

多模态医学影像配准、分割与可视化系统

系统测试计划

版本 <1.1>

修订历史记录

日期	版本	说明	作者
<日/月/年>		<详细信息>	<姓名>
12.05.2020	1.0	测试计划3	罗媚、赵语云
12.12.2020	1.1	测试计划3 排版	李陈豪

目录

1. 简介
 - 1.1 目的
 - 1.2 背景
 - 1.3 范围
 - 1.4 项目标识
2. 测试需求
3. 测试策略
 - 3.1 测试类型
 - 3.1.1 数据和数据库完整性测试
 - 3.1.2 功能测试
 - 3.1.3 多模态影像配准算法精度测试
 - 3.1.4 图像分割算法精度测试
 - 3.1.5 用户界面测试
 - 3.1.6 肿瘤类别预测测试
 - 3.1.7 性能测试
 - 3.1.8 负载测试
 - 3.1.9 安装测试
 - 3.2 工具
4. 资源
 - 4.1 角色
5. 项目里程碑
 - 5.1 测试模型
 - 5.2 测试日志
 - 5.3 缺陷报告
6. 可交付工件
7. 附录 A: 项目任务

1. 简介

1.1. 目的

本“测试计划”文档有助于帮助本系统的功能测试。

1.2. 背景

本文档以医学图像处理系统作为输入，对医学图像处理系统的功能性需求、易用性需求、兼容性需求、性能需求等多方面进行测试，以此来判断该系统是否符合预期目标

1.3. 范围

这份测试计划将集成测试与系统测试应用于医学图像处理系统的所有发布版本。这份测试文档针对医学图像处理系统的功能性需求、非功能性需求如易用性、兼容性、性能等进行测试。

假设网络通信情况良好，服务器端程序正常运行。

本文档测试的需求均来自需求规约文档。

1.4. 项目标识

下表列出了制定测试计划所用的文档，并标明了文档的可用性：

文档 (版本/日期)	已创建或可用	已被接受或 经过复审	作者或来源	备注
需求规约	<input checked="" type="checkbox"/> 是 <input type="checkbox"/> 否	<input type="checkbox"/> 是 <input type="checkbox"/> 否	迭代一	

用例报告	<input checked="" type="checkbox"/> 是 <input type="checkbox"/> 否	<input type="checkbox"/> 是 <input type="checkbox"/> 否	迭代一	
项目计划	<input checked="" type="checkbox"/> 是 <input type="checkbox"/> 否	<input type="checkbox"/> 是 <input type="checkbox"/> 否	迭代一	
原型	<input checked="" type="checkbox"/> 是 <input type="checkbox"/> 否	<input type="checkbox"/> 是 <input type="checkbox"/> 否	罗媚	

2. 测试需求

下面列出了那些已被确定为测试对象的项目（用例、功能性需求和非功能性需求）。此列表说明了测试的对象。

测试对象	测试项目
服务器端	性能需求
用户Mac端	功能性需求
用户Mac端	易用性
用户Mac端	兼容性
用户Mac端	界面
用户Windows端	功能性需求
用户Windows端	易用性
用户Windows端	兼容性
用户Windows端	界面
管理员网页端	功能性需求
管理员网页端	易用性
管理员网页端	兼容性
管理员网页端	界面

功能性测试

按优先级 p0 至 p2 测试功能需求，需求详见需求规约文档

易用性测试

当某项操作失败时（如图像处理由于滤波尺寸不合法而进行滤波处理失败），需要弹出对应的提示框。

用户可以在30分钟内掌握本软件的主要功能

兼容性需求

程序在Mac端与Windows端都可正常运行并实现主要功能

界面

界面美观，表意清晰

性能测试

在用户平均网络硬件水平的情况下，每个操作相应时间不超过2秒。

任何问题造成未响应，系统在5秒后在页面上显示未响应等提示信息。

吞吐量：系统能同时处理 50 事务/秒。

服务器内存为 4GB，CPU 2核，磁盘 50GB，公共带宽 100 MB/s。

3. 测试策略

3.1. 测试类型

3.1.1. 数据和数据库完整性测试

测试目标：	确保数据库访问方法和进程正常运行，数据不会遭到损坏。
方法：	<ol style="list-style-type: none"> 1. 调用各个数据库访问方法和进程，并在其中填充有效的和无效的数据或对数据的请求。 2. 检查数据库，确保数据已按预期的方式填充，并且所有数据库事件都按正常方式出现；或者检查所返回的数据，确保为正当的理由检索到了正确的数据
完成标准：	所有的数据库访问方法和进程都按照设计的方式运行，数据没有遭到损坏。
需考虑的特殊事项：	

3.1.2. 功能测试

测试目标：	确保医学图像处理系统的主要功能正常，其中包括登录、三视图展示、图像可视化、图像配准和图像分割等。
方法：	<ol style="list-style-type: none"> 1. 针对测试图像依次进行上述功能测试 2. 测试内容： <ol style="list-style-type: none"> a. 在使用有效数据时得到预期的结果，展示窗口展示处理后的图像。 b. 在使用无效数据时显示相应的错误消息或警告消息，如输入的处理参数不合法、输入的图像无法解析等。 c. 各业务规则都得到了正确的应用。
完成标准：	<ol style="list-style-type: none"> 1. 所计划的测试已全部执行。 2. 所发现的缺陷已全部解决。
需考虑的特殊事项：	确定或说明那些将对功能测试的实施和执行造成影响的事项或因素（内部的或外部的）

3.1.3. 多模态影像配准算法精度测试

测试目标:	测试本系统所涉及的配准算法的精度
方法:	<p>(1) 分别在CT图像与配准后的MR图像上寻找一定数量的解剖结构特征点，如鼻尖、耳尖、骨骼拐点、空腔交界处以及一些解剖组织交界点。这些解剖结构特征点需要能够在两种模态下明显清晰地观察到；</p> <p>(2) 分别在两种模态下拾取上述特征点，得到其在物理空间中的坐标值；</p> <p>(3) 计算第(2)步中拾取到的每一对特征点物理坐标值之间的欧氏距离以代表二者之间的误差距离，对同一例中的所有对特征点均计算误差距离，计算其均值、中位数与标准差；</p> <p>(4) 在融合后的影像上可视化所选取的解剖特征点。</p> <p>从开源MR/CT影像数据库中选择三组进行配准精度验证实验，在每组MR/CT数据上寻找9个解剖特征点</p>
测试结果	配准前后具体误差值见下表，观察表中“配准后”三行数据可发现，配准后解剖特征点间的误差平均值与中位数均较小，并且从三组数据的平均值可看出本文所设计的配准算法的误差大约在1.3mm左右，此配准实验所使用的数据在x轴与y轴上的分辨率为0.49mm，即本实验的配准误差大概为2.5个像素。

3.1.4. 图像分割算法精度测试

测试目标:	测试本系统所涉及的分割算法的精度
方法:	<p>在MICCAI BraTS2017数据集中进行分割精度验证实验。该数据集中包含有若干例高级别胶质瘤与低级别胶质瘤的多模态磁共振图像（共四种模态：FLAIR，T1，T2 和T1c），并且提供了由影像科专家手动勾画的脑肿瘤分割结果真值标签，选用该数据集进行分割算法精度验证实验。</p> <p>从该数据集中随机选出15例FLAIR模态磁共振影像（维度均为240*240*155）进行分割，计算分割结果与真值之间的精度。 选用分割精度指标分别为：Dice相似系数（DSC）、阳性预测值（PPV）与灵敏度（Sensitivity）</p>
测试结果	具体精度值见下表所示，三个指标的平均值与中位值都在0.89以上，标准差均在0.05以下，表明本项目所使用的算法有着较高的分割精度与稳定性，能够满足临床实际需求

3.1.5. 用户界面测试

测试目标:	确保用户界面美观且可正常使用，具体表现为： 1. 界面UI，若可点击，对应事件可正常运行 2. 整体界面展示美观，窗口的位置布置相对稳定 3. 主要功能的实现都有对应的UI入口
方法:	1. 测试每个主要功能对应的UI入口，测试功能是否可以正常运行 2. 输入不合法的参数，测试软件是否会有对应的错误信息提示
完成标准:	1. UI对应的功能都可正常运行，结果展示在窗口且结果正确 2. 输入不合法时，软件会弹出对应的错误提示
需考虑的特殊事项:	

3.1.6. 肿瘤类别预测测试

测试目标:	测试本系统所使用的的肿瘤类别预测功能的精度
方法:	使用开源医学肿瘤影像数据集进行图像分类预测。该数据集包含多种肿瘤类别以及其良恶性的类别标签。预测模型通过机器学习搭建神经网络进行实现，将通过python内开源机器学习框架pytorch进行预测效果的评估
完成标准:	1. 输入用户给出的肿瘤影像图片，能够输出预测结果 2. 预测模型在测试数据上能达到95%以上的精度
需考虑的特殊事项:	预测精度不达标需考虑更换机器学习模型

3.1.7 性能评价

测试目标：	测试软件可支持的工作量是否符合需求规约文档中的要求
方法：	<ol style="list-style-type: none">1. 通过修改数据文件来增加事务数量，或通过修改脚本来增加每项事务的迭代次数。2. 脚本应该在一台计算机上运行（最好是以单个用户、单个事务为基准），并在多台客户机上重复。
完成标准：	<ol style="list-style-type: none">1. 单个事务或单个用户：在每个事务所预期或要求的时间范围内成功地完成测试脚本，没有发生任何故障。2. 多个事务或多个用户：在可接受的时间范围内成功地完成测试脚本，没有发生任何故障。
需考虑的特殊事项：	<ol style="list-style-type: none">1. 性能测试应使用相关的测试工具2. 性能测试应在专用的计算机上执行，以便实现完全的控制和精确的评测3. 性能测试可在服务器应对不同流量的情况下进行

3.1.8. 负载测试

测试目标：	测试本系统在一定负载下仍可在规定时间内响应请求
方法：	<ol style="list-style-type: none">1. 可测试单一用户在不同QPS下的响应时间2. 可测试多个并发用户对于同一个API的响应时间
完成标准：	多个事务或多个用户：在可接受的时间范围内成功地完成测试，没有发生任何故障。
需考虑的特殊事项：	<ol style="list-style-type: none">1. 负载测试应该在专用的计算机上或在专用的机时内执行，以便实现完全的控制和精确的评测。2. 负载测试所用的数据库应该是与实际大小相同或等比例缩放的数据库。

3.1.9. 安装测试

测试目标：	核实在以下情况下，测试对象可正确地安装到各种所需的硬件配置中： <input type="checkbox"/> 首次安装。以前从未安装过的新计算机 <input type="checkbox"/> 更新。以前安装过相同版本的计算机 <input type="checkbox"/> 更新。以前安装过较早版本的计算机
方法：	1. 在同一计算机上依次进行安装、更新相同版本、更新较早版本的操作
完成标准：	事务成功执行，没有出现任何故障。
需考虑的特殊事项：	应用程序成功安装后是否能正常运行，是否所有主要功能都能正常运行

3.2. 工具

此项目将使用以下工具：

	工具	厂商/自行研制	版本
测试管理	腾讯文档	腾讯	
缺陷跟踪	腾讯文档	腾讯	
用于功能性测试的 ASQ 工具	QTest	开源框架	5.12
项目管理	Github	Github	

4. 资源

4.1. 角色

下表列出了在此项目的人员配备方面所作的各种假定。

人力资源		
角色	推荐的最少资源	具体职责或注释
测试经理， 测试项目经理	张诏威	进行管理监督。 职责： <ul style="list-style-type: none">● 提供技术指导● 获取适当的资源● 提供管理报告
测试设计员	张诏威，赵语云，李陈豪， 罗媚，张详宁	确定测试用例、确定测试用例的优先级并实施测试用例。 职责： <ul style="list-style-type: none">● 生成测试计划● 生成测试模型● 评估测试工作的有效性
测试员	张诏威，赵语云，李陈豪， 罗媚，张详宁	执行测试。 职责： <ul style="list-style-type: none">● 执行测试● 记录结果● 从错误中恢复● 记录变更请求
测试系统管理员	张诏威	确保测试环境和资产得到管理和维护。 职责： <ul style="list-style-type: none">● 管理测试系统● 授予和管理角色对测试系统的访问权
数据库管理员	张诏威	确保测试数据（数据库）环境和资产得到管理和维护。 职责： <ul style="list-style-type: none">● 管理测试数据（数据库）
设计员	张诏威，赵语云，李陈豪， 罗媚，张详宁	确定并定义测试类的操作、属性和关联。

	5. 项目里程碑	
--	----------	--

		职责： ● 确定并定义测试类 ● 确定并定义测试包
实施员	张诏威，赵语云，李陈豪，罗媚，张详宁	实施测试类和测试包，并对它们进行单元测试。 职责： ● 创建在测试模型中实施的测试类和测试包

5. 项目里程碑

里程碑任务		开始日期	结束日期
迭代一	制定测试计划	2020.10.09	2020.10.13
	设计测试	2020.10.13	2020.10.16
	实施测试	2020.10.16	2020.10.30
	执行测试	2020.10.16	2020.10.30
	评估测试	2020.10.25	2020.11.01
迭代二	制定测试计划	2020.11.02	2020.11.06
	设计测试	2020.11.06	2020.11.09
	实施测试	2020.11.09	2020.11.28
	执行测试	2020.11.16	2020.11.28
	评估测试	2020.11.25	2020.11.30
迭代三	制定测试计划	2020.12.01	2020.12.06
	设计测试	2020.12.06	2020.12.09
	实施测试	2020.12.09	2020.12.31
	执行测试	2020.12.16	2020.12.31
	评估测试	2020.12.28	2021.01.05

5.1. 测试模型

采用层次模型：单元测试，集成测试，系统测试。

5.2. 测试日志

使用腾讯文档记录测试日志，每个测试用例需要包含用例标题、测试优先级、前置条件、操作步骤、预期结果、实际记过、是否通过、错误严重等级、备注。若测试未通过，需要提bug单，表述bug产生的频率，复现的步骤。

5.3. 缺陷报告

使用腾讯文档记录产生的缺陷。

6. 可交付文件

	创建人员	交付对象	交付时间
测试用例	项目组	测试经理	每个迭代最后一周
测试报告	项目组	测试经理	每个迭代最后一周

附录 A：项目任务

以下是一些与测试有关的任务：

☐ 制定测试计划

- 确定测试需求
- 评估风险
- 制定测试策略
- 确定测试资源
- 创建时间表
- 生成测试计划

☐ 设计测试

- 准备工作量分析文档
- 确定并说明测试用例
- 确定并结构化测试过程
- 复审和评估测试覆盖

☐ 实施测试

- 记录或通过编程创建测试脚本
- 确定设计与实施模型中的测试专用功能
- 建立外部数据集

☐ 执行测试

- 执行测试过程
- 记录测试结果并报告

- 评估测试的执行情况
- 恢复暂停的测试

- 核实结果
- 调查意外结果
- 记录缺陷

☐ 评估测试

- 评估测试用例覆盖
- 评估代码覆盖
- 分析缺陷
- 确定是否达到了测试完成标准与成功标准

