**桥梁裂缝图像处理软件**

**设计说明书**

目录

[一、 简介 3](#_Toc490781224)

[1.1 目的 3](#_Toc490781225)

[1.2 背景 3](#_Toc490781226)

[1.2.1软件名称 3](#_Toc490781227)

[1.2.2 开发原因 3](#_Toc490781228)

[1.2.3 软件功能 4](#_Toc490781229)

[1.2.4 适用人群 4](#_Toc490781230)

[**二、 总体设计** 5](#_Toc490781231)

[2.1 系统整体功能模块 5](#_Toc490781232)

[2.2 总体结构 5](#_Toc490781233)

[2.3 分块描述 6](#_Toc490781234)

[**三、 详细设计** 8](#_Toc490781235)

[3.1 图像显示模块 8](#_Toc490781236)

[3.1.1 用户管理（功能点） 8](#_Toc490781237)

[3.1.2 依赖性描述 8](#_Toc490781238)

[3.2边缘检测模块 9](#_Toc490781239)

[3.2.1 用户管理（功能点） 9](#_Toc490781240)

[3.2.2 依赖性描述 9](#_Toc490781241)

[3.3二值化处理模块 10](#_Toc490781242)

[3.3.1 用户管理（功能点） 10](#_Toc490781243)

[3.3.2 依赖性描述 10](#_Toc490781244)

[3.4角点检测模块 11](#_Toc490781245)

[3.4.1 用户管理（功能点） 11](#_Toc490781246)

[3.4.2 依赖性描述 11](#_Toc490781247)

[四、运行设计 12](#_Toc490781248)

[4.1 设计概论 12](#_Toc490781249)

[4.1.1 交互关系 12](#_Toc490781250)

[4.1.2 实现功能 12](#_Toc490781251)

[4.2 详细设计 13](#_Toc490781252)

[4.2.1 实现步骤 13](#_Toc490781253)

[**五、结语** 15](#_Toc490781254)

[5.1 后期开发方向 15](#_Toc490781255)

[5.2 设计前景 15](#_Toc490781256)

一、 简介

## 1.1 目的

本软件设计说明书详细介绍了《桥梁裂缝图像处理软件》的总体结构、系统实现功能及各模块功能的设计理念，在设计方面，本软件集中考虑了学生学习，老师教学以及科研人员在机器学习学习、教授、科研等过程中对桥梁裂缝图像识别时出现“漏看”、“错看”等失误的问题。增强了桥梁裂缝图像中裂缝的特征，提高用户在桥梁裂缝图片中识别裂缝的准确度。同时也通过基于python 的Tkinter库，实现了软件界面图像显示功能；通过基于python的os库，实现文件夹中文件的打开及保存；通过基于opencv的角点检测和边缘检测方法，实现对桥梁裂缝图像的处理。进而提高桥梁裂缝图像中裂缝标注的准确度。在功能变得更加强大的同时，也是为了更好地解决上述问题。

方案重点是功能模块详细设计的描述和模块的执行流程。

## 1.2 背景

### 1.2.1软件名称

桥梁裂缝图像处理软件V1.0

### 1.2.2 开发原因

图像处理是近几年来机器语言深度学习的热潮，在计算机视觉化和机器学习中，桥梁裂缝识别都是一个很重要的方向。尤其是当桥梁裂缝识别软件需要进行市场投放前的测试时，更需要海量数据的处理。对于图像中信息的提取与识别，在进行深度算法研究之前，应首先进行数据提取。在本软件中，我们可以帮助人们更加快捷的进行桥梁裂缝图像的处理，及时增强桥梁裂缝的特点，方便后续的桥梁裂缝图像标注与识别。大大减少人们因为无法准确识别桥梁裂缝而带来的负担。我们也相信这个软件会填补当前市场桥梁裂缝图像处理软件的空白，成为人们进行桥梁裂缝图像处理时必不可少的软件。

### 1.2.3 软件功能

系统主要功能包括：

1、图像显示（单张图片显示，文件夹显示）；

2、边缘检测（canny算子、sobel算子、laplacian算子）；

3、 二值化处理（Gray Image、BINARY、BINARY\_INV、TRUNC、TOZERO、TOZERO\_INV）；

4、角点检测（Harris、Shi-Tomasi、Brisk）；

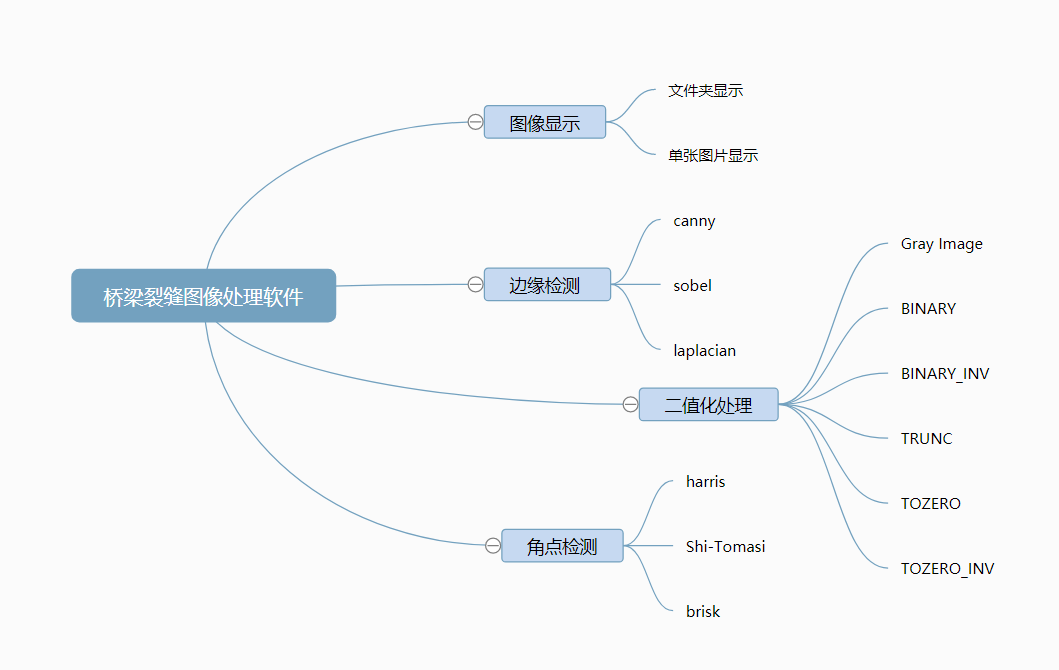
5、日志管理（记录操作情况、记录操作时间）；

### 1.2.4 适用人群

本系统适用正在进行桥梁裂缝图像处理的学生，以及做此方面科研工作的研究员、老师等。

二、 总体设计

## 2.1 系统整体功能模块



桥梁裂缝图像处理软件的整体模块框架如上图所示。每个模块根据系统所需功能编写，逐步实现图像显示、边缘检测、二值化处理、角点检测的功能。各个模块之间以图像流、信息流为基础相互联系，以实现软件的整体功能，又保持相对独立性，以方便系统维护和功能扩展。

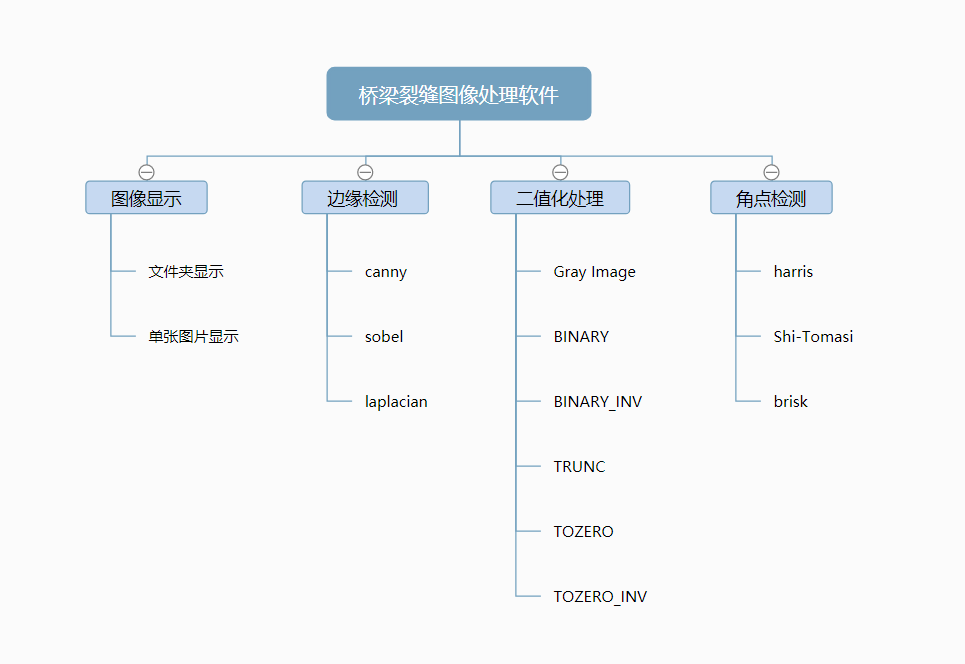
## 2.2 总体结构

**模块划分**

根据功能将程序划分为四个主要模块：

图像显示模块、边缘检测模块、二值化处理模块、角点检测模块。

各模块之间的组织结构如图所示：



## 2.3 分块描述

**各模块描述**

1. 图像显示

为实现用户对各种图像查看的需求，本模块提供了查看单张图片和图片文件夹的功能，可满足用户不同的查看方式。读取目录下的图片集，并将每张图片依次完整地显示在软件的GUI中，且提供按钮以实现图片间跳转的功能。

1. 边缘检测

该模块依托于opencv中的边缘检测功能，提供了canny算子、sobel算子、laplacian算子，为用户提供了多种边缘检测方法；

1. 二值化处理

该模块依托于opencv中的二值化处理功能，提供了Gray Image、BINARY、BINARY\_INV、TRUNC、TOZERO、TOZERO\_INV多种二值化处理的方法。

1. 角点检测

该模块依托于opencv中的角点检测功能，提供了Harris、Shi-Tomasi、Brisk多种角点检测方法。

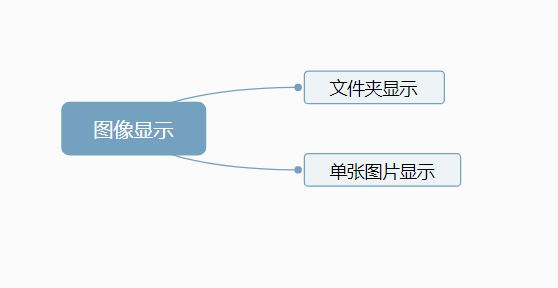
1. 日志显示

该模块提供了操作记录的显示以及操作时间的显示。

三、 详细设计

## 3.1 图像显示模块

### 3.1.1 用户管理（功能点）



图像显示模块包含文件夹显示及单张图片显示两部分。

该模块依托于PIL库中的图像处理功能，读取目录下的图片集，并将每张图片依次完整地显示在软件的GUI中，且提供按钮以实现图片间跳转的功能。

文件夹显示部分，支持用户直接打开根目录下Image文件夹并显示其中的图片。

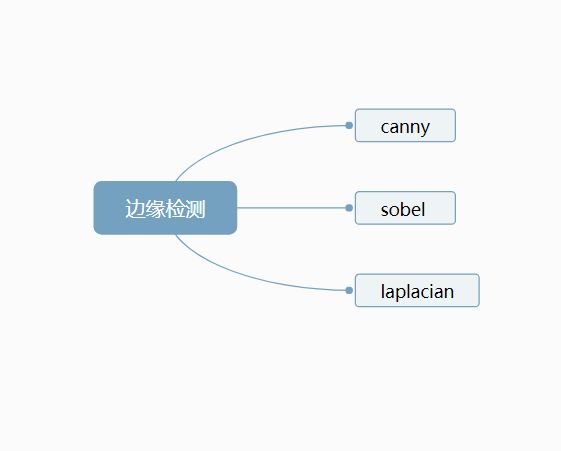
单张图片显示部分，支持用户在主界面单张查看图片，可选择上一张、下一张，或是跳转到某一张。

### 3.1.2 依赖性描述

基于python的tKinter库实现GUI界面，PIL库实现图像处理。

## 3.2边缘检测模块

### 3.2.1 用户管理（功能点）



边缘检测包括canny、sobel、laplacian三个功能。

Canny边缘检测算子使用4个mask检测水平、垂直以及对角线方向的边缘。原始图像与每个mask所作的卷积都存储起来。对于每个点都标识在这个点上的最大值以及生成的边缘的方向。这样就从原始图像生成了图像中每个点亮度梯度图以及亮度梯度的方向。

Sobel算子对于像素的位置的影响做了加权，可以降低边缘模糊程度，因此效果更好。Sobel算子是滤波算子的形式，用于提取边缘，可以利用快速卷积函数，简单有效，应用广泛。

Laplacian算子是n维欧几里得孔吉安 中的一个二阶微分算子，定义为梯度grad（）的散度div()。它具有各向同性，即与坐标轴方向无关。

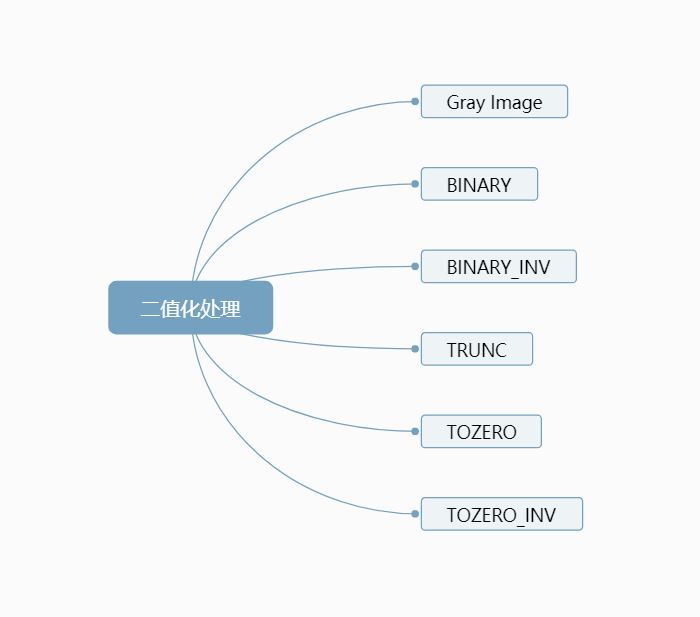
用户可以根据当前桥梁裂缝图像的特点，选择不同的边缘检测算子对裂缝图像进行处理。

### 3.2.2 依赖性描述

基于opencv的边缘检测算法。

## 3.3二值化处理模块

### 3.3.1 用户管理（功能点）



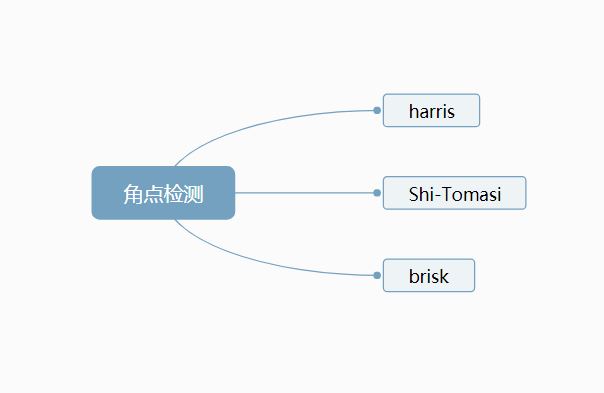
桥梁裂缝图片在处理之前需要进行预处理，也就是图像的二值化处理。根据这一特点，设计了二值化处理模块，方便用户处理图片。Opencv为我们提供了多种二值化处理的方法，例如Gray Image、BINARY、BINARY\_INV、TRUNC、TOZEO、TOZER\_INV用户可以根据需要选择对图片进行的二值化处理。

### 3.3.2 依赖性描述

基于opencv的图像二值化处理功能。

## 3.4角点检测模块

### 3.4.1 用户管理（功能点）



角点检测模块提供了Harris、Shi-Tomasi、Brisk三种角点检测方法。

Harris角点检测基于图像灰度的方法通过计算点的曲率及梯度来检测角点，Harris角点检测算子对亮度和对比度的变化不敏感，因为使用的微分算子对图像密度的拉升或收缩和对亮度的抬高或下降不敏感。

Shi-Tomasi算法主要是为了解决跟踪问题，用来衡量两幅图片的相似度，也是Harris算法的改进。

Brisk算法主要利用FAST9-16进行特征点检测，具有较好的旋转不变性、尺度不变形、较好的鲁棒性。

### 3.4.2 依赖性描述

基于opencv的角点检测功能。

四、运行设计

## 4.1 设计概论

### 4.1.1 交互关系

本地硬盘

存储图像

TkGUI界面

图像处理

加载图像

图像集

文件夹

系统流程信息交互如上图所示。通过根目录下的图像记，通过tKinter库绘制的GUI界面显示原始图片。也可以打开图片文件夹，加载出图片，之后进行诸如边缘检测、二值化处理、角点检测等图像处理方法，再选择存储图像。

### 4.1.2 实现功能

图像获取：

1. 图像集读取（loadDir）
2. 单个图像读取（loadImage）
3. 图像选择（setImage）

边缘检测：

1. Canny（canny）
2. Sobel（sobel）
3. Laplacian（laplacian）

二值化处理：

1. Gray Image（GrayImage）
2. BINARY（BINARY）
3. BINARY\_INV（BINARY\_INV）
4. TRUNC（TRUNC）
5. TOZERO（TOZERO）
6. TOZER\_INV（TOZERO\_INV）

## 4.2 详细设计

### 4.2.1 实现步骤

**环境配置：**

配置opencv

将opencv中的C:\opencv\build\python\2.7\x64下的cv2.pyd移动到python安装目录下的Lib\site-packages

安装pip

下载完成后解压

安装相关模块

Pip install numpy

配置python2.7编译环境  
把python2.7添加到环境变量path中  
1、变量名：path  
 变量值：C:\Python27;   
 C:\Python27\DLLs;   
 C:\Python27\Scripts;  
  
2、变量名：PYTHON\_INCLUDE   
 变量值：C:\Python27\include  
  
3、变量名：PYTHON\_LIB  
 变量值：C:\Python27\libs\python27.lib

配置Tkinter库

1. 下载和安装Tkinter库
2. 在项目中引入Tkinter库

配置PIL库

1. 下载和安装PIL库
2. 在项目中引入PIL库

**功能实现：**

1. 图像显示模块，通过Tkinter库实现GUI界面，借助其提供的事件响应机制实现布局与功能的连接,显示文件夹中的图片。
2. 图像处理模块，通过opencv自带的边缘检测，二值化处理，角点检测方法，对加载到软件中的图片进行处理，支持新建文件夹或选择任何硬盘中的文件夹进行图片的存储。

五、结语

## 5.1 后期开发方向

后期将继续优化本软件提供的图像处理的算法，改善裂缝图像的处理效果。

添加支持更多算法。

另外，对于软件界面的设计还有待进一步的提升。

## 5.2 设计前景

图像处理是近几年来机器语言深度学习的热潮，在计算机视觉化和机器学习中，桥梁裂缝识别都是一个很重要的方向。尤其是当桥梁裂缝识别软件需要进行市场投放前的测试时，更需要海量数据的处理。对于图像中信息的提取与识别，在进行深度算法研究之前，应首先进行数据提取。在本软件中，我们可以帮助人们更加快捷的进行桥梁裂缝图像的处理，及时增强桥梁裂缝的特点，方便后续的桥梁裂缝图像标注与识别。大大减少人们因为无法准确识别桥梁裂缝而带来的负担。我们也相信这个软件会填补当前市场桥梁裂缝图像处理软件的空白，成为人们进行桥梁裂缝图像处理时必不可少的软件。相信在不久的未来，对于我们产品的需求会越来越多，成为基于桥梁裂缝图像处理开发软件者必不可少的工具软件。