

第一章 需求分析

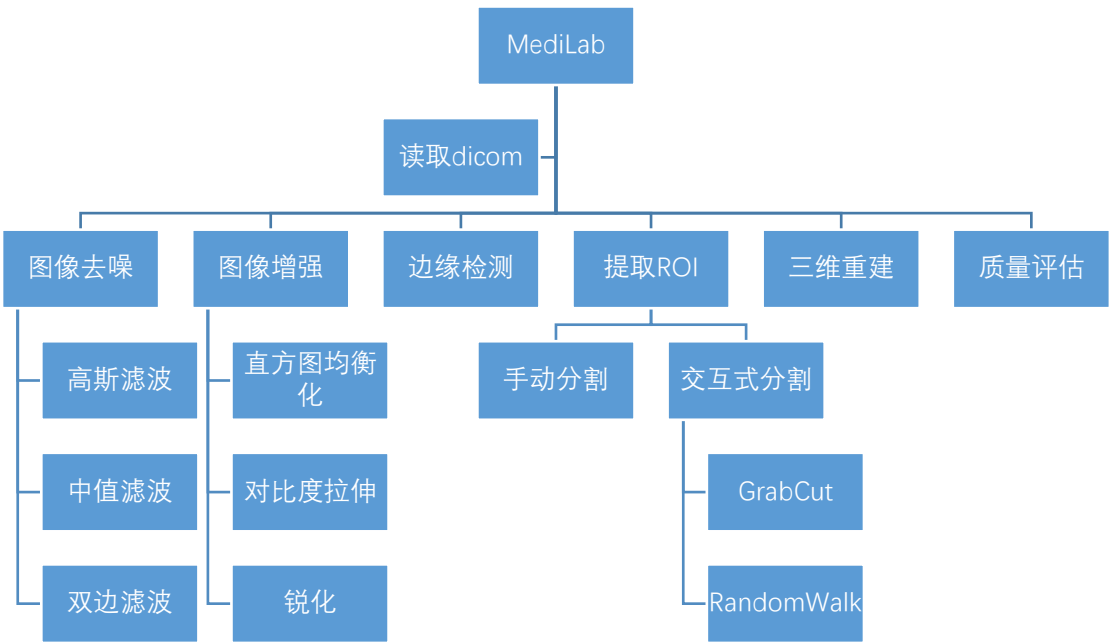
随着医学影像技术的不断发展，医学图像处理技术在医学领域的应用越来越广泛。通过对医学图像进行去噪、增强、分割等处理，可以更加清晰地观察病变部位，为临床诊断和治疗提供更加精准的依据。本作品的开发旨在满足医学图像处理的实际需求，为医生和科研人员提供一种方便、快捷、高效的图像处理工具。

市场上已经存在一些医学图像处理软件，例如 ImageJ、OsiriX MD、Mimics 等。这些软件各有优劣，有些功能丰富但操作复杂，有些界面友好但性能有限，有些专业度高但价格昂贵。

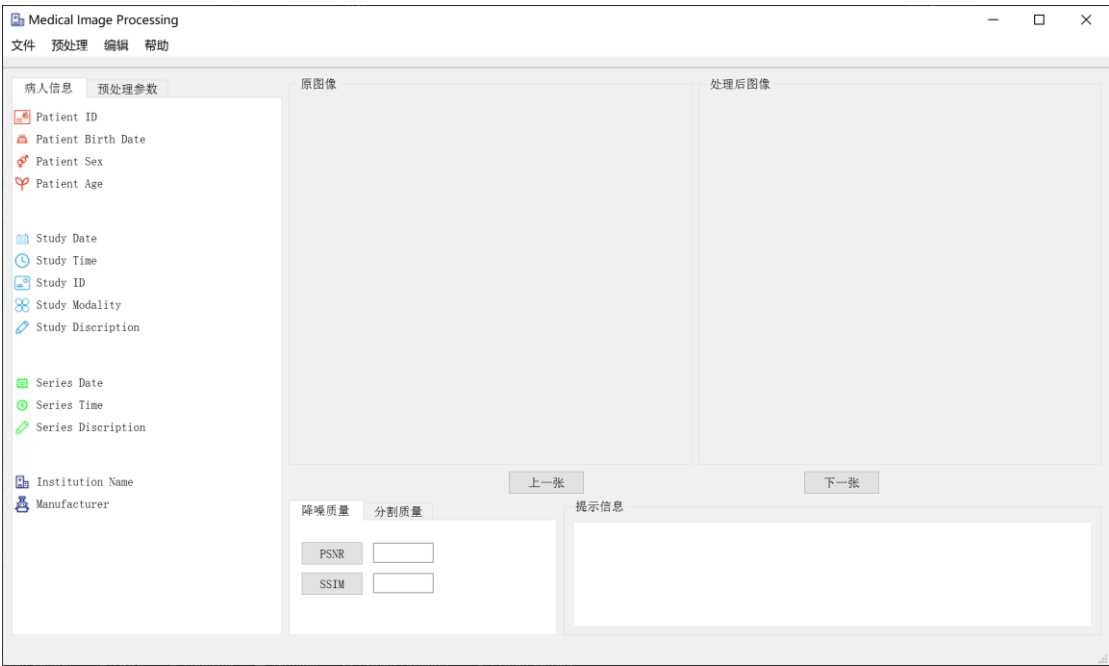
本作品主要面向医生、科研人员和医学生，他们需要对医学图像进行处理、分析和研究。同时，对于一些对医学图像处理感兴趣的普通用户，也可以使用本作品进行一些简单的操作和学习。

本产品的主要功能有：图像去噪、图像增强、手动勾画靶区、交互式分割、三维重建、图像质量评估等。

第二章 概要设计



第三章 详细设计



软件使用 Python3.8 环境开发，主要使用了 QtDesigner 设计界面，界面左边为病人信息，可以读取 dicom 文件中已标注的信息，如病人基本信息、临床基本信息等；还有“预处理参数”，可以调整滤波的卷积核大小，以便寻找更优的滤波效果。右边的主要部分为图像和处理后图像，有较好的对比效果。下方布局了图像质量评估（降噪质量和分割质量）、提示信息。上方的菜单栏中布局了“文件”、“预处理”、“编辑”和“帮助”，其中关键操作都在菜单栏中。

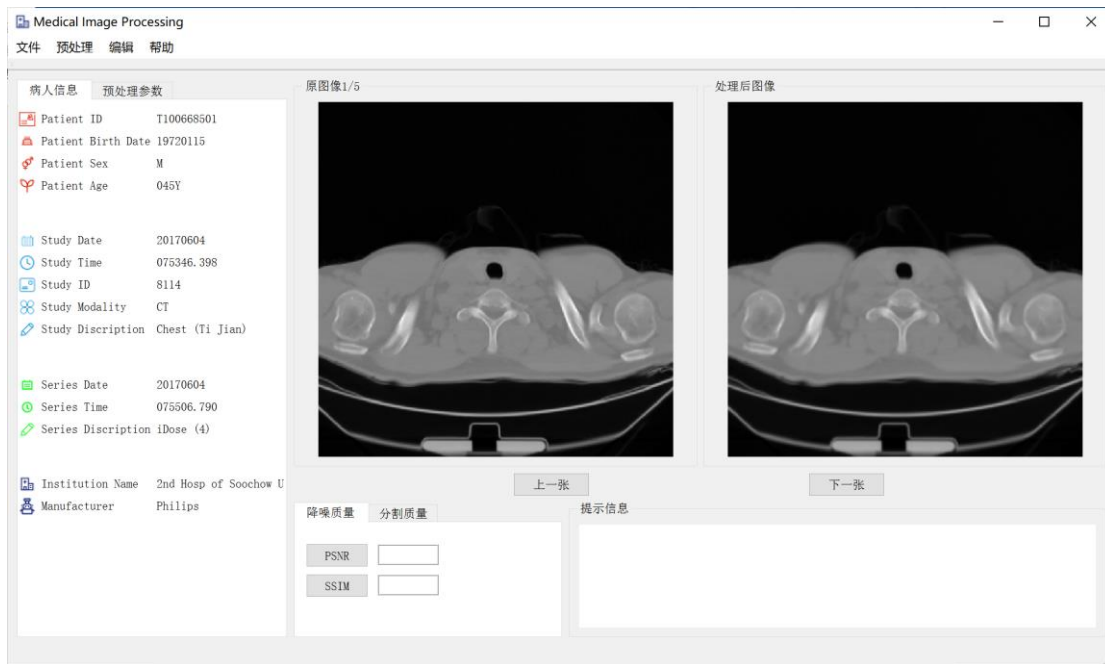
下面逐个介绍程序中应用到的算法：

➤ 高斯滤波：用高斯核对图像进行卷积，以减少图像中的噪声和细节。高斯滤波是一种图像模糊滤波器，它用正态分布计算图像中每个像素的变换。

高斯滤波的公式为：

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

其中，x 和 y 是像素点的坐标，σ是高斯函数的标准差。

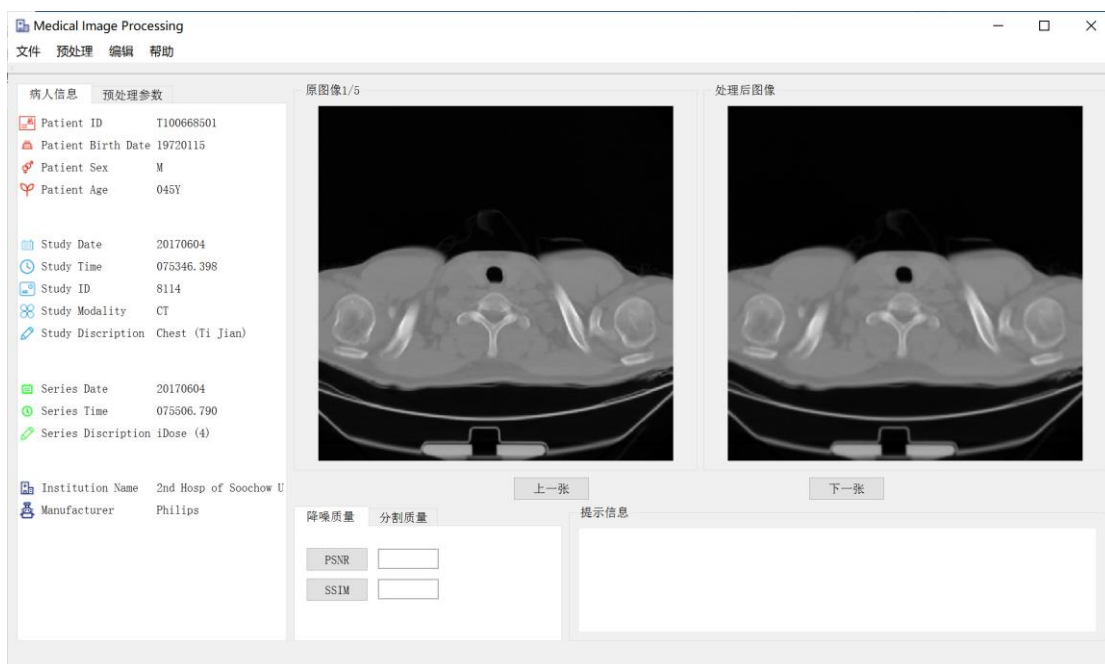


➤ 双边滤波：双边滤波是一种非线性的滤波方法，是结合图像的空间邻近度和像素值相似度的一种折衷处理，同时考虑空域信息和灰度相似性，达到保边去噪的目的。具有简单、非迭代、局部的特点。

双边滤波的公式为：

$$G(x, y) = \frac{1}{2\pi\sigma_s^2} e^{-\frac{(x-x')^2 + (y-y')^2}{2\sigma_s^2}} \cdot \frac{1}{2\pi\sigma_r^2} e^{-\frac{(I(x, y) - I(x', y'))^2}{2\sigma_r^2}}$$

其中， x 和 y 是像素点的坐标， $I(x, y)$ 是像素点的灰度值， σ_s 和 σ_r 分别是空间域和灰度域的标准差。



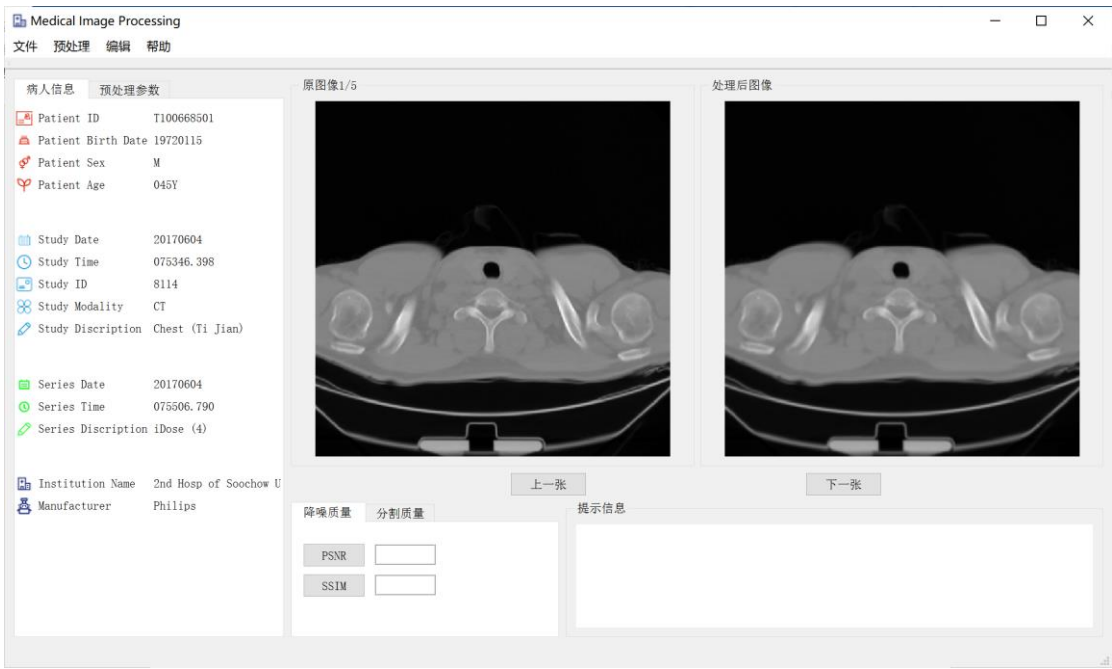
➤ 中值滤波：中值滤波是一种非线性的滤波方法，它是在图像中采用排序统计

的思想，用像素点周围的邻域灰度值的中值来代替该像素点的灰度值，从而达到去除噪声的目的。

中值滤波的公式为：

$$g(x,y)=median(f(x,y),f(x-1,y),f(x+1,y),f(x,y-1),f(x,y+1))$$

其中， x 和 y 是像素点的坐标， $f(x,y)$ 是像素点的灰度值， $g(x,y)$ 是处理后图像的灰度值。

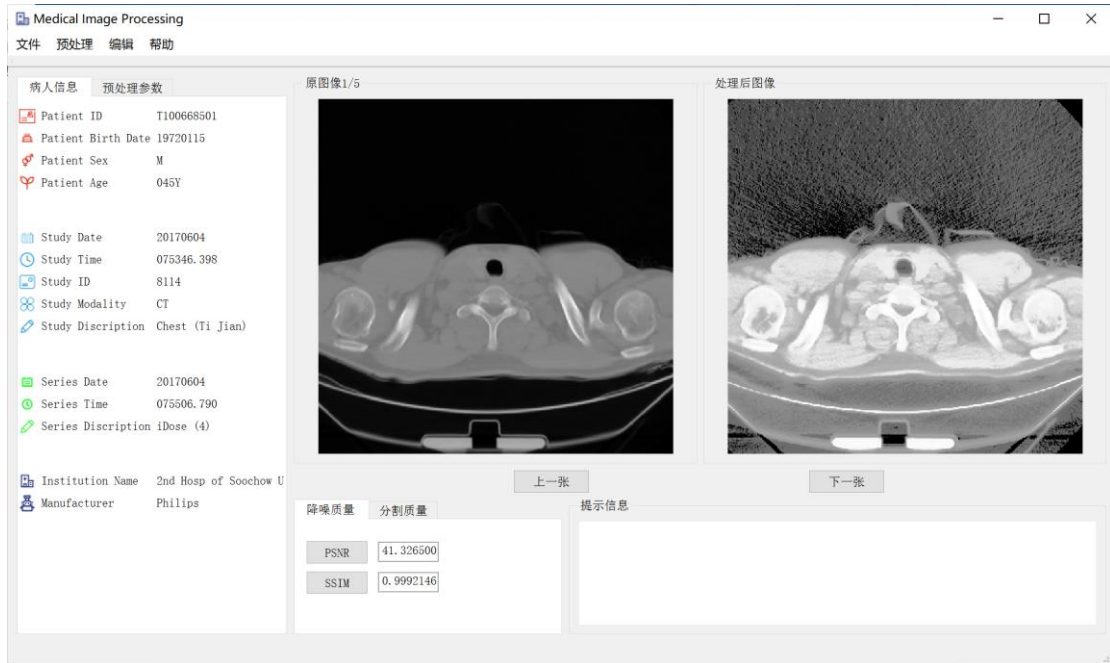


➤ 直方图均衡化：对图像的直方图进行变换，以增强图像的对比度和亮度。直方图均衡化是一种常见的图像增强方法，它可以通过对图像像素值的重新分配来增强图像的对比度。

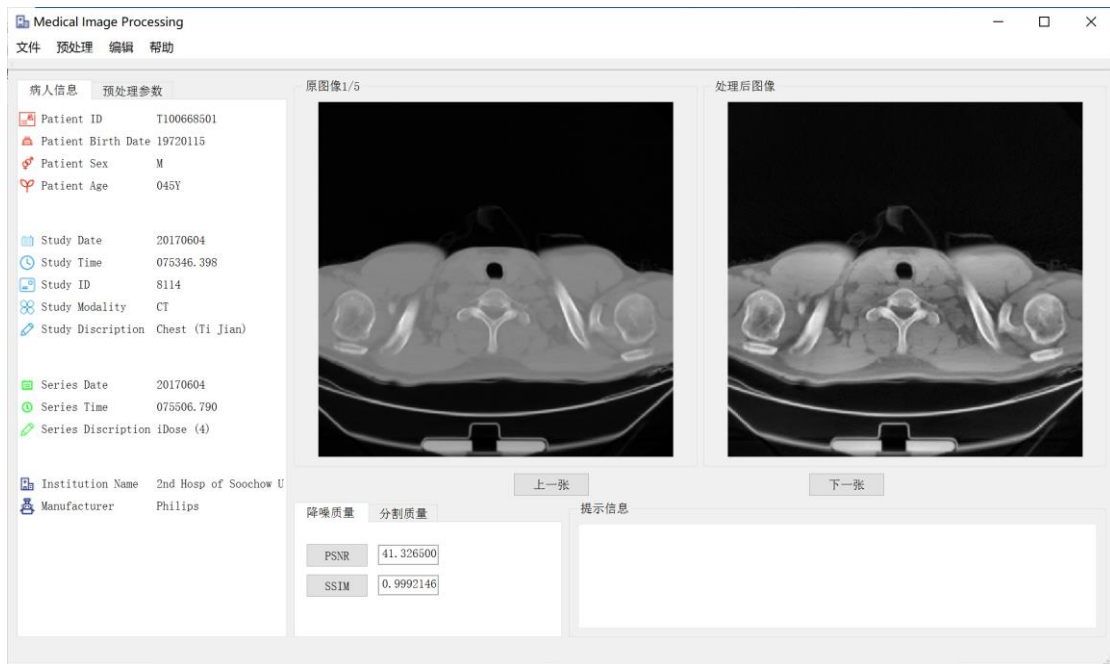
直方图均衡化的公式为：

$$s_k = \frac{(L-1)}{MN} \sum_{j=0}^k n_j$$

其中， s_k 是输出灰度级， L 是灰度级数， MN 是图像的总像素数， n_j 是灰度级 j 的像素数。



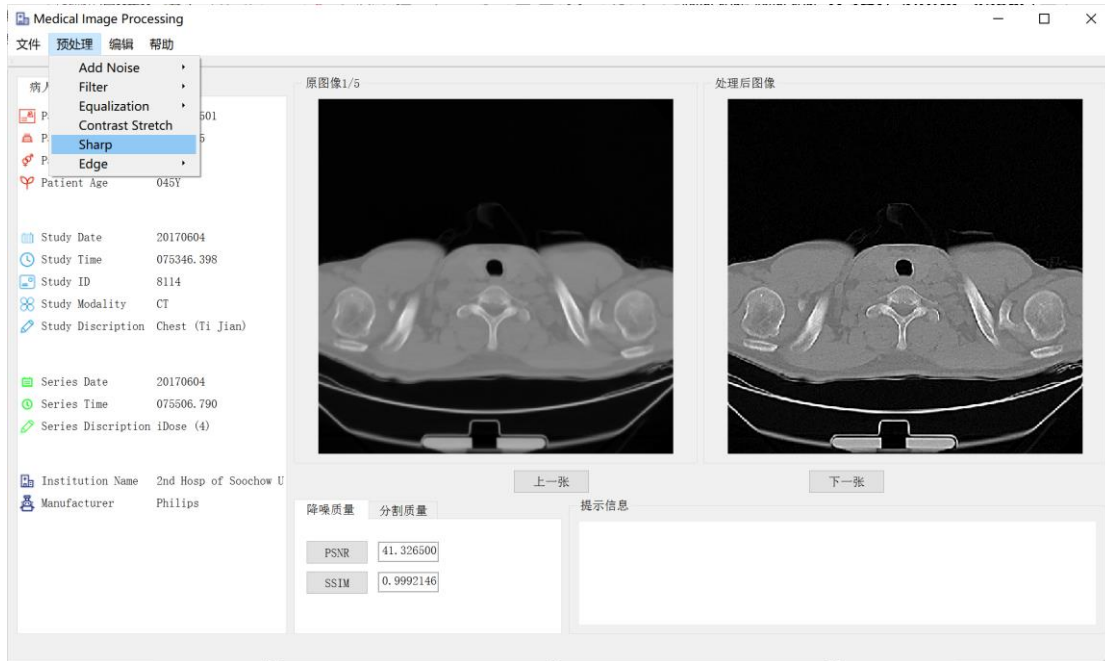
➤ 自适应直方图均衡化：自适应直方图均衡化与直方图均衡化不同之处在于，它将图像分成许多小区域，并对每个小区域进行直方图均衡化。这样可以避免在整个图像上进行全局直方图均衡化时出现过度增强和噪声放大的问题。



➤ 锐化：锐化是一种常见的图像增强方法，它可以通过增强图像的高频部分来使图像更加清晰。锐化的公式为：

$$g(x,y) = f(x,y) + k[f(x,y) - h(x,y)]$$

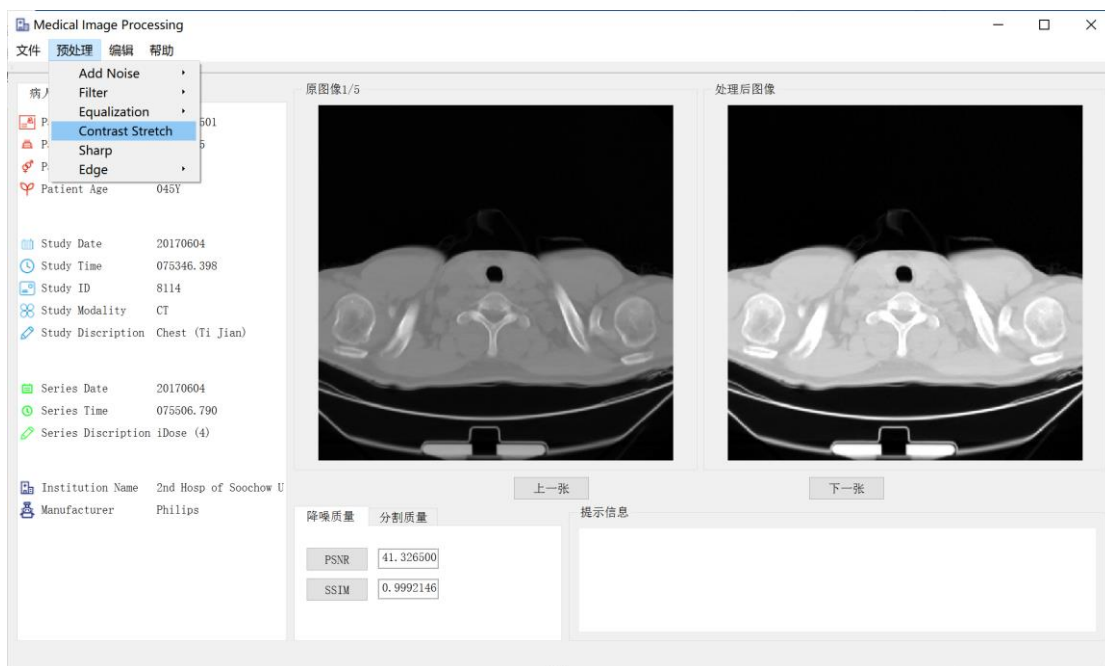
其中， $f(x,y)$ 是原始图像， $g(x,y)$ 是锐化后的图像， $h(x,y)$ 是平滑后的图像， k 是增益因子。



➤ 对比度拉伸：通过将图像中的像素值映射到一个更广泛的范围内，以增强图像的对比度。对比度拉伸的公式为：

$$g(x, y) = \frac{f(x, y) - \min}{\max - \min} \times L$$

其中， $f(x, y)$ 是原始图像， $g(x, y)$ 是拉伸后的图像， \min 和 \max 分别是原始图像中的最小值和最大值， L 是输出灰度级数。



➤ Canny 边缘检测：一种基于梯度的边缘检测算法，。它可以通过检测图像中的强度梯度来检测图像中的边缘。Canny 算子的公式为：

1. 高斯滤波

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}$$

$$f(x, y) * G(x, y)$$

其中, $f(x, y)$ 是原始图像, $G(x, y)$ 是高斯核。

2. 计算梯度幅值和方向

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} * f(x, y)$$

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} * f(x, y)$$

$$M(x, y) = \sqrt{G_x^2 + G_y^2}$$

$$\theta(x, y) = \tan^{-1} \frac{G_y}{G_x}$$

其中, $M(x, y)$ 是梯度幅值, $\theta(x, y)$ 是梯度方向。

3. 非极大值抑制

对于每个像素点, 沿着梯度方向上下两个像素点进行比较, 如果该像素点的梯度幅值不是相邻两个像素点中最大的, 则将该像素点的梯度幅值设为 0。

4. 双阈值检测

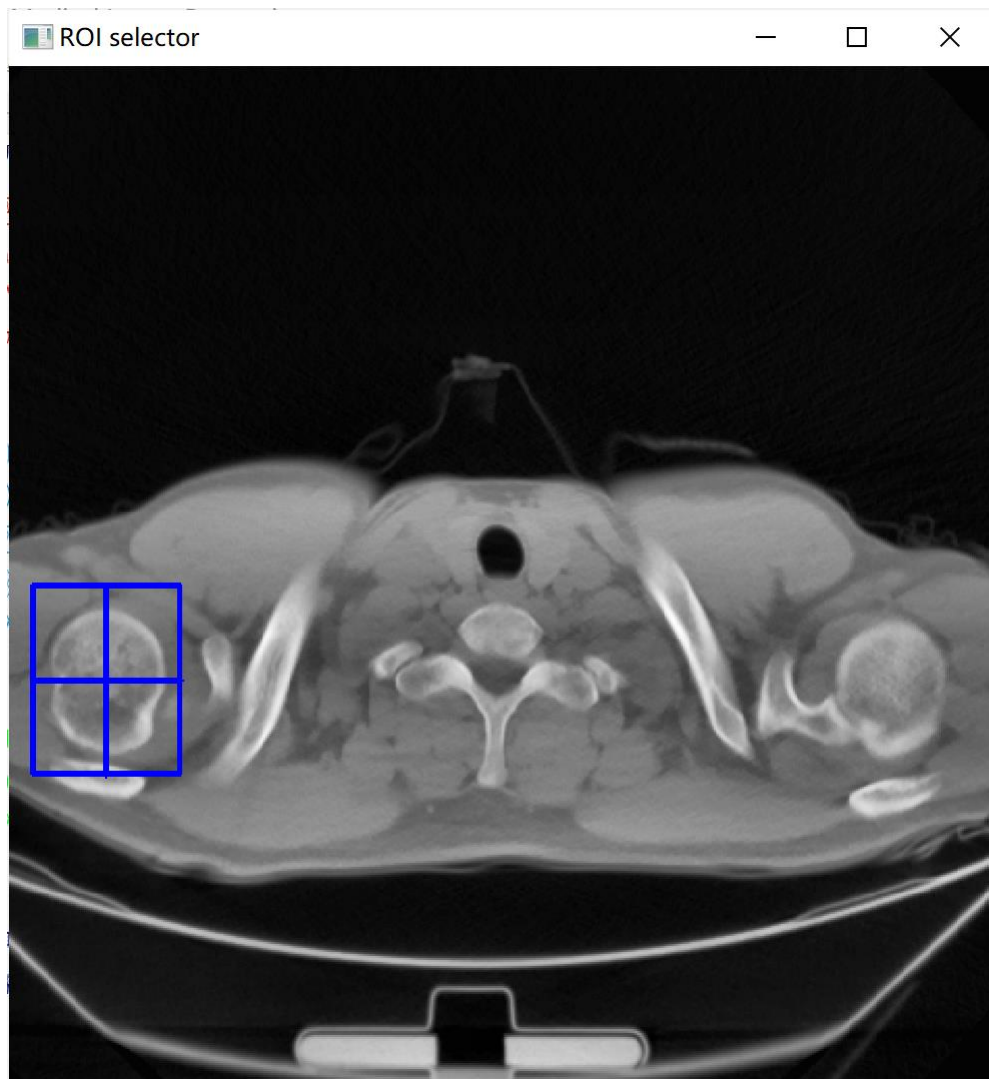
将所有非零像素点分为强边缘、弱边缘和非边缘三类。如果一个像素点的梯度幅值大于高阈值, 则将其归为强边缘; 如果一个像素点的梯度幅值小于低阈值, 则将其归为非边缘; 如果一个像素点的梯度幅值在高阈值和低阈值之间, 则将其归为弱边缘。然后, 对于所有弱边缘像素点, 如果其 8 邻域内有强边缘像素点, 则将其归为强边缘; 否则将其归为非边缘。

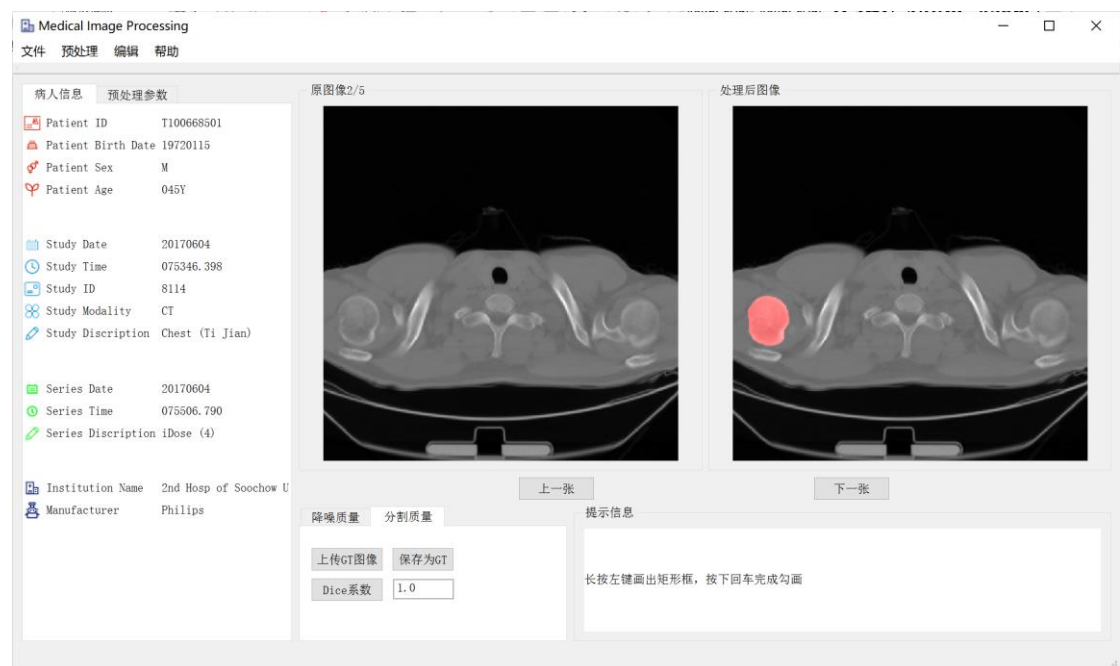


➤ GrabCut: 一种基于图割的交互式前景分割算法，生成黑白掩膜图像。

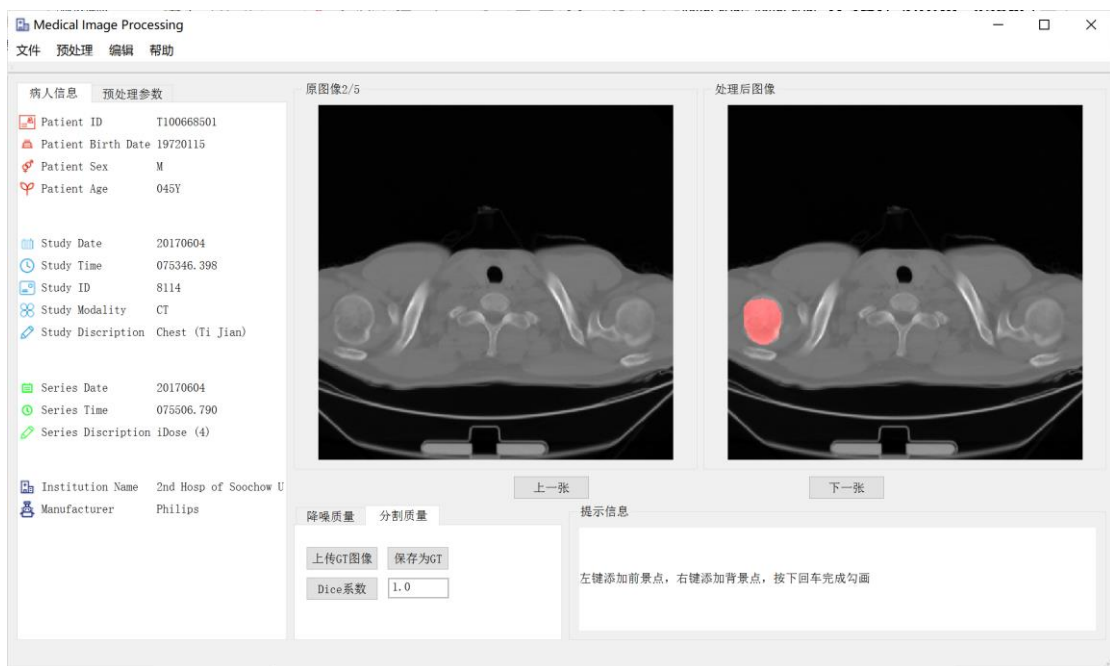
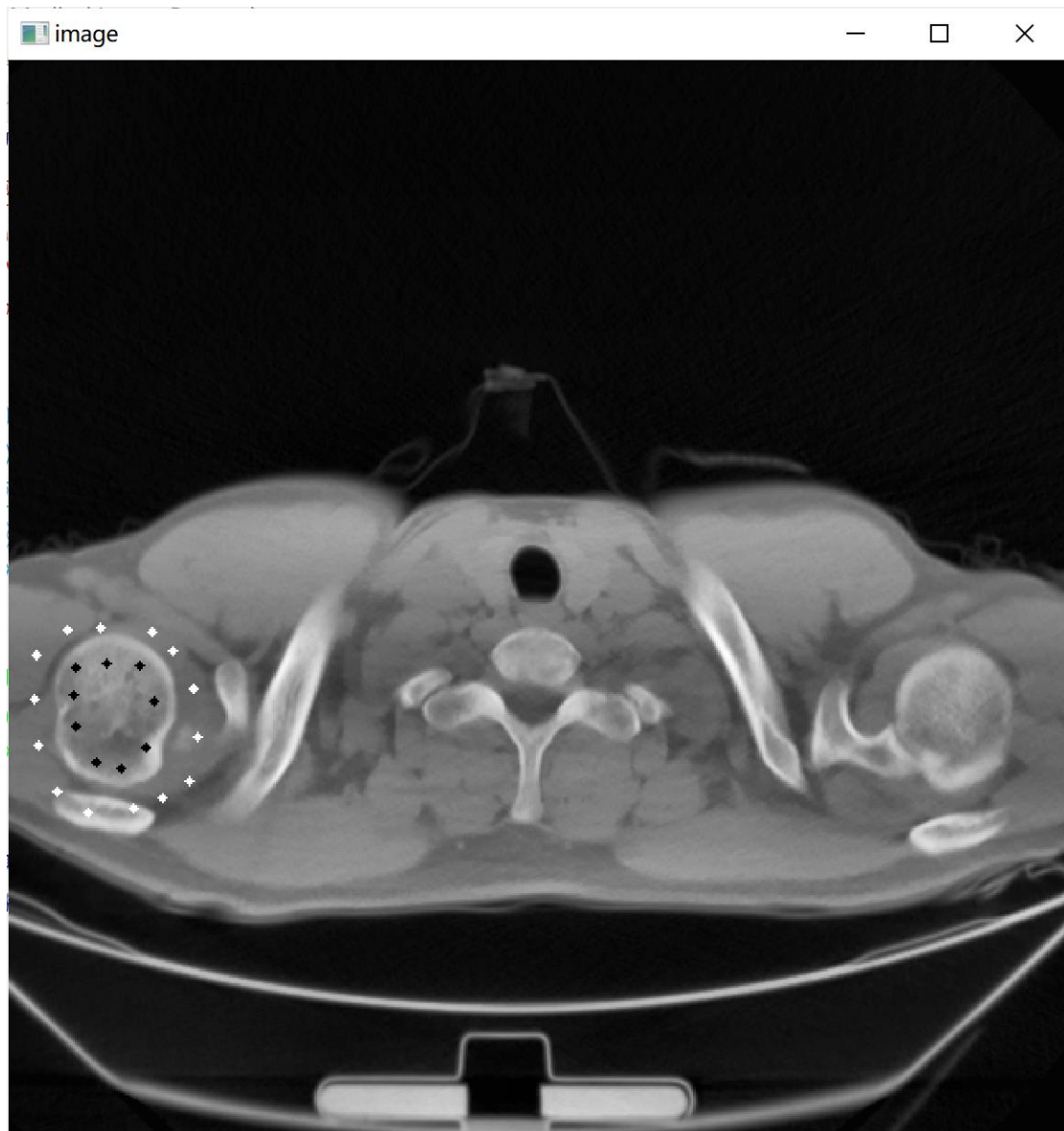
GrabCut 是一种常见的图像分割算法，它可以通过迭代优化来分割图像。将图像分为前景

和背景两部分，并将不确定区域标记为未知。对于前景和背景，分别使用高斯混合模型（GMM）来建模。GMM 模型包括高斯分布的均值、协方差矩阵和权重。迭代地更新 GMM 模型和标记，直到收敛。在每次迭代中，首先使用当前 GMM 模型来计算每个像素点属于前景或背景的概率，并根据概率更新标记。然后，使用当前标记来更新 GMM 模型。



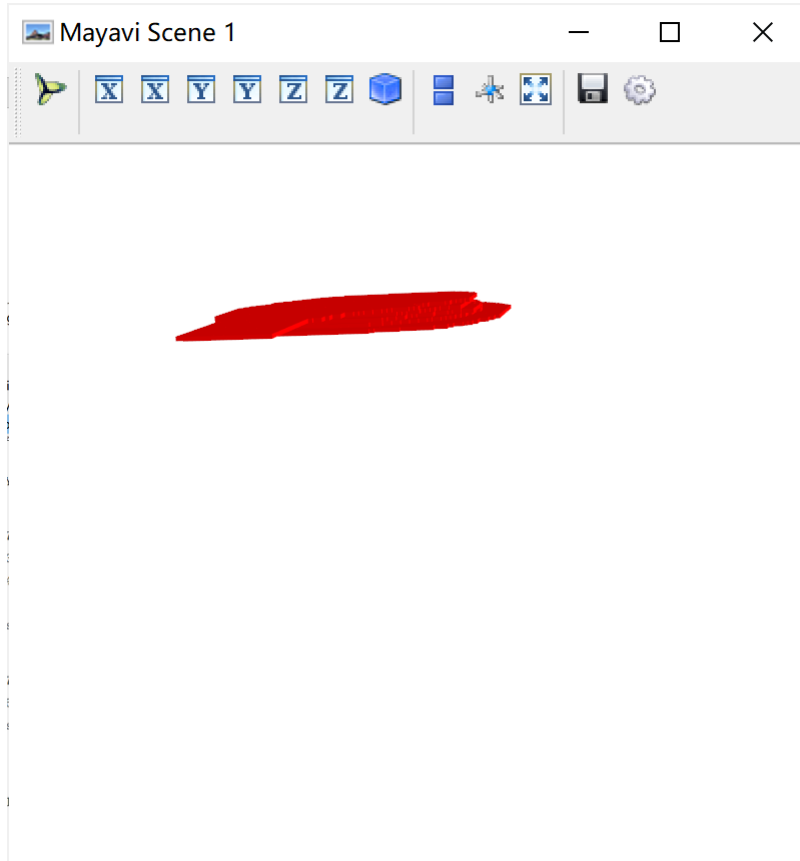


➤ Random Walk: 一种基于随机游走的图像分割算法，可以在一张图像中自动或交互式地将像素分成多个类别。Random Walk 将图像分为前景和背景两部分，并将不确定区域标记为未知。将图像转换为图，其中每个像素点对应一个节点，每个节点之间的权重表示它们之间的相似度。对于每个未知节点，计算其到前景和背景节点的最短路径，并根据路径长度来更新标记。



- **三维重建**：一种通过多个二维影像或点云数据，计算出物体表面几何形状和

结构的技术。



➤ SSIM: SSIM 是一种常见的图像质量评价指标，它可以通过比较两幅图像的结构、亮度和对比度来评价它们的相似度。SSIM 的公式为：

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

其中， x 和 y 分别表示两幅图像， μ_x 和 μ_y 分别表示它们的均值， σ_x 和 σ_y 分别表示它们的标准差， σ_{xy} 表示它们的协方差， c_1 和 c_2 是常数，用于避免分母为 0。

➤ PSNR: PSNR 是一种常见的图像质量评价指标，它可以通过比较两幅图像的峰值信噪比来评价它们的相似度。PSNR 的公式为：

$$PSNR = 10 \log_{10} \frac{MAX_I^2}{MSE}$$

其中， MAX_I 是像素值的最大值，MSE 是均方误差，定义为：

$$MSE = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [I(i, j) - K(i, j)]^2$$

其中， I 和 K 分别表示两幅图像， M 和 N 分别表示它们的宽度和高度。

第四章 测试报告

由于本软件使用 Python 开发，经测试在时间表现上存在一定不足。用“打开 dicom 文件夹”操作打开一个包含 661 张 dicom 文件（平均每张 531KB）的文件夹需要约 15s，打开 100 张以下的文件夹所需时间几乎可以忽略不计。

对于大部分图像处理操作，如加噪、滤波、增强等，运行所需时间小于 0.1s。

其中较慢的图像处理操作有交互式分割：GrabCut 时间复杂度为 $O(n^2)$ ，经过测试处理一张轮廓较清晰的图片需要 1~5s，若由于图像预处理不当等原因无法找到目标则平均在 3s 左右返回全黑图片；RandomWalk 时间复杂度为 $O(n^3)$ ，它需要对整张图像进行最小割分割，而种子点的数量越多，需要计算的最小割次数就越多，因此算法的运行时间也就越长，20 个前景点和 20 个背景点大约需要 10s 的等待时间。

三维重建操作涉及到多个二维图像的堆叠，通常需要大量的计算和内存，因此在处理大型数据时，可能会遇到内存不足或计算时间过长的问题。

本软件的鲁棒性较好，采用了 Python 的 traceback 库捕捉所有报错信息并在右下角“提示信息”框中输出，不会出现程序闪退崩溃等问题，方便维护。

本软件完全本地使用，无需联网，安全性有保障。适用 Windows 操作系统，在 Windows10 和 Windows11 均可正常运行。

附测试系统配置：

- CPU：第 11 代智能英特尔® 酷睿™ i5-11300H 处理器
- GPU：集成显卡 Intel® Iris® Xe Graphics
- 内存大小：16 GB LPDDR4x 双通道

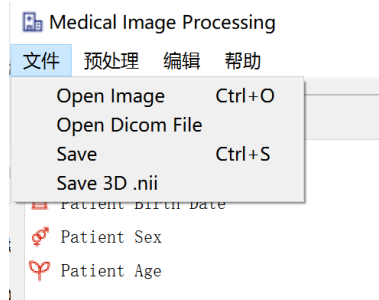
第五章 安装及使用

安装环境要求：Windows 操作系统均可

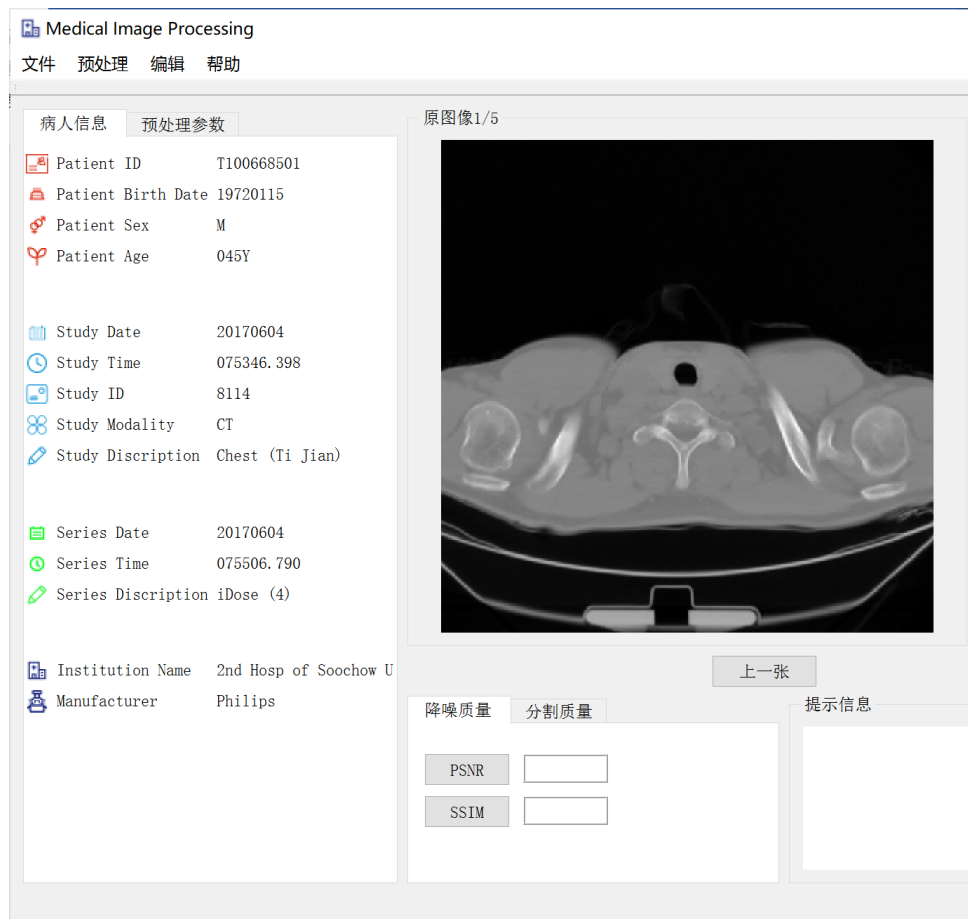
安装过程：解压缩 MediLab.zip 文件

使用流程：

1. 打开左上角菜单栏中的“文件”，选择“Open Image”（打开单张图片）或“Open Dicom File”（打开文件夹下的所有 dicom 文件）。



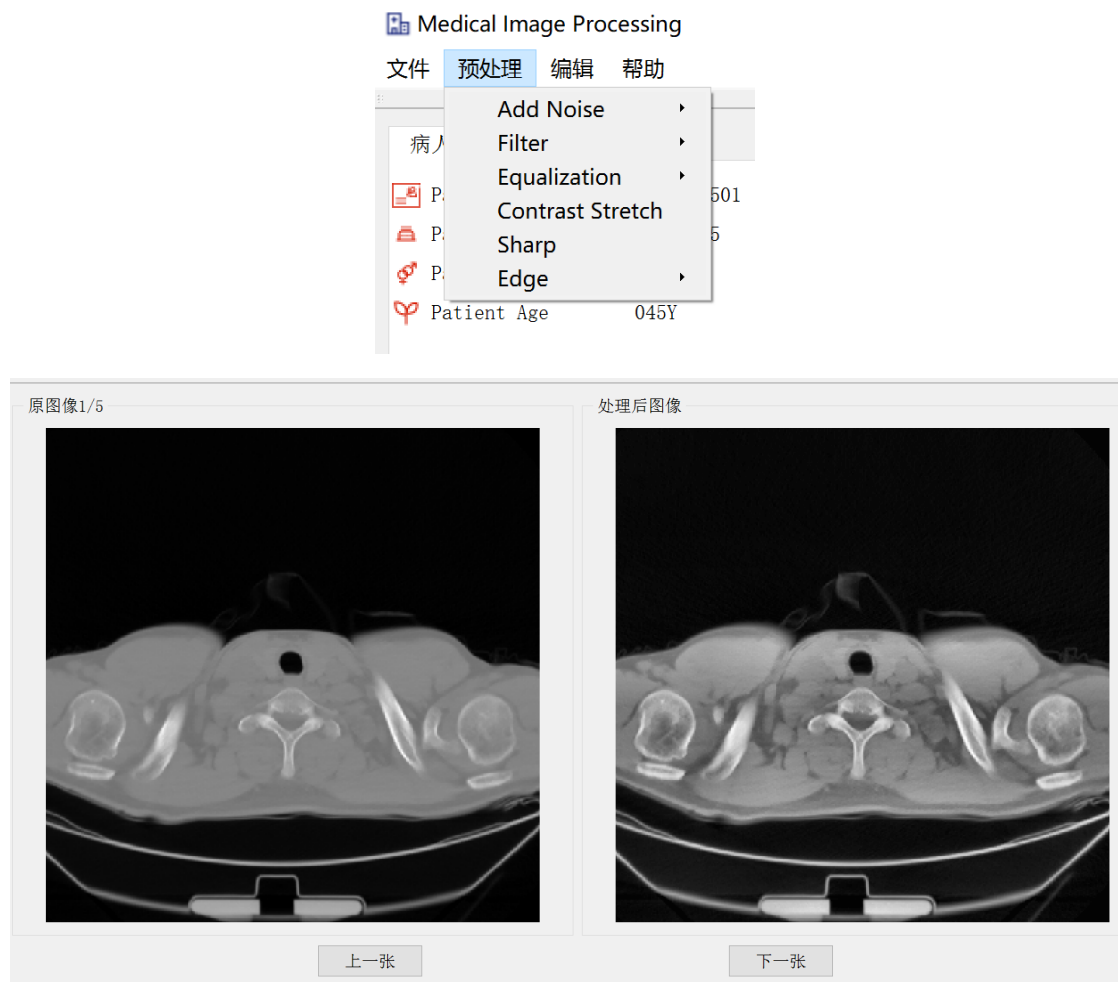
2. 查看图像，在左边“病人信息”栏读取 dicom 文件的信息。



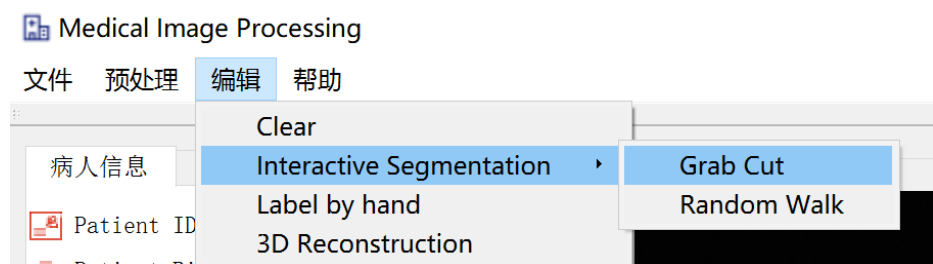
3. 调整预处理参数

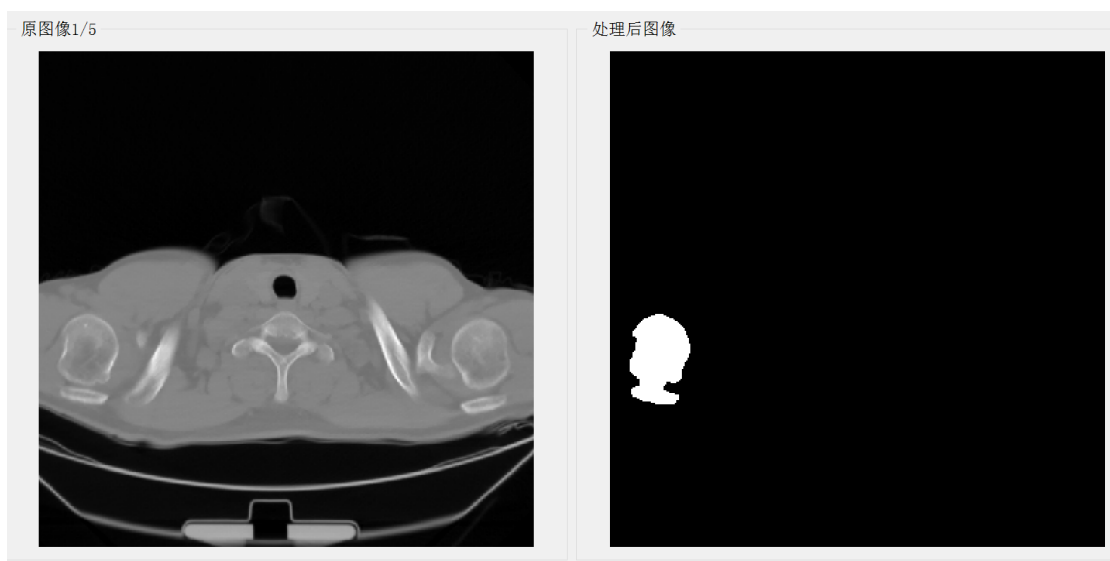
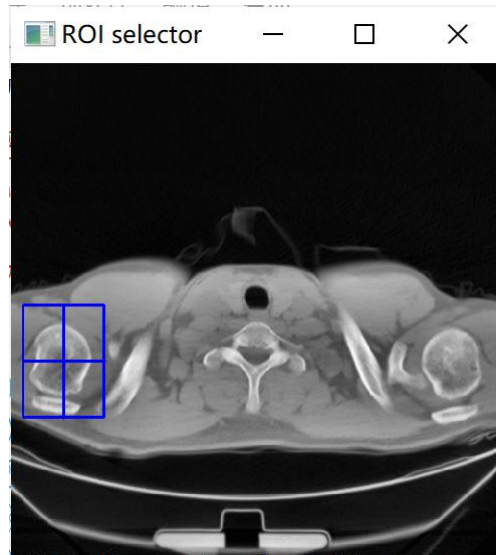


4. 在菜单栏“预处理”选择进行的预处理操作，以自适应直方图均衡化为例。

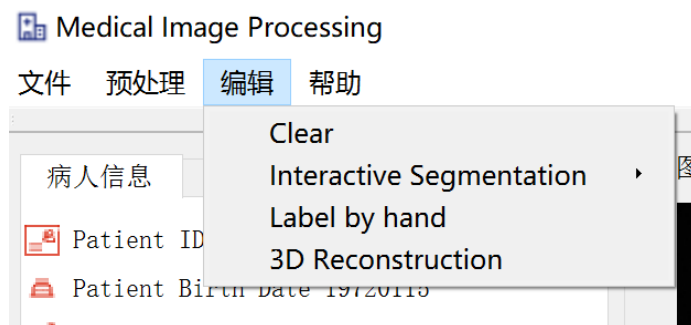


5. 在菜单栏中的“编辑”进行 roi 提取，以 GrabCut 为例，长按左键拖动矩形框，按下回车完成提取，生成黑白掩膜图像。

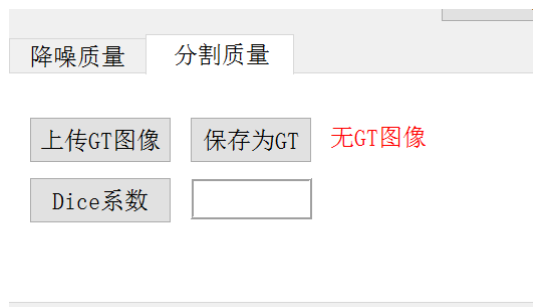




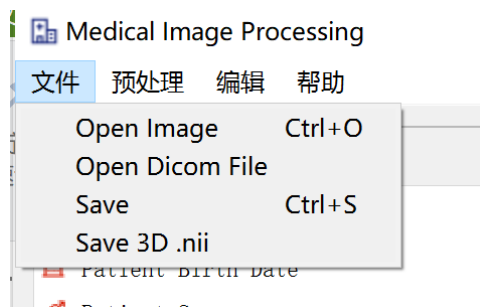
6. 对多个图像进行 roi 提取后三维重建。



7. 图像质量评估：上传 GT(groundtruth) 图像或把当前处理图片保存为 GT 图像，即可计算 Dice 系数，计算对象为处理后图片和已保存的 GT 图像。



8. 保存图片/3d 模型。



第六章 项目总结

在项目开始之前，我们有较好的规划，先手动绘制草稿，确定好项目的整体方向和目标，然后将其分解成可管理的小任务。每个小任务都有明确的目标和时间表，并且需要分配给合适的人员来完成。在整个项目周期中，彭涛老师定期评估进展情况，并及时做出调整。

在软件开发过程中，遇到了许多挑战，其中包括对医学图像处理技术的深入理解，PyQt 的学习和使用，用户界面的设计等等。我们通过积极沟通，学习和尝试，克服了这些挑战，并取得了项目的成功。

通过这个项目，我们的技能得到了提升。我们学会了使用 Python 和 PyQt 库，深入了解了医学图像处理的相关算法和技术，并掌握了软件开发过程的实际操作经验。我们通过不断学习和实践，不断提高自己的技能水平。

作为一款医学图像处理软件，我们计划在未来的版本中不断升级和改进，以更好地满足用户的需求。我们计划加入更多的算法和技术，提供更多的功能，增加用户友好性，提高软件的性能和稳定性。

在软件开发完成后，我们需要寻找更多的用户和市场。我们计划通过宣传和推广，扩大软件的知名度，吸引更多的用户。我们还计划与医疗机构和医生合作，以更好地推广软

件，并收集用户反馈以进一步改进软件。

总之，这个项目是一个充满挑战和机遇的过程。我们学会了协调和分解任务，克服了各种困难，提升了技能水平，并制定了未来的发展计划。我们相信这个软件将会有很好的前景，并帮助更多的医生和患者。

参考文献

1. [scikit-image 0.20.0 docs — skimage v0.20.0 docs](#)