智能医学图像分割系统

软件架构文档

版本 <1.0>

目录

1. 简介 3

1.1 目的 3

1.2 范围 3

1.3 参考资料 3

2. 架构目标和约束 3

3. 用例视图 3

3.1 提问 3

3.2 反馈 4

4. 逻辑视图 4

4.1 三层架构 4

4.2 表现层 5

*4.3* 智能分割处理层 6

*4.4* 数据层 7

5. 部署视图 8

6. 数据视图 9

7. 技术视图 12

8. 质量属性的设计 12

8.1 易用性设计 12

8.2 可靠性设计 12

8.3 性能设计 12

8.4 可支持性设计 12

软件架构文档

# 简介

## 目的

本文档将从架构方面对系统进行综合概述，其中会使用多种不同的架构视图来描述系统的各个方面。它用于记录并表述已对系统的构架方面做出的重要决策，各程序模块的设计考虑，为软件编程和系统维护提供基础。

## 范围

本文档用于高级软件开发课程第五小组正在开发中的智能医学图像分割系统。智能医学图像分割系统是通过web网页形式为医学专业人员或其他人员提供医学图像分割结果，数据共享平台的系统，从而为医疗诊断提供数据支持。本文档的预期读者包括系统设计人员、软件开发人员、软件测试人员和项目评审人员。

## 参考资料

1. 智能医学图像分割系统项目建议书1.0版本
2. 智能医学图像分割系统开发计划1.0版本
3. 引用文件、资料，软件开发标准等

# 架构目标和约束

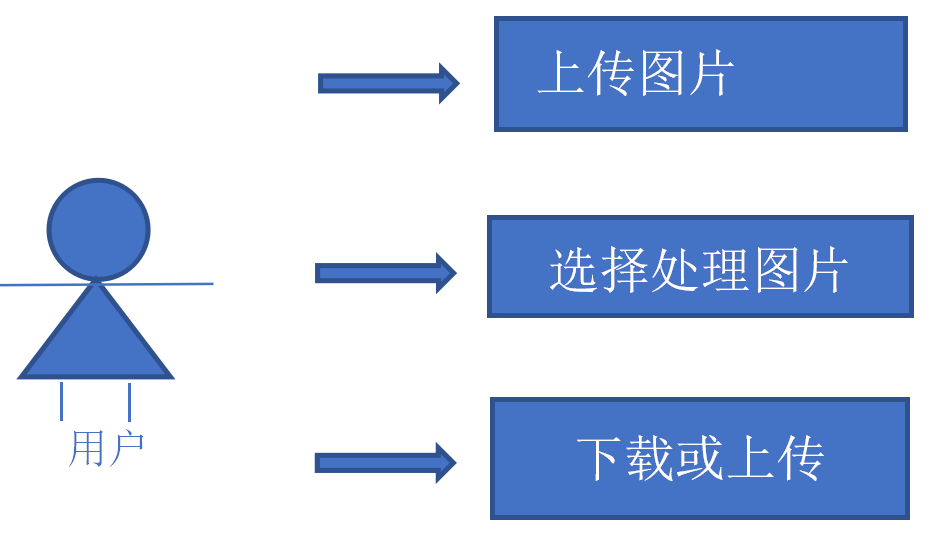
架构目标：

1. 支持本系统所有功能需求和非功能需求
2. 能够实现算法和模型的快速迭代和更新

架构约束：

系统在开发过程中有如下设计约束：应采用B/S架构，主要开发语言为python，Numpy系统存放和处理数据，采用基于tensorflow的CPU版keras作为环境平台，采用Flask作为服务器端网页开发工具。

# 用例视图

  
图1 系统用例图

## 上传图片

用户可以通过网页端，选择上传需要处理的图片。每次上传一张图片，后端的服务器会收到相应的通知。

## 选择处理图片

用户在选择上传了图片之后，可以选择不同的方式处理图片，后台通过服务器会把相应的图像处理结果返回到网页端。

用户可以选择的图像处理操作的种类主要包括如下三类：

（1）主动分割，通过服务器后端的机器学习算法完成对于各类医学图像的自动分割。

（2）基本图像处理操作，包括旋转、放大、缩小、反转、平移等功能，此项功能主要服务于下一项的手动分割。

（3）手动分割，使用者也可以选择手动来处理各种医学图像。

## 选择下载或上传处理后的图片

（1）医生可以选择将自动处理后的图像下载，以便于之后的诊断。

（2）选择手动分割后，若是医生对于分割的结果满意，可以将处理后的图片结果上传到服务器。服务器收集到一定量的手动分割结果后，将会再次运用机器学习的手段进行模型预测，新的模型将通过软件版本更新的方式传递给客户。

# 逻辑视图

## 三层架构

本项目软件逻辑架构分为三层，分别是表现层、智能分割处理层和数据层，如下图所示：

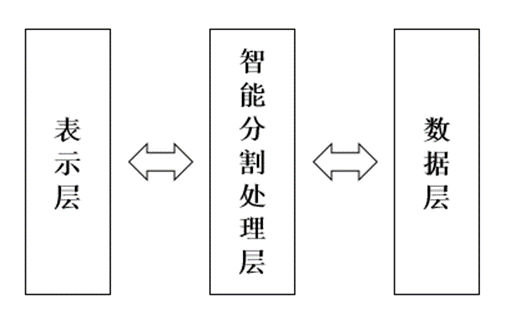


图2 三层架构

其中，表现层向用户呈现界面，接收用户请求发给智能分割处理层，并从智能分割处理层获取结果图像；智能分割处理层执行图像处理逻辑以处理用户请求，并将结果存入数据层或调用数据层得到图像分割网络参数；数据层负责样本图像的存储，以及准备智能分割处理层所需的数据。

## 表现层

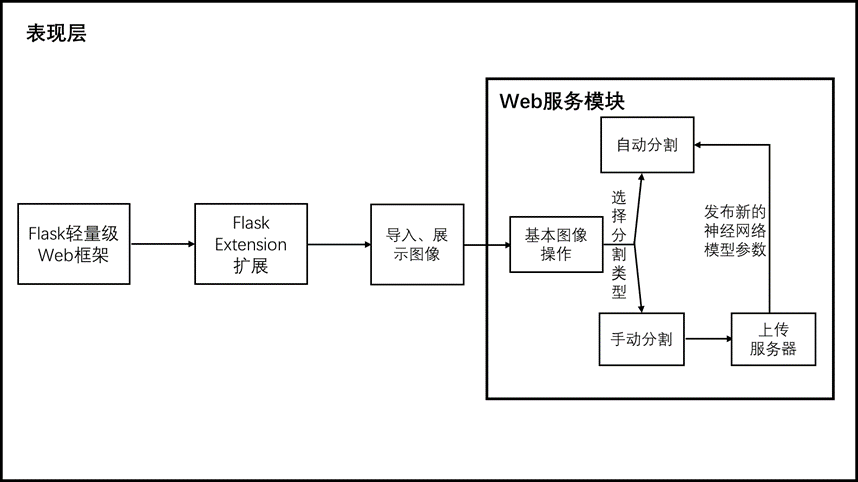


图3 表现层

在表现层中使用python中轻量级的Flask Web框架和其Extension扩展工具作为与用户交互的界面,部署在Web服务器中的Http服务负责提供Web中的自动分割、手动分割结果上传和更新神经网络模型参数的功能。表现层中包括如下几个关键步骤：

1. 导入并展示图像。
2. 提供基本的图像操作
3. 根据用户选择的图像种类和分割需求，进行图像的自动分割或手动分割。
4. 完成自动分割后，显示并导出分割结果
5. 完成手动分割后，上传图像和分割数据到服务器

## *智能分割处理层*

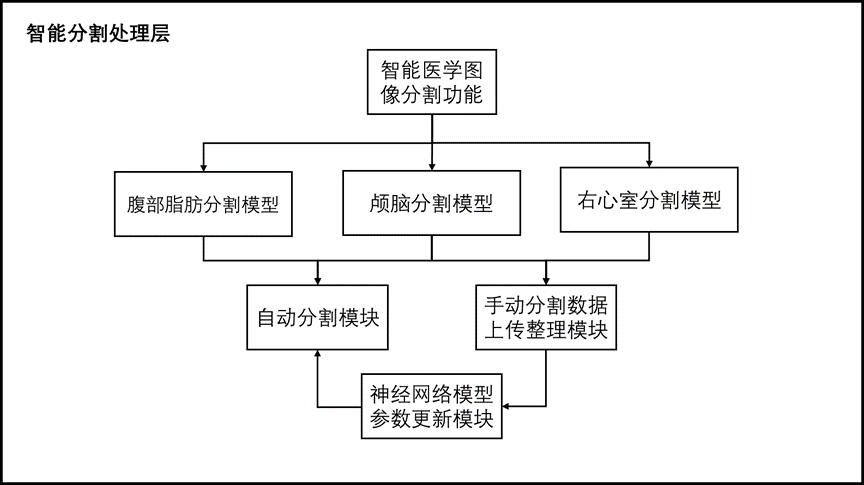


图4智能分割处理层

智能分割处理层部署在web服务器中，接受http请求，对收到的医学图像进行相应种类的自动分割处理，或对收到的手动分割图像和分割数据进行归并整理后训练新的神经网络模型。智能分割处理层规划有三种分割类型，分别为腹部脂肪分割、颅脑分割模型和右心室分割模型。

对应不同的分割类型，我们可以选择根据现有数据训练的神经网络模型进行预测，实现自动分割功能；也可以由医生进行图像手动分割后上传数据到服务器，由服务器整理后再次进行自动分割模型学习训练，更新自动分割模型参数。

## 数据层

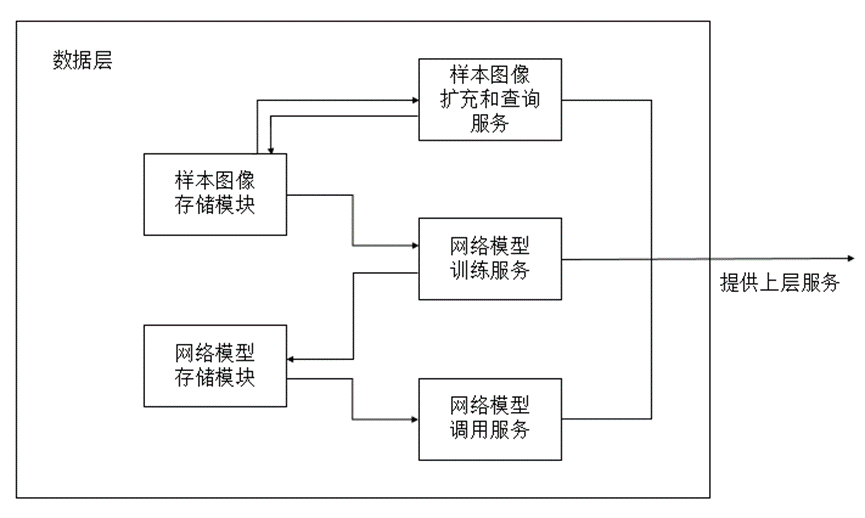


图7 数据层

数据层主要包括样本图像数据和网络模型数据，为上层提供服务。其中样本图像可根据上层请求，使用上层图像数据进行扩充，并提供查询服务。样本图像还用以提供神经网络模型训练服务，训练好的网络模型参数可存储在网络模型数据库中。网络模型数据库为智能分割处理层提供神经网络参数，提供调用服务，其中不同分割类型会对应不同的神经网络模型及相应参数。

# 部署视图

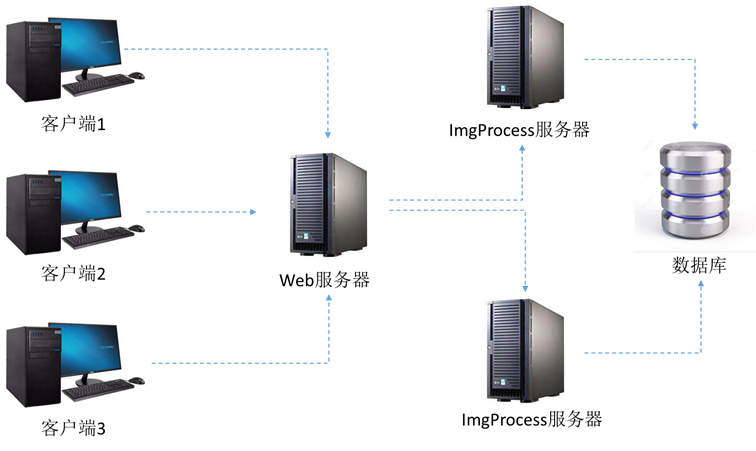


图8 系统部署视图

1. **客户端IDE**

用户通过客户端使用Http请求与Web服务器进行交互，用户需要联网才能进行图像上传、基本图像处理以及图像分割等操作。

1. **Web服务器**

Web服务器运行了一个响应用户请求的Web服务。该Web服务接到客户端请求后，通过http调用ImgProcess服务器接口，允许用户执行基本的图像操作，并选择图像分割的类型，之后返回自动分割的结果，或者将手动分割的图像上传到服务器，动态扩充训练模型，发布新的神经网络参数。Web服务器可调用不同的ImgProcess服务器，这些ImgProcess服务器可能基于不同的图像处理算法实现，也可能是同一算法的不同实例。考虑到被分割图像来源于人体不同生理组织，其特性在应用不同分割算法时会造成分割结果差异，因此将部署多个ImgProcess服务器，并由Web服务器进行最佳图像分割结果调度。

1. **ImgProcess服务器**

每个ImgProcess服务器运行了一种特定的图像分割算法，可能是基本图像操作算法、手动分割或者自动分割算法等。它通过http提供可输入不同参数的ImgProcess 接口，以对应用户在客户端中不同的图像需求。ImgProcess服务器通过局域网调用样本图像服务器。

1. **样本图像服务器**

样本图像服务器使用MySQL存储MRI数据，对外基于http提供MySql的新增、编辑、删除、显示等服务。因为我们的知识库构建完成后基本只允许用户对自己上传的图像进行基本图像处理和分割操作，所以这里不会由于多用户的读写冲突产生性能瓶颈。

# 数据视图

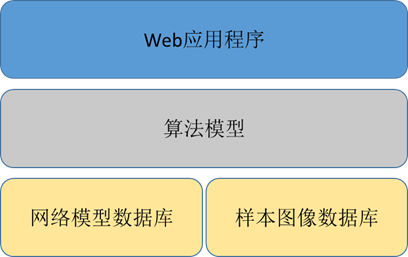


图9 自下而上的数据流

图9描述了数据如何在各模块之间流动的，数据自下而上被加工处理，最后被应用程序所使用，完成功能需求。

# 技术视图

本系统的技术选型如下：

编程语言：Python

使用Flask框架进行网页的开发

IDE支持PyCharm

使用Tensorflow框架进行深度学习的研究

# 质量属性的设计

## 易用性设计

界面友好简洁，方便用户操作

## 可扩展性设计

可以很方便的扩展软件系统的功能，各个模块独立性强，方便进行模块的扩展和功能的添加

## 可维护性

后期的维护容易

## 可支持性设计

Python代码根据Python的标准风格规范进行编写。组员编程参照Google 编程规范互相进行Code Review。