# 图像空域滤波算法对比研究

摘要：随着数字图像处理技术的迅猛发展，图像质量的提高和图像分析的准确性变得越来越重要。然而，图像在采集、传输和存储过程中不可避免地会受到各种噪声的干扰。为了改善图像质量，去除这些噪声，图像滤波技术应运而生。因此，本设计对常见的图像空域滤波算法进行算法实现，包括均值滤波器，中值滤波器，Sobel滤波器，Robert滤波器，Laplace算子，测试其对高斯噪声，椒盐噪声，泊松噪声加噪图片的处理性能。最终，通过常见图像空域滤波算法对不同噪声图像的性能，分析出不同算法更适用于哪一种现实场景。该设计为不同场景选择图像空域滤波算法提供参考，通过合理选择和组合，可以有效提高图像处理的效果和准确性。

关键词：均值滤波器，中值滤波器，Sobel滤波器，Robert滤波器，Laplace算子。

# 设计任务分析及设计框图

## 设计任务分析（2级标题顶格，4号黑体，段前5磅，段后2.5磅）

图像空域滤波算法是一类直接在图像像素值上进行操作的图像处理技术，用于实现各种滤波功能，如去噪、平滑、锐化和边缘检测。这些算法通过在图像的原始空间域（即像素级别）上进行运算，以改变像素值或像素之间的关系来达到不同的处理目的。图像空域滤波算法可以分为两类，一类是滤波，第二类是图像锐化。

### 基于滤波的图像空域滤波算法

去噪滤波旨在消除图像中的随机噪声，通过计算像素周围邻域的均值或加权平均值来平滑像素值，从而使图像更清晰和易于分析。平滑滤波则通过减少像素值的高频成分，如高频噪声或细节，来使图像看起来更加平滑和连续，适用于降低细节对分析或视觉感知的干扰。

### 基于锐化的图像空域滤波算法

锐化滤波通过增强图像中的高频细节，如边缘和纹理，来增强图像的清晰度和对比度，使其更具视觉冲击力和细节感。最后，边缘检测滤波技术则专注于识别图像中像素值突变的位置，以便精确地定位物体的边界或轮廓，为目标检测和分割提供重要信息。

## 设计框图

在本设计中，将编写代码程序实现图像空域滤波所涉及的滤波器，如均值滤波器、中值滤波器、Sobel滤波器、robert滤波器、Laplace算子锐化滤波器等。然后将代码设计的滤波器与opencv中库函数进行对比；之后，引入加噪声的图像，对以实现的滤波器进行性能测试，分析滤波器的适用场景。框图如图 1- 2。

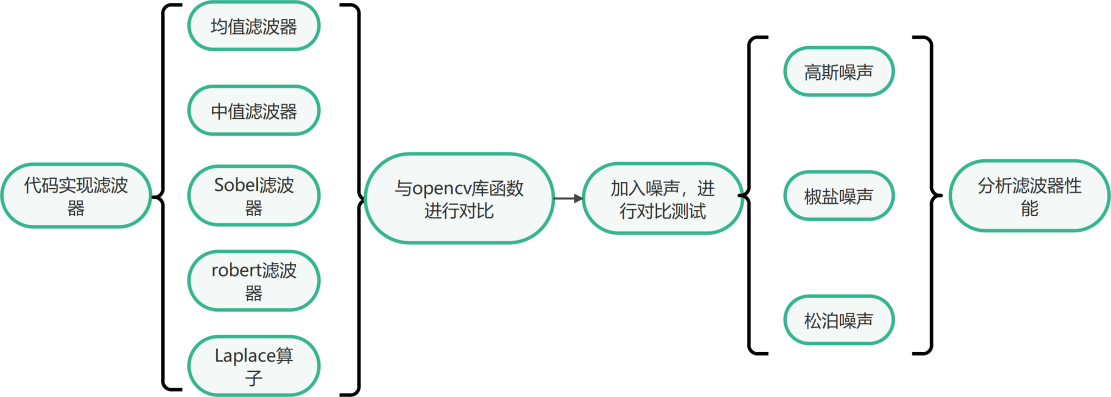


图 1- 2设计框图

# 图像空域滤波算子理论分析

## 空域滤波算子

### 理论知识

空域滤波算子是一种直接在图像空间域（像素域）进行操作的滤波器，通过对图像的像素值进行加权求和或其他操作来实现图像处理的目的。空域滤波算子在图像处理中的应用非常广泛，包括图像的平滑、锐化、边缘检测等。

### 基本原理

空域滤波是指在图像的空间域上进行滤波操作，即直接对图像的像素值进行处理。空域滤波器通过对图像中的每个像素及其邻域像素进行加权求和或其他运算来生成新的像素值，从而实现各种图像处理效果。

空域滤波的基本原理是卷积操作。卷积操作通过一个称为卷积核（或滤波器）的小矩阵在图像上滑动，并对核覆盖区域的像素值进行加权求和，从而生成新的图像。

## 各种空域滤波算子实现原理

### 平滑线性滤波器（均值滤波器）

平滑线性滤波器（均值滤波器）通过计算图像中每个像素邻域的平均值来平滑图像，减少噪声。具体来说，对于图像中的每一个像素点，将其邻域内的所有像素值相加并取平均值，然后用这个平均值替代原来的像素值。这种方法可以有效地平滑图像中的噪声。

假设窗口大小为k×k，均值滤波器的输出图像可表示为：

 （2-1）

其中， 是原始图像的像素值。

### 统计排序滤波器（中值滤波器）

统计排序滤波器是非线性滤波器的一种，最常见的是中值滤波器。中值滤波器通过取邻域内像素值的中值来替代中心像素值，从而去除噪声，特别是脉冲噪声（椒盐噪声）。中值滤波器能够有效地保留图像的边缘细节，同时去除噪声。

假设窗口大小为k×k，均值滤波器的输出图像可表示为：

 （2-2）

其中， 是原始图像的像素值，median表示取中值操作。

### Sobel滤波器

Sobel滤波器是一种边缘检测算子，主要用于图像处理中的边缘检测任务。它结合了微分运算和平滑操作，以增强图像中的边缘特征。

Sobel滤波器通过两个3x3的卷积核（分别用于检测水平方向和垂直方向的边缘）来计算图像每个像素点的梯度大小和方向。这两个卷积核分别是：

水平方向

 （2-3）

垂直方向

 （2-4）

计算每个像素点的梯度幅值

 （2-5）

将梯度幅值归一化到0到255的范围，并将其转换为8位无符号整数。将原始图像和梯度幅值图像进行加权求和，以达到锐化效果。

### 拉普拉斯算子

拉普拉斯算子是一种二阶导数算子，用于图像锐化和边缘检测。拉普拉斯算子通过计算图像的二阶导数来检测图像中的边缘。它对噪声比较敏感，因此在应用拉普拉斯算子之前，通常会对图像进行平滑处理。

连续形式的拉普拉斯算子:

 （2-6）

这个公式表示了在连续空间中，拉普拉斯算子是对图像函数关于x和y的二阶导数的求和。直观上，它测量了函数在每个点上曲率的总和，因此可以用于检测图像中的边缘和细节。

离散形式的拉普拉斯算子:

 （2-7）

这个公式是拉普拉斯算子在离散图像上的表示，它计算了当前像素与周围像素的差值之和减去当前像素的四倍。这样的计算方式可以突出图像中的边缘和细节，因为边缘处像素值的变化较大，而平均像素值较小。

### Roberts滤波器

Roberts 锐化是一种基于边缘检测的图像处理方法，通过计算图像的梯度来增强图像的边缘，从而达到锐化效果。Roberts 算子是一种简单且快速的边缘检测方法，其核心思想是使用两个 2x2 的卷积核来计算图像在对角线方向上的梯度。

Roberts 算子使用两个卷积核、，这两个核分别用于计算图像在x和y方向上的梯度。

 （2-8）

通过将图像与这两个核进行卷积运算，可以得到图像在 x 方向和 y 方向上的梯度：

 （2-9）

其中是输入图像。将 x 方向和 y 方向上的梯度组合起来，可以得到图像的梯度幅值：

 （2-10）

最后，将梯度幅值与原始图像组合，达到锐化效果：

 （2-11）

这种方法通过增强边缘信息，使得图像看起来更加清晰。

# 图像空域滤波算子算法实现

## 各种空域滤波算子与自带函数进行对比分析

### 平滑线性滤波器（均值滤波器）

平滑线性滤波器的目的是通过计算图像中每个像素邻域的平均值来减少噪声。均值滤波器的卷积核如下所示：

 （3-1）

实现代码如图 3- 1所示：

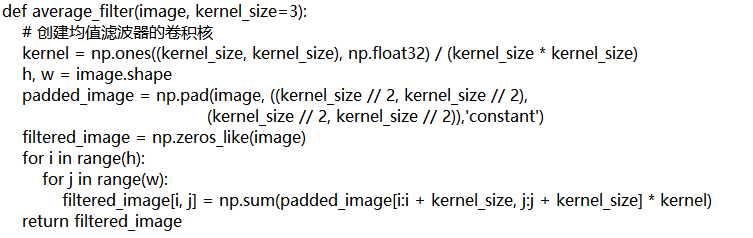


图 3- 1 均值滤波器关键代码

对比结果如图 3- 2所示：

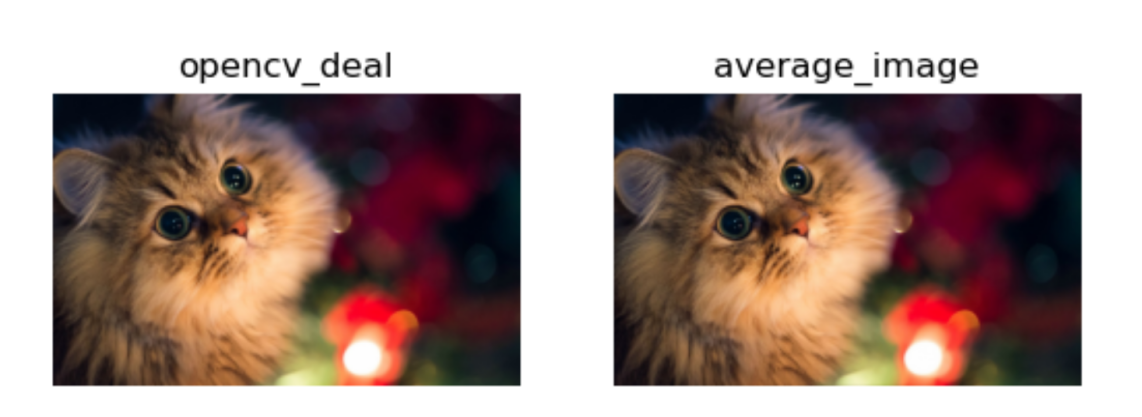


图 3- 2 均值滤波器与opencv库函数对比图

### 统计排序滤波器（中值滤波器）

统计排序滤波器通过取邻域内像素值的中值来替代中心像素值，能有效去除脉冲噪声。中值滤波器的实现如图 3- 3所示：

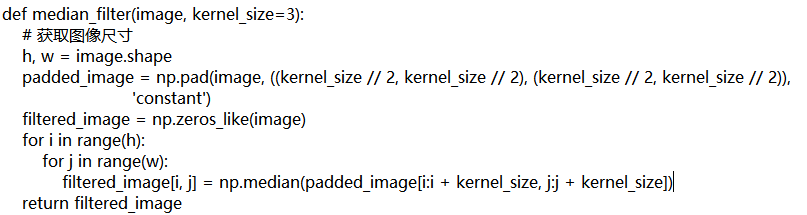


图 3- 3 中值滤波器关键代码

对比结果如下图 3- 4所示：



图 3- 4 中值滤波器与opencv库函数对比图

### Sobel滤波器

Sobel滤波器是一种用于边缘检测的图像处理算子。它使用两个3×3的卷积核分别计算图像在水平方向和垂直方向的梯度，通过这些梯度来突出图像中的边缘。水平方向的卷积核检测垂直边缘，垂直方向的卷积核检测水平边缘，最终结合这两个方向的结果即可得到图像的边缘信息。

Sobel滤波器的实现如图 3- 5所示：

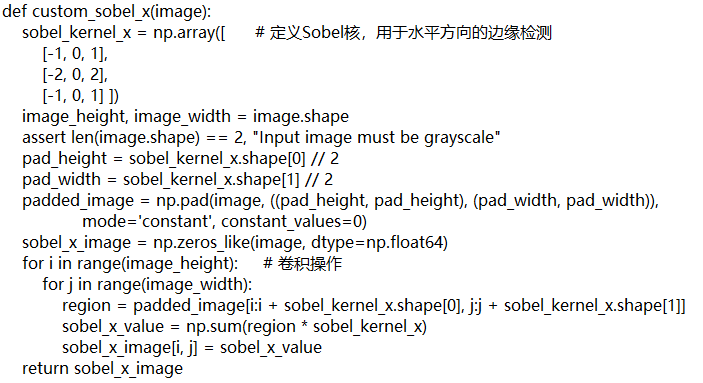


图 3- 5 Sobel滤波器关键代码

对比结果如下图所示：

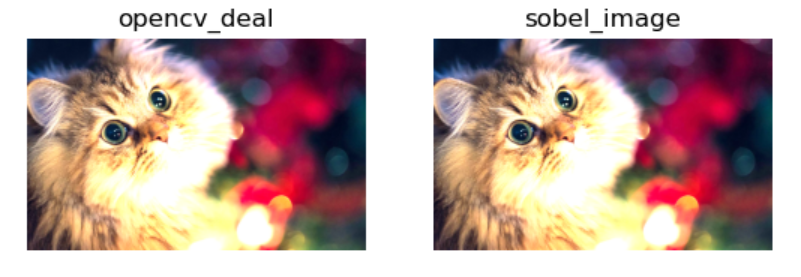


图 3- 6 Sobel滤波器与opencv库函数对比图

### 拉普拉斯算子

拉普拉斯算子通过计算图像的二阶导数来检测图像中的边缘。其卷积核如下所示：

 （3-2）

实现代码如图 3- 8所示：

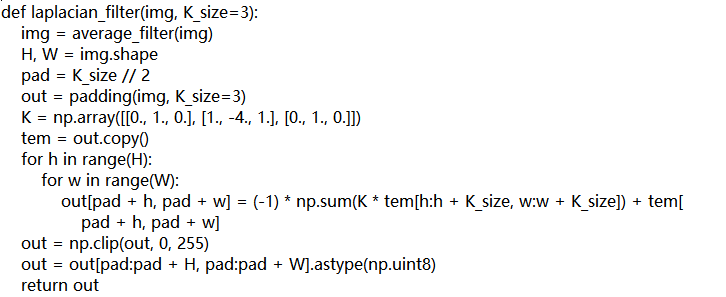


图 3- 7 Laplace算子锐化关键代码

对比结果如下图 3- 8所示：

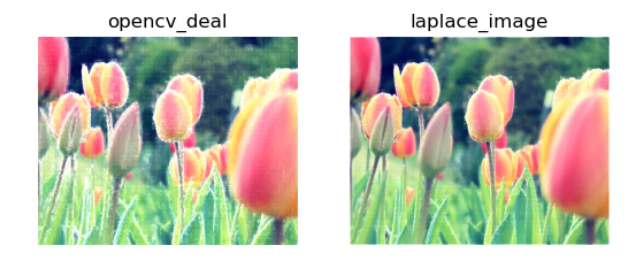


图 3- 8 Laplce算子锐化滤波器与opencv库函数对比图

### Roberts滤波器

Roberts滤波器是一种边缘检测算子，通过计算图像中相邻像素的对角差值来检测边缘。它使用两个2×2的卷积核，分别计算两个对角方向上的梯度，结合这些梯度可以突出图像中的边缘信息。这种方法简单且计算效率高，适用于实时边缘检测应用。

Roberts滤波器实现如图 3- 9所示：

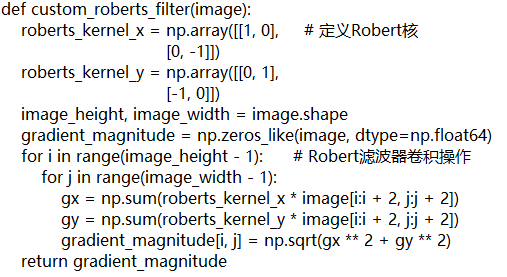


图 3- 9 Robert滤波器关键代码

对比结果如下图 3- 10所示：

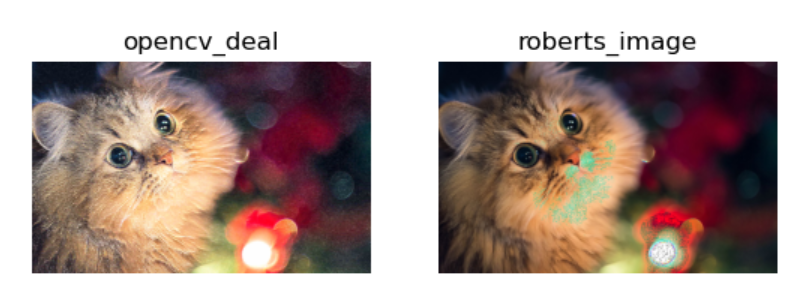


图 3- 10 Robert滤波器与opencv库函数对比图

# 空域滤波算子对不同噪声性能测试

## 噪声分析

### 高斯噪声

高斯噪声是一种常见的噪声类型，其特征是噪声值服从高斯（正态）分布。通常用于模拟图像传感器中的噪声和电子干扰。高斯噪声的数学模型可以表示为：

 （4-1）

其中，是带噪声的图像，是原始图像，是服从高斯分布的噪声。

在图像上添加高斯噪声会导致图像中每个像素值发生随机偏差，这种偏差服从高斯分布。高斯噪声常用于测试图像处理算法的鲁棒性和去噪效果。图像加噪效果如图 4- 1所示。



图 4- 1 高斯噪声效果图

### 椒盐噪声

椒盐噪声是一种在图像中随机分布的黑白点状脉冲干扰，其特征是黑色和白色像素的突然出现，用于测试图像处理算法的去噪性能，具有随机性和密度可控性，常见于模拟图像传输中的干扰情况。图像加噪效果如图 4- 2所示。



图 4- 2 椒盐噪声效果图

### 泊松噪声

泊松噪声是一种基于泊松分布的噪声类型，常用于模拟光子计数过程中的随机性，特别是在低光条件下的图像传感器中。泊松噪声遵循泊松分布，表现为随信号强度变化的随机波动。噪声的方差与信号强度成正比，信号强度越大，噪声越明显。在低光照条件下的图像传感器中常见，模拟光子计数的随机性。由于其依赖于信号强度，去除泊松噪声比其他类型噪声更为复杂。图像加噪效果如图 4- 3所示。



图 4- 3 泊松噪声效果图

## 各种滤波器对不同噪声类型处理效果

### 平滑线性滤波器（均值滤波器）效果

平滑线性滤波器，也称为均值滤波器，是一种常见的图像空域滤波算法，主要用于去除图像中的噪声和细节，使图像变得更加平滑和模糊。均值滤波器对不同噪声的处理效果图如图 4- 4、图 4- 5、图 4- 6所示。



图 4- 4 均值滤波器处理加噪图一

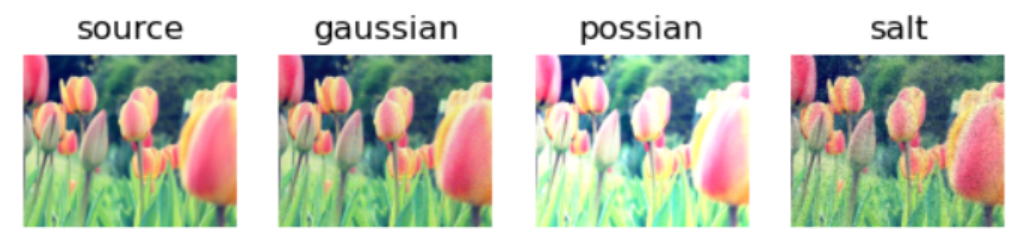


图 4- 5 均值滤波器处理加噪图二



图 4- 6 均值滤波器处理加噪图三

均值滤波器在处理高斯噪声时表现较好，因为高斯噪声的特点是随机分布且均值为零。通过对邻域像素进行平均，可以有效地减少这种随机噪声。处理高斯噪声后，图像会变得更平滑，但图像细节也会有所损失。

均值滤波器在处理椒盐噪声时效果不佳，因为椒盐噪声的值通常与邻域像素值有较大差异。简单的平均操作无法有效去除这种噪声，反而可能会导致噪声扩散。均值滤波器处理椒盐噪声后，图像会变得模糊，但噪声点可能仍然存在。

均值滤波器在处理泊松噪声时有一定效果，但不如高斯噪声处理效果好。由于泊松噪声的强度与信号强度相关，简单的均值操作不能完全去除噪声。均值滤波器能在一定程度上平滑泊松噪声，但也会导致图像细节的模糊。

### 统计排序滤波器（非线性滤波器）

中值滤波器是一种非线性滤波器，与平均滤波器不同，它不是简单地取邻域像素的平均值，而是选择邻域像素值的中值作为输出像素值。中值滤波器对不同加噪声图像的滤波效果图如图 4- 7、图 4- 8、图 4- 9所示。



图 4- 7 中值滤波器处理加噪图一

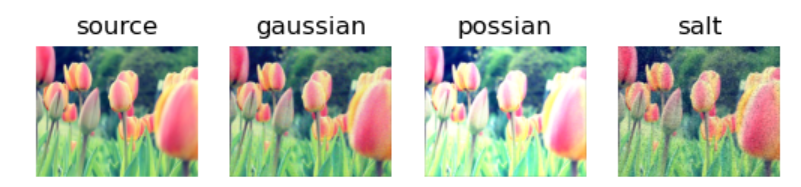


图 4- 8 中值滤波器处理加噪图二



图 4- 9 中值滤波器处理加噪图三

中值滤波器对高斯噪声的处理效果不如对椒盐噪声的处理效果好。尽管中值滤波器能在一定程度上减少高斯噪声，但它主要针对的是离群值，而高斯噪声是连续分布的随机噪声。中值滤波器在处理高斯噪声时能较好地保留图像的边缘和细节，这比均值滤波器表现得更好。

中值滤波器对椒盐噪声的处理效果非常好，因为这种噪声的像素值显著不同于其周围的像素值。中值滤波器能够有效地识别和去除这些异常值，同时保留图像的边缘和细节。

中值滤波器对泊松噪声的处理效果介于高斯噪声和椒盐噪声之间。由于泊松噪声的强度与信号强度相关，中值滤波器可以在一定程度上减少这种噪声，但效果不如椒盐噪声那样显著。

### Sobel滤波器

Sobel滤波器是一种常用的图像边缘检测滤波器，通过计算图像像素点的梯度来识别图像中的边缘。能够有效地帮助识别和定位图像中的边缘特征。Sobel滤波器对不同加噪图像的处理效果图如图 4- 10、图 4- 11、图 4- 12所示。



图 4- 10 Sobel滤波器处理加噪图一



图 4- 11 Sobel滤波器处理加噪图二



图 4- 12 Sobel滤波器处理加噪图三

Sobel滤波器对高斯噪声不太友好，因为它会计算图像的梯度，高斯噪声会被视为高频分量，从而被放大。图像的边缘和细节可能被噪声淹没。

Sobel滤波器对椒盐噪声也不太友好，因为这些噪声点会被视为强烈的梯度变化，从而被显著增强。由于噪声点的梯度值很大，它们会干扰图像的实际边缘检测结果，导致边缘检测的精度下降。因此，在应用Sobel滤波器之前，通常需要使用中值滤波器等方法去除椒盐噪声。

Sobel滤波器在处理泊松噪声时，同样会将噪声视为高频分量，从而放大这些噪声。

### Laplace算子锐化

**特点**: 拉普拉斯算子是一种常用的图像锐化算子，用于增强图像的边缘和细节，使图像看起来更加清晰和锐利。Laplace滤波器对不同加噪图像的处理效果图如图 4- 13、图 4- 14、图 4- 15所示。



图 4- 13 Laplace滤波器处理加噪图一

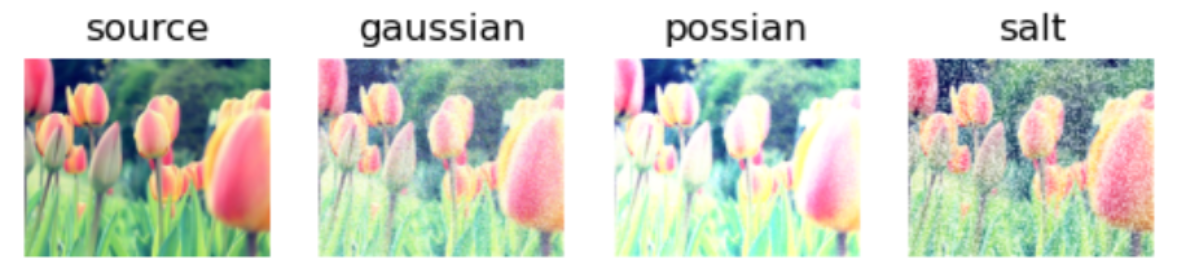


图 4- 14 Laplace滤波器处理加噪图二



图 4- 15 Laplace滤波器处理加噪图三

拉普拉斯算子会将高斯噪声视为高频成分，因此会显著放大这种噪声。这会导致图像中噪声显得更加明显，并干扰图像的边缘检测。拉普拉斯算子能增强图像中的边缘。

拉普拉斯算子对椒盐噪声的处理效果较差，因为这种噪声会被视为强烈的二阶梯度变化，从而被显著增强，导致图像中噪声点更加突出。椒盐噪声的放大会干扰图像的边缘检测结果，导致检测出的边缘不准确。

拉普拉斯算子在处理泊松噪声时，同样会将这种噪声视为高频成分，从而放大这些噪声。这会导致图像中的噪声显得更加明显，并影响边缘检测的准确性。

### Robert滤波器

Robert滤波器是一种边缘检测算子，用于识别图像中的边缘。它通过计算图像梯度的近似值来检测边缘，特别适合检测图像的细微边缘。Robert滤波器对加噪声图像的处理效果图如图 4- 16、图 4- 17、图 4- 18所示。



图 4- 16 Robert滤波器处理加噪图一

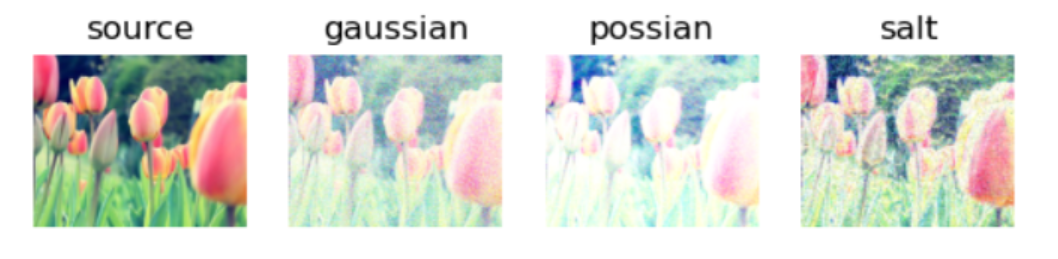


图 4- 17 Robert滤波器处理加噪图二



图 4- 18 Robert滤波器处理加噪图三

Roberts滤波器会将高斯噪声视为高频分量，从而放大这些噪声。Roberts滤波器仍能增强图像的边缘，但在存在高斯噪声时，边缘检测结果会受到噪声干扰。因此，建议在应用Roberts滤波器之前先进行高斯平滑滤波以减少噪声。

Roberts滤波器对椒盐噪声不太友好，因为这些噪声点会被视为强烈的梯度变化，从而被显著增强。图像中噪声点会变得更加突出。椒盐噪声的放大会干扰图像的实际边缘检测结果，导致边缘检测的精度下降。

Roberts滤波器在处理泊松噪声时，同样会将这种噪声视为高频分量，从而放大这些噪声。泊松噪声的放大会干扰图像的边缘检测结果。

# 总结

在现代数字信号处理领域，图像处理占据了重要的位置。随着各种图像获取设备和应用的普及，如何有效地处理和改善图像质量成为亟待解决的问题。本项目旨在研究几种常见的图像空域滤波算法，探讨它们在不同噪声环境下的性能表现，为后续图像处理工作提供技术支持。以下是我对设计此类算法的一些心得体会总结:

1. **理解设计背景与现实需求**

设计此类算法之前，需要深入理解空域滤波算法在现实世界中的应用场景，从实际出发，了解此类滤波器的应用目的，然后展开对算法的全面学习，这样会使我们更容易正确的理解此类滤波器设计的原理。

1. **代码编写能力**

算法的仿真时需要有一定的编码能力的。在设计过程中，会遇到许多算法原理之外的问题，一些编程语言特性导致的问题或是一些编程语言基础问题。例如，如何使用代码读取图片，如何使用代码展示图片。

1. **算法性能评估与优化：**

对设计的滤波算法进行系统的性能评估是必不可少的步骤。这包括量化算法在不同噪声环境下的处理效果，例如噪声去除能力和图像细节保留情况。通过量化和比较不同算法的性能，可以为选择最适合特定应用场景的滤波算法提供依据，并可能引发进一步的算法优化工作。

1. **应用实际与技术创新：**

将研究成果应用于实际图像处理任务中，并持续关注技术创新的发展，是项目的延续和拓展。利用新兴技术如深度学习结合空域滤波方法，可能能够进一步提升图像处理效率和质量，探索这些前沿技术的结合将为未来的研究和应用带来新的可能性。

在未来的工作中，可以进一步研究和优化滤波算法，提高其在噪声环境下的鲁棒性。此外，还可以结合频域滤波方法和机器学习技术，探索更为高效和智能的图像处理算法，为实际应用提供更优的解决方案。

# 参考文献

1. 赵辰裕.数字图像处理锐化技术的原理与实现[J].电子技术与软件工程,2018(04):60-62.
2. 关雪梅.基于Matlab的几种图像锐化处理算法研究[J].商丘师范学院学报,2018,34(12):12-14.
3. 王月新,刘明君.sobel算子与prewitt算子分析与研究[J].计算机与数字工程,2016,44(10):2029-2031+2042.
4. 袁修贵,孟正中,龚正.基于Sobel算子的小波去噪算法[J].电脑与信息技术,2009,17(05):1-3.DOI:10.19414/j.cnki.1005-1228.2009.05.001.
5. 彭宏,赵鹏博.边缘检测中的改进型均值滤波算法[J].计算机工程,2017,43(10):172-178.
6. 邢笑笑,李杰.空域滤波图像去噪算法研究[J].电子技术与软件工程,2022(16):144-147.
7. 楚智媛,吕闯.基于MATLAB下空域滤波算法的图像增强[J].中国科技信息,2020(19):91-92.
8. 郑丹,马尚昌,赵静.基于空域滤波的图像增强法的探讨[J].微型机与应用,2017,36(04):40-42+46.DOI:10.19358/j.issn.1674-7720.2017.04.012.
9. 王赛男.基于MATLAB下中值滤波算法的图像增强[J].智能计算机与应用,2014,4(06):87-89.

# 附录

import cv2  
import numpy as np  
def average(image):  
 blur\_image = cv2.blur(image, (5, 5))  
 return blur\_image  
def average\_self(image, kernel\_size=3):  
 # 创建均值滤波器的卷积核  
 kernel = np.ones((kernel\_size, kernel\_size), np.float32) / (kernel\_size \* kernel\_size)  
 # 获取图像尺寸  
 h, w, c = image.shape  
 # 为图像添加边界填充  
 padded\_image = np.pad(image, ((kernel\_size // 2, kernel\_size // 2),  
 (kernel\_size // 2, kernel\_size // 2),  
 (0, 0)), 'constant')  
 filtered\_image = np.zeros\_like(image)  
 for channel in range(c):  
 for i in range(h):  
 for j in range(w):  
 filtered\_image[i, j, channel] = np.sum(  
 padded\_image[i:i + kernel\_size, j:j + kernel\_size, channel] \* kernel)  
 return filtered\_image

import cv2  
import numpy as np  
  
  
def median(image):  
 blur\_image = cv2.medianBlur(image, 5)  
 return blur\_image  
  
  
def median\_self(image, kernel\_size=3):  
 # 获取图像尺寸  
 h, w, c = image.shape  
 # 为图像添加边界填充  
 padded\_image = np.pad(image, ((kernel\_size // 2, kernel\_size // 2),  
 (kernel\_size // 2, kernel\_size // 2),  
 (0, 0)), 'constant')  
 filtered\_image = np.zeros\_like(image)  
 for channel in range(c):  
 for i in range(h):  
 for j in range(w):  
 filtered\_image[i, j, channel] = np.median(padded\_image[i:i + kernel\_size, j:j + kernel\_size, channel])  
 return filtered\_image