在创建扩增任务中提供的扩增方法，包括方法原理、参数说明以及扩增示例 10a. 用户填写的扩增任务名称重复； 10. 系统显示任务名称已存在
特殊需求	无

UC3 执行扩增任务用例描述如表 3.5 所示。用户在新建扩增任务页面，选择待扩增数据集，填写任务名称和任务描述，并选择输出文件数量倍数。在选择扩增方法时，可同时选用通用规则扩增方法和医疗领域扩增方法。其中通用规则扩增方法表示根据图像基础特征进行变异的扩增方法，包括翻转、旋转、倾斜变换等 13 种扩增方法，医疗领域扩增方法表示基于医疗领域知识对图像进行变异

的方法，包括对病灶和非病灶区域分别进行变异的 8 种扩增方法。当用户完成扩增方法的选择后，系统会依次在页面下方显示各个扩增方法需要设置的参数，用户可将鼠标悬停在输入框上以查看参数的规定取值范围，并可通过扩增方法旁的帮助按钮查看扩增方法的相关介绍。在用户配置完所有方法参数并点击创建任务后，系统将跳转到管理扩增任务页面并显示当前任务进度等信息。

UC5 查看扩增任务信息用例描述如表 3.6 所示。用户首先从管理扩增任务页面查看扩增信息列表，并在需要查看的扩增任务一栏点击详情，系统跳转到扩增任务信息页面，显示用户创建该扩增任务时所配置的所有信息，包括数据集名称、数据项数量、任务名称、任务描述、输出文件数量倍数、扩增方法及其对应参数等。此外，系统提供了拷贝扩增任务的入口，即基于当前扩增任务的所有配置信息，重新创建一个新的扩增任务。

表 3.6: 查看扩增任务信息用例描述

ID	UC5
用例描述	查看扩增任务信息
参与者	用户
触发条件	用户在管理扩增任务页面，点击特定扩增任务对应的详情按钮进入查看扩增任务信息页面
前置条件	1. 用户需已被识别和授权； 2. 用户需有创建完成的扩增任务。
后置条件	无
优先级	中
正常流程	1. 用户点击扩增任务对应的详情操作； 2. 系统显示扩增任务相关信息，包括数据集名称、数据项数量、任务名称、任务描述、输出文件数量倍数、扩增方法及其对应参数，并提供扩增任务拷贝的跳转入口。
扩展流程	2a. 用户点击任务拷贝操作； 2. 系统跳转到新建扩增任务页面，并沿用当前扩增任务配置
特殊需求	无

UC7 查看扩增任务报告用例描述如表 3.7 所示。用户首先从管理扩增任务页面查看扩增信息列表，并在需要查看的扩增任务一栏点击报告按钮，进入查看扩增任务报告用例页面，系统会显示扩增任务执行后生成的报告信息，包括数据集名称、大小、数据项数量等数据集相关信息，扩增任务名称、描述、扩增数量、倍数等任务基本信息，以及任务运行时长、运行状态等任务运行信息，并提供原始数据集下载和扩增数据集下载链接，用户可自行导出。

UC8 下载扩增任务数据集用例描述如表 3.8 所示。在查看扩增任务页面，若该扩增任务已完成且执行成功，则用户可从系统提供的结果下载链接将扩增数

表 3.7: 查看扩增任务报告用例描述

ID	UC7
用例描述	查看扩增任务报告
参与者	用户
触发条件	用户在管理扩增任务页面，点击特定扩增任务对应的报告按钮进入查看扩增任务报告页面
前置条件	1. 用户需已被识别和授权； 2. 用户需有创建并成功执行完成的扩增任务。
后置条件	无
优先级	高
正常流程	1. 用户点击扩增任务对应的报告操作； 2. 系统显示数据集名称、大小、数据项数量、扩增任务名称、扩增数量、任务运行时长、运行状态等信息，并提供原始数据集下载和扩增数据集下载链接。
扩展流程	2a. 用户查看的扩增任务并未成功执行； 2. 系统显示任务失败原因和错误日志信息，并提供解决方案
特殊需求	无

数据集下载到本地，并可基于该数据集对已有模型进行训练或测试，以提高其性能表现或对模型进行评估。

表 3.8: 下载扩增数据集用例描述

ID	UC8
用例描述	下载扩增数据集
参与者	用户
触发条件	用户在查看扩增任务报告页面，点击扩增数据集下载链接下载数据
前置条件	1. 用户需已被识别和授权； 2. 用户需有执行成功的扩增任务。
后置条件	扩增数据集成功保存到本地
优先级	高
正常流程	1. 用户点击扩增数据集下载链接； 2. 系统跳转到本地下载器； 3. 用户选择存储路径后将数据集压缩包保存到目标路径。
扩展流程	无
特殊需求	无

UC9 查看评估结果用例描述如表 3.9 所示。在扩增多维评估页面，用户选择系统提供的多个评估指标，点击提交按钮执行扩增数据集质量评估流程。通过应用不同维度的评估指标公式计算数据集中图像的质量分数，并分别从数据集整体质量水平和内部质量分布来生成评估结果。用户可以查看数据集整体质量评估分数多维雷达图和图像质量评估分数分布图，获取对扩增任务效果的反馈。

表 3.9: 查看评估结果用例描述

ID	UC9
用例描述	查看评估结果
参与者	用户
触发条件	用户在管理扩增任务页面，点击特定扩增任务对应的评估按钮进入扩增质量评估页面
前置条件	1. 用户需已被识别和授权； 2. 用户需有执行成功的扩增任务。
后置条件	无
优先级	高
正常流程	1. 用户点击扩增任务对应的评估操作； 2. 系统跳转到扩增质量评估页面； 3. 用户可重新选择扩增任务，同时选择多个评估指标，并点击提交按钮； 4. 系统显示评估任务执行成功，并显示对应整体质量评估分数多维雷达图和图像质量评估分数分布图。
扩展流程	无
特殊需求	无

### 3.3 系统概要设计

#### 3.3.1 系统架构设计

医疗图像扩增系统的整体架构设计如图 3.3 所示，自底向上依次分为工具层、数据存储层、应用服务层和展示交互层。系统整体采用微服务架构进行开发，将业务划分成细粒度的服务，并由客户端通过轻量级 API 对微服务进行调用，达到服务内部的高内聚和服务之间的低耦合。下面分别对每一层进行详细介绍。

展示交互层即系统前端，整体采用 Vue 框架进行构造。基于 vue.js 并使用 Vue CLI 脚手架，快速完成原型开发和前端项目构建，以及前后端接口数据交互和整体项目配置。前端采用 Vue Router 实现路由管理，并将路径和组件进行映射，以完成切换和跳转页面的操作。同时引用 Element UI 组件库，完成前端整体布局、导航、表单等重要组件的开发。此外，为了保证系统的稳定性，前后端通过 Nginx 进行反向代理和负载均衡。

应用服务层整体采用 Spring Boot 框架，将前后端数据交互、业务逻辑实现和持久层数据访问进行分离，以达到高内聚低耦合、且具有较高可扩展性的系统架构。医疗图像扩增系统的功能模块主要包括扩增任务执行控制模块、医疗图像变异扩增模块和扩增质量多维评估模块，其中扩增任务执行控制模块提供用户交互功能，主要负责系统中数据集上传管理与基于数据集和自定义方法的扩增

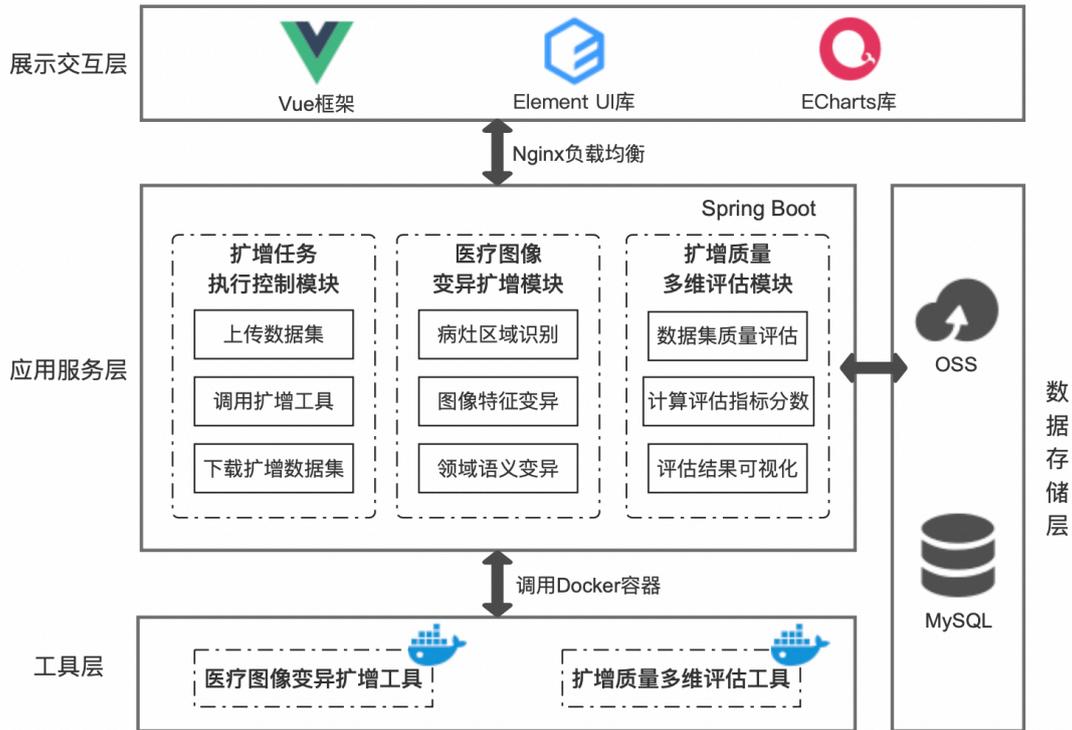


图 3.3: 系统架构设计图

任务的执行管理等业务，已上传的数据集和已创建的任务信息将存储到 MySQL 数据库中，以供用户查看下载。医疗图像变异扩增模块则通过 Docker 调用医疗图像变异扩增工具，实现了基于病灶征象和变异分析的医疗图像扩增技术，包括图像特征变异扩增方法和医疗领域语义变异扩增方法，通过叠加执行多种扩增方法，完成基于原始数据集的扩增操作。扩增质量多维评估模块则是采用多种评估维度指标对扩增得到的数据集进行质量评估，并在前端进行可视化，同样通过 Docker 调用扩增质量多维评估工具实现，对应评估分数也将存储到数据库中，供用户随时查看使用。

数据存储层采用 MySQL 数据库存储系统执行流程中的业务实体对象字段，包括扩增任务配置信息、数据集相关信息、用户身份信息，OSS 对象存储是由阿里云提供的海量、安全和高可靠的云存储服务，其中存储用户上传的数据集文件、扩增得到的数据集文件以及用于评估任务的模型文件等数据。

### 3.3.2 系统 4+1 视图

系统 4+1 视图是以用例为驱动，在软件生命周期各个阶段从不同视角对系统进行建模，从而形成对系统的整体架构描述。4+1 视图包括逻辑视图、开发视图、进程视图、物理视图和场景视图。其中场景视图与图 3.2 的系统用例图相对应。下面分别对每个视图进行详细介绍。

逻辑视图如图 3.4 所示，主要从最终用户的角度对系统架构进行描述，只关注系统提供给用户的功能集合。其中 `DataSetService` 负责数据集的上传、下载、查询、校验和更新等，`JobService` 负责异步任务的创建、执行、查询和修改，通过 `WorkerService` 创建基于特定异步任务的 `Worker`，并调用 `DockerHelper` 执行 shell 脚本，以创建执行扩增或评估任务的容器。`Job`、`DataSet` 和 `Worker` 分别是异步任务、数据集和任务执行者的抽象实体，`ZipAndRarHelper` 则负责对输入的原始数据集进行解压和压缩输出的扩增数据集，以供用户下载。

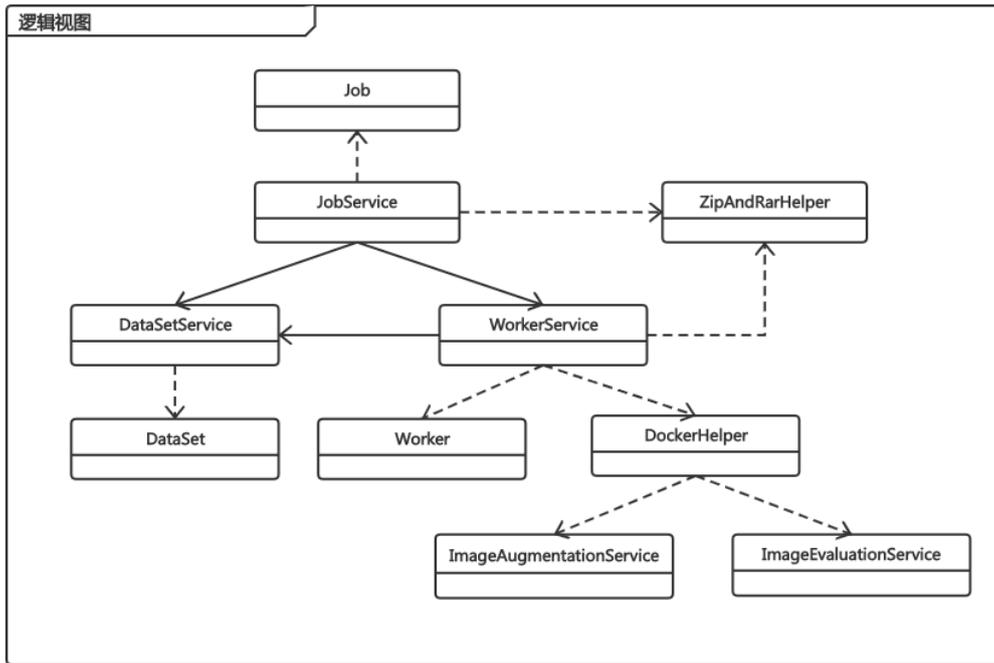


图 3.4: 系统逻辑视图

开发视图如图 3.5 所示，主要从开发人员的角度对系统代码架构进行描述，包括前端 UI、服务器端 `Server Logic` 和可供调用的技术服务 `Tech Services`。其中前端界面通过 `JavaScript` 和 `Components` 进行构建，并通过 `Echarts` 库实现对数据集质量的评估图表可视化。服务器端主要分为 `Controller` 包、`Service` 包、`Repository` 包和 `utils` 包，其中 `Controller` 负责处理外部请求，与前端进行数据交互，对请求

参数进行校验和处理，调用 **Service** 完成相关业务操作，并将得到的数据封装成用于前端渲染的视图对象或用于交互的数据传输对象。**Service** 内部实现具体业务逻辑，将逻辑和调用分离，以达到复用的效果，调用 **Repository** 完成对相关数据的访问和操作，同时调用 **utils** 中的类完成 **Docker** 容器创建删除、文件存取、文件压缩解压、类型转换、识别文件编码等相关操作。**Tech Service** 中包含图像扩增服务、图像质量评估服务和协助进行数据访问处理的 **Spring Data JPA**。

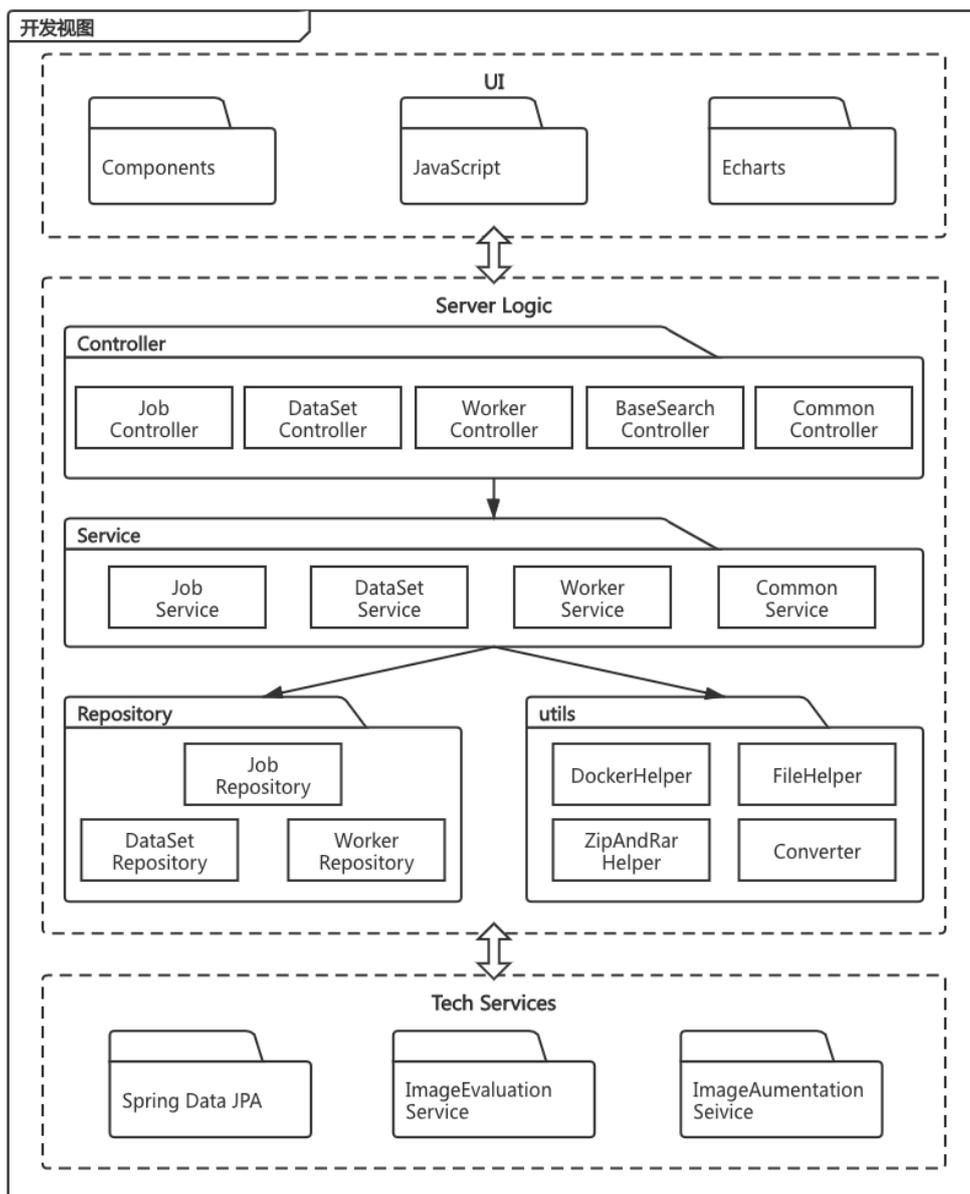


图 3.5: 系统开发视图

进程视图如图 3.6 所示，主要从系统集成及进程通信角度对系统进行描述，即在系统运行过程中进程及线程的并发、同步和传输等相关任务调度。系统主进程调用扩增或评估任务进程中的 `create()` 方法创建一个新任务，并将其任务配置、状态、存储路径等相关信息存储到 MySQL 数据库中。当系统开始执行扩增任务时，由扩增任务进程调用 `startWorker()` 方法启动异步任务线程，同时由异步任务创建 Docker 容器以执行具体扩增任务，并将扩增结果存储到 MySQL 数据库里该任务对应输出路径中。在异步任务执行过程中，支持对其运行状态进行轮询，一旦状态发生改变，则更新其任务状态，其间可以同时启动其他异步任务，并创建新的 Docker 容器执行任务操作，不同任务和容器之间相互独立。当查看到任务运行完成后，可由异步任务获取其对应扩增结果文件链接，并在前端显示以供用户下载。而当系统执行评估任务时，同样由任务进程调用 `startWorker()` 方法启动异步任务线程，并创建 Docker 容器计算相应指标评估分数结果，将结果存储在相应路径中，同时将其返回前端并进行可视化渲染。由于目前系统中均选用全参考评估指标，其结果不会由于多次评估发生改变，因此后续可直接从数据库中读取结果。

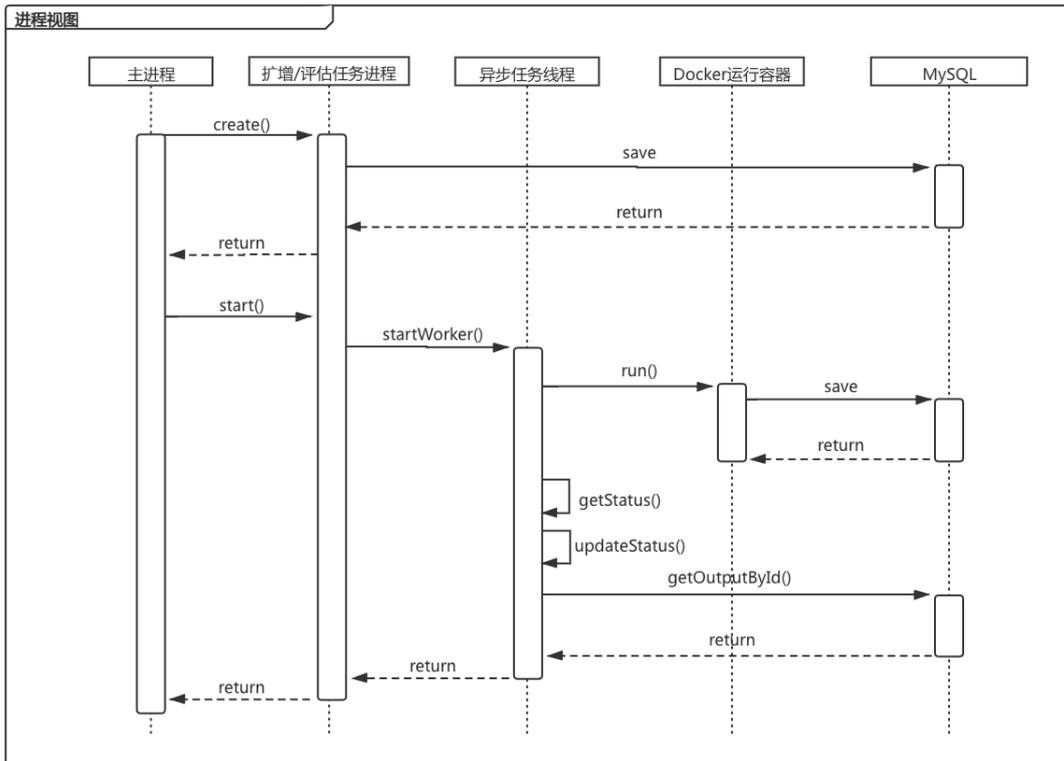


图 3.6: 系统进程视图

物理视图如图 3.7 所示，主要从系统运维、资源配置角度对系统进行描述，包括计算资源的分配、组件间的拓扑连接等。首先用户通过访问浏览器向 AI 数据扩增平台服务器发送 HTTP 请求，再由数据扩增平台服务器根据具体请求内容调用执行控制服务后端服务器中的相应方法。若需要创建和执行任务，则通过在 Docker 服务器中创建任务对应 Worker，并由该 Worker 调用医疗图像扩增服务后端服务器中的扩增或评估服务。同时，执行控制服务后端服务器还可通过 HTTP 连接在 OSS、MySQL 数据文件服务器中进行数据访问，其中数据集文件和评估结果文件都存储在阿里云对象存储（OSS）服务器中，任务参数等配置信息则存储在 MySQL 数据库中。

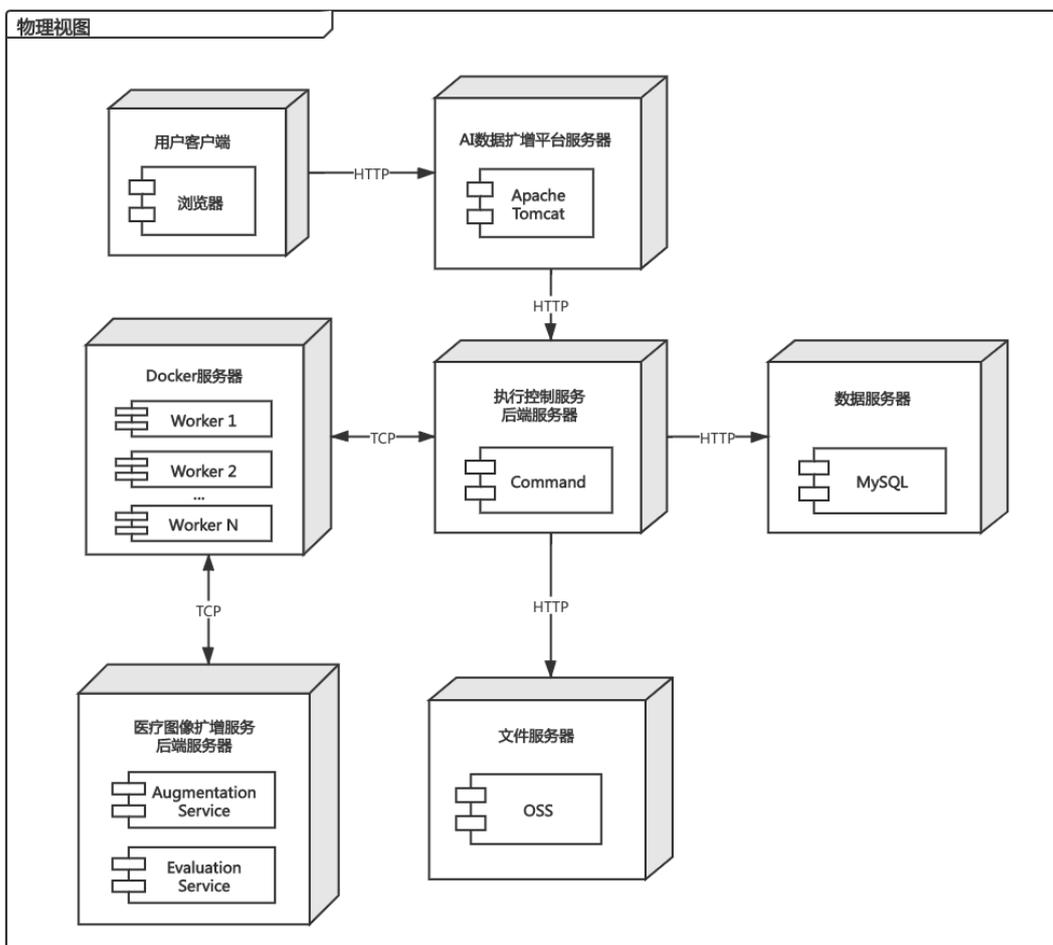


图 3.7: 系统物理视图

### 3.4 扩增任务执行控制模块

#### 3.4.1 架构设计

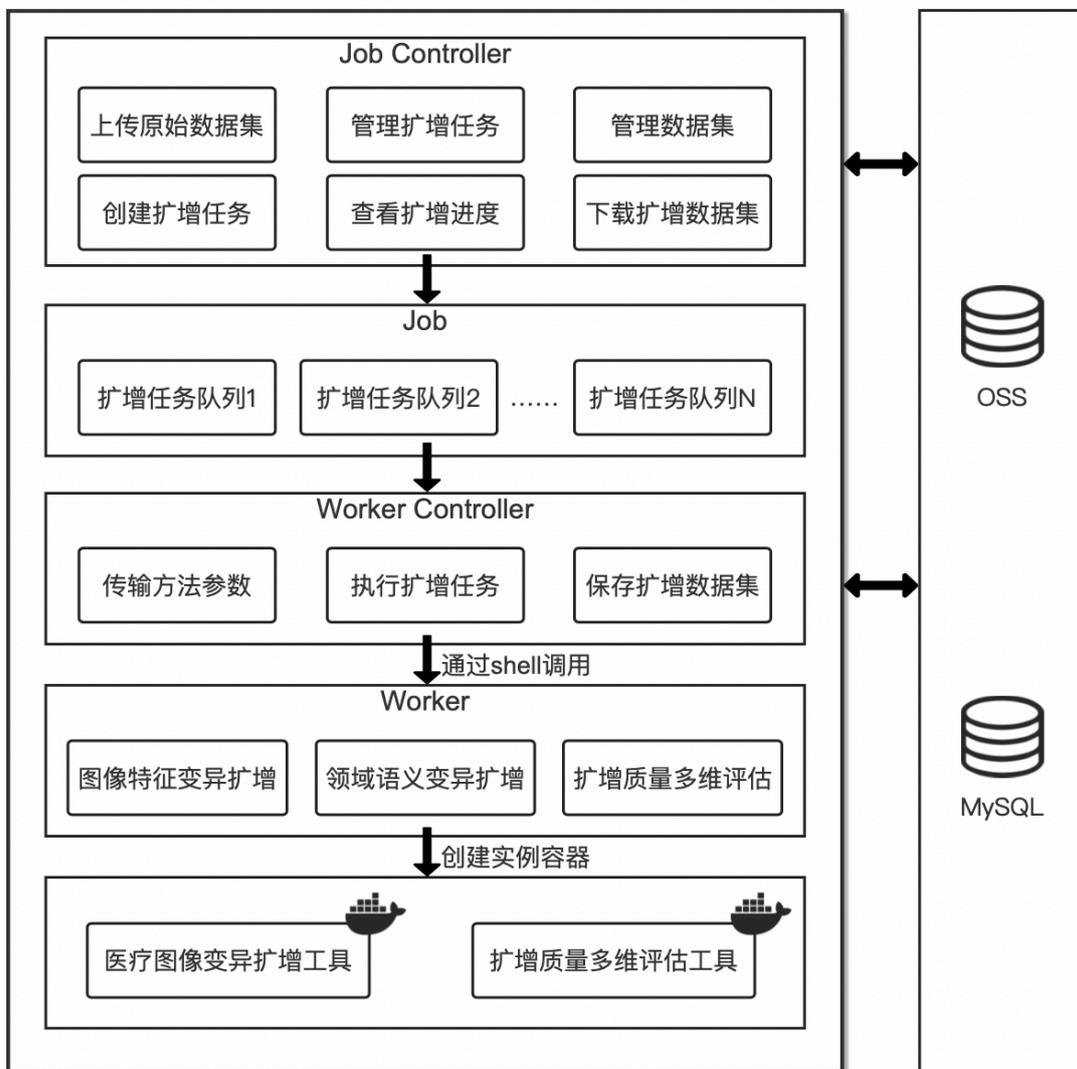


图 3.8: 扩增任务执行控制模块架构图

扩增任务执行控制模块架构设计如图 3.8 所示。其中 **Job** 是对异步任务的抽象，而 **Worker** 可以看成是具体任务的实现者。考虑到重用的问题，一个 **Worker** 可以完成很多任务，但不同的任务之间可能存在数据集或任务参数的差异，因此需要使用 **Job** 来进行区分。**Docker** 是 **Worker** 的底层，也是 **Worker** 的具体实现。模块中的 **Job Controller** 负责管理系统的扩增任务和数据集，以及系统整体的业务逻辑。在创建扩增任务后，系统将启动一个 **Job**，并由 **Worker Controller**

负责通过 Shell 脚本文件调用 Worker 中的扩增方法来执行相应的任务。Worker 则由医疗图像变异扩增 Docker 工具创建实例容器产生。

### 3.4.2 流程设计

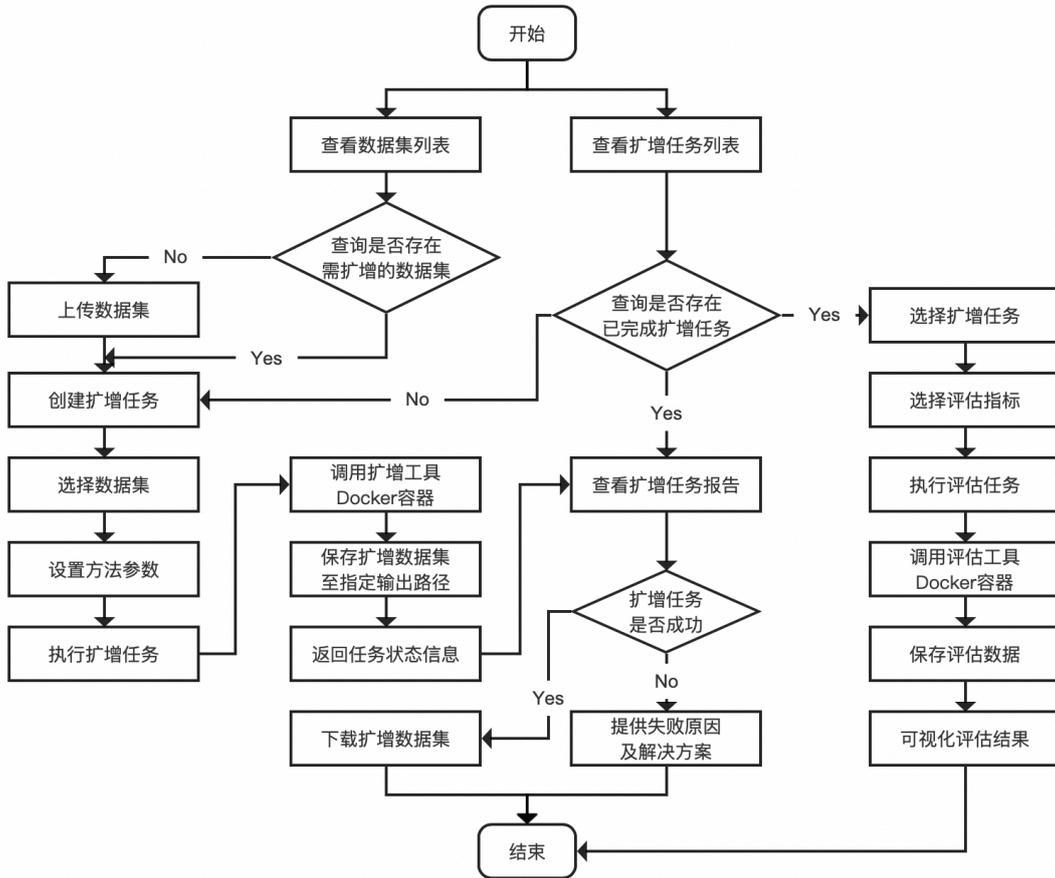


图 3.9: 扩增任务执行控制模块流程图

扩增任务执行控制模块流程图如图 3.9 所示。用户可在管理数据集页面查看已上传的所有数据集，包括数据集名称、类型、大小、数据项数量以及上传时间等信息，若其中并无用户想要扩增的数据集，则由用户将本地数据集上传到系统，创建扩增任务并设置扩增方法及相应参数，即可启动扩增任务，此时系统将调用图像扩增工具 Docker 容器，并将数据集路径、扩增任务配置信息等作为参数加载至容器中，由容器执行具体的扩增处理工作。扩增任务完成后，容器将扩增数据集保存到指定路径中，并将扩增任务完成信息回传到系统。同时，用户可在管理扩增任务页面查看已创建的所有扩增任务，包括任务名称、进度以

及创建时间等信息，还可查看特定扩增任务的结果报告，若扩增任务执行成功，则报告界面将提供下载扩增数据集的链接，若否，系统则显示失败信息并为用户提供解决方案。

### 3.5 医疗图像变异扩增模块

#### 3.5.1 架构设计

医疗图像变异扩增模块架构设计如图 3.10 所示。通过 AI 医疗图像扩增平台对医疗图像变异扩增工具进行调用，同时传递原始数据集路径、扩增方法及参数等信息，对原始数据集进行扩增处理。再将扩增后得到的数据集传输到指定输出路径，供用户下载使用。

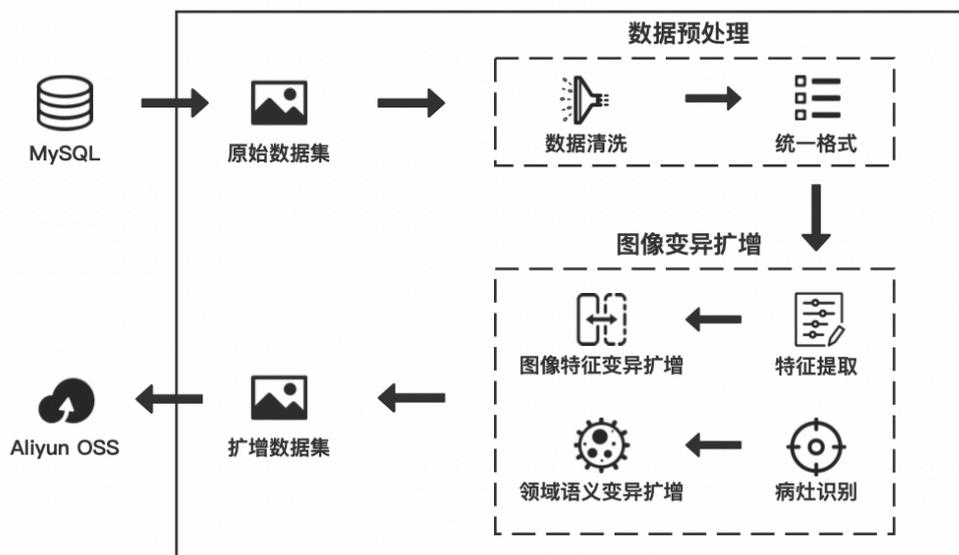


图 3.10: 医疗图像变异扩增模块架构图

首先从 MySQL 数据库中获取用户选择的数据集下载链接，并从阿里云对象存储服务下载对应数据集，再通过调用医疗图像扩增工具对输入的原始数据集进行预处理，通过数据清洗，排除无法读取、不完整的数据、无使用价值的数据和重复数据。此外，与自然图像不同，由于医疗图像的成像方式一般需要特殊的设备，因此其图像格式也多种多样。目前系统仅支持 png、bmp、gif 和 jpg 等图片格式，为了更有效率地进行数据扩增，需要将输入的各种格式的数据转换成统一的格式。

通过数据预处理后的数据集可根据选用的扩增方法进行变异扩增。其中图像特征变异扩增表示仅对图像本身特征进行变异的方法，图像特征包括像素空

间和值域特征、图像尺寸和纹理特征等。由于医学图像整体结构的高度相似性，同时为了在较大程度上保持医疗图像的完整度，基于无损和非刚性特性进行扩增处理。而领域语义扩增变异则是基于医疗领域特征，针对医疗图像中的关键区域，即病灶区域的扩增处理。传统的数据扩增是对整个图像进行处理，这在医疗图像中可能会改变图像的本质特征。因此提出了先对图像进行病灶识别，然后再制定数据扩增的方案。对图像非病灶区域的扩增，一般情况下不会改变图片的性质，而对图像病灶区域的扩增，则可能改变图像的病灶特征。通过对不同区域采用不同的数据扩增，产生不同的效果，也保证了扩增的有效性和多样性。最后将扩增得到的数据集存储到阿里云对象存储服务器中。

### 3.5.2 流程设计

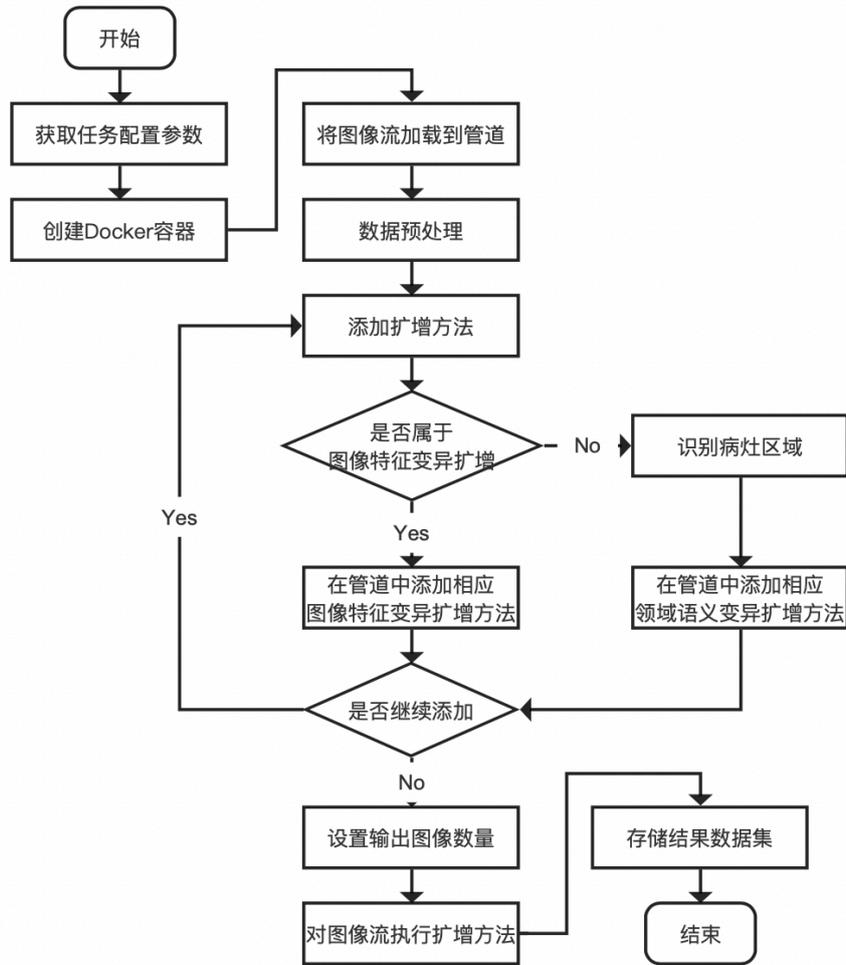


图 3.11: 医疗图像变异扩增模块流程图

医疗图像变异扩增模块流程设计如图 3.11 所示。系统首先获取扩增任务相关配置信息，包括原始数据集路径、扩增数据集存储路径、任务状态文件存储路径、扩增方法配置文件等参数，其中方法配置文件以 **Json** 文件形式传输。接收完所有配置信息后，模块创建执行扩增任务的 **Docker** 容器，并在其中通过 **Shell** 文件调用医疗图像变异扩增工具中的扩增方法。在调用过程中，首先解压原始数据集并将图像流加载到扩增管道中，然后对图像流进行预处理，剔除错误数据和空数据并将图像格式转换为 **RGB**。接着在扩增管道中逐个添加图像特征变异扩增方法和领域语义变异扩增方法，并设置输出图像数量，在管道中对图像流执行已添加的扩增方法，最后将输出的扩增数据集存储到指定路径。

### 3.6 扩增质量多维评估模块

#### 3.6.1 架构设计

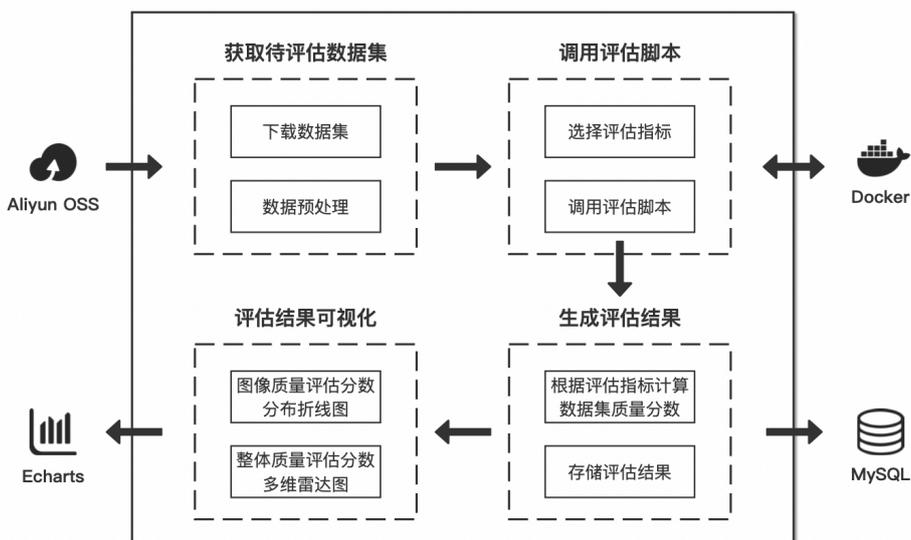


图 3.12: 扩增质量多维评估架构图

为了对扩增数据集的质量进行评估，系统加入了扩增质量多维评估模块，为用户提供对于各个扩增任务效果评估的定量度量结果。扩增质量多维评估模块架构设计如图 3.12 所示。启动扩增质量评估任务后，首先从阿里云对象存储服务中下载任务对应扩增数据集，同时进行预处理，剔除错误和重复的数据。然后基于用户在前端选择的评估指标，调用 **Docker** 评估脚本中对应参数的方法，并根据评估指标计算数据集各个维度的质量分数，同时存储到 **MySQL** 数据库中。基于计算出的各个维度的评估分数，通过 **Echarts** 在前端形成可视化评估视图结果，包括整体质量评估分数多维雷达图和图像质量评估分数分布折线图。

### 3.6.2 流程设计

扩增质量多维评估模块流程设计如图 3.13 所示。首先通过服务端传输的 Json 文件接收待评估数据集存储路径、评估指标、评估结果存储路径等参数，同时创建执行评估的 Docker 容器，并将相关参数传入容器中。其次从数据集存储路径中读取图像数据流，并对其进行预处理，即把图像全部设置为  $512 \times 512$  大小同时转换为 RGB 格式，以便于后续处理。然后根据传入的评估指标分别对数据集中的每张图像计算质量分数，同时针对数据集整体计算平均质量分数水平，并将评估结果存储到指定路径，供前端进行可视化图表渲染。

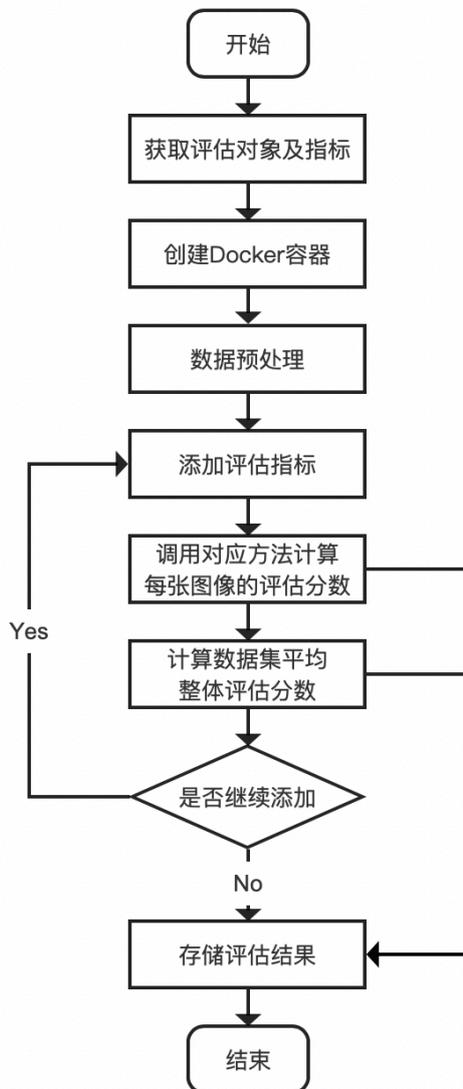


图 3.13: 扩增质量多维评估流程图

### 3.6.3 评估指标设计

扩增任务执行效果的评估主要通过对扩增数据集中图像的质量进行评估来实现。对图像的质量评估从方法上可分为主观评估和客观评估。主观评估就是从人的主观感知来评价图像的质量，首先给出扩增图像，让标注者给图像评分，一般采用平均主观得分 (Mean Opinion Score, MOS) 表示。客观评估则通过客观算法计算图像质量的量化值，与主观评估方法相比具有可批量处理、结果可重现的特点，并且不会因为人为原因产生偏差。客观评估方法根据评估时是否需要参考图像又分为无参考、半参考和全参考质量评估算法，其中无参考质量评估算法 [28] 不需要任何原始图像信息，所有评估流程都基于扩增图像进行处理。如利用质量排序图像数据集进行训练的 RankIQA 模型 [29]，可同时处理合成失真和真实失真的深层双线性网络 DBCNN[30]，以及结合主观内容依赖效应与时间记忆效应的视频质量评估模型 [31]。半参考质量评估算法 [32] 则是将原始图像和扩增图像中的部分对应特征进行比较。全参考质量评估算法 [33] 主要通过分析图像的视觉特征，对参考图像和扩增图像之间的差异进行量化，以计算扩增图像的质量评估分数，相关算法研究较为成熟。由于系统中需要对原始图像和扩增图像的整体质量变化幅度进行度量，因此主要选用全参考方法构建扩增质量评估指标体系。

根据评估维度和选择特征的差异，可将全参考质量评估方法大致分为基于误差敏感度的评估方法和基于感知模型和图像结构的评估方法。基于误差敏感度的评估方法通过线性变换分解图像信号，将扩增图像与原始图像进行对比，计算两个图像之间的差异。其中均方误差 (Mean Square Error, MSE) 和峰值信噪比 (Peak Signal to Noise Ratio, PSNR) 是图像处理领域目前应用最广的评估量化指标，计算简单高效且易于实现。MSE 计算原始图像和扩增图像间像素差值的均方值，然后通过均方值的大小来确定图像的失真程度，计算公式如下：

$$MSE = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W (X(i, j) - Y(i, j))^2$$

其中 H、W 分别为图像的高度和宽度， $X(i, j)$  和  $Y(i, j)$  分别表示扩增图像 X 和原始图像 Y 在  $(i, j)$  点的像素值。

PSNR 基于原始图像与扩增图像对应像素点间的误差计算量化评估分数值，其值越大图像质量越高，取值范围的最大值与图像分辨率有关，单位为 dB。计算公式如下：

$$PSNR = 10 \times \lg \frac{(2^n - 1)^2}{MSE}$$

其中  $n$  为每像素的比特数， $MSE$  为前述均方误差值。

基于感知模型和图像结构的评估方法是将图像质量下降转化为感知结构信息的变化。结构相似性 (**Structural Similarity, SSIM**) [34] 是衡量参考图像和原始图像相似度的指标，且已应用于医疗图像处理 [35]。由于图像具有结构性特征，表现在图像的像素间存在着较强的相关性，这些相关性往往会携带关于物体结构的重要信息，因此可以通过探测结构信息是否改变来感知图像失真的近似信息。**SSIM** 以每个像素点为中心的图像块 ( $11 \times 11$ ) 为单位，通过亮度 (**luminance**)、对比度 (**contrast**) 和结构 (**structure**) 信息对比图像块间的结构相似性，并分别应用均值、方差和协方差三种统计特征对上述信息进行度量。**SSIM** 取值范围为  $[0,1]$ ，值越大图像质量越高。其计算公式如下：

$$SSIM(x, y) = [l(x, y)]^\alpha \times [c(x, y)]^\beta \times [s(x, y)]^\gamma$$

式中

$$l(x, y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}$$

$$c(x, y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}$$

$$s(x, y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3}$$

其中  $x, y$  分别为参考图像和扩增图像， $\mu_x, \mu_y, \sigma_x, \sigma_y, \sigma_{xy}$  分别是图像  $x$  和  $y$  的局部均值、方差和协方差， $C_1, C_2, C_3$  是为了避免分母为零而设置的，通过参数  $\alpha, \beta, \gamma$  可调整亮度、对比度和结构信息在算法中所占的比例。实际使用时，为了简化计算，一般设置  $\alpha = \beta = \gamma = 1, C_3 = C_2/2$ ，得到：

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}$$

多尺度结构相似性 (**Multiscale Structural Similarity, MSSSIM**) [36] 在 **SSIM** 指标的基础上提出。通过不断对原图像进行低通滤波，不断得到不同分辨率尺度的对比度  $c(x, y)$  和结构  $s(x, y)$  特征值，亮度  $l(x, y)$  则保持不变，以模拟主观在不同分辨率下的质量评估情况。计算公式如下：

$$MSSSIM = [l_M(x, y)]^{\alpha M} \prod_{j=1}^M [c_j(x, y)]^{\beta_j} [s_j(x, y)]^{\gamma_j}$$

特征相似性 (Feature Similarity, FSIM) [37] 提出人类视觉系统主要根据图像低维特征对图像进行理解, 采用图像特征替代 SSIM 中的统计特征, 并选择相位一致性和梯度幅值特征构建 FSIM。相位一致性特征 (Phase Congruency, PC) 能够较好地刻画图像局部结构, 梯度幅值特征 (Gradient Magnitude, GM) 则用来弥补 PC 对于图像变化时的相对不变性。PC 采用 Kovessi[38] 的方法实现, PC 计算公式如下:

$$PC_{2D}(x) = \frac{\sum_j E_{\theta_j}(x)}{\varepsilon + \sum_n \sum_j A_{n,\theta_j}(x)}$$

梯度幅度特征 GM 可以通过 Sobel、Prewitt、Scharr 等算子分别计算水平和垂直方向的梯度, 然后计算最终梯度:

$$G(x) = \sqrt{G_x(x)^2 + G_y(x)^2}$$

通过上述公式, 分别计算参考图像 1 和扩增图像 2 的 PC 相似度  $S_{PC}(x)$ , GM 相似度  $S_G(x)$  和 PC、GM 融合相似度  $S_L(x)$ :

$$S_{PC}(x) = \frac{2PC_1(x) \cdot PC_2(x) + T_1}{PC_1^2(x) + PC_2^2(x) + T_1}$$

$$S_G(x) = \frac{2G_1(x) \cdot G_2(x) + T_2}{G_1^2(x) + G_2^2(x) + T_2}$$

$$S_L(x) = [S_{PC}(x)]^\alpha \cdot [S_G(x)]^\beta$$

最终 FSIM 值计算公式为:

$$FSIM = \frac{\sum_{x \in \Omega} S_L(x) PC_m(x)}{\sum_{x \in \Omega} PC_m(x)}$$

其中  $PC_m(x) = \max(PC_1(x), PC_2(x))$ 。

梯度幅值相似性偏差 (Gradient Magnitude Similarity Deviation, GMSD) [39] 基于图像中不同结构在失真时会产生不同梯度幅值退化的观点, 提出以局部梯度幅值相似性衡量局部图像质量, 再由局部图像质量的标准差对图像整体质量进行衡量。GMSD 值越小, 图像质量越高。首先采用 Prewitt 算子与图像进行卷积操作, 获取图像在水平和垂直方向的梯度幅值  $m_r, m_d$ , 则在像素点  $i$  处的梯度幅值相似性 GMS 为:

$$GMS(i) = \frac{2m_r(i)m_d(i) + c}{m_r^2(i) + m_d^2(i) + c}$$

由于图像中通常具有多种不同类型的局部结构，当图像在扩增过程中产生失真时，不同的局部结构可能受到不同程度的降解。因此，选用标准差池化方法计算整体图像的 GMSD：

$$GMSD = \sqrt{\frac{1}{N} \sum_{i=1}^N (GMS(i) - GMSM)^2}$$

其中 GMSM 为 GMS(i) 在图像上的平均池化值。

### 3.7 本章小结

本章基于医疗图像扩增系统的业务流程和应用场景，本章首先从功能需求和非功能需求两方面对系统需求进行建模并进行用例设计，其次通过系统架构和 4+1 视图进行系统整体设计，最后分别对系统中划分的扩增任务执行控制模块、医疗图像变异扩增模块和扩增质量多维评估模块进行初步的架构和流程设计。

## 第四章 系统详细设计与具体实现

### 4.1 扩增任务执行控制模块

扩增任务执行控制模块主要负责对数据集和扩增任务的管理，以及实现扩增任务执行控制的业务流程，即基于用户上传的扩增数据集，调用医疗图像变异扩增工具执行扩增任务，并输出可由用户下载的扩增数据集，可用于后续对医疗相关模型的训练和评估。同时也可调用扩增质量多维评估工具对扩增数据集的质量进行评估，并生成可视化评估结果。该模块作为系统的控制枢纽，链接前端实现与扩增工具，为不同的工具提供统一的接口规范，能够方便地将新工具集成到系统中，使系统具有较强的可扩展性和可维护性。

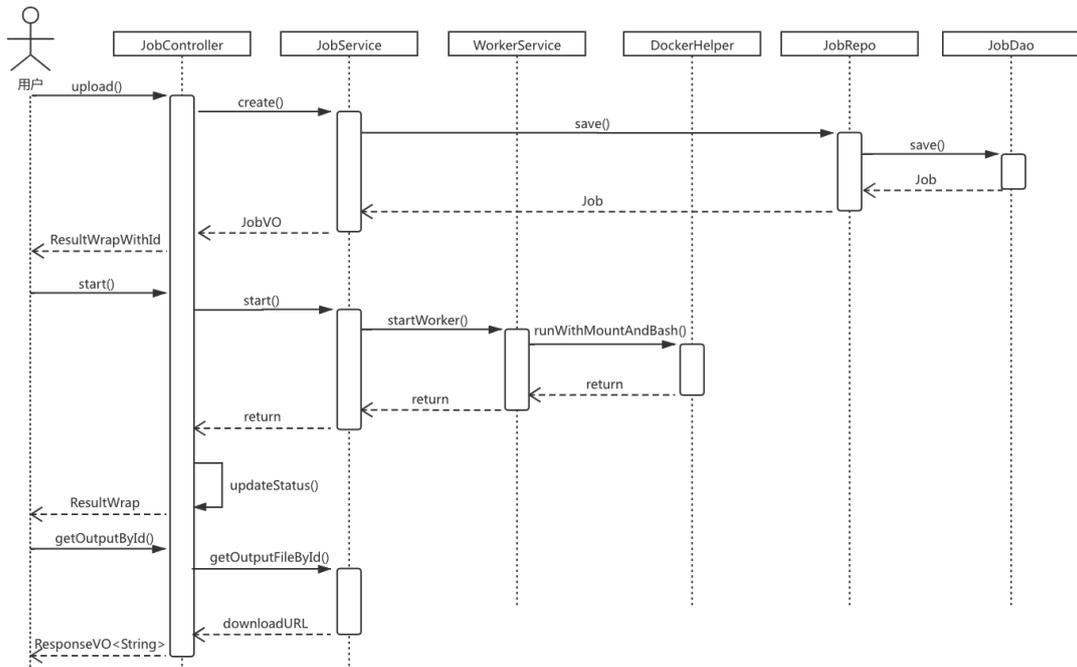


图 4.1: 扩增任务执行控制模块时序图

扩增任务执行控制模块时序图如图 4.1 所示，主要针对模块中创建和执行任务的主流程进行分析。用户在前端填写创建任务表单的相关信息后，系统将任务名称、数据集集合、方法参数配置等信息封装成 `JobUploadCommand` 对象，通过 `upload` 方法调用 `JobService` 中的 `create` 方法进行任务创建，将 `JobUploadCommand`

对象转换为 Job 对象，再由 JobRepository 和 JobDao 将新建的 Job 转换为持久化的 JobPO 对象并存储到 MySQL 数据库中。当用户填写完任务表单并点击创建任务后，系统将通过 JobService 中的 start() 方法启动该任务，并由 WorkerService 创建基于当前任务配置参数的 Worker，再由 Worker 创建 Docker 容器实例，实现扩增任务的执行。在扩增任务 Docker 容器运行期间，系统支持对于任务状态的轮询，例如当刷新扩增任务列表界面时，系统将对列表中获取到的每个任务进行状态的轮询和更新。当扩增任务完成后，系统将调用 JobController 中的 getOutputById 方法获取扩增结果数据集的下载链接，并将其返回前端，供用户进行结果下载。

扩增任务执行控制模块中主要类的信息如表 4.1 所示，包括控制器层、服务层、数据仓库层、数据访问层和数据层，其中数据层又包含持久化对象层、领域对象层和视图对象层。控制器层主要负责前后端数据的交互，通过调用服务层的方法将视图对象层的对象传入前端，并在前端进行显示和渲染。服务层则对业务逻辑中的具体方法进行实现，其中包括对数据集进行上传、查询、删除、下载、更新等操作的 DataSetService，对任务进行创建、执行、查询、更新任务状态等操作的 JobService，和对 Worker 进行创建、执行、查询等操作的 WorkerService。数据仓库层位于业务层和数据层之间，通过调用数据访问层的方法完成数据查询、存储等逻辑，数据访问层则向系统上层提供面向对象的数据访问接口，以完成对数据库的各种访问操作。在数据层部分，视图对象层负责将与前端相关的数据和方法进行封装；领域对象层将业务逻辑中的操作对象抽象成业务实体，其中封装了向持久化对象和视图对象的转换方法，以及对实体内部属性的相关设置方法；持久化对象层中类的属性与持久层数据库中的数据表结构形成一一对应，便于对数据进行持久化处理。

表 4.1: 扩增任务执行控制模块主要类

层级	主要类
控制器层	DataSetController、JobController、WorkerController、BaseSearchController、CommonController
服务层	DataSetService、JobService、WorkerService、CommonService
数据仓库层	DataSetRepo、JobRepo、WorkerRepo
数据访问层	DataSetDao、JobDao、WorkerDao
视图对象层	DataSetVO、JobVO、WorkerVO、SearchConditionVO
领域对象层	DataSet、Job、Worker、StatisticalData
持久化对象层	DataSetPO、JobPO、WorkerPO



**Job** 类的主要字段描述如表 4.2 所示。**Job** 表示由用户创建并管理的扩增任务，其中包含执行任务时需要读取或写入的文件路径、任务状态以及分别面向前后端的配置字段等。异步扩增任务通过其参数创建 **Worker** 并调用 **Docker** 扩增容器，完成扩增任务的创建和执行。

表 4.2: Job 类主要字段列表

字段	含义	类型	备注
id	任务编号	long	主键，不为空
createUserId	用户编号	long	创建该任务的用户编号
workerId	执行编号	long	用于执行该任务的 worker 编号
status	任务状态	String	任务对应的状态，包括 unstarted、running、finished、fail
dataSetIds	数据集编号集	String	任务需扩增的原始数据集集合
backConfig	后端配置参数	String	任务的配置参数，从前端接收，由后端存储，在调用 worker 时传入
frontConfig	前端配置参数	String	任务的配置参数，由 worker 提供，后端传给前端，前端用于展示
outputDirPath	结果输出路径	String	任务完成后扩增数据集的存储路径

**Worker** 类的主要字段描述如表 4.3 所示。**Worker** 在执行特定扩增任务时由 **Job** 创建，其主要字段包括任务执行类型、工具运行配置等。**Worker** 可以执行基于不同类型数据集的扩增、分类、评估等任务，具体任务参数需要通过 **Job** 传参指定。如对于医疗图像扩增任务，定义任务类型 **type** 为 **amplify**，数据集类型为图像，并通过 **docker** 运行容器执行任务。

表 4.3: Worker 类主要字段列表

字段	含义	类型	备注
id	执行编号	long	主键，不为空
type	任务类型	int	任务类型，包括 amplify、classify、estimate
inputDataSetType	数据集类型	int	任务输入的数据集类型，包括 image、video、text
execType	任务执行类型	int	执行任务的方法类型，包括 docker、rpc
execConfig	工具运行配置	String	可执行文件路径

**DataSet** 类的主要字段描述如表 4.4 所示。**DataSet** 类与用户上传的数据集一一对应，作为任务配置参数参与扩增任务创建和执行。其主要字段包括类型、上

传时间、文件数量等，用户上传数据集后，系统将其相关字段存入数据库中，并在前端数据集列表中显示当前用户已上传的所有数据集。

表 4.4: DataSet 类主要字段列表

字段	含义	类型	备注
id	数据集编号	long	主键，不为空
name	数据集名称	String	数据集对应名称，可修改
uploadUserId	用户编号	long	上传该数据集的用户编号
uploadTime	上传时间	Timestamp	数据集上传时间
type	数据集类型	int	数据集内文件类型，包括 image、video、text
fileCount	文件数量	long	数据集内文件数量
dirPath	数据集路径	String	数据集存储文件夹的路径

```
private void start(final Job job) throws IOException, InterruptedException {
    log.info(String.format("Prepare to start a job: %s", job.toString()));
    Worker worker = workerRepo.findByIdAndIsDeleteFalse(job.getWorkerId());
    if( worker == null || worker.isDelete() ) {
        throw new DataNotExistException("worker: id="+ job.getWorkerId());
    }
    // 检查数据集有效性
    for( long id : job.getDataSetIdsWithLongType() ) {
        DataSet dataSet = dataSetRepo.findByIdAndIsDeleteFalse(id);
        if( dataSet == null || dataSet.isDelete() ) {
            throw new DataNotExistException("dataset: id="+ job.getDataSetIds());
        }
    }
    int execType = worker.getExecType();
    if( job.getType() == JobRepo.AMPLIFY_TYPE ) {
        // 调用对应 worker 执行扩增任务
        workerService.startWorker( execType, this.job.getDataSetDirPaths(job),
            job.getOutputDirPath(), job.getStatusFilePath(), job.getLogFilePath(), job.getId(),
            job.getBackConfig(), worker.getName() );
    } else if( job.getType() == JobRepo.ESTIMATE_TYPE ) {
        ... //模型评估任务代码
    }
    // 设置任务状态
    job.setStatus( JobRepo.RUNNING_STATUS );
    jobRepo.update( job );
    log.info("start job success");
}
```

图 4.3: 执行扩增任务核心代码

下面对系统扩增任务执行核心流程进行详细描述。当用户点击创建任务按钮，系统调用 JobService 中的 create 方法将该任务内数据集、方法参数等相关信息封装成 Job 对象，并通过 JobRepo、JobDao 数据访问层将其存储到数据库中。然后系统将调用 JobService 中的 start 方法执行该任务，从数据库中找到 Job 对应 worker，同时判断该 worker 和任务中输入数据集的有效性，再根据任务执行类型，传入任务执行参数并调用对应 startWorker 方法执行图像扩增任务，最后

设置当前任务状态，并在数据库中更新对应任务的状态字段。其核心代码如图 4.3 所示。

为了保证多个扩增任务可以异步执行，提高系统效率，每个处于运行状态的扩增任务都将独占一个线程，系统可以在任意时间对任务状态进行轮询，一旦任务完成，就对前后端任务状态进行更新，同时在数据库中更新该任务的状态。对于扩增结果的读取，系统在调用 Docker 运行容器进行任务执行时，将会传入结果存储路径参数，即图 4.3 代码中的 `job.getOutputDirPath()`，当 Docker 容器完成扩增任务后，会将扩增数据集存储到该文件路径中。因此，当任务启动后，系统不需要对任务执行情况进行监控，只需要在查询到任务状态为已完成时，调用 `jobService` 接口中的 `getOutputFileById` 方法获取扩增结果，并封装成数据集下载链接传回前端。其核心代码如图 4.4 所示。首先获取对应任务的结果存储路径，再将扩增数据集文件夹压缩成 zip 文件，最后返回文件下载链接。

```
public String getOutputFileById(long id) throws IOException, InterruptedException {
    JobVO jobVO = findByIdAndIsDeleteFalseAndUpdateStatus( id );
    String outputDirPath = jobVO.getOutputDirPath();
    String outputDirName = FileHelper.getFileNameByPath( outputDirPath );
    // 将压缩的 zip 包放在服务器的/var/www/jobResult 目录下
    File outputZip = new File(outputZipPath + jobResultFolder + outputDirName+".zip");
    if(!outputZip.exists()){
        ZipAndRarHelper.toZip( outputDirPath, new FileOutputStream(outputZip), true );
    }
    return "http://" + serverIP + "/" + jobResultFolder + outputDirName+".zip";
}
```

图 4.4: 获取扩增结果核心代码

## 4.2 医疗图像变异扩增模块

医疗图像变异扩增模块用 Python 实现，主要负责实现执行扩增任务并实现具体的扩增方法，其本质是由 Dockerfile 自动化构建的 docker 镜像。首先从扩增任务执行控制模块接收原始数据集、扩增方法配置等信息，再调用相应的扩增方法对数据集进行扩增处理，最后再将扩增数据集及任务完成状态信息回传到扩增任务执行控制模块。在实际执行扩增任务时，先由 docker 镜像构建出执行具体扩增方法的容器，在容器根目录下，存放 `main.sh` 文件，作为容器的调用入口，当扩增任务执行完成后，容器将扩增结果保存到输出文件夹路径，然后自动清除容器相关内容。该模块将变异分析思想应用于图像扩增，实现的扩增方法包括图像特征变异扩增方法和领域语义变异扩增方法，分别针对图像基础属性特征和医疗领域语义特征进行变异分析，以挖掘其潜在信息，生成丰富且具有多样性的扩增数据集。

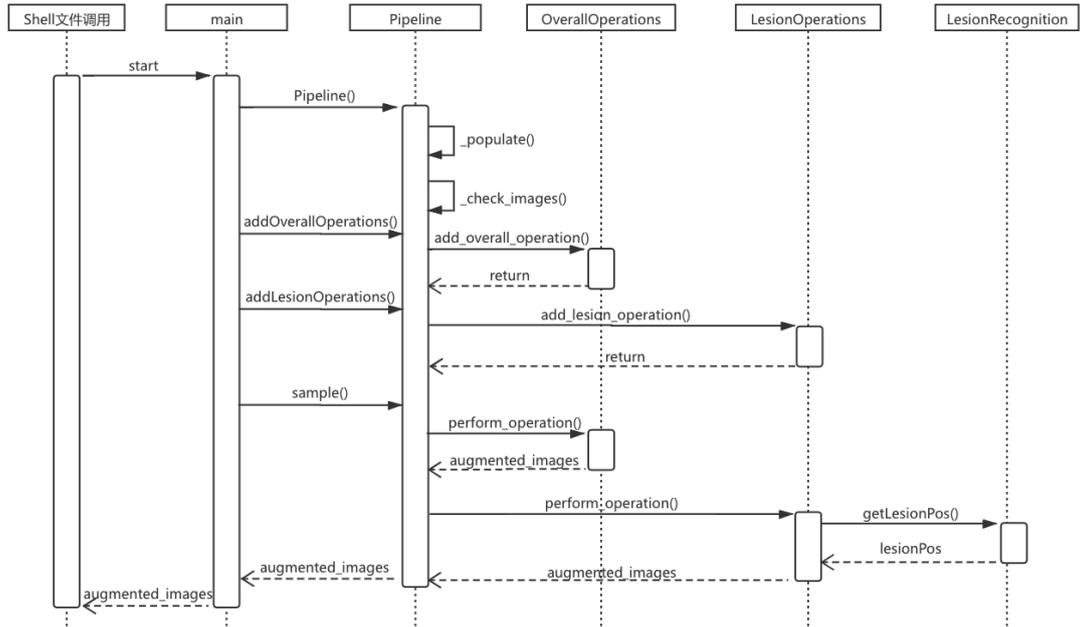


图 4.5: 医疗图像变异扩增模块时序图

医疗图像变异扩增模块时序图如图 4.5 所示，主要说明执行扩增任务的流程。首先由扩增任务执行控制模块通过 Shell 脚本文件调用扩增工具中的 main 接口，并由 main 接口通过 Pipeline() 方法生成一个新的扩增管道 pipeline，接着该管道将使用 Pipeline 类中私有的 populate 方法解压原始数据集并将其中的图像存储到临时文件夹中，同时将图像加载到管道的图像流 augmented\_images，以便进行后续扩增操作。在成功读取数据集后，该扩增管道用 Pipeline 类中私有的 check\_images 方法对图像进行预处理，剔除过小的图像、空数据和错误数据。完成数据的预处理过程后，main 接口将读取从扩增任务执行控制模块接收的扩增方法配置文件 content.json，并从中提取扩增方法列表，逐个将方法添加到扩增管道中。此时 pipeline 将调用对应扩增方法类的 add\_overall\_operation() 方法和 add\_lesion\_operation() 方法完成扩增方法流的添加工作。然后 main 接口调用 sample() 方法，再由 pipeline 调用相应方法的具体实现，即 perform\_operation()，将管道中的图像流依次通过各个扩增方法进行扩增，其中涉及 LesionOperation 的扩增方法还需要调用 LesionRecognition 中的 getLesionPos() 方法获取病灶区域 lesionPos，最终将扩增数据集 augmented\_images 保存到输出文件夹，进行回传。

医疗图像变异扩增模块类图如图 4.6 所示，其中 TauMedImage 为图像实体类，主要负责图像相关信息存储、数据集解压、数据预处理等任务。OverallOp-

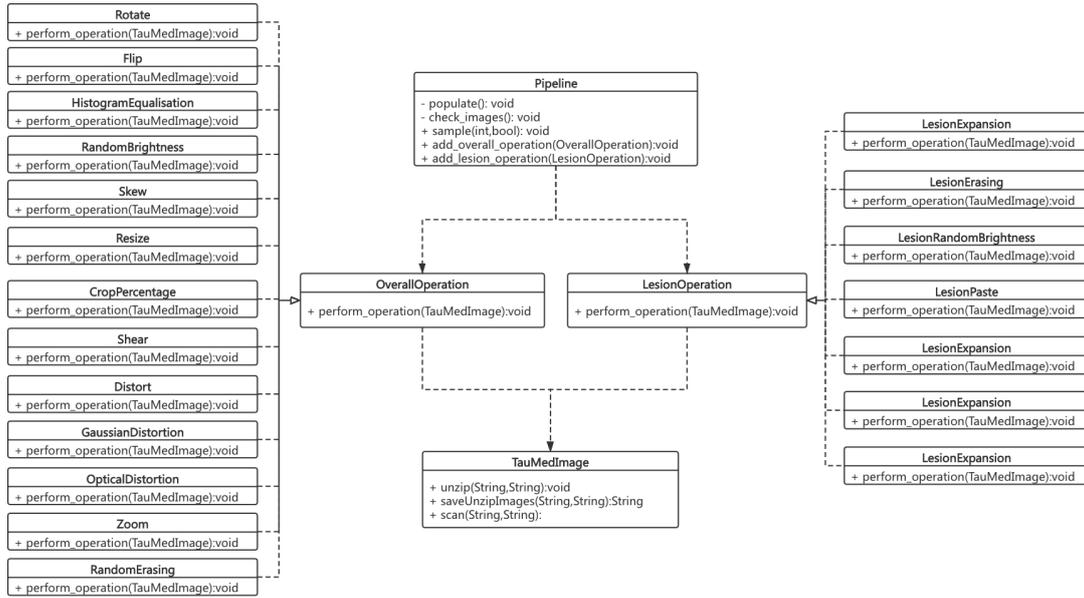


图 4.6: 医疗图像变异扩增模块类图

eration 为图像特征变异扩增方法的基类，基于医疗图像的整体相似性，实现了无损变换、非刚性变换等特征变异扩增方法。LesionOperation 为领域语义变异扩增方法的基类，基于医疗图像的局部多样性，实现了病灶区域变异和非病灶区域变异等语义变异方法。Pipeline 类为执行扩增任务的实体类，是在图像流上执行扩增任务的管道。在创建 Pipeline 对象时，程序同时将调用 Pipeline 中私有的 populate 方法和 check\_images 方法对数据集进行自动读取和预处理，并通过添加图像特征变异扩增方法和领域语义变异扩增方法的函数向 Pipeline 对象中添加若干扩增方法，最后通过 sample 方法完成对数据集的扩增处理。

表 4.5: Pipeline 类属性列表

字段	含义	类型	说明
augmented_images	图像集合	List<TauMedImage>	存储原始数据集中的图像以及扩增后的图像
save_format	存储类型	String	设置图像扩增后的存储格式
overall_operations	图像特征变异扩增方法集合	List<OverallOperation>	存储添加的特征变异扩增方法
lesion_operations	领域语义变异扩增方法集合	List<LesionOperations>	存储添加的语义变异扩增方法

Pipeline 类的详细描述如表 4.5 所示，包含图像集合和方法集合，便于在图像流上依次执行多种扩增方法。其中 `augmented_images` 中存储的是 `TauMed-Image` 类的对象集合，对象内部又包含原始图像路径 `image_path`、输出图像路径 `output_directory` 等属性，能更便捷地对图像进行读取和存储。`save_format` 表示扩增后图像的存储格式，可在创建 `Pipeline` 对象时作为参数进行赋值。`overall_operations` 和 `lesion_operations` 分别存储图像特征变异扩增方法集合和领域语义变异扩增方法集合，在 `Pipeline` 对象调用扩增方法时，将对应的扩增方法类添加到方法集合中，并在扩增任务实际执行过程中调用相应扩增方法类的 `perform_operation` 方法，对图像进行具体变换处理。

```
#!/bin/bash
if [ $# -gt 0 ]
then
  case $1 in
  start)
    tmp_path=$7
    start_time=$(date +%s%3N)
    status=0
    filepath=$4
    echo $filepath
    export UNRAR_LIB_PATH=/usr/lib/libunrar.so && python3 /TauMed/main.py 0 $2 $3 $5 $6
$7 2> "$tmp_pathErr_$job_id.txt"
    if [ $? -eq 0 ];then
      status=1
    else
      status=3
    fi
    echo "{" > $filepath
    echo "'version': 1.0," >> $filepath
    end_time=$(date +%s%3N)
    failMsg=$(echo $a | sed -n '$p' "$tmp_pathErr_$job_id.txt")
    echo "'state': $status," >> $filepath
    echo "'startTime': $start_time," >> $filepath
    echo "'endTime': $end_time," >> $filepath
    echo "'params': 'python3 /TauMed/main.py 0 $2 $3 $5 $6 $7'," >> $filepath
    echo "'failMsg': '$failMsg'," >> $filepath
    echo "}" >> $filepath
    ;;
  *)
    echo "The operation is not exist!"
    ;;
  esac
else
  echo "Please input operation!"
fi
```

图 4.7: 扩增工具调用核心代码

由于每创建一个新的扩增任务，都需要对医疗图像变异扩增模块进行调用，即由 `Docker` 镜像生成具体执行任务的容器。为了简化流程，便于参数传递，将工具封装为标准 `Linux` 命令行接口，并使用文本进行数据传输，同时在容器根目录下存放 `main.sh` 文件作为容器的调用入口，其核心代码如图 4.7 所示。`main.sh`

用 `start` 操作原语完成对扩增任务的启动操作，其中实际执行的是工具中面向系统的扩增接口 `main.py`，同时将数据集路径、配置文件等信息通过命令行参数传输到工具中，并将任务运行状态、运行时间、错误信息等写入状态文件中。

根据医疗图像变异扩增模块的业务流程，扩增工具需要根据创建任务时用户选择的扩增方法及其相应参数才能实现对实际方法的调用，因此将方法配置信息写入 `content.json` 文件中，并在命令行调用时进行传输，其示例代码如图 4.8 所示。其中 `output_num` 用于指定图像扩增数量，`operations` 中存储用户选择的扩增方法列表，包括方法类型 `type` 和对应参数列表 `param`。

```
{
  "output_num":100,
  "operations": [{"type": "Flip", "param": [1,0]},
                 {"type": "Rotate", "param": [1,23,2]},
                 {"type": "LesionErasing", "param": [1]}
                ]
}
```

图 4.8: 方法配置文件示例代码

#### 4.2.1 医疗图像病灶识别子模块

医学图像是解剖学区域的内部结构或功能的表示。在医疗图像中，最重要的病理信息集中表现在病灶区域，即一个具有病原微生物的局部病变组织，其中征象则是不同病症在病灶区域的不同病理表现。由于医疗图像的特殊性，其细节和征象特征更多依赖于图像纹理、像素值等低层特征，因此对于征象的识别和定位需要更多地考虑图像低层特征，同时也需要结合高层特征中的轮廓、类别等信息。

因此，针对以上医疗图像的特性，在选择病灶识别模型时，需要更关注图像底层特征的网络。在经过大量调研后，最终选用 `PANet` 模型完成对医疗图像的病灶区域识别。一方面，模型在特征金字塔 `FPN` 后添加自下而上的路径增强，将图像底层特征传导到高层，增强底层特征在高层的表达。另一方面，模型采用适应性特征池化技术，将不同层级的特征网格通过区域特征聚集方式进行池化，然后再将各个特征层级的特征信息进行融合，即结合图像底层纹理特征和高层轮廓特征，使最终的病灶区域识别结果更符合实际。

`PANet`[23] 是基于 `Mask R-CNN` 进行改进后的实例分割框架。首先通过自下而上的路径增强 (`Bottom-up Path Augmentation`) 来缩短信息路径和用低层级的

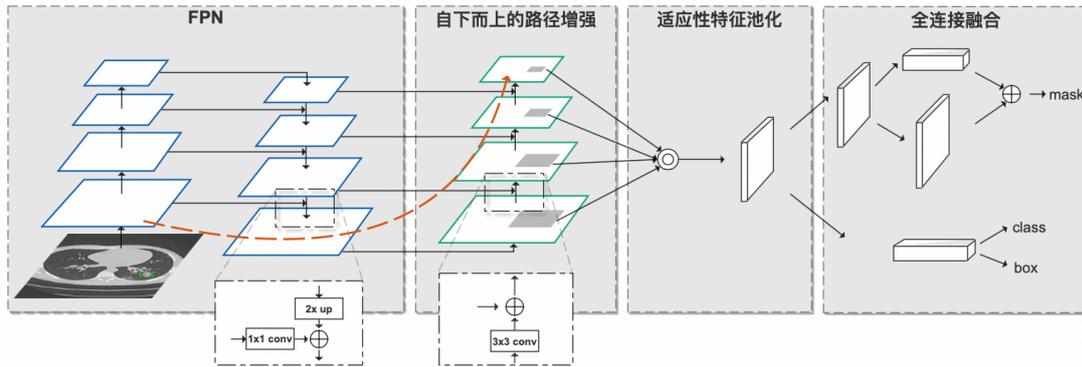


图 4.9: PANet 模型算法结构图

准确定位信息增强特征金字塔。其次，为了恢复每个建议区域和所有特征层级之间被破坏的信息，提出了适应性特征池化（Adaptive Feature Pooling）技术，将所有特征层级中的特征整合到每个建议区域中。最后，为了捕捉每个建议区域的不同视野，使用全连接层融合（Fully-connected Fusion）来增强 Mask 预测，其算法网络结构如图 4.9 所示。下面对 PANet 整体算法框架各个部分进行简要介绍。

FPN（Feature Pyramid Networks, 特征金字塔网络）通过自底向上（Bottom-up），自顶向下（Top-down）以及横向连接（Lateral Connection）将特征图进行高效整合，在尽可能不影响算法效率的同时提升精度。自底向上即卷积网络逐层浓缩表达特征的过程，其中较低层反映较浅层次的图片信息特征如纹理、边缘等，较高层则反映较深层次的图片特征如物体轮廓、类别等。结合自顶向下和横向连接，将自底向上获得的多个特征图进行整合，并将高层的强语义特征向下传递，增强语义信息。其中自顶向下和横向连接的计算过程如图中 FPN 部分虚线框所示，将各层输入进行  $1 \times 1$  卷积并与上采样后的高层特征图的对位元素进行相加。

由于医疗图像中的征象识别分类与图像纹理、像素值等较低层特征变化相关，因此病灶分割时更需要低层维度的信息。自下而上的路径增强（Bottom-up Path Augmentation）可以传播低层的纹理、像素定位等特征信息并增强整个特征层，从而提升实例分割质量。其具体流程如虚线框中所示，将低层输入进行  $3 \times 3$  卷积后与左侧输入相加后的结果传输到高层。自下而上的路径增强将低层特征信息传导到高层，同时由于横向连接操作减少了高层到低层的中间信息传输过程，即如图中橙色虚线所示。

将自下而上的路径增强后得到的感兴趣区域 (ROI) 进行适应性特征池化 (Adaptive Feature Pooling)。对于每个候选 ROI, 即图中每层的灰色区域, 将其映射到不同的特征层次, 再使用 ROI Align 池化来自不同层次的特征网格, 最后将多层次的 ROI 特征进行融合, 使其包含更加丰富的特征信息。

最后, 在 FCN (Fully Convolutional Network, 全卷积网络) 中添加全连接层融合支路。FCN 层根据局部接受域给出每个像素的预测, 并且在不同的空间位置共享参数。相反, 全连接层是位置敏感的, 因为不同空间位置的预测是通过不同的参数集实现的。前者关注局部信息, 后者关注整体信息, 从而提升最终 Mask 的质量。

本模块使用 PANet 模型在由 32120 张包含病灶区域信息的医疗图像组成的 DeepLesion 数据集下进行训练, 并用训练得到的模型对图像的病灶区域进行预测, 最后得到病灶区域矩形框的顶点坐标值, 并基于该参数完成领域语义变异扩增操作。医疗图像病灶识别子模块首先加载已训练完成的 PANet 模型, 然后将图像输入该模型进行病灶区域预测, 最后得到病灶矩形区域左上角和右下角的坐标值。

#### 4.2.2 图像特征变异扩增子模块

特征变异指通过提取数据不同维度的特征, 并在该特征上进行不同方向不同程度的变化。图像特征一般包括像素空间特征、像素值域特征、纹理特征、形状特征等, 分别基于上述特征对图像进行不同方面的变异, 以达到扩增图像在不同特征维度的差异化和多样性。

表 4.6 中列举了系统中实现的 13 种图像特征变异扩增方法。对于医疗图像而言, 基于图像特征变异的图像扩增技术主要是针对医疗图像在单样本上表现出的图像本质特征进行变异, 以模拟现实中病症的多样性以及医疗图像成像的多样性。由于人体器官组织本身相似度较高, 因此绝大多数同部位、同体态的医疗图像具有较高的相似度, 因此医疗图像具有整体相似性的特征。一方面, 由于一张医学图像中包含的所有信息都具有潜在利用价值, 为了保证医疗图像的整体相似性, 避免破坏医疗图像中的关键结构, 造成重要信息的缺失, 基于图像特征变异规则, 选用最大程度保留图像内容的扩增方法对整体的医疗图像进行处理。另一方面, 宏观来看, 人体组织器官是非刚性的, 并且微观上细胞生长或病变的过程也是非刚性、非线性的, 因此非刚性变换也有助于增强数据, 可以使扩增后的医疗图像具有时域变化的特征。因此, 在该模块中, 选择实现无损变换、非刚性变换和基本图像变换方法对图像进行扩增处理。

表 4.6: 图像特征变异扩增方法列表

扩增方法	方法介绍	方法原理
水平翻转	沿图像水平方向翻转	翻转后图像的每个像素的横坐标相对于图像的垂直中心线对称于翻转前相应像素的横坐标
垂直翻转	沿图像垂直方向翻转	翻转后图像的每个像素的横坐标相对于垂直翻转中图像的水平中心线对称于翻转前相应像素的横坐标
随机旋转	将图像旋转任意角度	以旋转中心为原点, 将图像旋转角度 $\theta$
倾斜变换	将图像向任意方向倾斜	在水平和垂直方向上将单元内的每个点扭曲一定的角度
弹性变换	在保持图像的纵横比下对图像进行弹性变换	由网格宽度和高度控制扭曲的精细程度
高斯畸变	在保持图像的纵横比下对图像进行高斯畸变	使用高斯滤波过滤像素每个维度的偏差矩阵, 并用新像素值替代原始像素位置的值
光学失真	在保持图像的纵横比下对图像进行光学失真	包括桶形和枕形畸变
随机裁剪	根据要获取的区域百分比裁剪图像的随机区域	在图像的随机区域进行裁剪
错切变换	将图像错切指定的度数	以某一系数对单方向的像素点进行变换
随机擦除	对图像任意区域进行随机擦除	通过用随机像素值替换任意位置的矩形区域
缩放变换	对图像大小进行调整	对图像进行放大或缩小, 同时保持其原尺寸大小
直方图均衡化	利用图像直方图调整图像对比度	把原始图的直方图变换为均匀分布的形式
随机亮度	对图像亮度进行调整	随机改变图像亮度

图 4.10 是基于原始肺部 CT 图像通过上述 13 种图像特征变异扩增方法处理后的扩增图像示例, 分别实现了针对像素空间特征、像素值域特征、图像尺寸特征和图像纹理特征的变异扩增处理, 从中可直观看出各种特征变异方法并未大幅度破坏医疗图像原有的空间结构。

### (1) 基于像素空间特征的变异扩增方法

基于像素空间特征的变异扩增方法表示在不改变图像蕴含的疾病特征的前提下, 对像素矩阵中的各个像素的空间位置进行变换操作, 以避免破坏医疗图像的关键结构。空间特征变异扩增方法包括水平翻转、垂直翻转、随机旋转、倾斜变换和错切变换, 基本采用矩阵仿射变换操作进行实现。

水平翻转即沿图像水平方向翻转, 翻转后图像的每个像素的横坐标相对于图像的垂直中心线对称于翻转前相应像素的横坐标。假设一幅图像的大小为  $3 \times 1$ , 其矩阵表示为  $\begin{bmatrix} x & y & z \end{bmatrix}$ , 其中  $x, y, z$  分别为该图像中像素点的值, 则其水平翻转后的结果为:

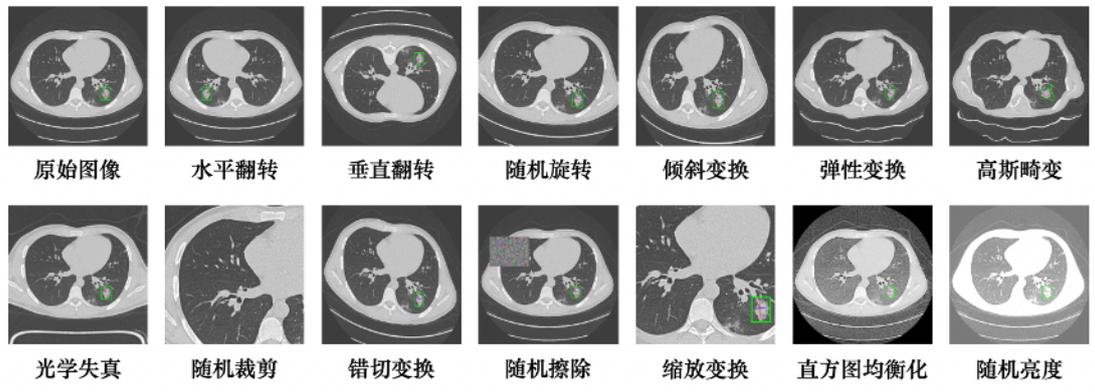


图 4.10: 图像特征变异扩增方法示例

$$HorizontalFlip\left(\begin{bmatrix} x & y & z \end{bmatrix}\right) = \begin{bmatrix} z & y & x \end{bmatrix}$$

与水平翻转类似，垂直翻转是沿图像垂直方向翻转，翻转后图像的每个像素的横坐标相对于垂直翻转中图像的水平中心线对称于翻转前相应像素的横坐标。同样假设一幅图像的大小为  $1 \times 3$ ，其矩阵表示为  $\begin{bmatrix} x \\ y \\ z \end{bmatrix}$ ，其中  $x, y, z$  分别为该图像中像素点的值，则其垂直翻转后的结果为：

$$HorizontalFlip\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) = \begin{bmatrix} z \\ y \\ x \end{bmatrix}$$

随机旋转即将图像旋转任意角度，以旋转中心为原点，将图像旋转角度  $\theta$ 。用户可以指定向左旋转最大角度和向右旋转最大角度，工具将在这两个角度范围内将输入图像随机旋转，并裁剪为原图像尺寸。旋转通过变换矩阵实现，即

$$M = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ -cX & cY & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ left & top & 1 \end{bmatrix}$$

其中  $cX$  为图像中心点  $x$  坐标， $cY$  为图像中心点  $y$  坐标， $\theta$  为旋转角度。该变换矩阵的变换过程为：首先将坐标的原点从左上角变换到图像的中心（第一个矩阵），然后以旋转中心为原点，将图像旋转角度  $\theta$ （第二个矩阵），旋转完成

后，再将坐标原点恢复到旋转图像的左上角（第三个矩阵）。通过将图像矩阵和变换矩阵  $M$  相乘完成旋转操作。

倾斜变换即在三维视觉上将图像向任意方向倾斜，通过在水平和垂直方向上将单元内的每个点扭曲一定的角度实现。每个点的坐标根据指定的角度以及到原点的距离，进行成比例的值调整。倾斜变换可以模拟从不同角度查看图像的情况。倾斜变换同样通过变换矩阵实现，设图像中某一像素点坐标为  $(x, y)$ ，则其变换后位置坐标为：

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & b & t_x \\ c & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

其中  $t_x$  为对  $x$  轴逆时针旋转的角度， $t_y$  为对  $y$  轴顺时针旋转的角度。设原图像大小为  $h * w$ ，则  $b = \frac{t_x}{h}$ ， $c = \frac{t_y}{w}$

错切变换即将图像错切指定的度数，以某一系数对单方向的像素点进行变换，类似于四边形的不稳定性。错切变换分为  $X$  方向和  $Y$  方向的变换，设原始图像中任意点坐标为  $(x, y)$ ，则其经过  $X$  方向  $\theta$  角度错切变换后的坐标  $(x', y')$  为：

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & \tan \theta & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

经过  $Y$  方向  $\theta$  角度错切变换后的坐标  $(x', y')$  为：

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ \tan \theta & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

## (2) 基于像素值域特征的变异扩增方法

基于像素值域特征的变异扩增方法是指在一定程度内，对图像内局部或整体的像素值进行调整，在视觉层面上表现为颜色的改变。像素值域特征变异扩增方法包括随机擦除、直方图均衡化和随机亮度，由于医疗图像一般为灰度图，因此对多种颜色的变换方式并不适用。

随机擦除即对图像任意位置进行指定区域大小的擦除，通过用随机像素值替换任意位置的矩形区域，使模型对遮挡具有鲁棒性。用户可指定擦除区域占

图像的最大百分比，工具将在最小占比（0.1）和该指定值之间取随机大小的区域，生成随机像素值生成的擦除区域，并覆盖在图像的任意位置。

直方图均衡化是利用图像直方图对图像的对比度进行调整，即通过拉伸像素强度分布范围把原始图的直方图变换为均匀分布的形式，增加了像素灰度值的动态范围，从而提高图像整体对比度，达到改善图像视觉效果的目的。对于医疗图像，这种方法可以提高 CT 图像中器官结构显示效果，以及在曝光过度或不足的图像中增强其细节表现。直方图均衡化的计算公式为：

$$h(v) = \text{round}\left(\frac{cdf(v) - cdf_{min}}{(M * N) - cdf_{min}} * (L - 1)\right)$$

其中  $v$  为原始图像中某点的灰度值， $cdf(v)$  为灰度值等于  $v$  的累积概率分布函数， $cdf_{min}$  为该函数最小值， $M$  和  $N$  分别代表了图像的长宽像素个数， $L$  则为灰度级数，一般为 256。

随机亮度是指对图像整体亮度进行调整，可以模拟不同光照下医生阅片的情况或不同医疗成像设备的输出差异。对比度和亮度一般通过以下变换公式进行调整：

$$g(x) = \alpha f(x) + \beta$$

其中  $\alpha$  和  $\beta$  一般称为偏置和增益参数，分别对对比度和亮度进行控制， $f(x)$  为源图像像素，而  $g(x)$  为输出图像像素。

### (3) 基于图像尺寸特征的变异扩增方法

图像尺寸特征是指只针对图像的整体尺寸进行改变，而不影响像素等较低维度的特征。图像尺寸特征变异扩增方法包括随机裁剪和缩放变换，这两种方法都可能会改变图像内容特征。

随机裁剪是根据获取的区域百分比值对该图像区域进行裁剪和保留。其中裁剪区域百分比由用户进行指定，而系统将在图像的随机区域执行裁剪操作。

缩放变换是对图像整体进行放大或缩小，并保持其原尺寸大小不变。

### (4) 基于图像纹理特征的变异扩增方法

纹理特征体现图像中具有缓慢变化或周期性变化的像素结构组织排列属性，与上述基于单个像素的图像特征不同，纹理特征通过像素及其周围空间邻域的灰度分布体现。纹理特征由纹理基元构成，纹理基元是指一定的形状和大小的多种像素基元的组合，通过纹理基元的排列组合以及疏密、周期性和方向性的不同，对图像纹理特征进行变异，以达到扩增的效果。其中包括弹性变换、高斯

畸变以及光学失真，这些方法同样属于非刚性变换。非刚性变换表示通过非线性的拉伸、扩张等方法改变图像的内部信息，而不改变其几何形状位置的变换。根据医疗图像的特征来看，非刚性变换有助于医疗领域数据扩增，可以对医疗图像中的细胞、器官变化特征进行模拟，以产生在病症诊断中具有实际意义的新数据。

弹性变换是在保持图像的纵横比下对图像进行弹性变换，其中由纹理基元宽度和高度控制扭曲的精细程度。首先对图像中的每个像素点  $(x, y)$  产生两个-1到1之间的随机数—— $\delta x(x, y)$  和  $\delta y(x, y)$ ，分别表示该像素点的  $x$  方向和  $y$  方向的移动距离，然后生成一个以0为均值，以  $\sigma$  为标准差的大小为  $n \times n$  的高斯核  $knn$ ，并用前面的随机数与之做卷积，并将结果作用于原图像。其中  $\sigma$  的大小与弹性变换的处理结果息息相关，如果  $\sigma$  过小，则生成的结果类似与对图像每个像素进行随机移动，而如果  $\sigma$  过大，则生成的结果与原图基本类似。

高斯畸变是将图像通过网格划分成若干个顶点，通过移动顶点的方式让几何形状发生改变。为像素的每个维度生成一个随机的标准偏差  $(-1, 1)$  间隔，然后使用高斯滤波  $(0, \sigma)$  过滤每个维度的偏差矩阵，最后控制偏差范围放大系数  $\alpha$ 。最后，从  $A(xy)$  获得  $A'(x + \Delta X, y + \Delta Y)$ ，并用  $A'$  的值替代原始  $A$  位置的值。

光学失真是指光学系统在物体上形成的图像相对于物体本身的失真程度，其程度是由光学理论计算出的畸变程度，其概念通常应用于摄影领域，与镜头本身属性有关。应用光学失真扩增方法可以模拟由不同医学影像设备拍摄和成像的医疗图像。光学失真包括桶形和枕形畸变，未变形的图像中的直线以特征性的桶形方式凸出，或者以枕形表示形式向内弯曲。

### 4.2.3 领域语义变异扩增子模块

不同于其他领域，医疗领域图像的重要信息点集中于病灶区域。因此针对医疗图像的特殊性，提出以病灶为核心的扩增思路。首先从医疗图像中识别出不同征象，通过在图像中对识别出的征象进行多样化处理，从同一个体的病情发展阶段和不同个体的病情表现形式两个维度对医疗图像研究扩增方法，以模拟现实中病人患病时可能产生的病症影像学表现。表 4.7 中列举了系统中实现的 8 种领域语义变异扩增方法。

图 4.11 中列举了基于肺部 CT 图像通过上述 8 种领域语义变异扩增方法处理后的图像示例，其中绿框内部为病灶区域。可以发现前五种方法均只针对病灶区域的征象进行了变异处理，而并未改变非病灶区域。后三种方法则相反，保持病灶区域不变以保留其关键信息，并对非病灶区域进行变异处理。

表 4.7: 领域语义变异扩增方法列表

扩增方法	方法介绍	方法原理
病灶擦除	对图像中的病灶区域进行擦除	使用随机像素值替换病灶区域，以创建没有病灶区域的新图像
病灶亮度	随机改变图像病灶区域亮度	将病灶区域随机变亮或变暗，以模拟医生实际读片时光线对病灶区域的影响
病灶复制	对图像中的病灶区域进行复制	增加医学图像中病灶的数目，模拟疾病在不同个体或发展阶段的多样性
病灶扩张	扩张图像中的病灶区域	扩大病变区域，即扩大病变范围，纵向模拟疾病随时间的发展
病灶移动	移动图像中的病灶区域	将提取的病灶区域移动到图像中的随机位置，模拟病灶的活动性或个体表现差异
噪声处理	在图像的非病灶区域进行噪声处理	在图像中非病灶区域随机添加噪声。噪声根据图像的信噪比，随机生成一些图像内部的像素位置，并且对这些像素点赋值
对比度处理	改变非病灶区域对比度	在图像中的非病灶区域改变对比度。在医疗影像中，对比度会对实际读片效果产生影响
锐度处理	改变非病灶区域锐度	在图像中的非病灶区域改变锐度。锐度由不同色调或颜色区域之间的边界定义，反应了图像细节数量的多少

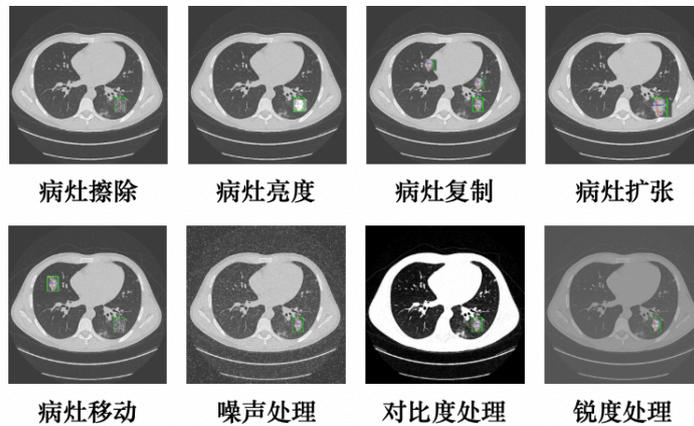


图 4.11: 领域语义变异扩增方法示例

### (1) 基于病情表现形式的变异扩增方法

医疗领域病症种类繁多，且每一种病症表现都包含多种不同的征象。根据病情某个时期可能出现的征象集合，对其中包含的征象进行随机组合，以生成包含不同表现形式的病情图像。同时还可改变征象在图像中的位置以及出现的数量，以增加图像多样性。基于病情表现形式的变异扩增方法包括病灶亮度和

病灶复制。

病灶亮度指随机改变图像病灶区域亮度，基于用户设置的变化范围将病灶区域随机变亮或变暗，以模拟医生实际读片时光线对病灶区域的影响。病灶复制则是对图像中的病灶区域进行复制，增加医学图像中病灶的数目，模拟疾病在不同个体或发展阶段的多样性。即提取病灶区域图像，并基于用户设置的复制数量将其在图像中任意位置进行复制。其实现代码如图 4.12 所示。

```
class LesionPaste(LesionOperations):
    """
    在原始图像中添加多个病灶
    lesion_num: 病灶数
    """
    def __init__(self, probability, lesion_num):
        LesionOperations.__init__(self, probability)
        self.lesion_num = lesion_num

    def perform_operation(self, images):
        def do(image):
            start_x, start_y, end_x, end_y = TauMed.getLesionPos(image)
            box = (start_x, start_y, end_x, end_y)
            lesion_image = image.crop(box)
            w, h = image.size
            count = self.lesion_num
            while count > 0:
                x = np.random.randint(w/4, 3*w/4)
                y = np.random.randint(h/4, 3*h/4)
                image.paste(lesion_image, (x, y))
                count -= 1
            return image

        augmented_images = []
        for image in images:
            augmented_images.append(do(image))
        return augmented_images
```

图 4.12: 病灶复制方法代码

## (2) 基于病情发展阶段的变异扩增方法

此外，从病情发展阶段来说，病症在影像学表现上的大致发展分为早期、进展期、重症期和消散期四个阶段，每个阶段的病症表现都不一样。通过对一张图

像的病灶区域覆盖范围进行变化来模拟病症发展的范围，通过对一张图像上某一征象的程度进行变化，如肺部 CT 图像中磨玻璃密度影的密度增强或减弱，来模拟病症表现的程度。基于病情发展阶段的变异扩增方法包括病灶擦除、病灶移动和病灶扩张方法。

病灶擦除指对图像中的病灶区域进行擦除，即使用随机像素值覆盖病灶区域，以创建没有病灶区域的新图像，模拟处于未得病状态的医疗图像。病灶移动则是在病灶擦除的基础上，将原始病灶再复制到图像中的随机区域，以模拟病灶的活动特性。病灶扩张指对图像中的病灶区域面积进行扩张，扩大病变范围，以纵向模拟疾病随时间的发展。即基于用户设置的幅度值，通过 `resize` 方法改变提取出的病灶区域大小，再将其覆盖到原位置。

### (3) 基于非病灶区域的变异扩增方法

非病灶区域的扩增是在保持病灶区域不变的前提下对非病灶区域进行处理，在保留关键信息的同时扩充数据集，在较大程度上保证了扩增数据的有效性，并且可以据此测试医疗图像分类模型针对非病灶区域的分类结果是否具有不变性。其方法包括噪声处理、锐度处理和对比度处理等。

噪声处理是在图像中非病灶区域随机添加噪声。噪声是根据图像的信噪比，随机生成一些图像内部的像素位置，并且对这些像素点赋值。添加适量的噪声可以增强学习能力。对比度处理是在图像中的非病灶区域改变对比度。在医疗影像中，对比度会对实际读片效果产生影响。锐度处理则是在图像中的非病灶区域改变锐度。锐度由不同色调或颜色区域之间的边界定义，反应了图像细节数量的多少。

## 4.3 扩增质量多维评估模块

扩增质量多维评估模块主要采用不同维度的评估指标对扩增数据集的质量进行定量分析，以评估扩增任务的效果。模块整体由 `Docker` 封装为镜像，首先从扩增任务控制处理模块获取对应任务的扩增数据集和用户选择的评估指标，再调用评估模块中的对应方法进行计算，最后将计算结果回传到扩增任务执行控制模块，同时存储到服务器中。该模块提供了 `MSE`、`PSNR`、`SSIM`、`MSSSIM`、`FSIM` 和 `GMSD` 六个主流质量评估指标，在实际执行评估时，通过 `Docker` 镜像创建实例容器，完成相应计算任务后自动将容器销毁，防止占用过多磁盘空间。

扩增质量多维评估模块时序图如 4.13 所示，并选择 `SSIM` 和 `GMSD` 两个评估指标作为示例。首先通过 `Shell` 文件调用 `EvaluateService`，同时通过 `loadImages` 方法从存储路径中读取待评估数据集，并进行数据预处理。然后在 `Evaluator` 抽象类中通过 `calculateScore` 方法调用 `SSIMEvaluator` 中的具体计算逻辑完成评估

分数的计算，并将评估结果以 Json 形式存储到指定路径中，同时将评估结果数据及执行状态进行回传，以完成基于 Echarts 库的前端图表可视化。

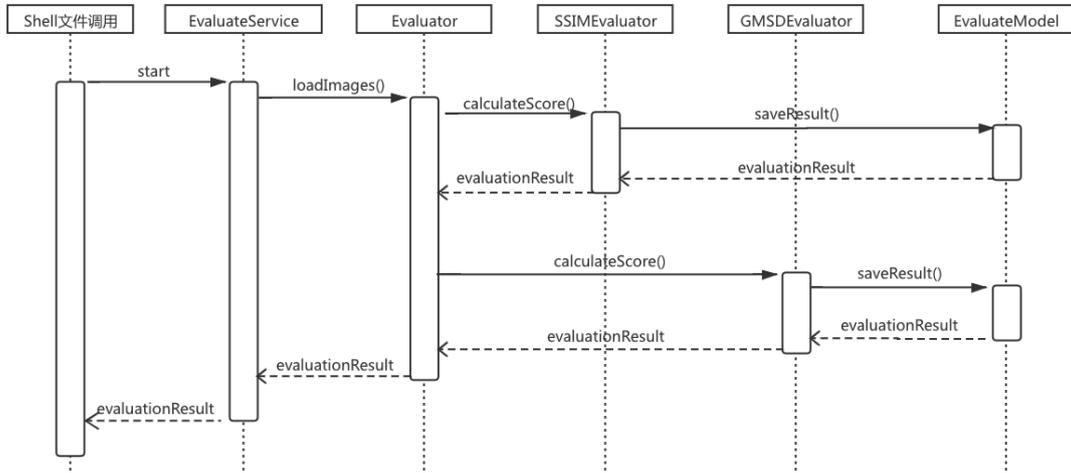


图 4.13: 扩增质量多维评估模块时序图

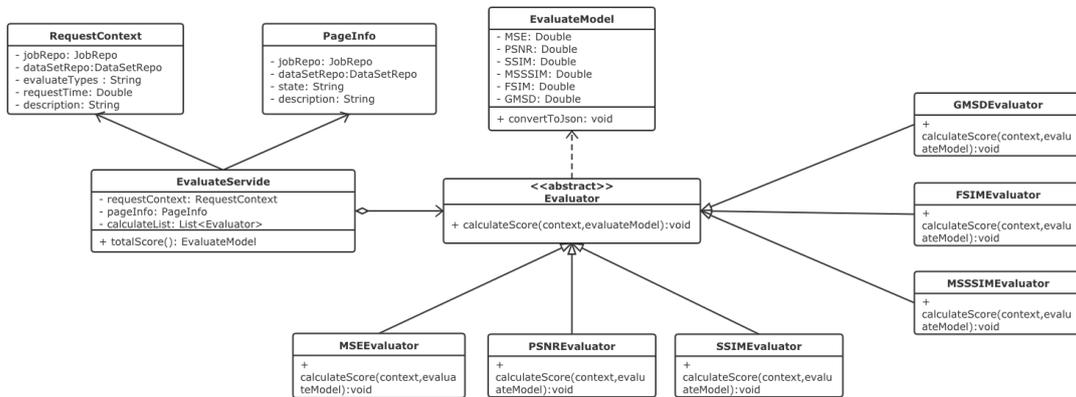


图 4.14: 扩增质量多维评估模块类图

扩增质量多维评估模块类图如 4.14 所示，模块主要采用策略设计模式，有良好的易扩展性和可复用性，便于对算法族的管理。其中 Evaluator 作为抽象策略类，为所有具体评估指标的计算类声明了抽象方法 calculateScore，同时具体策略类 MSSEvaluator、PSNREvaluator、SSIMEvaluator、MSSSIMEvaluator、FSIMEvaluator 和 GMSDEvaluator 通过实现抽象方法，完成对具体评估算法的封装。EvaluateService 则根据具体请求参数，通过 Evaluator 调用相应算法生成评估分数计

算结果。此外，将各个评估指标的计算结果封装为 `EvaluateModel`，便于数据传输和存储，同时利用 `RequestContext` 和 `PageInfo` 类完成评估请求体和页面显示数据的操作支持。

扩增质量多维评估模块页面设计如图 4.15 所示，用户首先选择需要评估的扩增任务和评估指标，点击提交后在可视化评估结果部分将生成数据集整体质量评估分数多维雷达图和图像质量评估分数分布图。其中整体质量评估分数多维雷达图是数据集中各个评估指标计算的平均值，体现数据集整体质量水平。而图像质量评估分数分布图则是根据数据集中每张图像的质量分数生成，体现数据集中质量的分布情况，同时可以根据该分布图查看质量显著降低样本的具体评估值，以从数据集中剔除质量较差的图像。

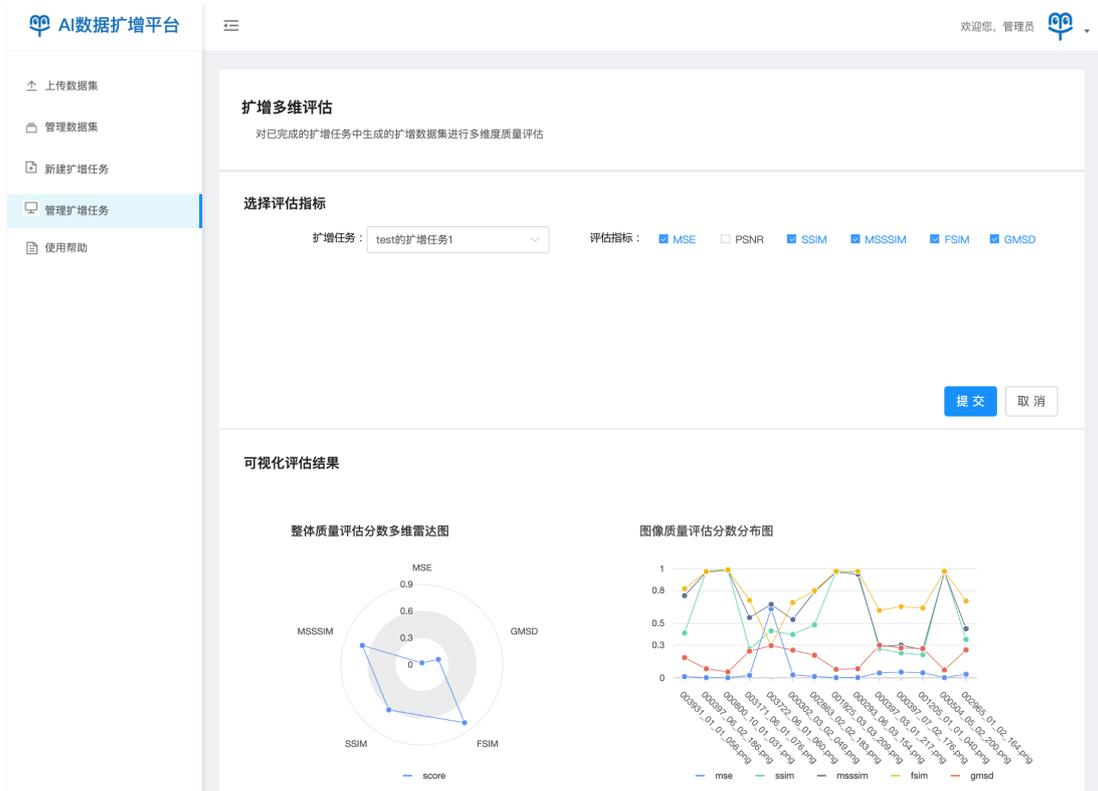


图 4.15: 扩增质量多维评估模块页面

#### 4.4 本章小结

本章基于前章的系统概要设计，对扩增任务执行控制模块、医疗图像变异扩增模块和扩增质量多维评估模块的详细设计和具体实现进行了阐述，并通过时序图、类图等表现形式对模块内部业务逻辑和数据交互进行了说明。其中将

医疗图像变异扩增模块分为医疗图像病灶识别子模块、图像特征变异扩增子模块和领域语义变异扩增子模块，对各个子模块中设计实现的识别和扩增算法进行了详细阐述，并给出了关键代码和页面截图。



## 第五章 系统测试与实验分析

### 5.1 系统测试

#### 5.1.1 测试目标及测试环境

系统测试的目标是基于测试环境，通过对各个模块功能设计并执行测试用例，查看其输出与期望结果是否一致，系统功能是否能够覆盖所有需求。主要测试的功能包括数据集上传与管理、扩增任务执行与管理以及扩增结果下载等。

系统测试环境参数如表 5.1 所示，服务器搭设在基于 Ubuntu 操作系统的 4 核 16G 阿里云 ECS 服务器上，项目前后端通过 Nginx 进行反向代理和负载均衡，同时采用 MySQL 数据库存储数据集、任务配置参数等信息，图像扩增服务打包成 Docker 镜像文件以供系统调用。

表 5.1: 测试环境参数列表

参数	参数详情
云服务器	阿里云 ECS 服务器
操作系统	Ubuntu 16.04.6 LTS
CPU	4
内存	16G
数据库	MySQL 5.7.18
容器	Docker 19.3.8
其他	Nginx 1.15.7

#### 5.1.2 功能测试

功能测试是对系统的各个功能进行验证，根据需求编写测试用例，并比较预期结果与实际输出是否一致，以验证系统内部逻辑的正确性和完备性，同时测试系统是否能对异常输入做出正确反馈，以提高系统可用性和健壮性。该节基于第三章需求分析进行测试用例设计和执行，测试用例中将提供测试过程、预期输出和测试结果等相关信息。下面根据表 3.3 系统用例列表分别对每个用例设计测试用例，并据此进行功能测试。

对于用例 UC1 上传数据集的测试用例如表 5.2 所示，主要测试上传数据集的流程是否符合预期，其测试项包括文件上传方式、链接上传方式等，并对上传不规范格式文件、输入不符合系统规范等情况进行测试，验证系统是否能够正确上传数据集文件，并对错误输入进行提示。设计的测试用例执行后全部通过，

表明该功能符合预期需求。

对于用例 UC2 查看数据集列表的测试用例如表 5.3 所示，主要测试管理数据集页面的相关操作是否具有正确输出。测试项包括查看数据集列表，对数据集的下载、删除、重命名、搜索以及通过数据集新建扩增任务等，验证系统是否能够正确支持数据集的相关操作，并对错误操作做出反馈。设计的测试用例执行后全部通过，表明该功能符合预期需求。

表 5.2: 上传数据集测试用例

测试用例编号	测试项	操作/输入	预期结果	测试结果
TC1-1-1	选择文件方式	点击选择文件上传方式	系统显示文件上传框	通过
TC1-1-2	拖拽文件上传	将本地 zip 文件拖拽到上传框处	系统显示数据集上传成功	通过
TC1-1-3	点击文件上传	点击文件上传框并选择本地 zip 文件	系统显示文件选择框并显示数据集上传成功	通过
TC1-1-4	上传非 zip 文件	选择非 zip 文件进行上传	系统提示仅支持上传扩展名为.zip 的数据集压缩包	通过
TC1-1-5	上传大于 500MB 的文件	选择 900MB 的文件进行上传	系统提示仅支持上传小于 500MB 的文件	通过
TC1-1-6	上传空文件	选择空文件进行上传	系统提示压缩包文件为空	通过
TC1-1-7	上传不符合格式规范的文件	选择包含不符合格式数据的文件进行上传	系统提示数据集中包含不符合要求的图片格式	通过
TC1-2-1	选择链接方式	点击选择链接上传方式	系统显示数据集名称和数据集链接输入框	通过
TC1-2-2	输入数据集名称及链接	用户输入数据集名称和链接，并点击确认上传	系统显示数据集上传成功	通过
TC1-2-3	不输入数据集名称	不输入数据集名称并点击确认上传	系统提示请填写上传数据集的名称	通过
TC1-2-4	不输入数据集链接	不输入数据集链接并点击确认上传	系统提示请填写上传数据集的链接	通过

对于 UC7 查看扩增结果报告和 UC8 下载扩增数据集的测试用例如表 5.4 所示，主要测试系统针对查看扩增结果相关操作是否具有正确输出。测试项包括查看成功和失败扩增结果报告，以及对原始和扩增数据集的下载，以保证系统在扩增结果模块的业务逻辑正确性。设计的测试用例执行后全部通过，表明该功能符合预期需求。

对于用例 UC3 执行扩增任务的测试用例如表 5.5 所示，主要测试执行扩增任务的业务流程是否正确。测试项包括创建扩增任务的各个步骤，如选择数据

表 5.3: 查看数据集列表测试用例

测试用例编号	测试项	操作/输入	预期结果	测试结果
TC2-1-1	查看数据集列表	点击菜单栏进入管理数据集页面	系统按时间倒序分页显示该用户已上传的数据集列表	通过
TC2-1-2	下载数据集	点击数据集对应下载按钮	系统跳转至本地下载	通过
TC2-1-3	重命名数据集	点击数据集对应重命名按钮,并在对话框中输入新名称,点击确认	系统显示输入框,并显示修改成功	通过
TC2-1-4	删除数据集	点击数据集对应删除按钮,并在对话框中点击确认	系统显示是否确认删除该数据集的提示对话框,并显示删除成功	通过
TC2-1-5	批量删除数据集	选中一至多个数据集点击批量删除按钮,并在对话框中点击确定	系统提示此操作将永久删除选中的数据集,是否继续的对话框,并显示删除成功	通过
TC2-1-6	批量删除数据集不选中	未选择数据集并点击批量删除按钮	系统提示请先选中一项或多项数据集	通过
TC2-1-7	新建扩增任务	点击数据集对应新建任务按钮	系统跳转到新增扩增任务页面,并默认选择当前数据集	通过
TC2-1-8	搜索数据集	输入数据集名称模糊值	系统显示匹配到的数据集列表	通过

表 5.4: 查看扩增结果相关测试用例

测试用例编号	测试项	操作/输入	预期结果	测试结果
TC5-1-1	查看成功扩增结果报告	管理扩增任务页面点击成功任务对应报告按钮	系统跳转到扩增任务报告页面,显示扩增结果报告信息,包括任务基本和运行信息	通过
TC5-1-2	查看失败扩增结果报告	管理扩增任务页面点击失败任务对应报告按钮	系统跳转到扩增任务报告页面,显示扩增结果报告信息,给出失败原因及解决方案	通过
TC5-1-3	下载原始数据集	点击原始数据集下载链接	系统跳转至本地下载	通过
TC5-1-4	下载扩增数据集	点击扩增结果下载链接	系统跳转至本地下载	通过

集、填写任务名称、添加扩增方法等，同时对用户的错误输入进行测试，验证系统是否能对其做出正确提示。设计的测试用例执行后全部通过，表明该功能符合预期需求。

表 5.5: 执行扩增任务测试用例

测试用例编号	测试项	操作/输入	预期结果	测试结果
TC3-1-1	选择数据集	点击下拉框选择数据集	系统在下拉框中选择用户已上传数据集	通过
TC3-1-2	未选择数据集	未选择数据集并点击创建任务	系统提示请选择数据集	通过
TC3-1-3	输入通用配置	输入任务名称、输出文件数量倍数	系统显示输入信息	通过
TC3-1-4	未填写任务名称	未填写任务名称并点击创建任务	系统提示请填写任务名称	通过
TC3-1-5	指定输出文件数量倍数大于 20	输入文件数量倍数为 50	系统提示最多只输出 20 张图片	通过
TC3-1-6	添加扩增方法	从树形菜单多选框中选择一至多个扩增方法，并填写对应方法参数	系统在扩增方法栏显示已选择的扩增方法，并在设置方法参数栏显示每个方法的对应参数设置框	通过
TC3-1-7	执行扩增任务	点击创建任务按钮	系统显示创建成功，并跳转到管理扩增任务页面	通过
TC3-1-8	查看使用帮助	点击扩增方法旁的帮助按钮	系统跳转到使用帮助页面	通过

对于用例 UC4 查看扩增任务列表、UC5 查看扩增任务信息、UC6 拷贝扩增任务的测试用例如表 5.6 所示，主要测试系统中扩增任务相关操作的正确性。测试项包括查看扩增任务列表，以及对扩增任务的查看、删除、重命名、搜索、拷贝等操作，验证系统是否能够正确支持扩增任务的相关操作，并对错误操作做出反馈。设计的测试用例执行后全部通过，表明该功能符合预期需求。

## 5.2 实验分析

在验证了系统从上传数据集、管理数据集到创建扩增任务、管理扩增任务并查看扩增结果业务流程的正确性后，需要通过多个控制变量实验对系统的实际效果进行验证和评估，主要包括扩增带来的数据量提升是否能提高模型表现，扩增技术中定义的图像特征变异和领域语义变异方法是否真的有效，以及本扩增系统与其他主流扩增工具的效果相比是否有所提升等问题。

表 5.6: 查看扩增任务相关测试用例

测试用例编号	测试项	操作/输入	预期结果	测试结果
TC4-1-1	查看扩增任务列表	点击菜单栏进入管理扩增任务页面	系统按时间倒序分页显示该用户已创建的扩增任务列表	通过
TC4-1-2	查看扩增任务信息	点击扩增任务对应详情按钮	系统跳转到扩增任务详情页面,显示扩增任务详细信息	通过
TC4-1-2	重命名扩增任务	点击扩增任务对应重命名按钮,并在对话框中输入新名称,点击确认	系统显示输入框,并显示修改成功	通过
TC4-1-3	删除扩增任务	点击扩增任务对应删除按钮,并在对话框中点击确认	系统显示是否确认删除该扩增任务的提示对话框,并显示删除成功	通过
TC4-1-4	批量删除扩增任务	选中一至多个扩增任务点击批量删除按钮,并在对话框中点击确定	系统提示此操作将永久删除选中的扩增任务,是否继续的对话框,并显示删除成功	通过
TC4-1-5	批量删除扩增任务不选中	未选择扩增任务并点击批量删除按钮	系统提示请先选中一项或多项任务	通过
TC4-1-6	搜索扩增任务	输入任务名称模糊值	系统显示匹配到的扩增任务列表	通过
TC4-2-1	拷贝扩增任务	点击扩增任务对应拷贝按钮	系统跳转到新建扩增任务页面,并默认沿用原扩增任务配置	通过

根据以上问题,本节拟通过四个实验来验证该扩增技术及系统的有效性和优越性,包括扩增数量提升实验、扩增方法设置实验、扩增效果对比实验以及扩增质量对比实验。

### 5.2.1 实验对象

医学图像数据集的类型非常广泛,本系列实验中仅以 DeepLesion 数据集作为实验对象,并选择了其中的一个子集——肺部和肾脏的 CT 图像进行实验。DeepLesion 数据集 [40] 包含来自 10000 多个病例研究的超过 32000 个病变标注,现在是面向所有人可公开获得的最大的医学图像数据集,其中包括多种病变类型,包括肾脏病变、骨病变、肺结节和淋巴结肿大等。由于 DeepLesion 数据集中的图像上已经标注了病灶区域,便于在实验中判断病灶区域识别的结果是否准确,并且能够准确定位领域语义方法扩增的结果。但在实际应用中,可以输入

无标注病灶区域的原始医疗图像进行处理。

将实验数据集按 1:1:14 的比例分为训练集、验证集和测试集，其中训练集和验证集中又包含肺部和肾脏两个类别的数据，肺部和肾脏的 CT 图像示例如图 5.1 所示，可见其形态差别较为明显，其中绿框部分为病灶区域。此外，选取 ResNet-50 图像分类模型并基于上述数据集进行实验，以验证扩增数据集对于深度学习模型性能的提升效果。

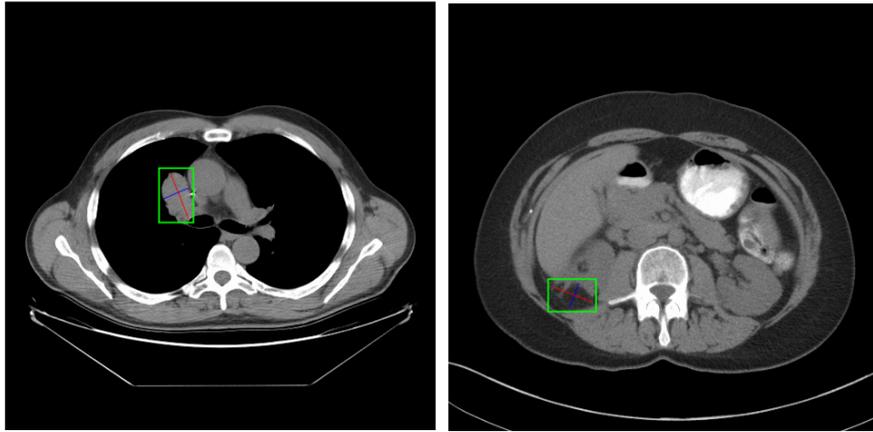


图 5.1: DeepLesion 数据集示例图（左：肺部；右：肾脏）

### 5.2.2 实验设计及结果分析

下面分别对四个实验的具体设计进行介绍。

#### (1) 扩增数量提升实验

该实验主要针对扩增带来的数据集文件数量提升能否提高深度学习模型表现进行设计。分别基于原始数据集、系统扩增 2 倍得到的数据集以及系统扩增 3 倍得到的数据集在 ResNet-50 图像分类模型上进行实验，并通过 Accuracy、Top-1 Accuracy、Precision、Recall、F1-Score 五个指标从不同维度评估实验效果。

表 5.7: 扩增数量提升实验结果

实验数据集	图像数量	Accuracy	Top-1 Accuracy	Precision	Recall	F1-Score
原始数据集	2863	90.6368	94.7539	77.3537	65.8009	71.1111
扩增 2 倍数据集	5726	93.6884	98.3488	84.4005	78.4632	81.3236
扩增 3 倍数据集	8589	<b>95.0847</b>	<b>99.0933</b>	<b>87.8799</b>	<b>83.4416</b>	<b>85.6033</b>

表 5.7 中记录了该实验得到的结果数据，从表中可以看出，当数据集的数量逐步增加时，模型训练后得到的模型准确率、Top-1 准确率、精度、召回率和

F1 指标都有较明显的提升。其中 Accuracy 和 Top-1 Accuracy 均提高了约 5%，Precision、Recall 和 F1-Score 则有超过 10% 的提升。因此实验结果表明，扩增数据集的数量提升对于模型性能表现有明显的积极影响。在实际应用中，医疗图像扩增系统可降低深度学习模型训练起步需要的数据量，同时能够在一定程度上提高模型的准确率等性能表现。

### (2) 扩增方法效果实验

该实验主要针对系统中图像特征变异和领域语义变异扩增方法的实际效果进行实验的设计和实验。在实验中，分别选择无扩增方法、仅图像特征变异扩增方法、仅领域语义变异扩增方法以及结合图像特征变异和领域语义变异的扩增方法四种组合进行实验变量设置，并基于 ResNet-50 图像分类模型进行训练和实验，最终通过 Accuracy、Top-1 Accuracy、Precision、Recall 以及 F1-Score 五个指标评估实验效果。

表 5.8: 扩增方法效果实验结果

实验扩增方法	Accuracy	Top-1 Accuracy	Precision	Recall	F1-Score
无扩增方法	90.6368	94.7539	77.3537	65.8009	71.1111
图像特征变异扩增方法	95.2616	99.0285	<b>88.6468</b>	83.2251	<b>86.0802</b>
领域语义变异扩增方法	<b>95.7849</b>	<b>99.4171</b>	88.4125	82.5758	85.3945
结合图像特征变异 和领域语义变异扩增方法	95.0531	98.6399	87.8857	<b>83.6580</b>	85.4919

对 DeepLesion 数据集中 2863 张医疗图像的实验结果示于表 5.8 中。从总体表现上看，单独应用图像特征变异扩增方法或领域语义变异扩增方法，以及结合图像特征变异和领域语义变异的扩增方法进行实验的效果均比不使用扩增方法的实验效果更好，具体体现在各个指标均有较大幅度的提高。而对于不同方法配置内部的实验效果比较，各个指标表现出的差距较小，差值均低于 1%。实验结果表明，不管采用怎样的扩增方法组合，其效果都高于无扩增方法的情况，同时各种扩增方法组合之间的表现差距小，也为用户自定义扩增规则赋予了更多的变化。

### (3) 扩增工具效果对比实验

该实验选择两种主流的图像扩增工具 Alumentation 和 Augmentor 作为对比实验，以验证医疗图像扩增系统是否在医疗图像扩增方面与主流扩增工具相比具有优越性。各组实验数据集是基于原始数据集分别通过无扩增、Alumentation、Augmentor 和医疗图像扩增系统进行扩增得到，同样在 ResNet-50 图像分类模型上进行实验，并由 Accuracy、Top-1 Accuracy、Precision、Recall 和 F1-Score 五个

指标评估实验效果。

表 5.9 中记录了基于 DeepLesion 数据集中 2863 张医疗图像的实验结果。从表中数据可以看出，相较于 Alumentation 和 Augmentor，医疗图像扩增系统在每个指标上都有最好的表现，而 Augmentor 的准确率、精度等各个指标值均较低。实验结果表明针对医疗图像数据集，医疗图像扩增系统与通用扩增工具相比具有较高的优越性和准确性，相比之下，Augmentor 扩增工具最不适合用于医疗图像扩增。

表 5.9: 扩增工具效果对比实验结果

实验扩增工具	Accuracy	Top-1 Accuracy	Precision	Recall	F1-Score
无扩增工具	90.6368	94.7539	77.3537	65.8009	71.1111
Alumentation	92.3048	97.215	81.4320	72.6190	76.7735
Augmentor	85.3867	75.1943	61.7874	43.3983	50.9854
医疗图像扩增系统	<b>95.0531</b>	<b>98.6399</b>	<b>87.8857</b>	<b>83.2251</b>	<b>85.4919</b>

#### (4) 扩增工具质量评估实验

该实验通过多个图像评估指标对 Alumentation、Augmentor 和医疗图像扩增系统处理得到的扩增数据集的质量进行评估。与扩增工具效果对比实验类似，同样通过三个扩增工具处理获得各组实验数据集，并由基于误差的评估指标和基于感知模型与图像结构信息的评估指标对其进行质量评估。基于误差的评估指标指与原始图像进行对比，计算两个图像之间的差异，包括均方误差 (MSE) 和峰值信噪比 (PSNR)。基于感知模型与图像结构信息的评估指标将图像质量下降转化为感知结构信息的变化，包括结构相似性 (SSIM)、多尺度结构相似性 (MSSSIM)、特征相似性 (FSIM) 和梯度幅度相似性偏差 (GMSD)。

表 5.10: 扩增工具质量评估实验结果

实验扩增工具	MSE	PSNR	SSIM	MSSSIM	FSIM	GMSD
Alumentation	0.0548	14.2614	0.3148	0.3471	0.638	0.277
Augmentor	0.0909	10.7638	0.2518	0.1616	0.5401	0.3237
医疗图像扩增系统	<b>0.0206</b>	<b>19.54</b>	<b>0.6263</b>	<b>0.6971</b>	<b>0.8034</b>	<b>0.1913</b>

表 5.10 中记录了该实验得到的结果数据。其中 MSE 和 GMSD 越小，表示扩增图像与原始图像相关性越高，质量越高。而 PSNR、SSIM、MSSSIM 和 FSIM 则是值越高表明图像质量越高。从表中可以看出，相较于 Alumentation 和 Augmentor，通过医疗图像扩增系统生成的扩增数据集在基于误差的评估指标和

基于感知模型与图像结构信息的评估指标上均有较好的表现，即通过医疗图像扩增系统处理得到的扩增数据集质量最高，同时通过 Augmentor 扩增产生的医疗数据集质量较低。

从以上四个方面的实验结果整体表现来看，医疗图像扩增系统能够在提高原始数据集的数据量的同时更大程度地保证扩增数据集的质量，在将扩增数据集输入模型训练时，也能更好地提升模型在准确率、精度等方面的性能表现，此外还通过图像特征变异扩增方法和领域语义变异扩增方法为用户提供了多样化的扩增方法组合。因此该系统在面向医疗图像扩增方面的技术是有效且优越的。

### 5.2.3 案例分析

本节将对系统核心业务流程即执行扩增任务流程进行详细说明，并对系统各功能进行简要介绍。图 5.2 和图 5.3 为新建扩增任务页面，即根据自定义的扩增规则进行数据扩增任务。首先从数据集列表中选择需要扩增的数据集，完善任务名称等基本信息，并指定输出文件的数量倍数，且该数值不能超过 20 倍。

图 5.2: 新建扩增任务——完善任务信息页面

在完善扩增任务基本信息后，用户可通过多选框添加一系列扩增方法，其中通用规则扩增和医疗领域扩增方法可以叠加执行。在添加完所有方法后，需要在下方对各个方法参数进行设置，将鼠标悬停在输入框上方可查看该参数值的规定取值范围，此外还可通过菜单栏进入使用帮助页面查看各个扩增方法的原理及扩增效果。

成功执行完成的扩增任务报告如图 5.4 所示，可以查看数据集及任务的基本信息，以及任务运行时长、状态等信息，并提供了扩增结果数据集的下载链接。



图 5.3: 新建扩增任务——添加扩增方法页面



图 5.4: 查看扩增任务报告页面

### 5.3 本章小结

本章首先对系统进行功能测试，以验证系统内部逻辑的正确性和完备性。然后通过扩增数量提升实验、扩增方法设置实验、扩增效果对比实验以及扩增质量对比实验对医疗图像扩增系统的有效性和优越性进行了验证。

## 第六章 总结与展望

### 6.1 总结

由于病人隐私等问题阻碍着医疗领域的信息和数据共享，且采用通用图像扩增技术无法保证医疗图像的扩增质量，因此基于医疗图像的特殊性，提出基于病灶征象和变异分析的扩增技术，并以此为核心设计并实现了面向医疗领域的医疗图像扩增系统。

本文首先介绍了系统研究背景及意义，并分析了国内外图像扩增技术、扩增工具以及病灶识别技术的研究现状，其后介绍了医疗图像、病灶、征象等医疗领域相关概念，以及前后端开发相关技术，包括用于前端开发的 **Element UI** 框架和 **Vue** 生态、用于项目服务端开发的 **Spring Boot** 框架和用于封装和调用服务的 **Docker**。根据系统的应用场景和业务流程，对系统进行需求分析和概要设计，并将系统功能划分为扩增任务执行控制、医疗图像变异扩增和扩增效果多维评估三个模块，其中扩增任务执行控制模块通过对医疗图像变异扩增模块和扩增效果多维评估模块进行调用，以执行图像扩增和多维评估任务。医疗图像变异扩增模块将变异分析思想应用于图像扩增，分别针对图像基础属性特征和医疗领域语义特征进行变异分析，以挖掘其潜在信息，生成丰富且具有多样性的扩增数据集。扩增效果多维评估模块则主要采用 **MSE**、**PSNR**、**SSIM**、**MSSSIM**、**FSIM** 和 **GMSD** 等不同维度的评估指标对扩增数据集的质量进行定量分析，以评估扩增任务的效果。最后通过对各个模块功能进行测试，验证系统功能是否能够覆盖所有需求，并通过扩增数量提升、扩增方法设置、扩增效果对比和扩增质量对比四个实验验证系统的有效性和优越性。

系统提出了以病灶为核心的扩增思路，即通过基于病灶的医疗图像特征识别，提取其中的病灶区域，并从同一个体的病情发展阶段和不同个体的病情表现形式两个维度分别对病灶区域和非病灶区域进行扩增处理。针对病灶区域的扩增能够增加病理表现的多样性，而针对非病灶区域的扩增则可以用来测试模型的不变性。此外，可将扩增后的优质数据集输入辅助诊断模型进行训练，可以优化模型并提高其准确度，形成“扩增——优化”的良性生态。此外，以扩增形成的数据集为基础建立标准数据库，可以服务于广泛的医疗智能诊断模型，提高其泛用性，也能为数据扩增提供更多的原始数据来源。该系统可在一定程度上缓解人工分析误诊漏诊、医生资源缺口、医疗数据量不足等痛点问题，加深数据的积累以及技术上的创新，提升医疗质量，降低医疗差错。

## 6.2 展望

医疗图像扩增系统已经实现了 21 种面向医疗领域的扩增技术，并已作为 AI 数据扩增平台的服务内测使用，但在以下几个方面还可进行优化和改进：

(1) 针对特定病症设计扩增技术。目前系统实现的扩增技术是将医疗领域相关知识维度加入通用图像扩增技术中，即通过对病灶征象的识别和处理，从病症发展阶段和不同个体病症表现两个维度进行医疗图像的扩增处理。但由于医疗领域不同病症的影像学表现之间存在较大差异，因此需要在识别征象的基础上对其进行分类，例如肺部病症中的肺结节征象就包括形态学特征中的圆形肿块征、分叶征、棘状突起、结节征、空泡征和空洞征，病灶密度特征中的纯磨玻璃结节、混合磨玻璃结节和实性结节，以及病灶表面特征中的毛刺、分叶和晕征，基于更细化的医疗图像征象特征设计扩增技术，可以实现更精确的扩增结果。

(2) 制定病理特征与图像扩增的关联机制。由于病症的复杂性和多样性，病理及其对应的影像学表现也是及其复杂的。针对不同病症应制定一套医疗病理与图像扩增技术之间的关联机制，即提取病理对应影像学表现中的重要特征，并根据该特征选择适宜的扩增技术或设计合适的扩增组合，以达到更好的扩增效果，得到优质的扩增医疗图像，缓解由于隐私或伦理带来的医疗数据获取困难的问题。

(3) 添加扩增质量评估方式。在现有六种主流评估指标的基础上，可根据医疗图像质量评估需求调研并添加更多评估指标，以提高扩增效果多维评估体系的多样性和完整性。同时，还可以加入模型评估方式，即通过将扩增数据集输入测试模型中，并评估模型的准确率、召回率等指标变化情况，以达到扩增效果评估目的。

## 参考文献

- [1] F. Perez, C. Vasconcelos, S. Avila, E. Valle, Data augmentation for skin lesion analysis, in: D. Stoyanov, Z. Taylor, D. Sarikaya, A. J. McLeod, M. Á. G. Ballester, N. C. F. Codella, A. L. Martel, L. Maier-Hein, A. Malpani, M. A. Zenati, S. de Ribaupierre, X. Luó, T. Collins, T. Reichl, K. Drechsler, M. Erdt, M. G. Linguraru, C. O. Laura, R. Shekhar, S. Wesarg, M. E. Celebi, K. J. Dana, A. Halpern (Eds.), OR 2.0 Context-Aware Operating Theaters, Computer Assisted Robotic Endoscopy, Clinical Image-Based Procedures, - and - Skin Image Analysis - First International Workshop, OR 2.0 2018, 5th International Workshop, CARE 2018, 7th International Workshop, CLIP 2018, Third International Workshop, ISIC 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 16 and 20, 2018, Proceedings, Vol. 11041 of Lecture Notes in Computer Science, Springer, 2018, pp. 303–311.  
URL [https://doi.org/10.1007/978-3-030-01201-4\\_33](https://doi.org/10.1007/978-3-030-01201-4_33)
- [2] L. Perez, J. Wang, The effectiveness of data augmentation in image classification using deep learning, CoRR abs/1712.04621.  
URL <http://arxiv.org/abs/1712.04621>
- [3] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, Y. Bengio, Generative adversarial networks, CoRR abs/1406.2661.  
URL <http://arxiv.org/abs/1406.2661>
- [4] T. DeVries, G. W. Taylor, Dataset augmentation in feature space, in: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings, OpenReview.net, 2017.  
URL <https://openreview.net/forum?id=HyaF53XYx>
- [5] S. K. Lim, Y. Loo, N. Tran, N. Cheung, G. Roig, Y. Elovici, DOPING: generative data augmentation for unsupervised anomaly detection with GAN, in: IEEE International Conference on Data Mining, ICDM 2018, Singapore, November 17-20, 2018, IEEE Computer Society, 2018, pp. 1122–1127.  
URL <https://doi.org/10.1109/ICDM.2018.00146>

- 
- [6] E. D. Cubuk, B. Zoph, D. Mané, V. Vasudevan, Q. V. Le, Autoaugment: Learning augmentation strategies from data, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019, Computer Vision Foundation / IEEE, 2019, pp. 113–123.  
URL [http://openaccess.thecvf.com/content\\_CVPR\\_2019/html/Cubuk\\_AutoAugment\\_Learning\\_Augmentation\\_Strategies\\_From\\_Data\\_CVPR\\_2019\\_paper.html](http://openaccess.thecvf.com/content_CVPR_2019/html/Cubuk_AutoAugment_Learning_Augmentation_Strategies_From_Data_CVPR_2019_paper.html)
- [7] X. Zhang, Z. Wang, D. Liu, Q. Ling, DADA: deep adversarial data augmentation for extremely low data regime classification, in: IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2019, Brighton, United Kingdom, May 12-17, 2019, IEEE, 2019, pp. 2807–2811.  
URL <https://doi.org/10.1109/ICASSP.2019.8683197>
- [8] Z. Zhong, L. Zheng, G. Kang, S. Li, Y. Yang, Random erasing data augmentation, in: The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020, AAAI Press, 2020, pp. 13001–13008.  
URL <https://aaai.org/ojs/index.php/AAAI/article/view/7000>
- [9] H. Inoue, Data augmentation by pairing samples for images classification, CoRR abs/1801.02929.  
URL <http://arxiv.org/abs/1801.02929>
- [10] H. Zhang, M. Cissé, Y. N. Dauphin, D. Lopez-Paz, mixup: Beyond empirical risk minimization, in: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings, OpenReview.net, 2018.  
URL <https://openreview.net/forum?id=r1Ddp1-Rb>
- [11] A. H. Ornek, M. Ceylan, Comparison of traditional transformations for data augmentation in deep learning of medical thermography, in: N. Herencsar (Ed.), 42nd International Conference on Telecommunications and Signal Processing, TSP 2019, Budapest, Hungary, July 1-3, 2019, IEEE, 2019, pp. 191–194.  
URL <https://doi.org/10.1109/TSP.2019.8769068>

- [12] Z. Hussain, F. Gimenez, D. Yi, D. L. Rubin, Differential data augmentation techniques for medical imaging classification tasks, in: AMIA 2017, American Medical Informatics Association Annual Symposium, Washington, DC, USA, November 4-8, 2017, AMIA, 2017.  
URL <http://knowledge.amia.org/65881-amia-1.3897810/t003-1.3901461/f003-1.3901462/2728928-1.3901765/2730723-1.3901762>
- [13] A. Zhao, G. Balakrishnan, F. Durand, J. V. Guttag, A. V. Dalca, Data augmentation using learned transformations for one-shot medical image segmentation, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019, Computer Vision Foundation / IEEE, 2019, pp. 8543–8553.  
URL [http://openaccess.thecvf.com/content\\_CVPR\\_2019/html/Zhao\\_Data\\_Augmentation\\_Using\\_Learned\\_Transformations\\_for\\_One-Shot\\_Medical\\_Image\\_Segmentation\\_CVPR\\_2019\\_paper.html](http://openaccess.thecvf.com/content_CVPR_2019/html/Zhao_Data_Augmentation_Using_Learned_Transformations_for_One-Shot_Medical_Image_Segmentation_CVPR_2019_paper.html)
- [14] H. Shin, N. A. Tenenholtz, J. K. Rogers, C. G. Schwarz, M. L. Senjem, J. L. Gunter, K. P. Andriole, M. Michalski, Medical image synthesis for data augmentation and anonymization using generative adversarial networks, in: A. Gooya, O. Goksel, I. Oguz, N. Burgos (Eds.), Simulation and Synthesis in Medical Imaging - Third International Workshop, SASHIMI 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 16, 2018, Proceedings, Vol. 11037 of Lecture Notes in Computer Science, Springer, 2018, pp. 1–11.  
URL [https://doi.org/10.1007/978-3-030-00536-8\\_1](https://doi.org/10.1007/978-3-030-00536-8_1)
- [15] M. Frid-Adar, I. Diamant, E. Klang, M. Amitai, J. Goldberger, H. Greenspan, Gan-based synthetic medical image augmentation for increased CNN performance in liver lesion classification, *Neurocomputing* 321 (2018) 321–331.  
URL <https://doi.org/10.1016/j.neucom.2018.09.013>
- [16] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin, A. A. Kalinin, Albumentations: Fast and flexible image augmentations, *Inf.* 11 (2) (2020) 125.  
URL <https://doi.org/10.3390/info11020125>

- 
- [17] M. D. Bloice, C. Stocker, A. Holzinger, Augmentor: An image augmentation library for machine learning, *J. Open Source Softw.* 2 (19) (2017) 432.  
URL <https://doi.org/10.21105/joss.00432>
- [18] S. B. Lo, S. A. Lou, J. Lin, M. T. Freedman, M. V. Chien, S. K. Mun, Artificial convolution neural network techniques and applications for lung nodule detection, *IEEE Trans. Medical Imaging* 14 (4) (1995) 711–718.  
URL <https://doi.org/10.1109/42.476112>
- [19] A. Barbu, L. Lu, H. Roth, A. Seff, R. M. Summers, An analysis of robust cost functions for CNN in computer-aided diagnosis, *Comput. methods Biomech. Biomed. Eng. Imaging Vis.* 6 (3) (2018) 253–258.  
URL <https://doi.org/10.1080/21681163.2016.1138240>
- [20] H. R. Roth, L. Lu, N. Lay, A. P. Harrison, A. Farag, A. Sohn, R. M. Summers, Spatial aggregation of holistically-nested convolutional neural networks for automated pancreas localization and segmentation, *Medical Image Anal.* 45 (2018) 94–107.  
URL <https://doi.org/10.1016/j.media.2018.01.006>
- [21] A. Varghese, M. S. K. P., S. S. Chennamsetty, G. Krishnamurthi, Generative adversarial networks for brain lesion detection, in: M. A. Styner, E. D. Angelini (Eds.), *Medical Imaging 2017: Image Processing*, Orlando, Florida, United States, 11-16 February 2017, Vol. 10133 of SPIE Proceedings, SPIE, 2017, p. 101330G.  
URL <https://doi.org/10.1117/12.2254487>
- [22] X. Chen, E. Konukoglu, Unsupervised detection of lesions in brain MRI using constrained adversarial auto-encoders, *CoRR* abs/1806.04972.  
URL <http://arxiv.org/abs/1806.04972>
- [23] S. Liu, L. Qi, H. Qin, J. Shi, J. Jia, Path aggregation network for instance segmentation, in: *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018*, Salt Lake City, UT, USA, June 18-22, 2018, IEEE Computer Society, 2018, pp. 8759–8768.  
URL [http://openaccess.thecvf.com/content\\_cvpr\\_2018/html/Liu\\_Path\\_Aggregation\\_Network\\_CVPR\\_2018\\_paper.html](http://openaccess.thecvf.com/content_cvpr_2018/html/Liu_Path_Aggregation_Network_CVPR_2018_paper.html)

- [24] 王艺培, 闫雯, 张益肇, 来茂德, 许燕, 面向精准医疗的人工智能在病理图像中的应用, 现代信息科技 v.2 (05) (2018) 178–180.
- [25] R. A. Robb, X-ray computed tomography: from basic principles to applications, *Annual review of biophysics and bioengineering* 11 (1) (1982) 177–201.
- [26] E. Andrew, Nuclear magnetic resonance and the brain, *Brain topography* 5 (2) (1992) 129–133.
- [27] 董金宝, 张小亚, 西安创联超声技术有限责任公司, 超声波电机在核磁共振医疗设备中的应用设计, in: 中国电工技术学会, 2010.
- [28] A. Mittal, A. K. Moorthy, A. C. Bovik, No-reference image quality assessment in the spatial domain, *IEEE Transactions on image processing* 21 (12) (2012) 4695–4708.
- [29] X. Liu, J. van de Weijer, A. D. Bagdanov, Rankiqa: Learning from rankings for no-reference image quality assessment, in: *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, IEEE Computer Society, 2017, pp. 1040–1049.  
URL <https://doi.org/10.1109/ICCV.2017.118>
- [30] W. Zhang, K. Ma, J. Yan, D. Deng, Z. Wang, Blind image quality assessment using a deep bilinear convolutional neural network, *IEEE Trans. Circuits Syst. Video Technol.* 30 (1) (2020) 36–47.  
URL <https://doi.org/10.1109/TCSVT.2018.2886771>
- [31] D. Li, T. Jiang, M. Jiang, Quality assessment of in-the-wild videos, in: L. Amsaleg, B. Huet, M. A. Larson, G. Gravier, H. Hung, C. Ngo, W. T. Ooi (Eds.), *Proceedings of the 27th ACM International Conference on Multimedia, MM 2019, Nice, France, October 21-25, 2019*, ACM, 2019, pp. 2351–2359.  
URL <https://doi.org/10.1145/3343031.3351028>
- [32] A. Rehman, Z. Wang, Reduced-reference image quality assessment by structural similarity estimation, *IEEE transactions on image processing* 21 (8) (2012) 3378–3389.

- [33] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, Image quality assessment: from error visibility to structural similarity, *IEEE transactions on image processing* 13 (4) (2004) 600–612.
- [34] Z. Wang, A. C. Bovik, H. R. Sheikh, E. P. Simoncelli, Image quality assessment: from error visibility to structural similarity, *IEEE Trans. Image Process.* 13 (4) (2004) 600–612.  
URL <https://doi.org/10.1109/TIP.2003.819861>
- [35] C. P. Loizou, C. S. Pattichis, C. I. Christodoulou, R. S. Istepanian, M. Pantziaris, A. Nicolaides, Comparative evaluation of despeckle filtering in ultrasound imaging of the carotid artery, *IEEE transactions on ultrasonics, ferroelectrics, and frequency control* 52 (10) (2005) 1653–1669.
- [36] Z. Wang, E. P. Simoncelli, A. C. Bovik, Multiscale structural similarity for image quality assessment, in: *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003, Vol. 2, Ieee, 2003*, pp. 1398–1402.
- [37] L. Zhang, L. Zhang, X. Mou, D. Zhang, FSIM: A feature similarity index for image quality assessment, *IEEE Trans. Image Process.* 20 (8) (2011) 2378–2386.  
URL <https://doi.org/10.1109/TIP.2011.2109730>
- [38] P. Kovesei, et al., Image features from phase congruency, *Videre: Journal of computer vision research* 1 (3) (1999) 1–26.
- [39] W. Xue, L. Zhang, X. Mou, A. C. Bovik, Gradient magnitude similarity deviation: A highly efficient perceptual image quality index, *IEEE Trans. Image Process.* 23 (2) (2014) 684–695.  
URL <https://doi.org/10.1109/TIP.2013.2293423>
- [40] K. Yan, X. Wang, L. Lu, R. M. Summers, Deeplesion: Automated deep mining, categorization and detection of significant radiology image findings using large-scale clinical lesion annotations, *CoRR abs/1710.01766*.  
URL <http://arxiv.org/abs/1710.01766>

## 简历与科研成果

基本情况 王黛薇，女，汉族，1996年8月出生，四川省南充市人。

### 教育背景

**2019.9~2021.6** 南京大学软件学院 硕士

**2014.9~2019.6** 厦门大学软件学院 本科

### 读研期间成果

1. 国家自然科学基金项目（重点项目）：智能软件系统的数据驱动测试方法与技术（61932012），2020-2024



## 致 谢

时光飞逝，转眼间两年的研究生生活已接近尾声。回首两年的求学时光漫长而又短暂，有迷惘彷徨，但更多的是收获和成长。在此论文完成之际，向所有帮助和关心过我的老师朋友和家人们致以最诚挚的谢意。

首先要感谢我的研究生导师陈振宇教授和房春荣老师，从确定研究方向到系统的具体细节实现，老师们都给予了非常细致的指导，也提出了非常多宝贵的意见。两位老师严谨求实的治学态度、孜孜以求的工作作风以及大胆创新的进取精神也教会了我学习和工作的准则，这也将对我未来的发展道路产生深远的影响。同时也要感谢实验室的黄勇老师，在系统的架构设计和技术选型上提供了非常多的帮助，同时也经常会非常耐心地和我们一起讨论技术问题。

其次我要感谢实验室的各位同学们，从你们身上我学到了很多在学习和研究方面的知识以及认真坚持的态度，非常幸运能够加入 iSE 大家庭，实验室里丰富的项目实践机会和互帮互助的学习氛围也促使我能够快速提升和成长。同样要感谢我最亲爱的朋友们在我最迷茫和低谷时候的安慰和陪伴，是你们给了我前进的信心和动力，让我能够一直坚持自己想做的事情。

最后要感谢我的父母和家人们在千里之外的坚定支持和无言奉献，你们对我的信任理解和支持是我前进的最大动力。同时还要感谢参加论文评审和答辩的所有专家老师们的辛苦付出。