

Dip 软件系统功能实现及使用说明

数字图像处理课程设计

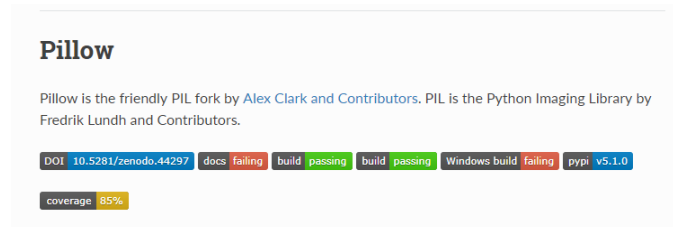
71115327 罗皓

目录

1. 技术路线	2
1.1 Pillow python 图形库	2
1.2 Scikit-image python 科学图形科学计算库	2
1.3 Electron 平台桌面应用程序.....	2
1.4 Orpc 服务器进程分布式通讯.....	3
2 功能实现	3
2.1 需求抽离	3
2.2 功能实现	4
2.2.1 图像空域增强	4
2.2.2 图像频域增强	7
2.2.3 图像变换操作	7
2.2.4 边缘检测算子	8
2.2.5 高级	16
3 界面设计	18
3.1 首页	18
3.2 菜单设计	19
3.3 功能页	20
4 总结	21
4.1 整个项目总结	21
4.2 数字图像处理相关	21
4.3 一些感想	22
5 参考	22

1. 技术路线

1.1 Pillow python 图形库



<https://github.com/python-pillow>

提供基础的数字图像处理功能。Python Imaging Library 为您的 Python 解释器添加图像处理功能。该库提供了广泛的文件格式支持，高效的内部表示和相当强大的图像处理功能。核心图像库专为快速访问以几种基本像素格式存储的数据而设计。它应该为一般的图像处理工具提供坚实的基础。

1.2 Scikit-image python 科学图形科学计算库



<http://scikit-image.org/download.html>

提供进阶的数字图像处理功能，能够进行很大程度的科学计算。skimage 包的子模块 skimage 包的全称是 scikit-image SciKit (toolkit for SciPy)，它对 scipy.ndimage 进行了扩展，提供了更多的图片处理功能。它是由 python 语言编写的，由 scipy 社区开发和维护。skimage 包由许多的子模块组成，各个子模块提供不同的功能。

1.3 Electron 平台桌面应用程序



<https://electronjs.org/>

作为前端界面开发的前端技术平台。Electron 是由 Github 开发，用 HTML，CSS 和 JavaScript 来构建跨平台桌面应用程序的一个开源库。Electron 通过将 Chromium 和 Node.js 合并到同一个运行时环境中，并将其打包为 Mac，Windows 和 Linux 系统下的应用来实现这一目的。

1.4 Orpc 服务器进程分布式通讯



<http://www.zerorpc.io/>

提供前后端通信的 RPC 模块。zerorpc 是一个轻量级，可靠且语言无关的库，用于服务器端进程之间的分布式通信。它建立在 ZeroMQ 和 MessagePack 之上。支持流式响应 - 类似于 Python 生成器 - 使 zerorpc 不仅仅是一个典型的 RPC 引擎。内置的心跳和超时检测并从失败的请求中恢复。内省功能，一流例外和命令行实用程序使调试变得简单。

2 功能实现

2.1 需求抽离

1. 图像增强 空域增强 pillow
 - a. ☒ 直方图修正(绘制，均衡化)和均衡化
 - b. ☒ 彩色增强
 - c. ☒ 平滑和锐化处理
 - d. ☒ 对比度和亮度调整
2. 图像增强 频域操作
 - a. ☒ 低通滤波
 - b. ☒ 高通滤波
 - c. ☒ 同态滤波
3. 图像变换操作
 - a. ☒ 旋转平移
 - b. ☒ 拉伸(调整尺寸)
 - c. ☒ 放大缩小
4. 边缘检测算子
 - a. ☒ Sobel 算子
 - b. ☒ Laplace 算子
 - c. ☒ Prewitt 算子
 - d. ☒ Roberts 算子

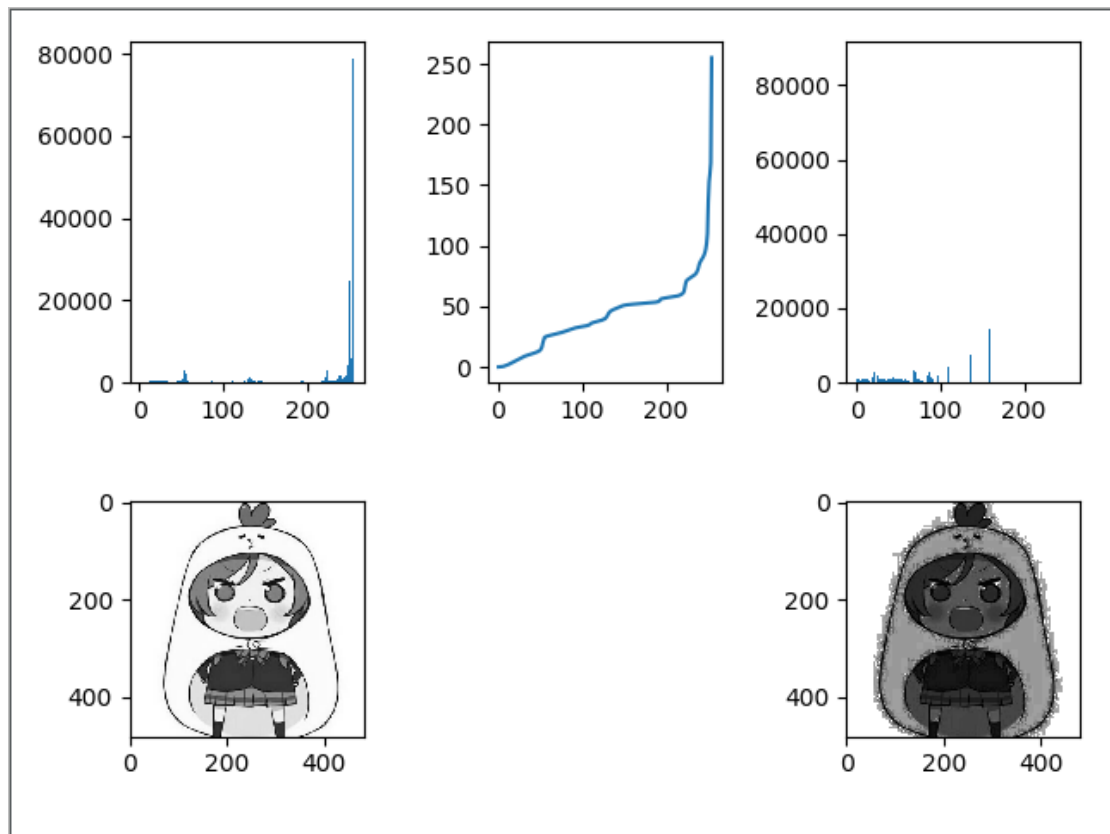
- e. ☒ Canny 算子
 - f. ☒ Gabor 算子
 - g. ☒ 水平、垂直边缘检测
 - h. ☒ 交叉边缘检测
 - i. ☒ 高斯滤波
 - j. ☒ 中值滤波
 - k. ☐ 维纳滤波
 - l. ☐ Log 算子
5. 高级
- a. ☒ 图像复原
 - b. ☒ 图像分割
 - c. ☐ 图像压缩
6. 文件操作

2.2 功能实现

2.2.1 图像空域增强

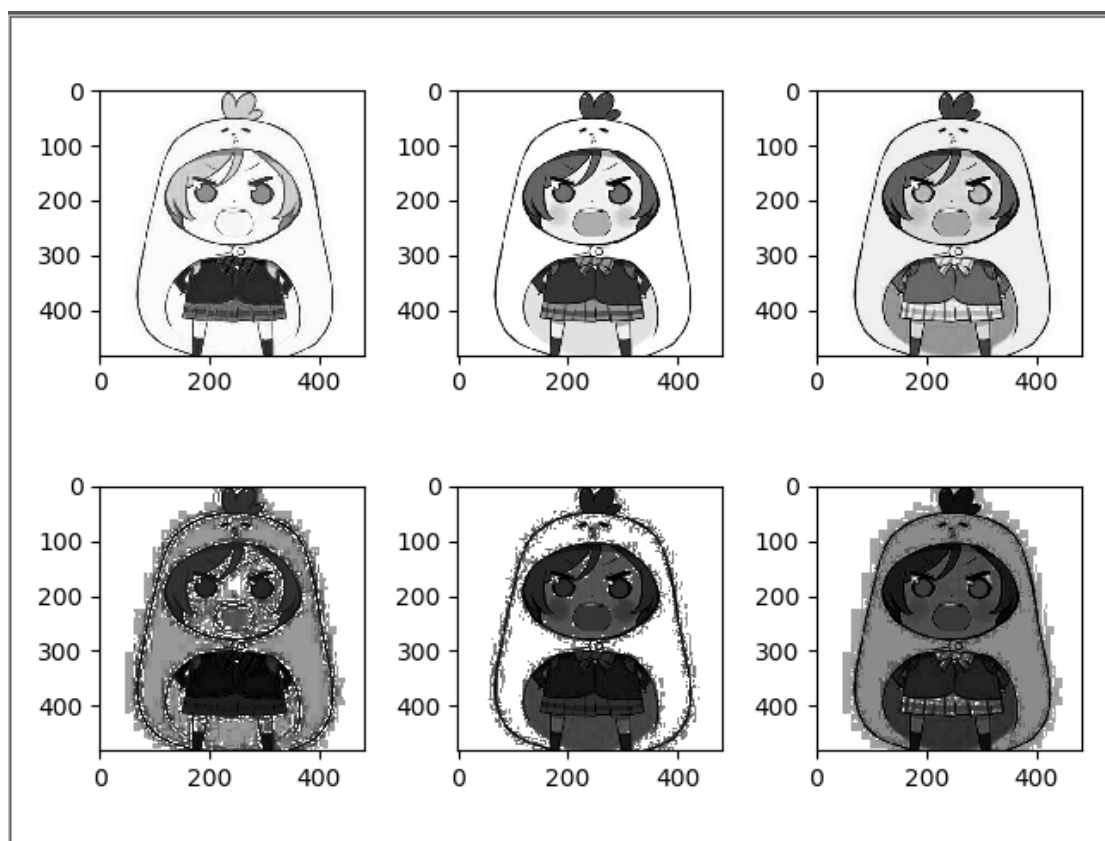
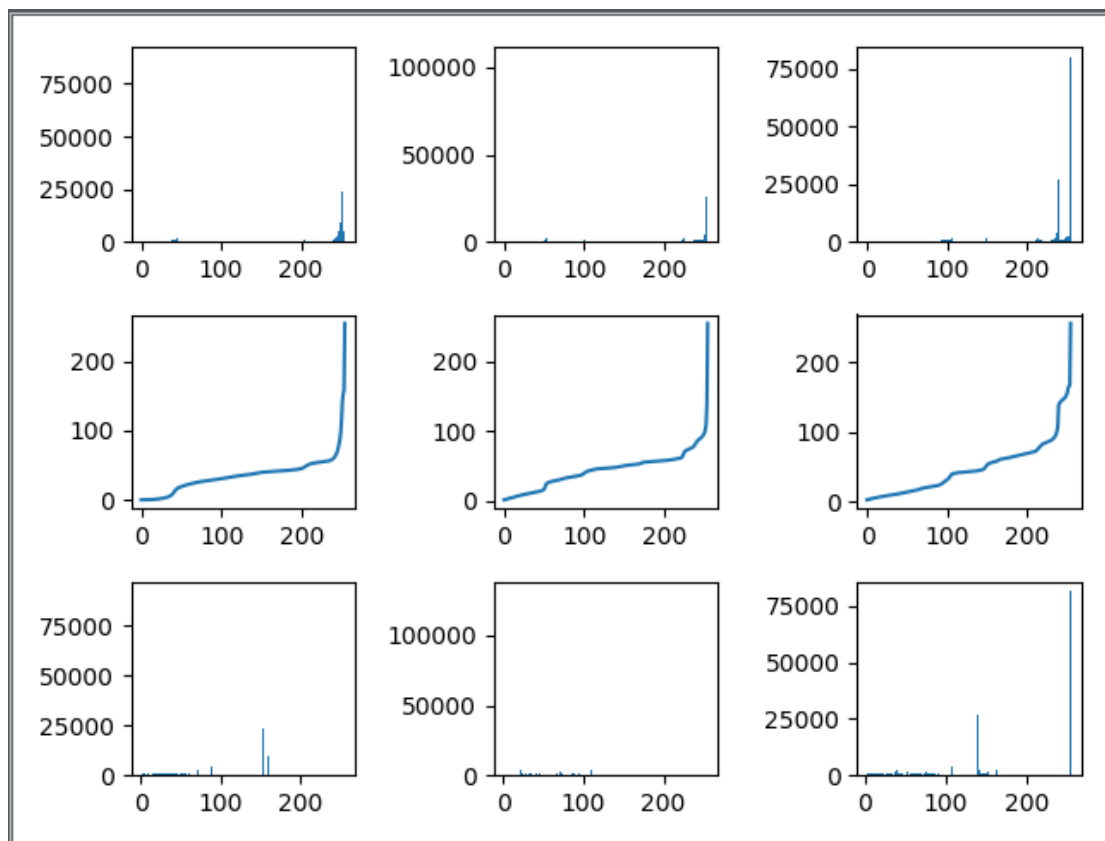
- 直方图修正和均衡化

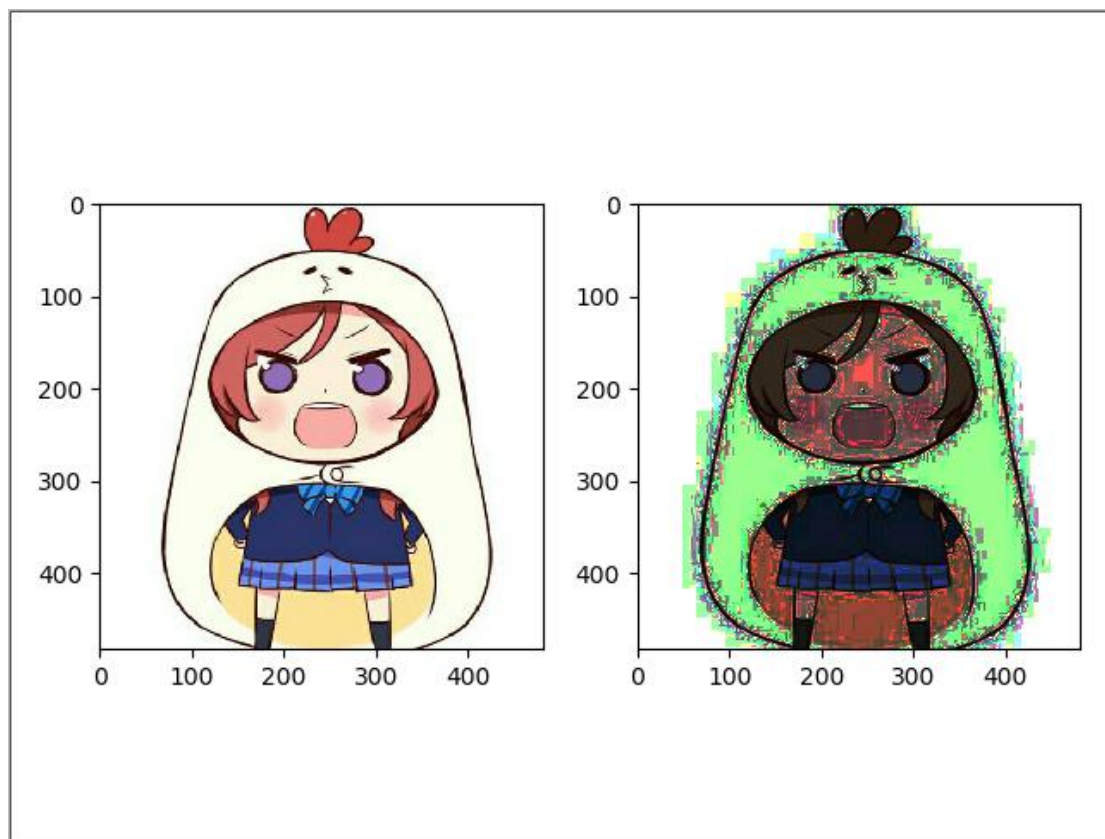
采用的是 python pillow 图像库中的 api 接口，对应代码文件为 pillow/imhist/solv1.py。
效果图如下：



- 彩色直方图均衡化

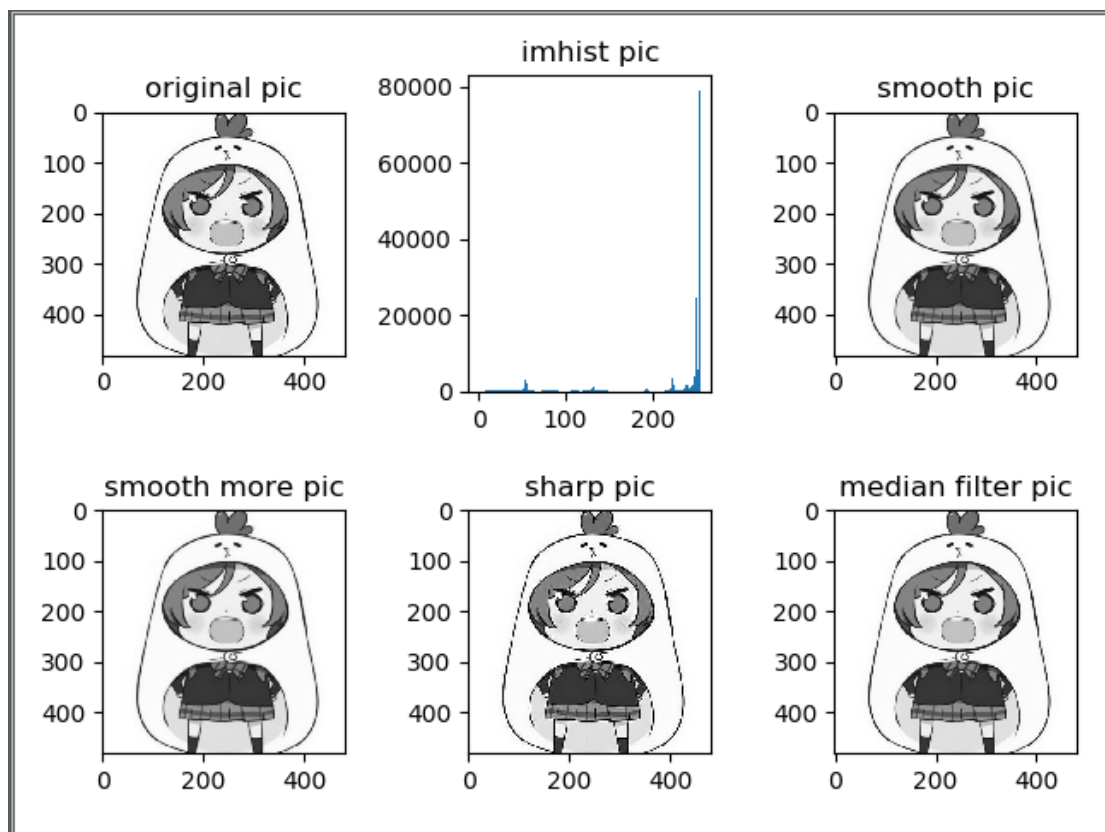
采用的是 python pillow 图像库中的 Image 类，通过 RGB 三通道对彩色图像进行直方图均衡化后合并，对应代码文件为 pillow/imhist/solv2.py。效果图如下：





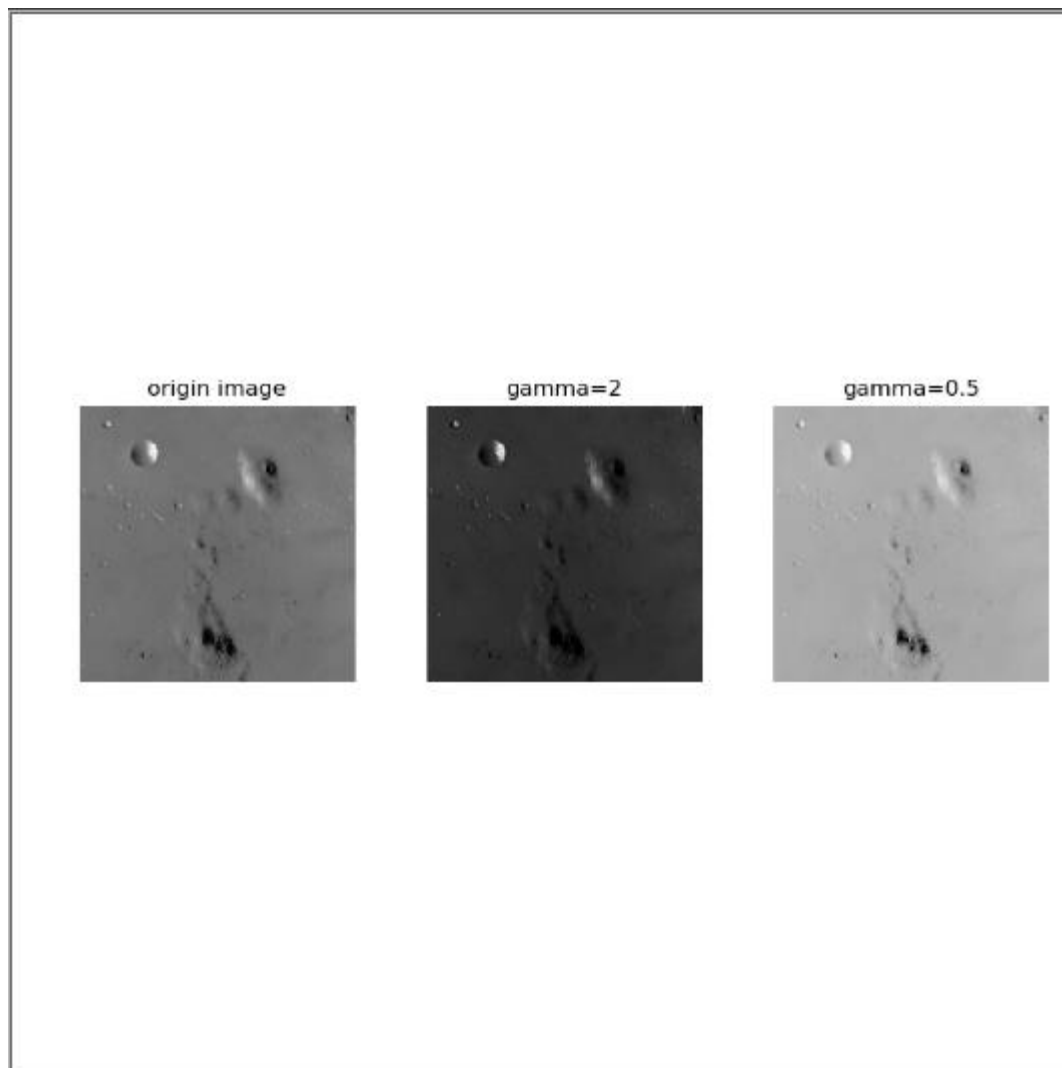
- 平滑和锐化处理

采用的是 python pillow 图像库中的 Image Filter 类，提供各种 filter 滤波器对图像进行处理其中包括平滑和锐化滤波器，对应代码文件为 pillow/filter/filterpic.py。效果图如下：



- 对比度和亮度调整

采用的是 python skimage 图像库中的 exposure 类，exposure 模块提供图像 gamma 值的调整，log 对数调整以及调整强度，对应代码文件为 scikitfunc/exposure.py。效果图如下：



2.2.2 图像频域增强

- 低通滤波，高通滤波和同态滤波

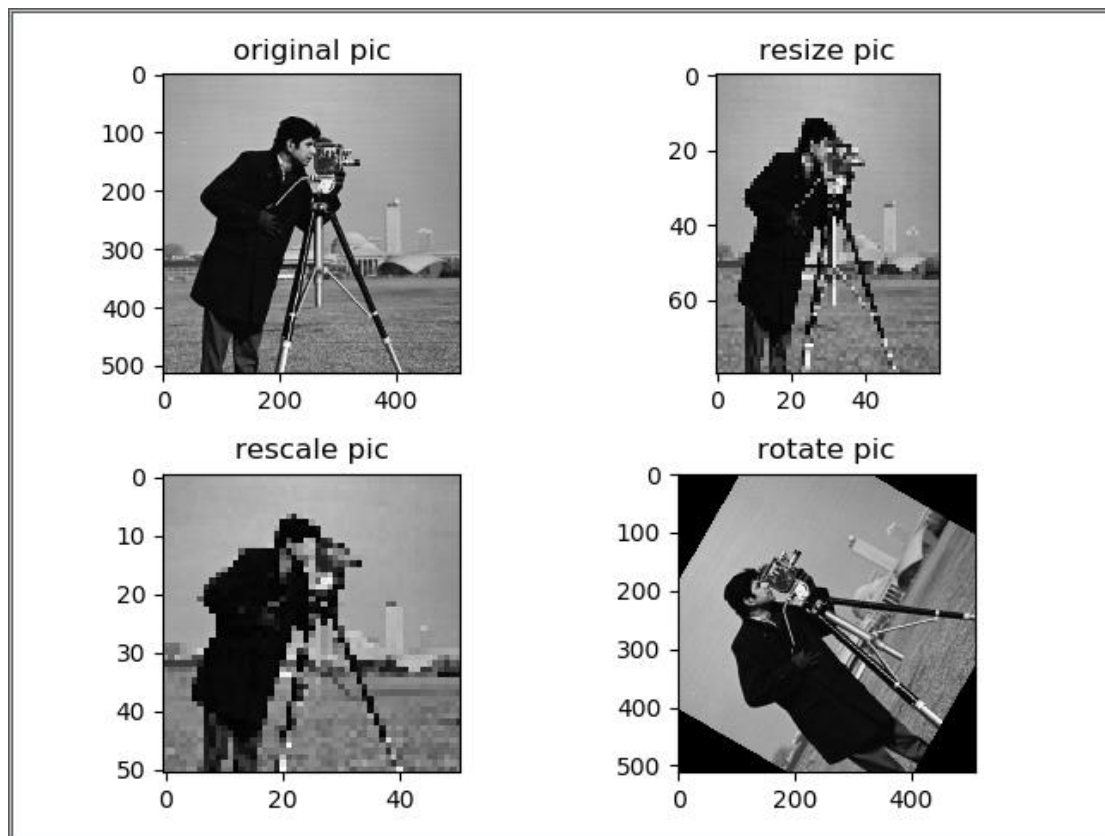
采用的是 python scipy.signal.filter_design 库中 filter_design 类，提供 butter 函数进行巴特沃斯滤波处理，Butterworth digital and analog filter design，同时进行参数选择有，{'lowpass', 'highpass', 'bandpass', 'bandstop'}，分别对应低通滤波，高通滤波和同态滤波以及截止频率滤波。

2.2.3 图像变换操作

- 旋转平移、拉伸（调整尺寸）和放大缩小

采用的是 python skimage 图像库中的 transform 类，提供各种对图像的基本处理，利用 rescale，resize 和 rotate 函数来对图像进行平移缩放和旋转的操作，对应代码文件为

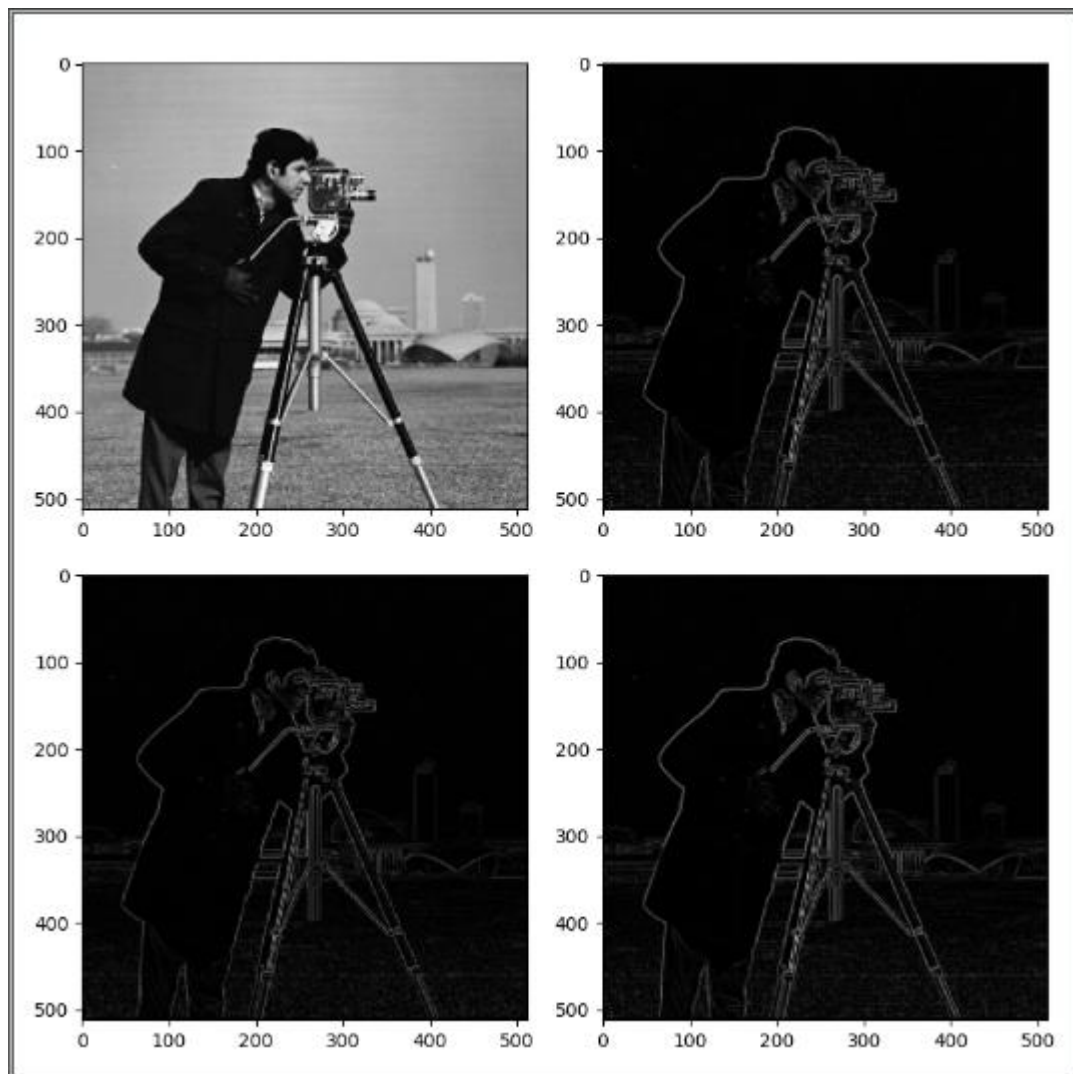
scikifunc/transform.py。效果图如下：



2.2.4 边缘检测算子

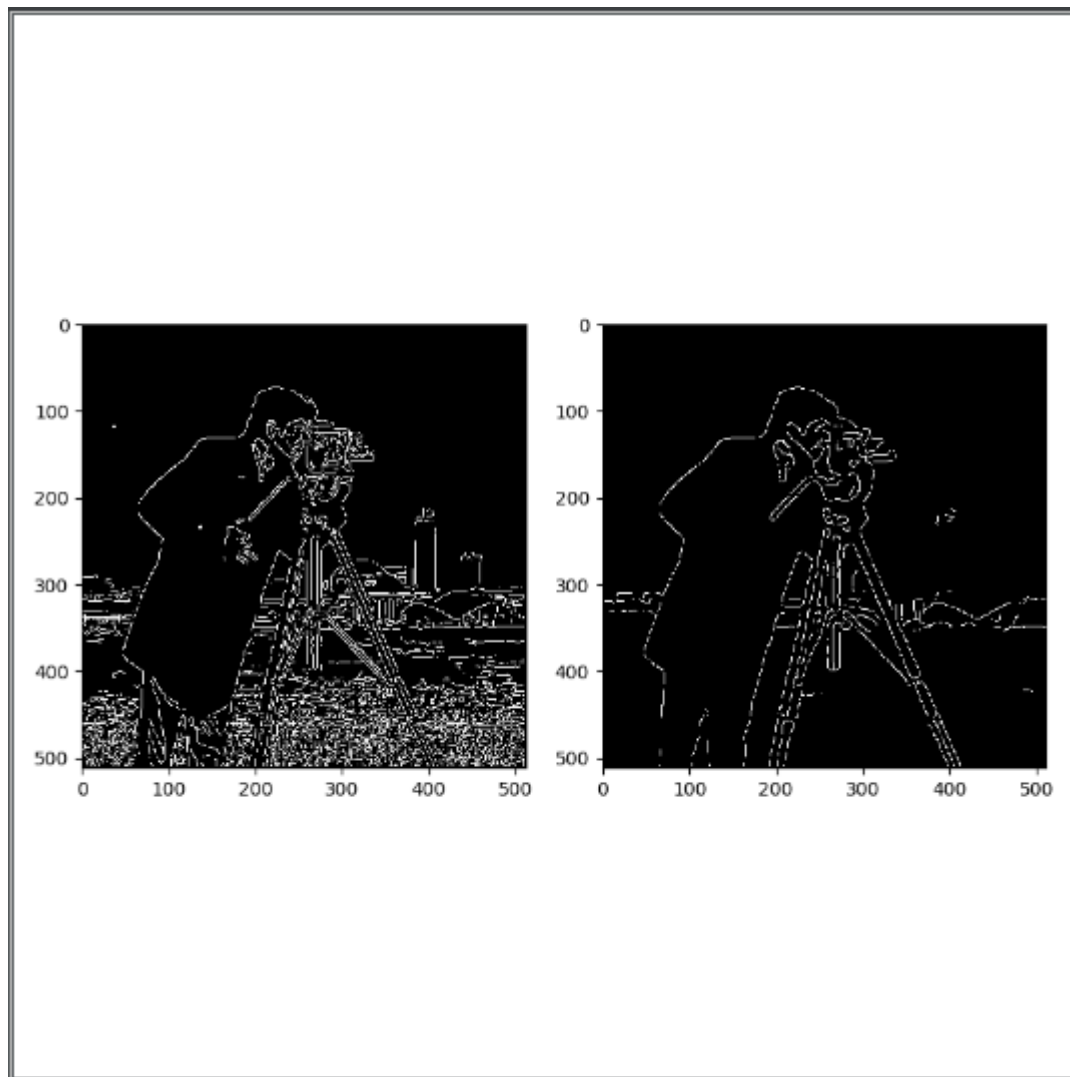
- Sobel, Laplace, Prewitt, Roberts 算子

采用的是 python skimage 图像库中的 filter 类，提供各种 filter 滤波器对图像进行处理。其中包括了各种边缘检测算子，对应代码文件为 scikifunc/filter.py。效果图如下：



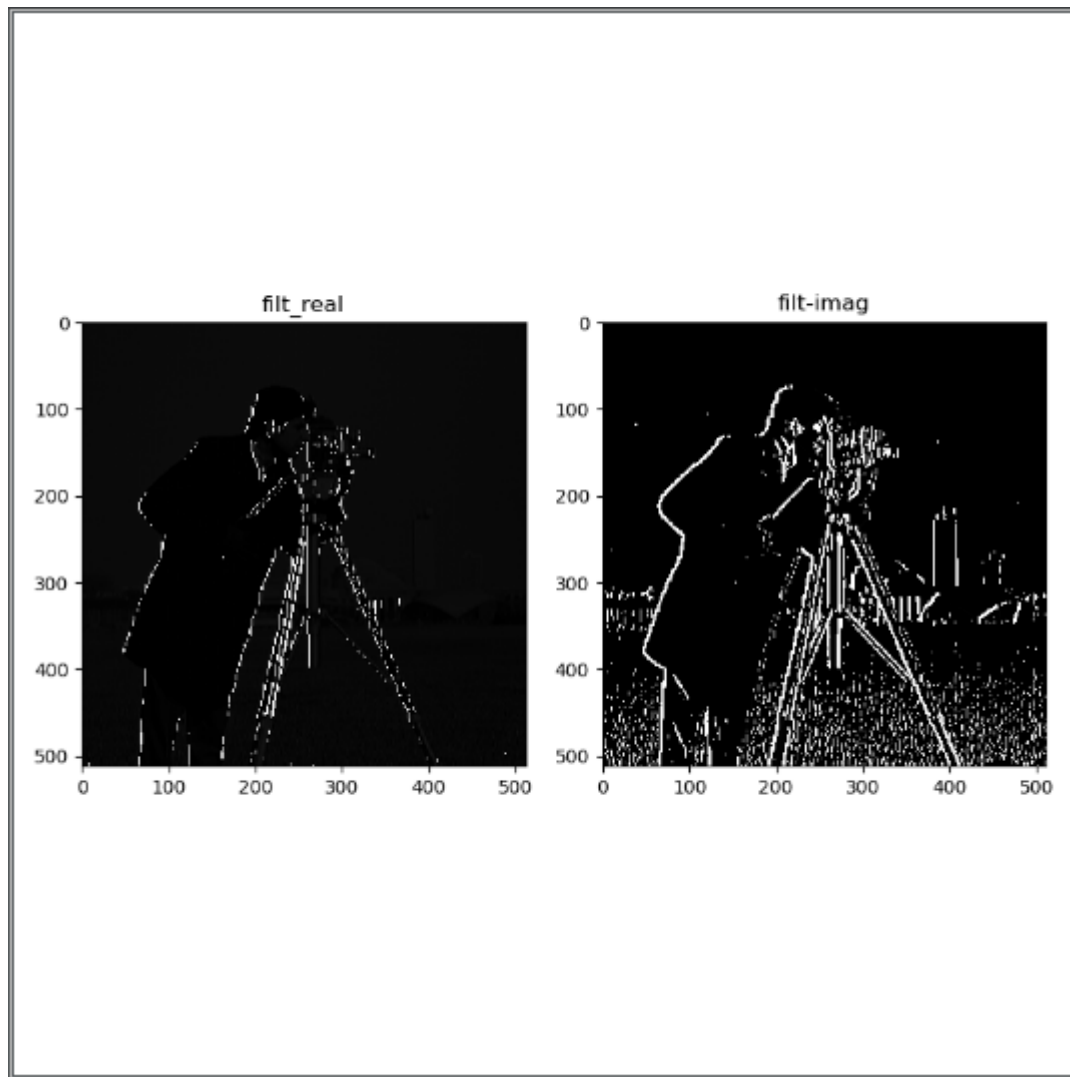
- Canny 算子

采用的是 python skimage 图像库中的 feature 类，提供对图像进行处理的 canny 边缘检测算子，对应代码文件为 scikifunc/filter.py。效果图如下：



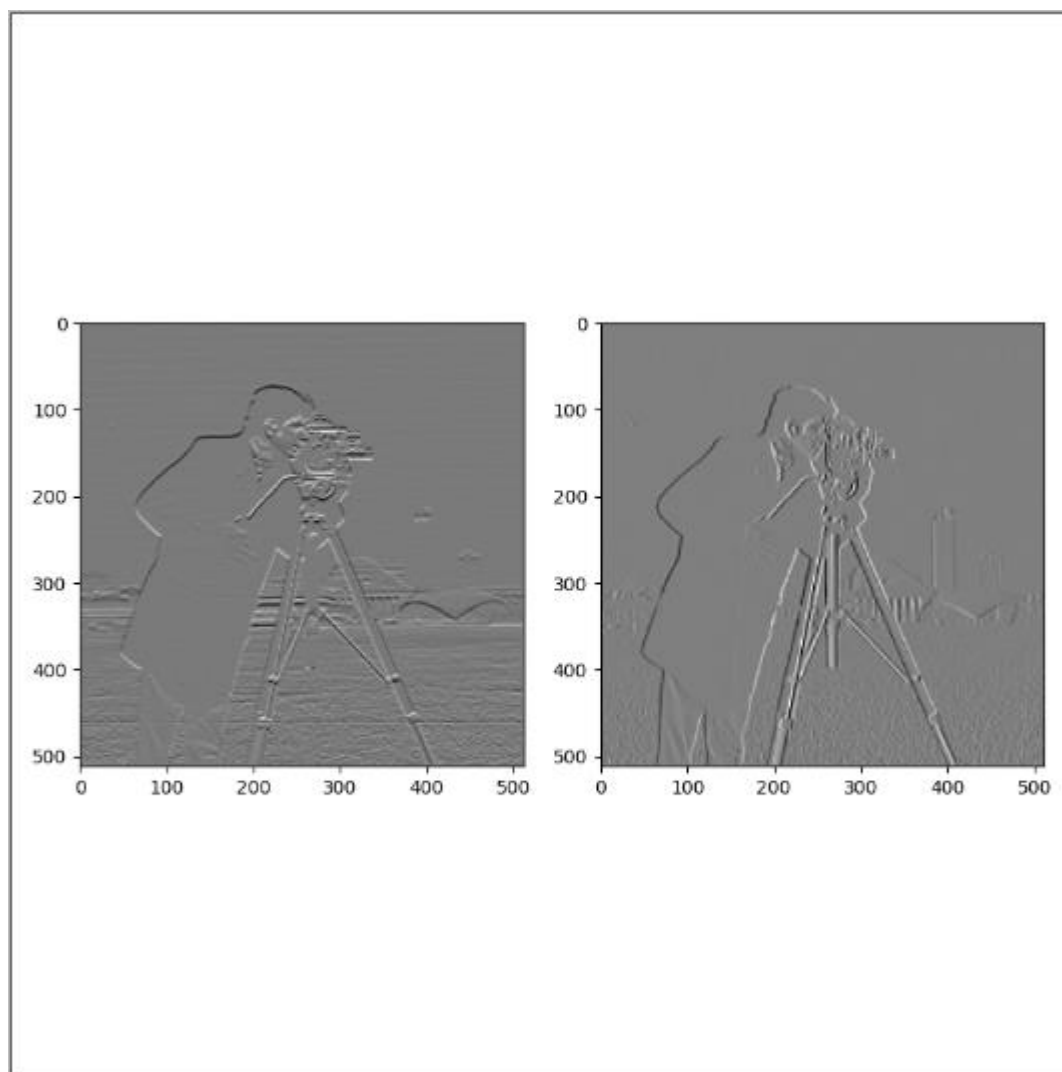
- Gabor 算子

采用的是 python skimage 图像库中的 filter 类，提供各种 filter 滤波器对图像进行处理。其中包括了各种边缘检测算子，Gabor 滤波可用于来进行边缘检测和纹理特征提取。对应代码文件为 scikifunc/filter.py。效果图如下：



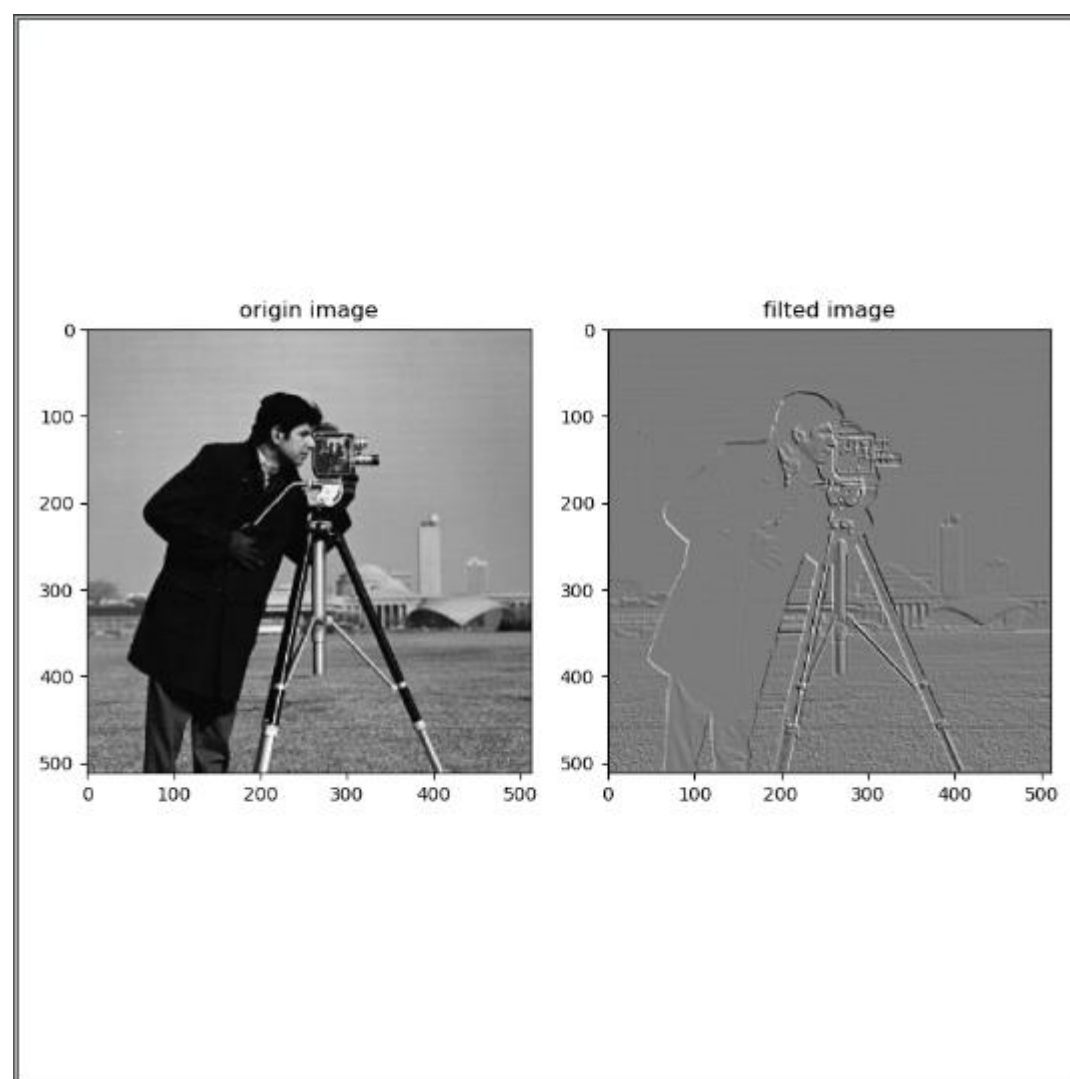
- 水平垂直边缘检测

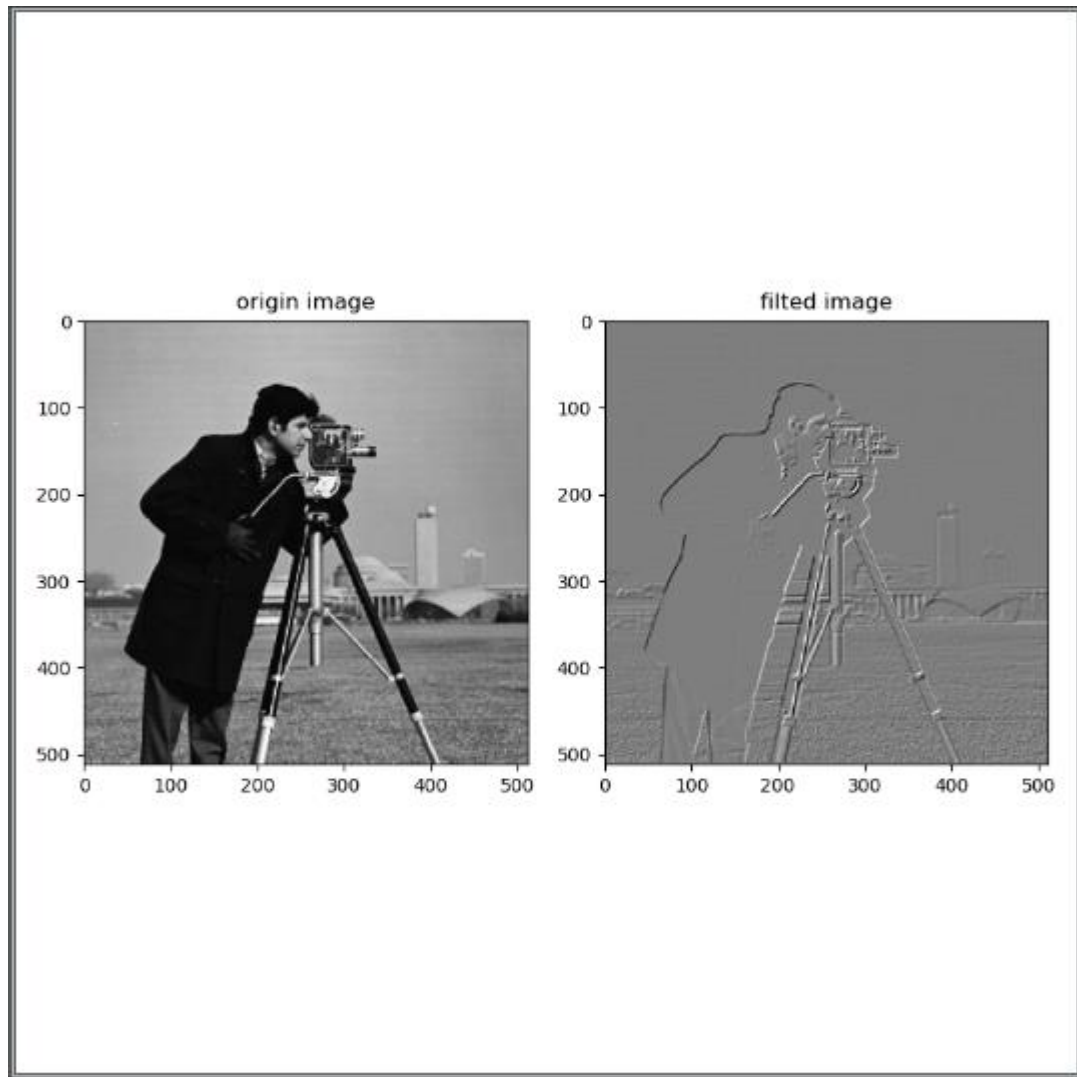
采用的是 python skimage 图像库中的 filters 类，利用 sobel 函数 api 进行水平边缘检测：
sobel_h, prewitt_h, scharr_h 垂直边缘检测： sobel_v, prewitt_v, scharr_v 对应代码文件为
scikifunc/filters.py。效果图如下：



- 交叉边缘检测

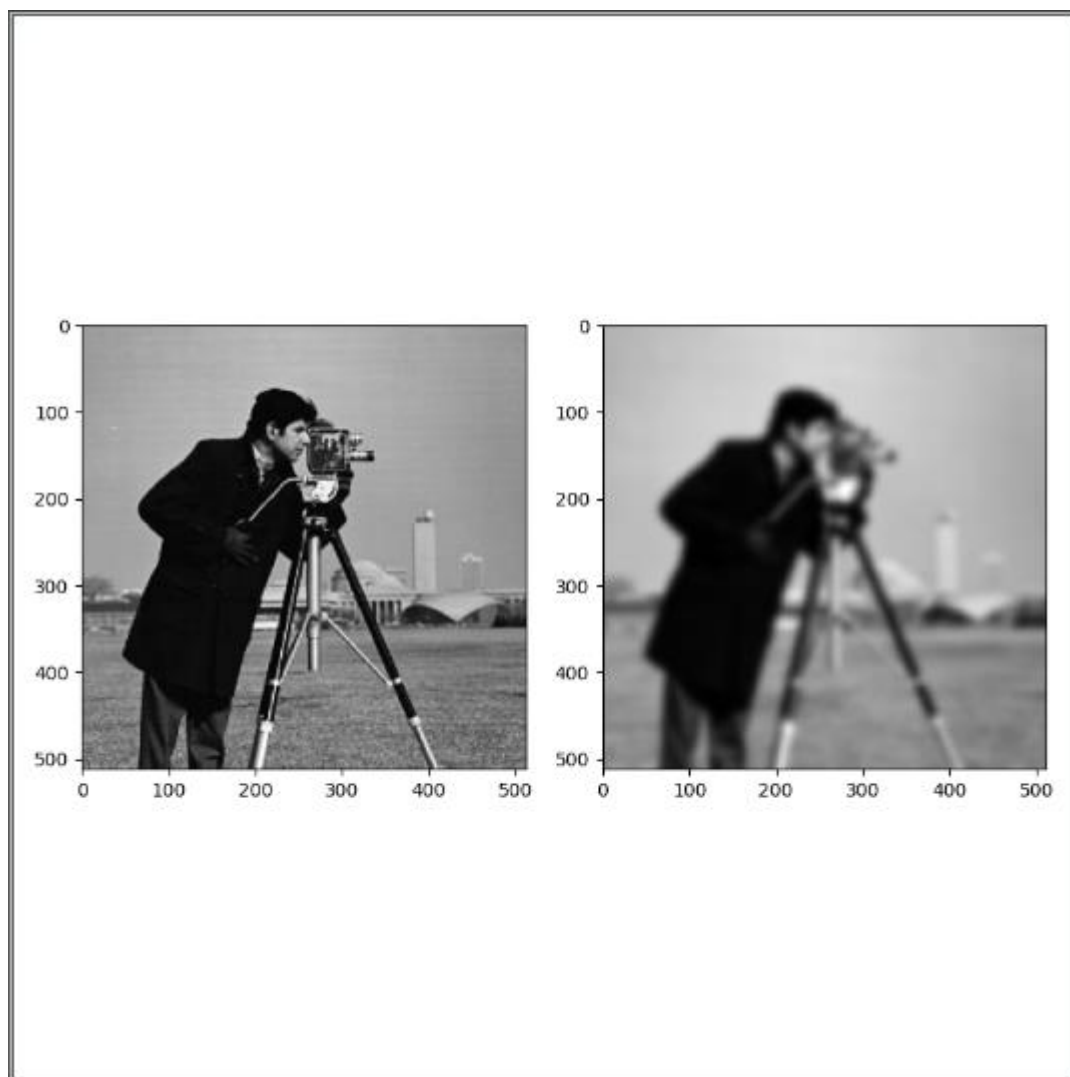
采用的是 python skimage 图像库中的 filters 类，利用 Roberts 的十字交叉来进行交叉边缘检测，通过 `roberts_neg_diag` 和 `roberts_pos_diag` 进行两个侧重点不同的核进行交叉边缘检测。对应代码文件为 `scikifunc/filters.py`。效果图如下：





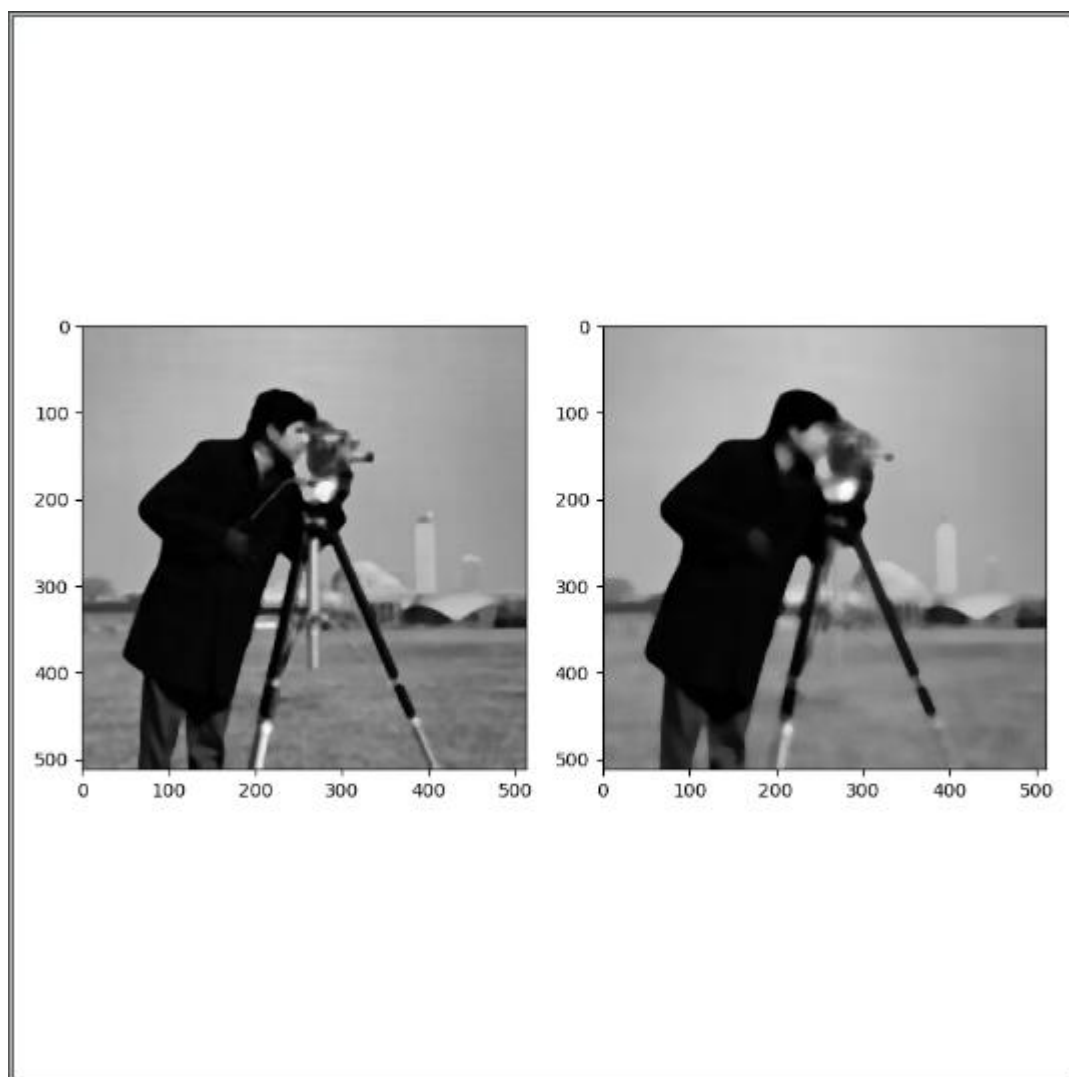
- 高斯滤波

采用的是 python skimage 图像库中的 filters 类，通过 gaussian_filter 函数直接对图像进行高斯滤波处理，对应代码文件为 scikifunc/filters.py。效果图如下：



- 中值滤波

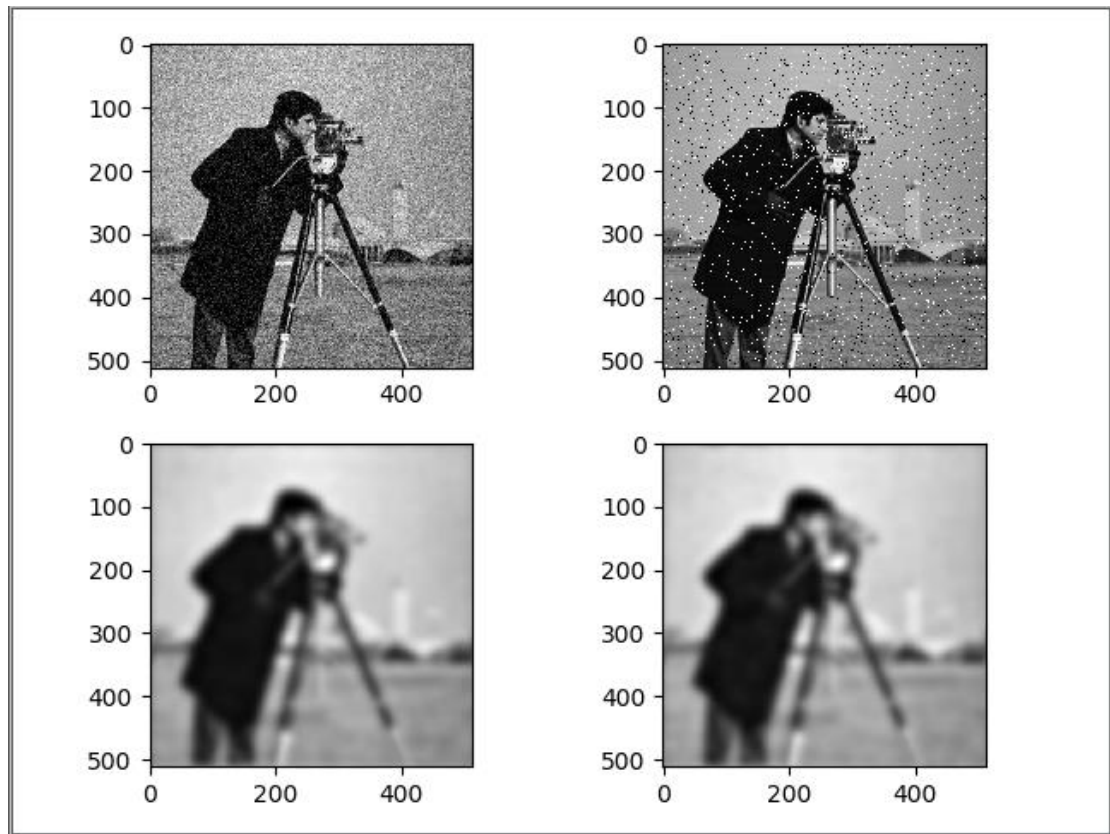
采用的是 python skimage 图像库中的 filters 类，通过 median 函数对图像进行中值滤波处理，对应代码文件为 scikifunc/filters.py。效果图如下：



2.2.5 高级

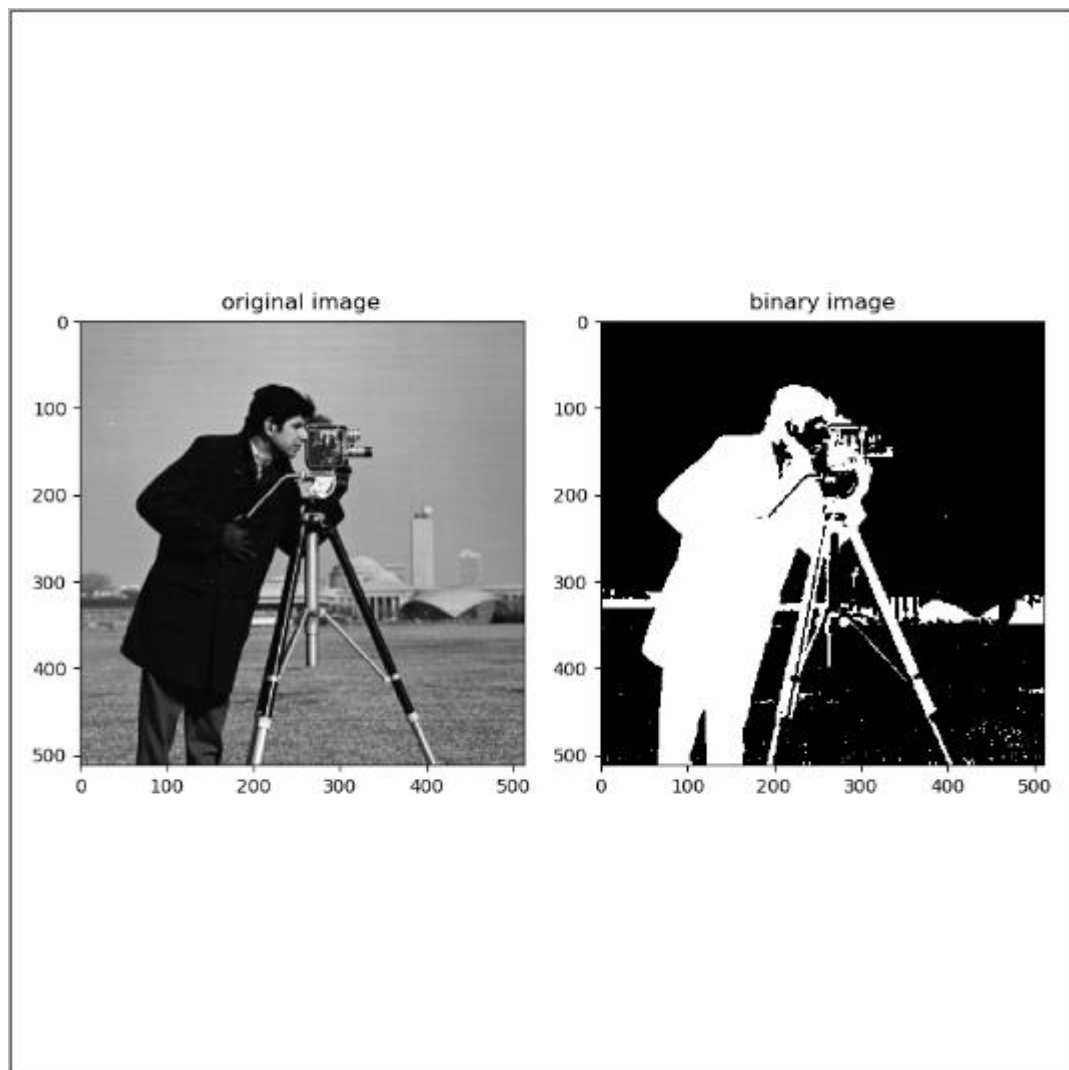
- 图像复原

采用的是python skimage 图像库中的 restoration 类,提供各种对图像的基本复原函数,利用 random noise 函数来对图像进行添加高斯噪声和椒盐噪声的操作,同时利用 wiener 函数对加噪声的图像进行一个复原,对应代码文件为 scikifunc/restoration.py。效果图如下:



- 图像阈值分割

采用的是 python skimage 图像库中的 filters 类,提供各种对图像的基本阈值分割函数,利用 `threshold_otsu`, `threshold_adaptive`, `threshold_isodata` 等函数来对图像进行一定的阈值分割操作,对应代码文件为 `scikifunc/binary.py`。效果图如下:

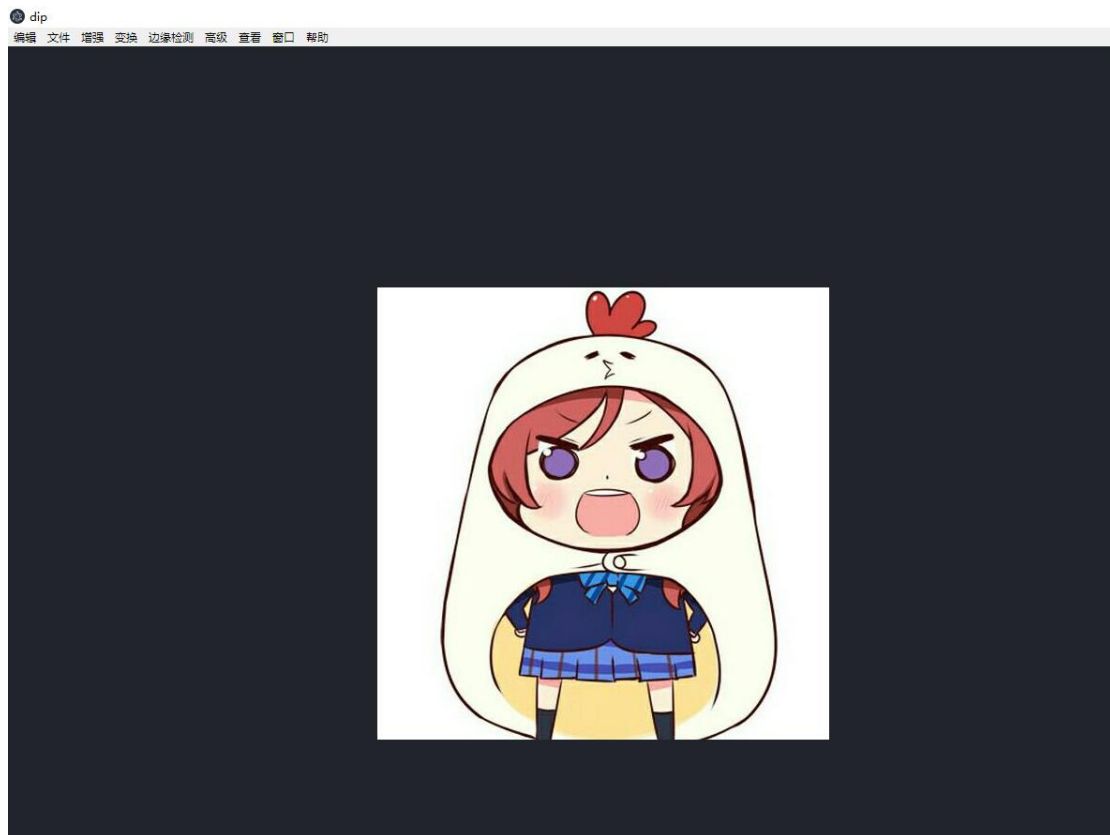


3 界面设计

3.1 首页



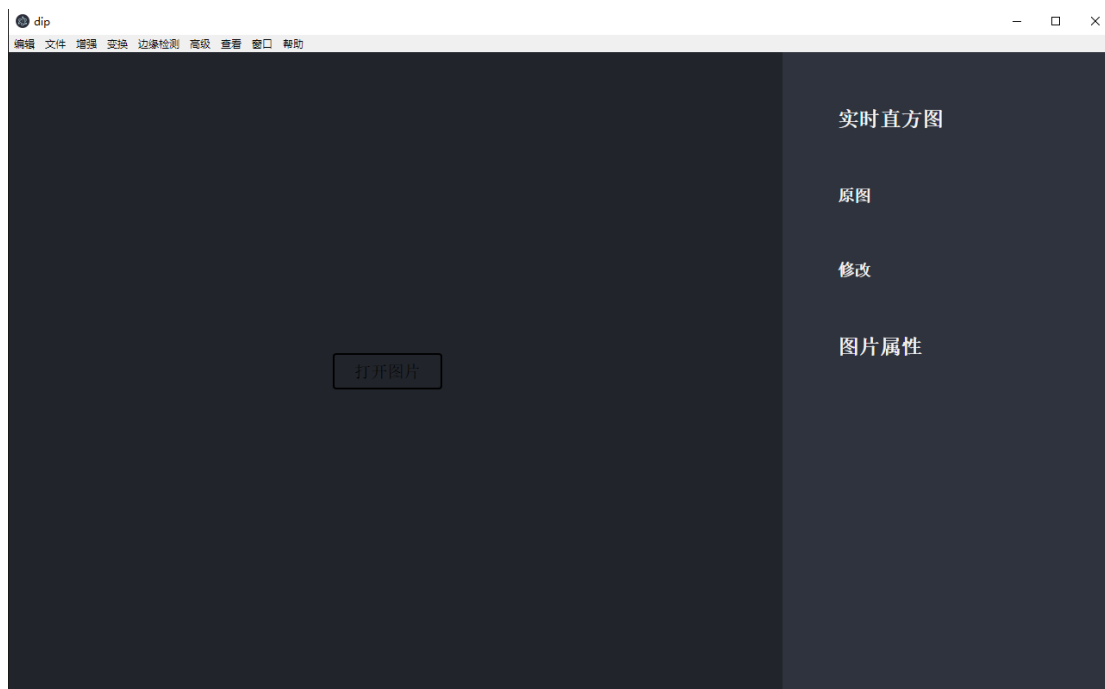
3.2 菜单设计



- 文件

- 打开图片
 - 转化为灰度图
 - 保存图片
- 增强
 - 直方图修正
 - 灰度图增强
 - 彩色图增强
 - 低通滤波
 - 高通滤波
 - 同态滤波
- 变换
 - 旋转平移
 - 拉伸（调整尺寸）
 - 放大
 - 缩小
- 边缘检测
 - Sobel
 - Laplace
 - Prewitt
 - Roberts
 - Canny
 - Gabor
 - 水平、垂直边缘检测
 - 交叉边缘检测
 - 高斯滤波
 - 中值滤波
 - 维纳滤波
- 高级
 - 图像复原
 - 图像分割
 - 图像压缩

3.3 功能页



4 总结

4.1 整个项目总结

在实现该软件系统的过程中遇到了比较多的问题,很惭愧并没有完整的实现整个软件系统,一开始的想法是利用 electron 进行开发能够不受运行平台的限制,同时也能够打包成 exe 运行程序,因为采用 MVVC 的模式进行开发,技术栈采用的 python 科学计算库和 nodejs 环境下的 electron 作为后端和前端开发环境,因为连接 python 和 nodejs 的技术比较少见,一方面接触的是之前并未了解的 RPC 技术,采用的 zerorpc 作为跨语言沟通的桥梁,另一方面可能是自己技术方面不够到位,最终导致整个项目的流产。流产的因素包括以下: zerorpc 的运行机制的不够了解和深入,对 python 和 JavaScript 以及 nodejs 等技术并不是很深入的研究过,代码经验不够,以及自己的一些惰性吧…最终导致的结果就是, MVVC 模式下的前端和后端确实是实现了分离,但因为关键点 zerorpc 没有攻破,导致 python 下的数字图像处理的库得到的结果无法反馈到前端 GUI 上。

4.2 数字图像处理相关

在 python 实现各种图像处理的功能时,通过对 pillow 和 scikit-image 两个由浅入深的数字图像处理库的 API 的了解,虽然基本上都不是自己实现的,但是在实现各个课堂上提到的大部分功能的同时,也了解了数字图像处理的一些方方面面的基本知识吧。通过阅读源码,也是对图像在内存中的保存形式以及像素,颜色等内容对应的内存结构有部分的了解,可以说对数字图像的基本知识和功能有一个比较大概的认识吧。

4.3 一些感想

另一方面，因为自己对 electron 比较感兴趣，在前端的开发过程中也是遇到了比较多的问题，但在 electron 的开发中其实发现自己的学习的薄弱点，以及对 electron 文档的开发方式有点生涩，通过在 GitHub 上学习两个比较优秀的项目（electron-zh 和 python-node-example）也是了解到比较优秀的 electron 的项目目录结构和开发代码结构。同时也因为选择了一条目前看来只有一个人走过的道路（node+python 实现 electron+python 服务），遇到了挺多的技术难题，也了解到一种新的机制——RPC，接触到 zerorpc 这样一个十分便利的神器（虽然对我不太友好），可以作为以后这样一条技术路线的参考。

5 参考

基础 pillow <http://pythonguidecn.readthedocs.io/zh/latest/>

直方图均衡化 <https://blog.csdn.net/sunmc1204953974/article/details/50606395>

API 接口, PIL 介绍 <https://blog.csdn.net/gzlaiyonghao/article/details/1852726>

pillow 官方 doc <http://pillow.readthedocs.io/en/4.2.x/reference/ImageFilter.html>

pillow filter 详解 <https://blog.csdn.net/FloatDreame/article/details/79015551>

进阶 scikit-image <https://www.cnblogs.com/denny402/p/5121501.html>

scikit-image 文档 <http://scikit-image.org/docs/dev/api/skimage.filters.html>

scikitimage <https://wizzardforcel.gitbooks.io/scipy-lecture-notes/content/16.html>

通用了解 https://blog.csdn.net/mao_xiao_feng/article/details/73251440

一些 blog 内容 <https://www.cnblogs.com/denny402/p/5131004.html>