

**图像处理课程设计**

**考查报告**

|  |  |
| --- | --- |
| 姓 名： | 孙鹤轩 |
| 学 号： | 20211000156 |
| 班 级： | 191211 |
| 学 院： | 计算机学院 |
| 题 目： | 图像处理系统开发 |
| 指导 教师： | 左博新 |

2023 年 12 月

目 录

[摘 要 I](#_Toc155743253)

[Abstract II](#_Toc155743254)

[1 引言 1](#_Toc155743255)

[2 实验与方法 1](#_Toc155743256)

[2.1 直方图均衡化 2](#_Toc155743257)

[2.1.1直方图均衡化概述 2](#_Toc155743258)

[2.1.2直方图均衡化算法原理 2](#_Toc155743259)

[2.1.3直方图均衡化算法实现步骤 2](#_Toc155743260)

[2.2 频率域滤波实现 3](#_Toc155743261)

[2.2.1频率域滤波概述 3](#_Toc155743262)

[2.2.2频域图像平滑实现 3](#_Toc155743263)

[2.2.3频域图像锐化 5](#_Toc155743264)

[2.3空域滤波实现 6](#_Toc155743265)

[2.3.1空域滤波概述 6](#_Toc155743266)

[2.3.2空域图像平滑 6](#_Toc155743267)

[2.3.3空域锐化滤波 9](#_Toc155743268)

[2.4图像去噪 11](#_Toc155743269)

[2.5小波变换 11](#_Toc155743270)

[2.5.1小波变换概述 11](#_Toc155743271)

[2.5.2小波变换应用 12](#_Toc155743272)

[3 结果展示（标题可自拟） 13](#_Toc155743273)

[3.1直方图均衡化 13](#_Toc155743274)

[3.2频率域滤波 14](#_Toc155743275)

[3.3空域滤波 15](#_Toc155743276)

[3.4图像去噪 16](#_Toc155743277)

[3.5小波变换 16](#_Toc155743278)

[4 讨论与结论 17](#_Toc155743279)

[参考文献 18](#_Toc155743280)

基于Python的数字图像处理系统

# 摘 要

本论文介绍了一个基于Python的数字图像处理系统的设计与实现。该系统包括直方图均衡化、频率域滤波、空域滤波、图像去噪和小波变换等功能模块，旨在提供一种方便、高效且功能丰富的图像处理工具。在直方图均衡化模块中，我们使用了直方图分析和调整像素灰度级的方法，实现了图像的对比度增强和亮度均衡化。频率域滤波模块基于傅里叶变换，通过滤波器的设计和应用，实现了图像的频域特征增强和噪声去除。空域滤波模块提供了多种滤波器选项，包括均值滤波、中值滤波和高斯滤波等，可根据具体需求选择不同的滤波方法进行图像改善和噪声消除。图像去噪模块采用了经典的降噪算法，如均值滤波、中值滤波和小波阈值去噪等，有效地减少了图像中的噪声干扰。最后，我们实现了小波变换模块，通过分解和重构图像，提取了不同尺度和方向的频域信息，用于图像的去噪应用演示。该系统经过实验验证，在处理各类数字图像时表现出良好的性能和效果。通过本论文的研究，我们对数字图像处理技术有了更深入的理解，并为图像处理领域的进一步研究和应用提供了有益的参考。

**关键字：**数字图像处理，直方图均衡化，频率域滤波，空域滤波，图像去噪，小波变换，Python

**Python-based digital image processing system**

# Abstract

This paper introduces the design and implementation of a digital image processing system based on Python. The system includes functional modules such as histogram equalization, frequency domain filtering, spatial domain filtering, image denoising and wavelet transform, etc., aiming to provide a convenient, efficient and feature-rich image processing tool. In the histogram equalization module, we use the method of histogram analysis and adjustment of pixel gray levels to achieve contrast enhancement and brightness equalization of the image. The frequency domain filtering module is based on Fourier transform and realizes frequency domain feature enhancement and noise removal of images through the design and application of filters. The spatial filtering module provides a variety of filter options, including mean filtering, median filtering, and Gaussian filtering. Different filtering methods can be selected according to specific needs for image improvement and noise elimination. The image denoising module uses classic noise reduction algorithms, such as mean filtering, median filtering, and wavelet threshold denoising, to effectively reduce noise interference in images. Finally, we implemented the wavelet transform module to extract frequency domain information of different scales and directions by decomposing and reconstructing the image for image denoising application demonstration. The system has been experimentally verified and shows good performance and effect when processing various types of digital images. Through the research of this paper, we have a deeper understanding of digital image processing technology and provide a useful reference for further research and application in the field of image processing.

**Key words:** Digital image processing, histogram equalization, frequency domain filtering, spatial domain filtering, image denoising, wavelet transform, Python

# 1 引言

数字图像处理是计算机视觉和图像分析领域中的重要研究方向，其应用涵盖了图像增强、特征提取、目标识别等多个领域。随着计算机技术的不断进步和数字图像处理算法的发展，人们对于高质量和高效率的图像处理方法的需求也越来越迫切。

针对数字图像处理的需求，许多研究者已经提出了许多有效的方法和算法。其中，直方图均衡化、频率域滤波、空域滤波、图像去噪和小波变换等技术在数字图像处理中得到了广泛应用。直方图均衡化通过调整图像的灰度级分布，提高图像的对比度和亮度均衡性，从而改善图像的视觉效果。频率域滤波基于傅里叶变换，可以通过滤波器的设计和应用，增强图像的频域特征，去除图像中的噪声和干扰。空域滤波则通过在图像的像素域上进行滤波操作，实现图像的平滑、锐化和边缘增强等目的。图像去噪技术致力于消除图像中的噪声，提高图像的质量和清晰度。而小波变换作为一种多尺度分析方法，在图像处理中具有出色的特征提取和去噪能力。

然而，尽管已经有许多优秀的数字图像处理方法存在，但现有的图像处理系统往往缺乏一种集成化、易用性好且功能丰富的解决方案。因此，本论文旨在设计和实现一个基于Python的数字图像处理系统，该系统将直方图均衡化、频率域滤波、空域滤波、图像去噪和小波变换等模块集成在一起，提供一种方便、高效、且功能强大的图像处理工具。

在本论文中，我们将详细介绍系统的设计和实现过程，并对各个模块进行深入的研究和分析。具体而言，我们将探讨直方图均衡化模块中的灰度级分布调整方法，频率域滤波模块中的滤波器设计和应用，空域滤波模块中的不同滤波器选项，图像去噪模块中的降噪算法选择，以及小波变换模块中的分解和重构过程等。通过实验验证和性能评估，我们将展示系统在各类数字图像处理任务上的优越性能和有效性。

通过本论文的研究，我们期望为数字图像处理领域的进一步研究和应用提供有益的参考和指导，推动数字图像处理技术的发展和应用。

# 2 实验与方法

（描述自己具体所作的工作和内容）

## 2.1 直方图均衡化

### 2.1.1直方图均衡化概述

直方图均衡化是一种常用的图像增强技术，旨在改善图像的对比度和亮度分布。图像的直方图表示了像素灰度级的分布情况，直方图均衡化通过重新分配像素的灰度级，使得图像的直方图在整个灰度级范围内均匀分布，从而增强图像的细节和视觉效果。直方图是对图像像素分布的一种统计，它将每个像素值出现的次数记录下来，并以图形的方式呈现出来。在一幅图像中，像素值的范围通常是有限的。例如，对于一个 8 位灰度图像，像素值的范围是 0 到 255。

### 2.1.2直方图均衡化算法原理

直方图均衡化是一种点运算，通过变换函数将原灰度值映射为目标灰度值。假如已知原图像的直方图(即图像灰度级的概率密度函数)，原始图像灰度级r归一化为[0, 1],即0≤r<1，是目标图像的灰度级概率密度函数。直方图均衡化是要找到一个变换函数，让目标图像的直方图趋向于一个常量，其中*L*为图像的灰度级，也就是使得映射后的灰度分布趋向均匀。

### 2.1.3直方图均衡化算法实现步骤

直方图均衡化具体步骤如下：

1. 将原图像的灰度级归一化，使得;
2. 统计原图像的灰度级概率分布,其中n为图像总像素数，代表第k个灰度级出现的频数(像素数)
3. 计算图像的累积分布函数
4. 根据累积分布函数，建立输入图像与输出图像之间的对应关系，也就是将映射为。由于所计算得到的很可能不是量化灰度级，因此需要量化处理，即在归一化灰度级中，寻找与最接近的得到量化的。

## 2.2 频率域滤波实现

### 2.2.1频率域滤波概述

频率域滤波的基本原理是将信号从时域转换到频域，然后在频域上对信号进行滤波操作，最后再将滤波后的信号转换回时域。这样可以有效地处理一些时域滤波难以解决的问题。

### 2.2.2频域图像平滑实现

#### 1理想低通滤波器

1. 简介：理想低通滤波器是一种假想的低通滤波器，其对于高于截止频率的信号完全截止，而对于截止频率的信号完全无失真通过。
2. 传递函数：，式中,其中是直流分量的坐标，对于低频中心化后的傅里叶频谱，；是截止频率
3. 算法实现：理想低通滤波器伪代码实现如下：

*function create\_ideal\_lowpass\_filter(image, frequency)*

*# 获取图像的行数和列数*

*rows, cols = image.shape*

*# 计算图像中心点的坐标*

*crow = rows div 2*

*ccol = cols div 2*

*# 创建一个与图像大小相同的全零数组作为掩膜*

*mask = new array[rows, cols] initialized with 0*

*# 在掩膜中心的正方形区域中设置为1，其余部分为0*

*for i = crow - frequency to crow + frequency do*

*for j = ccol - frequency to ccol + frequency do*

*mask[i, j] = 1*

*# 将图像与掩膜相乘，实现低通滤波效果*

*filtered\_image = elementwise\_multiply(image, mask)*

*return filtered\_image*

#### 2巴特沃夫低通滤波器

1. 简介：巴特沃斯低通滤波器又称为最大平坦滤波器。与理想低通滤波器不同，它的通带与阻带之间没有明显的不连续性。
2. 传递函数：，式中，是截止频率，即下降到最大值的时的频率；n是阶数，取正整数。
3. 算法实现：巴特沃斯低通滤波器伪代码实现如下：

*function create\_blft\_lowpass\_filter(fshift, D0, n)*

*# 获取频域图像的行数和列数*

*rows, cols = fshift.shape*

*# 计算频域图像中心点的坐标*

*crow = rows div 2*

*ccol = cols div 2*

*# 创建一个与频域图像大小相同的全零数组作为滤波器*

*H = new array[rows, cols] initialized with 0*

*# 遍历频域图像中的每个位置*

*for u = 0 to rows - 1 do*

*for v = 0 to cols - 1 do*

*# 计算当前位置到中心点的距离*

*D = sqrt((u - crow)^2 + (v - ccol)^2)*

*# 根据巴特沃斯滤波器公式计算频域滤波器的值*

*H[u, v] = 1 / (1 + (D / D0)^(2 \* n))*

*# 将频域图像与滤波器相乘，实现低通滤波效果*

*filtered\_fshift = elementwise\_multiply(fshift, H)*

*return filtered\_fshift*

#### 3高斯低通滤波器

1. 简介：高斯低通滤波器是一种常用的图像处理滤波器，用于平滑图像并降低高频噪声。它基于高斯分布函数，通过在频域中对图像进行卷积操作来实现。
2. 传递函数：，式中，是截止频率，即下降到最大值的时的频率。
3. 算法实现：高斯低通滤波器伪代码实现如下：

*function create\_gaussian\_lowpass\_filter(fshift, sigma)*

*# 获取频域图像的行数和列数*

*rows, cols = get\_dimensions(fshift)*

*# 计算频域图像中心点的坐标*

*crow = rows div 2*

*ccol = cols div 2*

*# 创建一个与频域图像大小相同的全零数组作为滤波器*

*mask = new array[rows, cols] initialized with 0*

*# 遍历频域图像中的每个位置*

*for i = 0 to rows - 1 do*

*for j = 0 to cols - 1 do*

*# 计算当前位置到中心点的距离的平方*

*distance\_squared = (i - crow)^2 + (j - ccol)^2*

*# 根据高斯滤波器公式计算频域滤波器的值*

*mask[i, j] = exp(-distance\_squared / (2 \* sigma^2))*

*# 将频域图像与滤波器相乘，实现低通滤波效果*

*filtered\_fshift = elementwise\_multiply(fshift, mask)*

*return filtered\_fshift*

### 2.2.3频域图像锐化

#### 1理想高通滤波器

理想高通滤波器的传递函数与理想低通滤波器传递函数正好相反为：

式中各符号与理想低通滤波器传递函数一致，实现方法相似更改传递函数即可。

#### 2巴特沃夫高通滤波器

巴特沃夫高通滤波器的传递函数为：

式中各符号与巴特沃夫低通滤波器一致，实现方法相似更改传递函数即可。

#### 3高斯高通滤波器

高斯高通滤波器传递函数为

与高斯低通滤波器相似，实现方法更改传递函数即可。

## 2.3空域滤波实现

### 2.3.1空域滤波概述

利用邻域像素点的高相关性，借助邻域像素点的灰度值来推测受污染像素点的灰度值，这种图像增强技术成为空域滤波，它是空间坐标域直接对像素点及其指定邻域的灰度值进行操作，可表达为：，其中，是输入图像的灰度值，是输出图像的灰度值，是对输入图像的像素点及其指定邻域的灰度值进行处理的操作符。

### 2.3.2空域图像平滑

#### 1均值滤波

1. 简介：均值滤波（又称为邻域平均法）就是对原始图像的每个像素点取一个邻域*S*，计算*S*中所有像素灰度值的平均值，作为空域平均处理后图象个的像素值，输出图像的灰度值等于输入图像灰度值的局部平均。
2. 表达式：，式中是的矩阵，；*S*是以点为中心的邻域的集合；*M*是邻域*S*中的像素点阵。
3. 算法实现：以下是均值滤波实现的伪代码

*function meanFilter(image, filterSize):*

*filteredImage = 创建与image相同大小的新图像*

*halfSize = filterSize // 2*

*for each 像素 (x, y) in image:*

*total = 0*

*count = 0*

*for each 邻居像素 (i, j) in filterSize范围内:*

*if (x + i >= 0 and x + i < image.width and y + j >= 0 and y + j < image.height):*

*total += image[x + i, y + j]*

*count += 1*

*filteredImage[x, y] = total / count*

*return filteredImage*

#### 2高斯滤波

1. 简介：高斯滤波是加权的平均滤波，高斯滤波是一种线性平滑滤波，适用于消除高斯噪声。高斯滤波运用了高斯正态分布的密度函数来计算滤波核上的权重。
2. 正态分布函数表达式：
3. 算法实现：以下是均值滤波实现的伪代码

*function gaussianFilter(image, filterSize, sigma):*

*filteredImage = 创建与image相同大小的新图像*

*halfSize = filterSize // 2*

*kernel = 生成高斯卷积核(filterSize, sigma)*

*for each 像素 (x, y) in image:*

*total = 0.0*

*for each 邻居像素 (i, j) in filterSize范围内:*

*if (x + i >= 0 and x + i < image.width and y + j >= 0 and y + j < image.height):*

*weight = kernel[i, j]*

*total += weight \* image[x + i, y + j]*

*filteredImage[x, y] = total*

*return filteredImage*

*function 生成高斯卷积核(filterSize, sigma):*

*kernel = 创建大小为filterSize的二维数组*

*sum = 0.*

*for each 像素 (i, j) in filterSize范围内:*

*x = i - filterSize // 2*

*y = j - filterSize // 2*

*exponent = - (x \* x + y \* y) / (2 \* sigma \* sigma)*

*weight = exp(exponent)*

*kernel[i, j] = weight*

*sum += weight*

*for each 像素 (i, j) in filterSize范围内:*

*kernel[i, j] /= sum*

*return kernel*

#### 3阈值邻域平滑滤波

1. 简介：阈值平滑滤波是人们为了减轻图像平滑后的模糊情况而诞生的一种空域滤波，其思想是将原图像原像素点的灰度值与均值滤波得到的像素点的灰度值进行比较，如果它们的绝对差小于设定的阈值，则认为当前像素点不是噪声点，像素点的灰度值采用原像素点的值；如果它们的绝对差大于设定的阈值，则认为当前像素点已被噪声污染，像素点的灰度值采用均值滤波得到的值。
2. 表达式：

式中，是原像素点的灰度值，是均值滤波得到的灰度值，*T*是根据经验设定的一个阈值。

1. 算法实现：以下是阈值邻域平滑滤波的伪代码：

*function thresholdSmoothFilter(image, filterSize, threshold):*

*//image 是输入图像，filterSize 是滤波器的尺寸，threshold 是阈值*

*filteredImage = 创建与image相同大小的新图像*

*halfSize = filterSize // 2*

*for each 像素 (x, y) in image:*

*total = 0.0*

*count = 0*

*for each 邻居像素 (i, j) in filterSize范围内:*

*if (x + i >= 0 and x + i < image.width and y + j >= 0 and y + j < image.height):*

*total += image[x + i, y + j]*

*count += 1*

*mean = total / count*

*if abs(image[x, y] - mean) <= threshold:*

*filteredImage[x, y] = image[x, y]*

*else:*

*filteredImage[x, y] = mean*

*return filteredImage*

#### 4中值滤波

1. 简介：中值滤波是一种非线性统计滤波。首先它是非线性运算，也就是；另外它是一种统计滤波，此滤波运算需要对滤波集合(滤波核对应的像素所形成的像素集合)中的灰度值进行统计排序，然后选择滤波集合中的最大值、最小值或中值，由此产生了最大值滤波、最小值滤波和中值滤波。非线性统计滤波的滤波核没有权重，它只需要确定滤波核大小，从而确定滤波集合。
2. 表达式：

式中，是以为中心的邻域集合，函数就是取集合的中间值。

1. 算法实现：以下是中值滤波实现的伪代码：

*function medianFilter(image, filterSize):*

*filteredImage = 创建与image相同大小的新图像*

*halfSize = filterSize // 2*

*for each 像素 (x, y) in image:*

*neighbors = []*

*for each 邻居像素 (i, j) in filterSize范围内:*

*if (x + i >= 0 and x + i < image.width and y + j >= 0 and y + j < image.height):*

*neighbors.append(image[x + i, y + j])*

*neighbors.sort()*

*medianIndex = len(neighbors) // 2*

*filteredImage[x, y] = neighbors[medianIndex]*

*return filteredImage*

在这个伪代码中，image 是输入图像，filterSize 是滤波器的尺寸。

算法遍历输入图像的每个像素，并将其邻域像素的值收集到一个列表中。然后，对邻域像素的值进行排序。接下来，找到排序后列表中的中间值，并将其作为当前像素的值。

### 2.3.3空域锐化滤波

图像锐化与图像平滑是相反的操作，图像平滑是去除图像的细节，让图像变得模糊，而图像锐化是突出图像中的细节，让图像变得清晰。从数学上看，图像模糊的实质就是图像受到平均运算或者积分运算，让图像的细节消失。因此如果执行积分运算的逆运算(即微分运算)，就可以使图像清晰，因为微分运算是求信号的变化率，有加强高频分量的作用，可以使图像轮廓清晰。图像平滑是基于空域积分运算，而图像锐化是基于空域微分运算。

在图像处理中，由于通常使用的是梯度的大小，习惯上把梯度的大小称为“梯度”。在计算图像的梯度时，需要计算图像在某个方向上的偏微分，对于离散图像，可以采用差分的方式来近似一阶微分

#### 1一阶水平垂直差分

通常用一阶差分来近似一阶微分，即：

这种近似计算一阶微分的方法成为水平垂直差分法，因此梯度的近似计算公式可以表示为：

算法实现如下：

*function sharpenImage(image):*

*sharpened = 创建与image相同大小的新图像*

*for each 像素 (x, y) in image:*

*if (x > 0 and x < image.width - 1 and y > 0 and y < image.height - 1):*

*gx = image[x + 1, y] - image[x - 1, y] // 水平方向差分*

*gy = image[x, y + 1] - image[x, y - 1] // 垂直方向差分*

*gradient = abs(gx) + abs(gy) // 梯度值*

*sharpened[x, y] = image[x, y] + gradient*

*return sharpened*

算法遍历输入图像的每个像素，除了图像边缘像素外，对于每个像素，计算其水平方向和垂直方向的差分值（gx和gy）。然后，计算梯度值（gradient）作为水平方向和垂直方向差分的绝对值之和。最后，将原始像素值和梯度值相加，得到锐化后的像素值。

#### 2二阶拉普拉斯微分算子

二阶拉普拉斯微分算子是一种常用的空域锐化算子，用于增强图像的边缘和细节。它基于图像的二阶导数，通过对像素周围的邻域进行加权求和来计算锐化后的像素值。

二阶拉普拉斯微分算子可以使用以下模板表示：

在这个模板中，中心元素的权重为-4，表示对当前像素的负二阶导数的贡献。周围的四个元素的权重为1，表示对当前像素的邻域像素的负一阶导数的贡献。

使用二阶拉普拉斯微分算子进行空域锐化可以使边缘更加锐利，并增强图像中的高频细节。然而，它也可能增加图像中的噪声和干扰。因此，在实际应用中，通常会采用一些预处理和后处理步骤，如平滑滤波和边缘保护，以控制锐化效果并减少噪声。

拉普拉斯算子的梯度计算公式为：

则其算法实现可以表示为如下伪代码：

*function laplacianSharpen(image):*

*sharpened = 创建与image相同大小的新图像*

*for each 像素 (x, y) in image:*

*if (x > 0 and x < image.width - 1 and y > 0 and y < image.height - 1):*

*laplacian = -4 \* image[x, y] + image[x - 1, y] + image[x + 1, y] + image[x, y - 1] + image[x, y + 1]*

*sharpened[x, y] = image[x, y] - laplacian*

*return sharpened*

算法遍历输入图像的每个像素，除了图像边缘像素外，对于每个像素，根据二阶拉普拉斯微分算子的模板，计算其周围像素的加权和（laplacian）。然后，将该加权和从原始像素值中减去，得到锐化后的像素值。

## 2.4图像去噪

图像去噪的技术有很多，前文介绍的频域平滑滤波和空域平滑滤波在图像去噪中均有不同表现，这里不再做过多的阐述。

## 2.5小波变换

### 2.5.1小波变换概述

小波变换（Wavelet Transform）是一种在数字图像处理中广泛应用的数学工具，用于分析和处理图像的频率和空间信息。与传统的傅里叶变换相比，小波变换具有更好的时域和频域局部化特性，能够更好地捕捉图像中的局部细节和边缘信息。

小波变换的基本思想是通过将原始图像分解成不同尺度的图像细节和低频部分，来表示图像的不同频率成分。这个分解过程可以通过使用一组小波基函数来实现，这些小波基函数是原始小波函数的平移和缩放版本。常用的小波基函数有Haar小波、Daubechies小波、db2小波等。

小波变换的过程可以分为两个主要步骤：分解和重构。

1. 分解：将原始图像分解成不同尺度的细节和低频部分。这可以通过将原始图像与小波基函数进行卷积运算来实现。卷积的结果给出了原始图像在不同频率尺度上的响应。
2. 重构：根据分解得到的细节和低频部分，通过将它们与相应的小波基函数的逆变换进行卷积运算，重构出原始图像。这个过程可以通过使用逆小波变换来实现。

小波变换的优点在于它能够提供多尺度的频率分析，使得图像的不同频率成分可以得到更好的表示。此外，小波变换还具有局部化特性，可以更好地捕捉图像中的局部细节和边缘信息。因此，小波变换在图像压缩、图像增强、图像去噪和边缘检测等方面有着广泛的应用。

### 2.5.2小波变换应用

小波变换在数字图像处理中有许多应用，例如：图像压缩、图像增强、图像去噪、边缘检测、图像分割、特征提取等。

本次实验主要完成了图像去噪，去噪流程如下：

1. 读取图片
2. 为图片添加噪声
3. 选择小波变换基函数对图像进行小波变换
4. 提取小波系数cA(图像低频部分系数)、cH(水平方向上的高频细节系数)、cV(垂直方向上的高频细节系数)、cD(对角线方向上的高频细节系数)
5. 设定噪声阈值，将低于阈值的系数设置为零，保留高于阈值的系数
6. 对图像进行逆变换显示图像

# 3 结果展示（标题可自拟）

## 3.1直方图均衡化

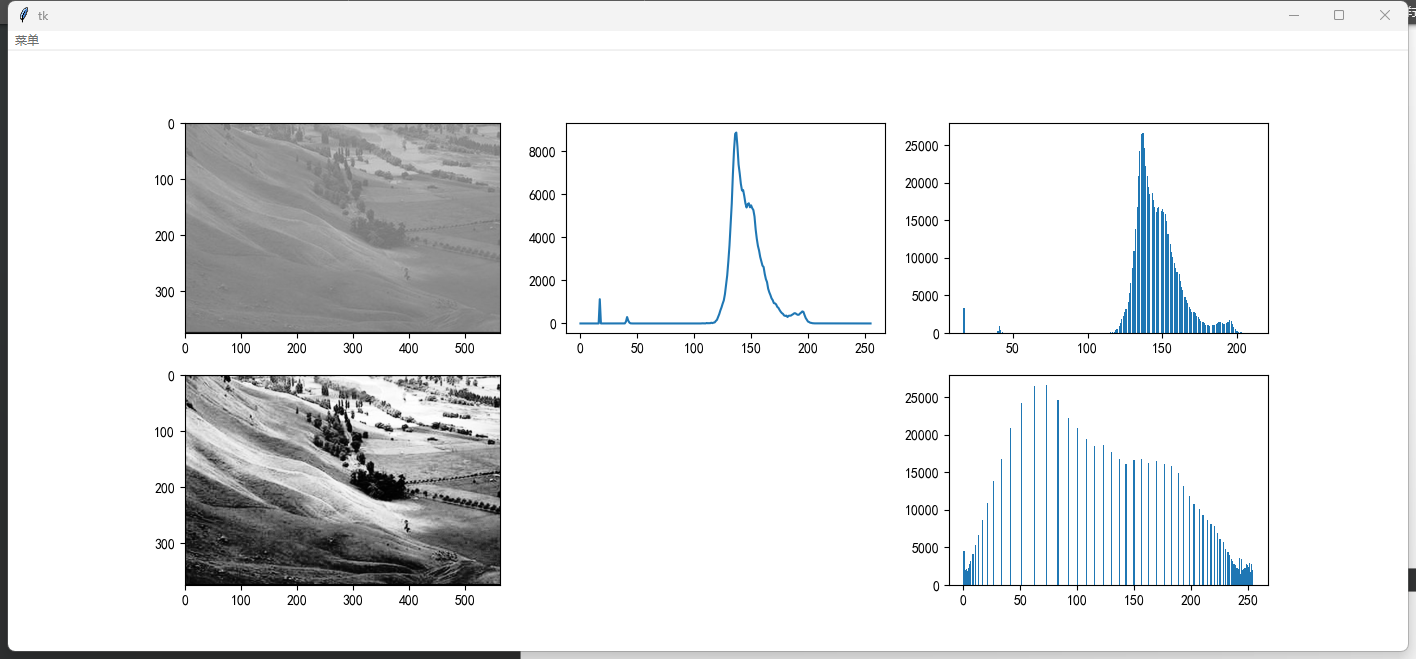


图 1直方图均衡化实验结果

从两个图像的对比中我们可以看出均衡化后的图片图像对比度有了明显提升，从中直方图中也可以看出灰度值分布较原先分布更加平均。

## 3.2频率域滤波

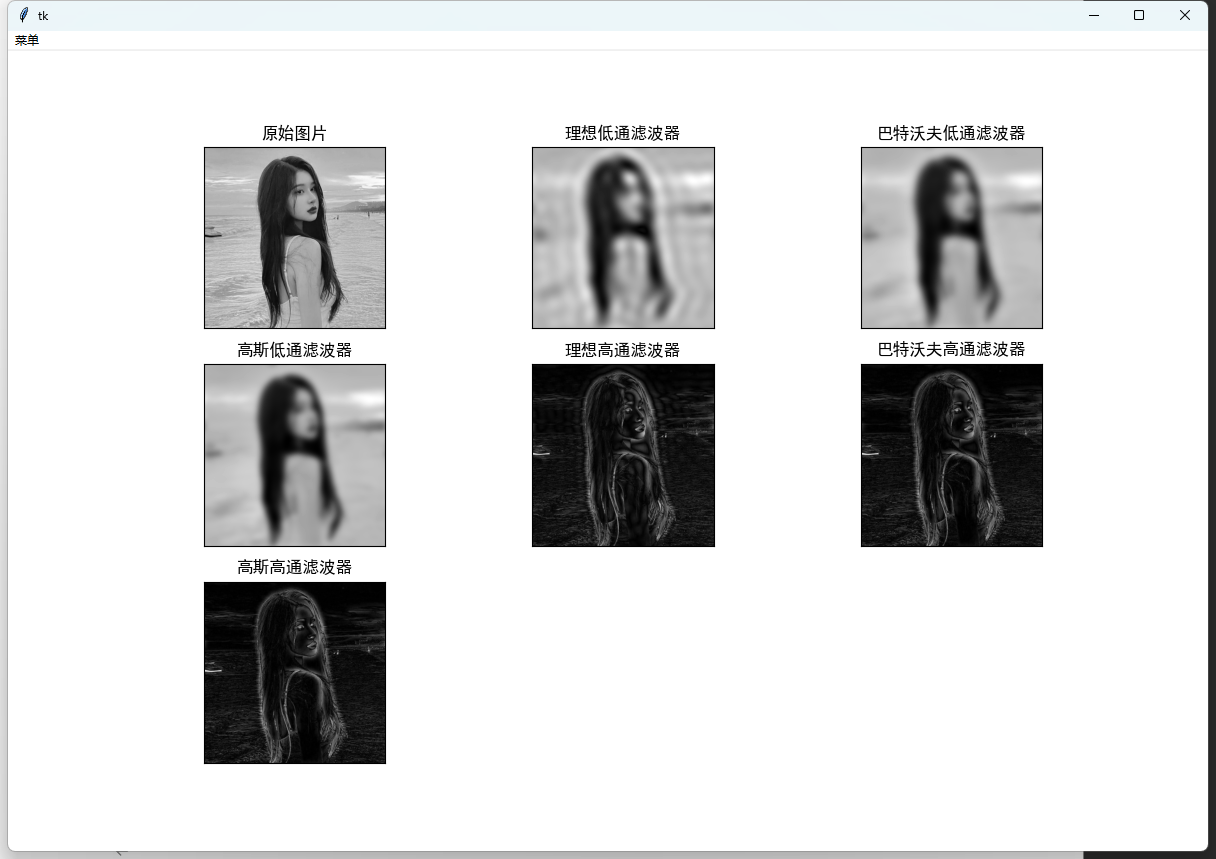


图 2频率域滤波实验结果

从实验结果看频率域滤波低通滤波器使图像变得更加平滑，但图像变得模糊，而高通滤波器对图像做的是锐化处理，凸显图像边缘信息。

## 3.3空域滤波

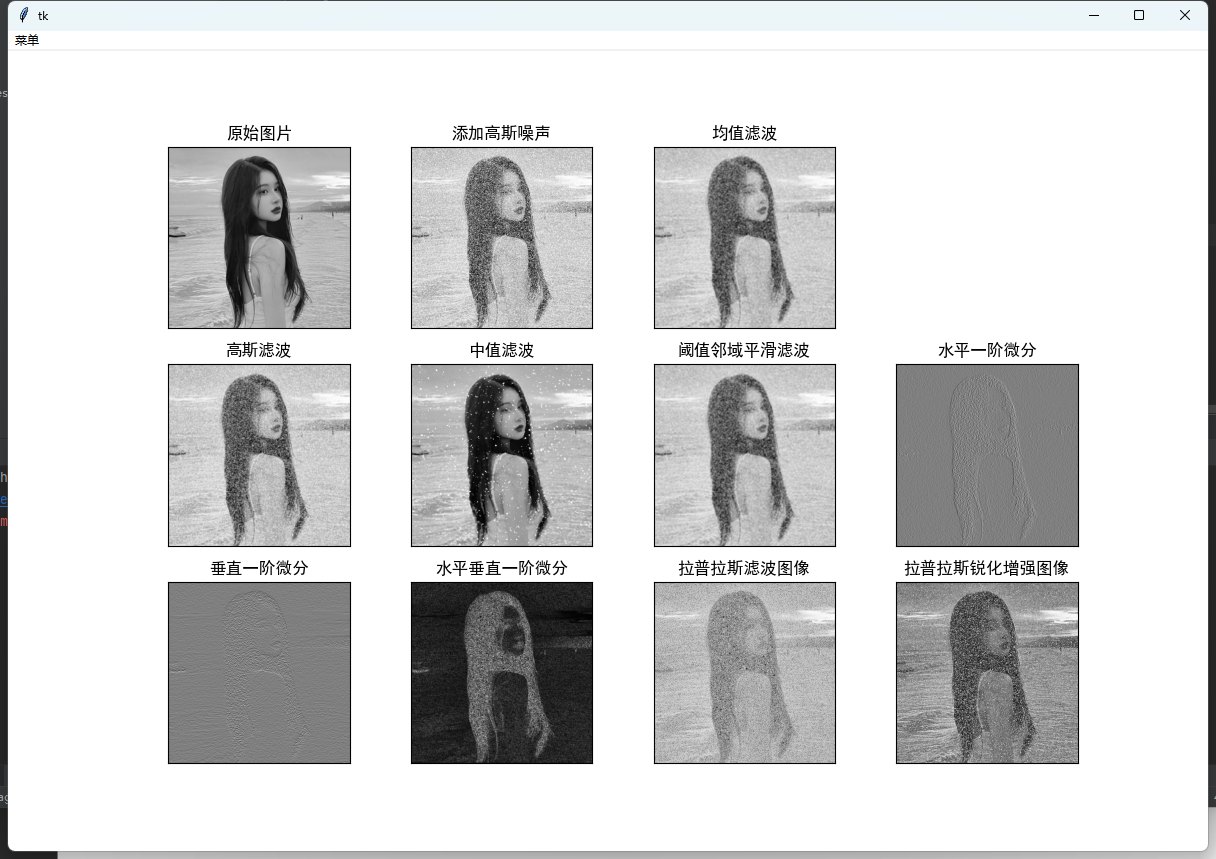


图 3空域滤波实验结果

在实验中我们添加了高斯噪声来观察不同滤波器的效果，从图片中我们可以看出空域平滑滤波对噪声去除均有一定效果，但如果噪声过多活着将滤波器的阈值调高，图像的去噪效果并不理想，而且对于均值滤波这种线性滤波器会使图片变得模糊，而空域图像锐化中我们展示了一阶微分的水平一阶微分、垂直一阶微分和水平垂直一阶微分的锐化处理结果，从不同梯度方向展示了锐化结果，同时展示了二阶拉普拉斯算子的滤波图像和增强图像，均有明显的处理痕迹。

## 3.4图像去噪



图 4图像去噪实验结果

在图像去噪实验中我们设定添加了两种噪声高斯噪声(1强度)、椒盐噪声(0.1密集度)，分别使用高斯滤波和中值滤波进行了去噪实验，从实验结果看高斯滤波有一定效果但也让图像变得模糊，中值滤波为非线性滤波器所以其对椒盐噪声这种离散点处理有明显效果。

## 3.5小波变换

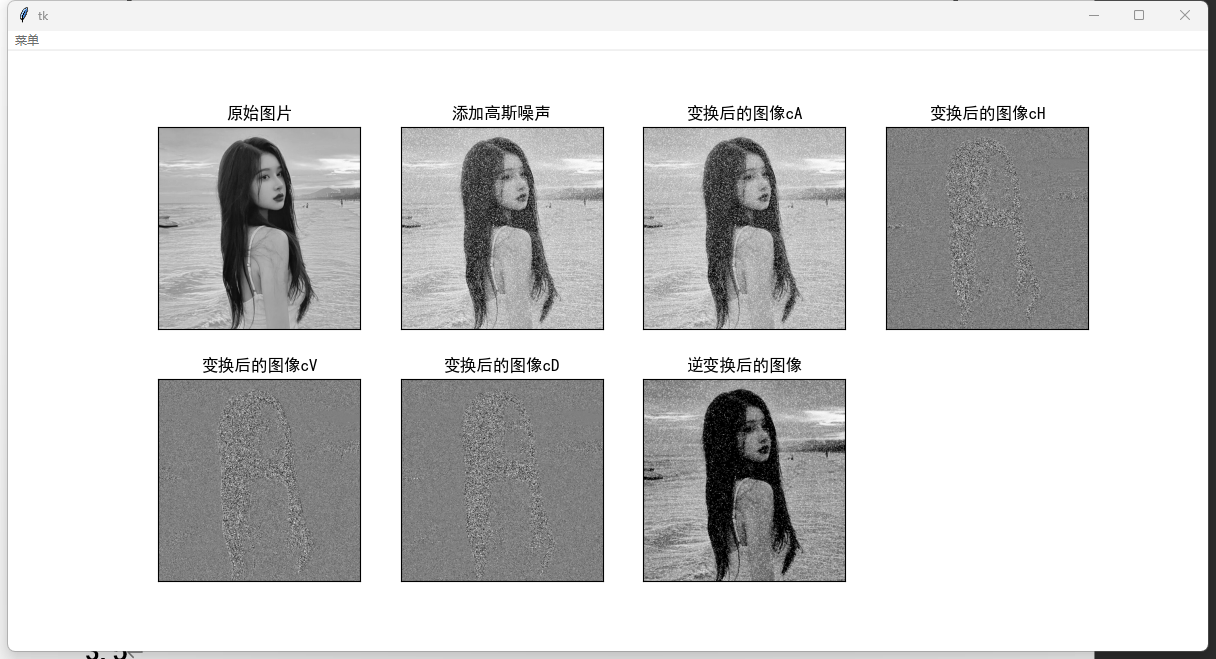


图 5小波变化实验结果

上图为小波变换的实验结果，我们主要是应用小波变化对图像进行特征提取来进行去噪实验，cA为变换后提取到的低频部分所以图片较为平滑，而cH、cV、cD分别从水平、垂直以及对角线方向提取图像高频部分，所以体现的是图像的边缘锐化特征，由变换后的图像我们也看出小波变换同样可以进行有效的图像去噪，并且不会使图片变得模糊。

# 4 讨论与结论

总的来说这次课程设计实验探讨了直方图均衡化模块中的灰度级分布调整方法，频率域滤波模块中的滤波器设计和应用，空域滤波模块中的不同滤波器选项，图像去噪模块中的降噪算法选择，以及小波变换模块中的分解和重构过程等。让我对数字图像处理有了进一步的了解，也学习到了多种现代数字图像处理算法与应用前景。同时系统的功能设计大部分使根据课本流程来做，缺乏一定的创新性。

需要改进的地方有很多，例如算法的性能在系统中没有体现，因为系统规模较小，图片的大小也比较小所以算法的性能比较比较困难，没有选择大尺寸图片进行对比试验；还有就是系统交互UI也没有精心设计，比较简陋；没有进行算法改进创新来得到更好的试验效果等等。今后我会更加留意相关方面知识，做出改进。

# 参考文献

[1]数字图像处理-基于python，蔡体健 刘伟主编

[2]博客：[https://blog.csdn.net/hu\_666666/article/details/127306483?ops\_request\_misc=%257B%2522request%255Fid%2522%253A%2522170482222116800182110019%2522%252C%2522scm%2522%253A%252220140713.130102334..%2522%257D&request\_id=170482222116800182110019&biz\_id=0&utm\_medium=distribute.pc\_search\_result.none-task-blog-2~all~sobaiduend~default-1-127306483-null-null.142^v99^pc\_search\_result\_base7&utm\_term=python%E6%95%B0%E5%AD%97%E5%9B%BE%E5%83%8F%E5%A4%84%E7%90%86%E7%9B%B4%E6%96%B9%E5%9B%BE%E5%9D%87%E8%A1%A1%E5%8C%96&spm=1018.2226.3001.4187](https://blog.csdn.net/hu_666666/article/details/127306483?ops_request_misc=%257B%2522request%255Fid%2522%253A%2522170482222116800182110019%2522%252C%2522scm%2522%253A%252220140713.130102334..%2522%257D&request_id=170482222116800182110019&biz_id=0&utm_medium=distribute.pc_search_result.none-task-blog-2~all~sobaiduend~default-1-127306483-null-null.142%5ev99%5epc_search_result_base7&utm_term=python%E6%95%B0%E5%AD%97%E5%9B%BE%E5%83%8F%E5%A4%84%E7%90%86%E7%9B%B4%E6%96%B9%E5%9B%BE%E5%9D%87%E8%A1%A1%E5%8C%96&spm=1018.2226.3001.4187)

[3]博客