

数字图像处理基础知识题

1、说出最感兴趣的图像处理应用及涉及的课程章节和知识点

我最感兴趣的图像处理应用是目标检测，在课程章节中对应第十四讲深度学习中的目标检测，相关知识点包括 RCNN 系列，FPN，YOLO 系列，SSD 系列等。

2、产生右图所示亮块图像 $f_1(x,y)$ (128×128 大小，暗处=0，亮处=255)，对其进行 FFT：



(1) 同屏显示原图 f_1 和 $\text{FFT}(f_1)$ 的幅度谱图：

(2) 若令 $f_2(x,y) = (-1)^{x+y} f_1(x,y)$ ，重复以上过程，比较二者幅度谱的异同，简述理由：

(3) 若将 $f_2(x,y)$ 顺时针旋转 45 度得到 $f_3(x,y)$ ，试显示 $\text{FFT}(f_3)$ 的幅度谱，并与 $\text{FFT}(f_2)$ 的幅度谱进行比较。

(1) 原图 f_1 和 $\text{FFT}(f_1)$ 的幅度谱图如图 1 所示。

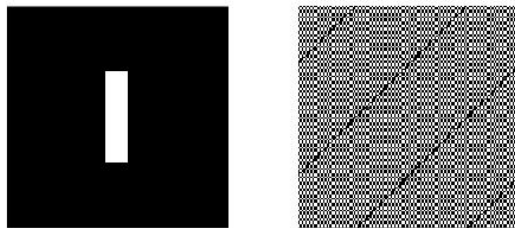


图 1. 原图 f_1 和 $\text{FFT}(f_1)$ 的幅度谱图

(2) 令 $f_2(x,y) = (-1)^{x+y} f_1(x,y)$ ， f_2 和 $\text{FFT}(f_2)$ 的幅度谱图如图 2 所示。

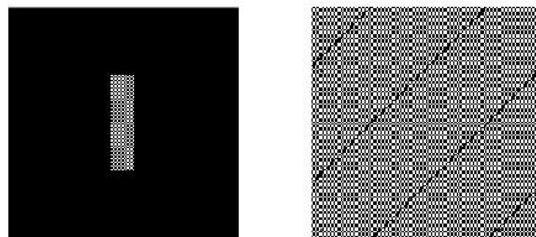


图 2. 原图 f_2 和 $\text{FFT}(f_2)$ 的幅度谱图

两者的异同：

不同点： f_2 的频谱是对 f_1 频谱的移位。

相同点：频谱的实质没有改变，幅度等都没有发生变化。

(3) 将 $f_2(x,y)$ 顺时针旋转 45 度得到 $f_3(x,y)$ ， $\text{FFT}(f_2)$ 的幅度谱和 $\text{FFT}(f_3)$ 的幅度谱如图 3 所示。

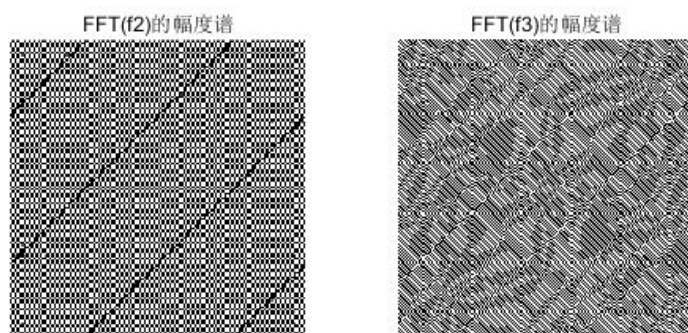


图 3. FFT(f2)的幅度谱图和 FFT(f3)的幅度谱图

- 3、对 512×512 大小、256 级灰度的数字图像 lena 进行频域的理想低通、高通滤波，同屏显示原图、幅度谱图和低通、高通滤波的结果图。

本题采用了高斯滤波器来对图像进行低通和高通滤波。滤波器公式如下：

低通滤波(D_0 为截止频率，本实验中取值为 30Hz)：

$$H(u,v) = e^{-\frac{D^2(u,v)}{2D_0^2}}$$

高通滤波(D_0 为截止频率，本实验中取值为 30Hz)：

$$H(u,v) = 1 - e^{-\frac{D^2(u,v)}{2D_0^2}}$$

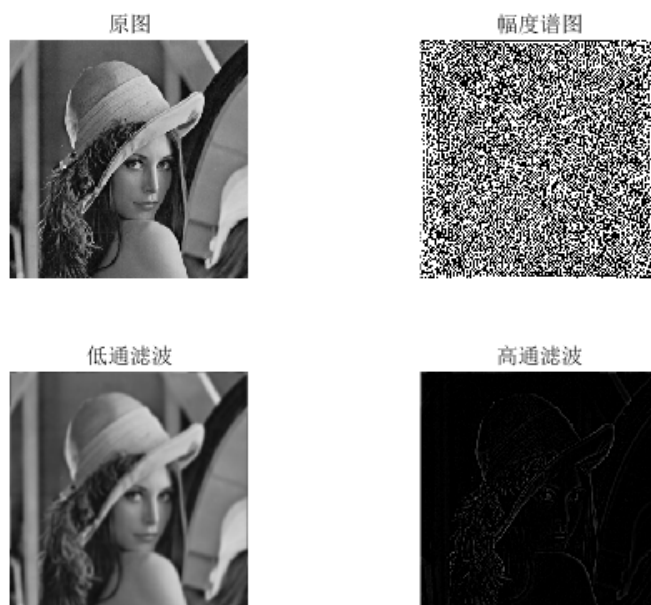


图 4. lena 原图、幅度谱图和低通、高通滤波结果图

如图所示，图像经过低通滤波后，图像主体仍然保留，但羽毛、轮廓等高频信息被滤除，图像经过高通滤波后，图像主体信息被滤除，图像轮廓信息被提取出来。

4、对图像 pollen 进行如下处理：

(1)使用直方图均衡化及直方图规定化，展示对比度拉伸的效果，要求先转化为灰度级 $L=8$ ，再按要求转化，写出详细的过程，其中规定直方图为 0,0,0,0.15,0.20,0.30,0.20,0.15

首先将图像 pollen 转化为灰度级 $L=8$ ，如图 5 所示：

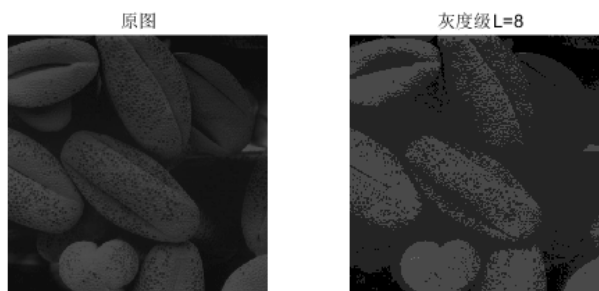


图 5. Pollen 原图和灰度级 $L=8$ 转换图

之后将其进行直方图均衡化和规定化，如图 6 所示：

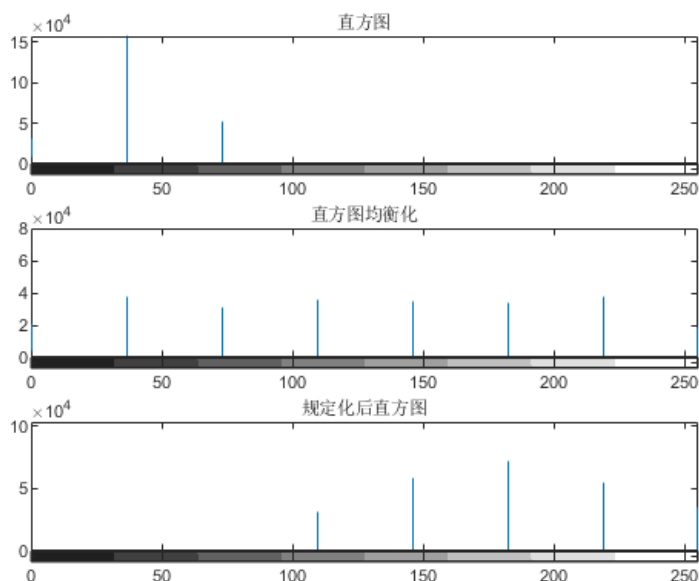


图 6. Pollen 直方图和均衡化、规定化直方图

从图中可以看出，原图中像素点大多为 $2/8$ 和 $3/8$ 灰度级，直方图均衡化之后，原图中像素点在各灰度级上均匀分布。直方图规定化之后，像素点的分布情况满足题目预设的分布条件。

(2)对直方图均衡化后的图像加入高斯噪声，用 4-邻域平均法和中值滤波平滑加噪声图像（图像四周边界不处理，下同），同屏显示原图像、加噪图像和处理后的图像。

① 不加门限；

② 加门限 $T = 2\overline{f(m,n)}$, (其中 $\overline{f(m,n)} = \frac{1}{N^2} \sum_i \sum_j f(i,j)$)

原图像、加噪图像、不加门限平滑后图像，加门限后平滑后图像如图 7 所示：

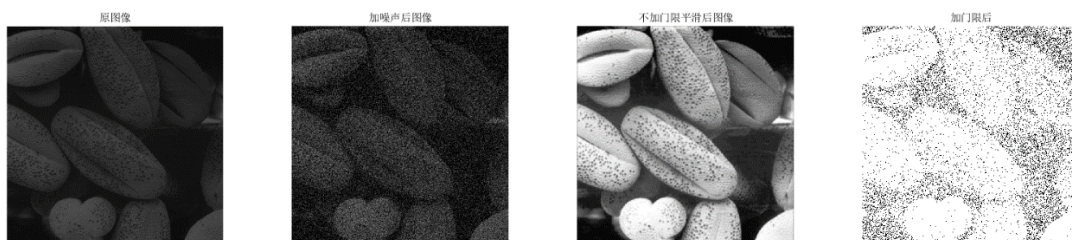


图 7. Pollen 原图、加噪声图像、处理后图像

5、(1) 用 Laplacian 锐化算子 (分 $\alpha = 1$ 和 $\alpha = 2$ 两种情况) 对 512×512 大小、256 级灰度的数字图像 lena 进行锐化处理，显示处理前、后图像。

处理前后的图像如图 8 所示：



图 8. Laplacian 锐化处理前后图像

从图中可以看出，当 $\alpha = 2$ 时，Laplacian 锐化的效果更为明显。

(2) 若令

$$g_1(m,n) = f(m,n) - \alpha \nabla^2 f,$$

$$g_2(m,n) = 4\alpha f(m,n) - \alpha[f(m-1,n) + f(m+1,n) + f(m,n-1) + f(m,n+1)]$$

则回答如下问题：

① $f(m,n)$ 、 $g_1(m,n)$ 和 $g_2(m,n)$ 之间有何关系？

三者的关系为： $g_1(m,n) = f(m,n) + g_2(m,n)$

② $g_2(m,n)$ 代表图像中的哪些信息？

$g_2(m,n)$ 代表了原图像中的二阶梯度信息

③ 由此得出图像锐化的实质是什么？

图像锐化的实质是将原图像与梯度信息叠加，对目标边缘进行增强。

6、分别利用 Roberts、Prewitt、Sobel、Laplacian、Canny 边缘检测算子，对 256 级灰度的自然图像 iris-Na、光学遥感图像 bridge-RS、SAR 图像 road-SAR 进行边缘检测，显示处理前、后图像，并阐述不同算法和数据源对结果的影响。

自然图像 iris-Na:

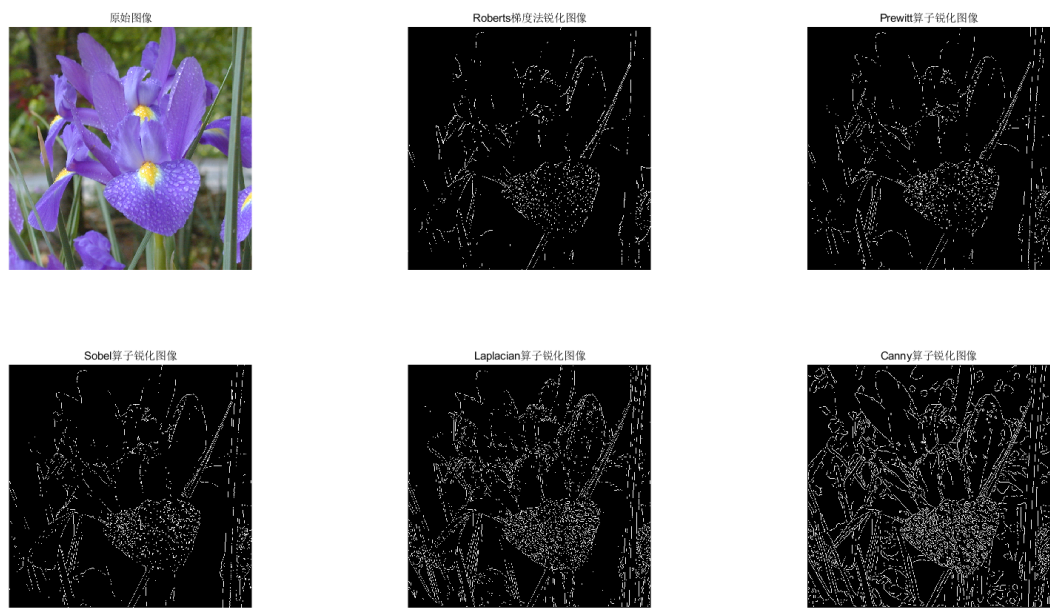


图 9. iris-Na 处理前后图像

光学遥感图像 bridge-RS:

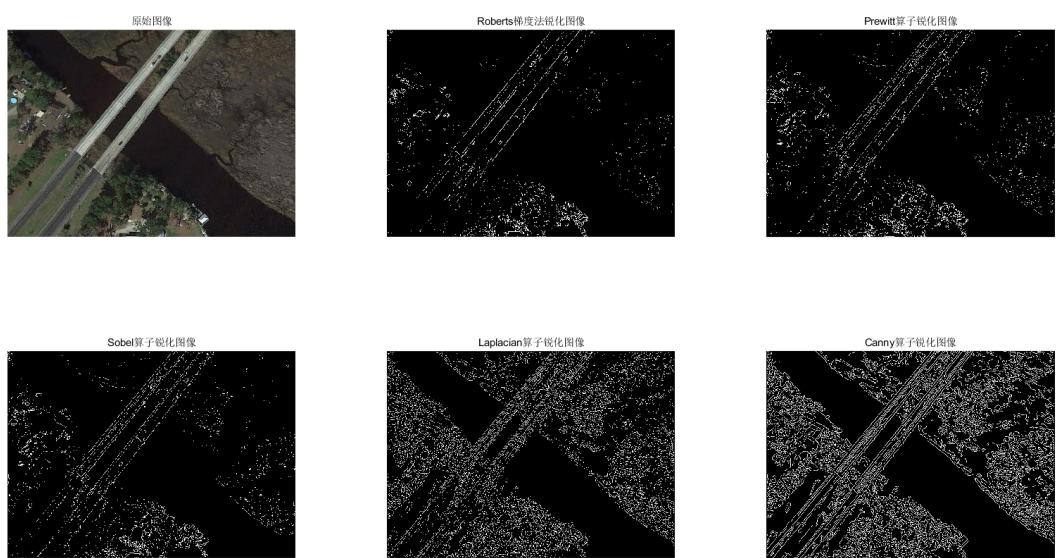


图 10. bridge-RS 处理前后图像

SAR 图像 road-SAR:

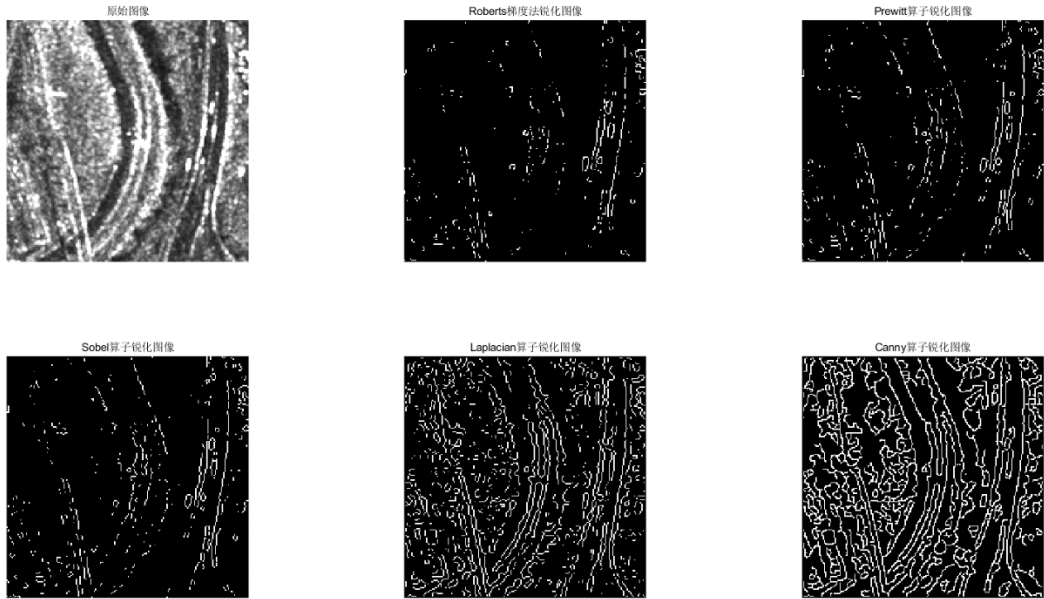


图 11. road-SAR 处理前后图像

不同算法结果分析:

Roberts 算子采用对角方向相邻两像素之差，也被称作 4 点差分法，对噪声较敏感。从三种图像上可以看出，使用 **Roberts** 算子进行边缘检测，检测出的结果较为稀疏，仅包含部分图像主题信息。

Prewitt 算子先求平均，再求差分来计算梯度，也被称作平均差分法。从三种图像上可以看出，使用 **Prewitt** 算子进行边缘检测，检测出的结果比 **Roberts** 算子的结果更加稠密一些，同时能够一定程度上抑制噪声的干扰。

Sobel 算子当前行或列对应的值加权后，再进行平均和差分，也被称作加权平均差分法。从三种图像上可以看出，使用 **Sobel** 算子结果和使用 **Prewitt** 算子的结果近似，但其边缘模糊的程度要略低于 **Prewitt** 算子。

Laplacian 算子是求像素点的二阶偏导，优点是各向同性、线性和唯一不变性；对孤立点及线段的检测效果好，缺点是对噪声敏感，对噪声有双倍加强作用；不能检测出边的方向，常产生双像素的边缘。相比与上述三种算子，**Laplacian** 算子得到的结果得到的细节信息更多，但同时对于噪声较多的 road-SAR 效果较差。

Canny 算子基本思想是找寻一幅图像中灰度强度变化最强的位置，首先它用高斯滤波来平滑图像，然后找寻图像的强度梯度，之后应用非最大抑制技术来消除边误检，最后应用双阈值的方法来检测和连接边界。从结果图上可以发现，**Canny** 算子检测的结果最为丰富，很多细节信息都被检测出来，同时由于它独特的处理方法，边缘部分比较清晰。

不同数据源分析:

本题中使用的三种数据源，格式并不相同。检测时首先将图像转换成灰度图，然后再

进行检测。其中，iris-Na 和 bridge-RS 图像的噪声较少，road-SAR 图像的噪声较多。

对于 iris-Na 图像，图像主体是鸢尾花，背景包含比较杂乱的枝叶信息。对于该图像，Roberts 算子，Prewitt 算子和 Sobel 算子都能较好地将图像主体边缘检测出来。Laplacian 算子和 Canny 算子在此基础上能够检测出更多的细节信息。

对于 bridge-RS 图像，Roberts 算子，Prewitt 算子和 Sobel 算子同样能够将图像的主体桥梁检测出来，但对于两岸细节信息较为忽略，Laplacian 算子和 Canny 算子能够捕捉到主体和细节信息。

对于 road-SAR 图像，由于包含了较多的噪声信息，Roberts 算子，Prewitt 算子的检测效果不太理想，而 Sobel 算子能够较少地受到噪声的影响，检测出图中两条公路的主体信息。Laplacian 算子和 Canny 算子都在一定程度上将噪声信息进行了放大增强，导致最终结果的可识别性不如 Sobel 算子。

7、学习数字图像处理课程的心得体会，该课程在哪些方面需要改进，对该课程或者任课老师有哪些意见或建议。

在本课程中，我学习到了传统图像处理的各种方式，并了解了深度学习在计算机视觉方面的各种应用。通过课程大作业，我对目标检测领域进行了进一步学习和探索，并利用 yolov5 模型解决了安全帽识别的实际问题。课程方面，我认为目前课程内容比较粗浅，我觉得老师可以在目前课程安排的基础上，深入讲解某一方面，并布置相应练习，以便学生更好地理解。

附录

问题二 matlab 代码:

```
% 生成第一问图像
f1=zeros(128,128);
for i=38:1:90
    for j=58:1:70
        f1(i,j)=255;
    end
end
% 第一小问
figure(1)
subplot(1,2,1);
imshow (f1);
```



```

subplot(1,2,2);
imshow (fft2(f1));
% 第二小问
for i=1:1:128
    for j=1:1:128
        f2(i,j)=(-1)^(i+j)*f1(i,j);
    end
end
figure(2);
subplot(1,2,1);
imshow (f2);
subplot(1,2,2);
imshow (fft2(f2));
% 第三小问
figure(3);
f3=imrotate(f2,-45,'bilinear');%将 f2 顺时针旋转 45 度
subplot(1,2,1);
imshow(fft2(f2));%显示 f2 的频谱
title('FFT(f2)的幅度谱');
subplot(1,2,2);
imshow(fft2(f3));%显示 f3 的频谱
title('FFT(f3)的幅度谱');

```

问题三 matlab 代码:

```

data=imread('D:\Desktop\计算机视觉\平时作业\img\lena.bmp');
figure(1);
subplot(2,2,1)
imshow(data);
title('原图');
subplot(2,2,2);
imshow(fft2(data));
title('幅度谱图');
%低通滤波
s=fftshift(fft2(data));
[M,N]=size(s);
n=2;
%GLPF 滤波 d0=30
d0=30;
n1=floor(M/2);           %对 M/2 进行取整
n2=floor(N/2);           %对 N/2 进行取整

```



```

for i=1:M
    for j=1:N
        d=sqrt((i-n1)^2+(j-n2)^2);          %点 (i,j) 到傅立叶变换中心的距离
        h=1*exp(-1/2*(d^2/d0^2));          %GLPF 滤波函数
        s(i,j)=h*s(i,j);                    %GLPF 滤波后的频域表示
    end
end
s=ifftshift(s);                             %对 s 进行反 FFT 移动
s=uint8(real(ifft2(s)));
subplot(2,2,3);
imshow(s);
title('低通滤波');
%高通滤波
p=fftshift(fft2(data));
[M,N]=size(p);                             %分别返回 p 的行数到 M 中，列数到 N 中
n=2;                                         %对 n 赋初值
%GHPF 滤波 d1=30
d1=30;
n3=floor(M/2);
n4=floor(N/2);
for i=1:M
    for j=1:N
        dd=sqrt((i-n3)^2+(j-n4)^2);        %点 (i,j) 到傅立叶变换中心的距离
        h1=1-exp(-1/2*(dd^2/d1^2));        %GHPF 滤波函数
        p(i,j)=h1*p(i,j);                  %GHPF 滤波后的频域表示
    end
end
p=ifftshift(p);                             %对 p 进行反 FFT 移动
p=uint8(real(ifft2(p)));
subplot(2,2,4);
imshow(p);
title('高通滤波');

```

问题四 matlab 代码:

% 第一小问

```

I=imread('D:\Desktop\计算机视觉\平时作业\img\pollen.tif');
figure(1);
subplot(1,2,1);
imshow(I);
title('原图');

```

```

% L=8 灰度级
[wid, hei] = size(I);
img8 = zeros(wid, hei);
for i=1:wid
    for j=1:hei
        img8(i,j)=floor(I(i,j)/32);
    end
end
subplot(1,2,2);
imshow(uint8(img8),[0,7]);
title('灰度级 L=8');
figure(2);
subplot(3,2,[1,2]);
imhist(I,8);
title('直方图');
J=histeq(I);
subplot(3,2,[3,4]);
imhist(J,8);
title('直方图均衡化');
subplot(3,2,[5,6]);
hgram=[0,0,0,0.15,0.20,0.30,0.20,0.15];
H=histeq(I,hgram);
imhist(H,8);
title('规定化后直方图');
% 第二小问
figure(3);
I1=imnoise(I,'gaussian'); %加高斯噪声
H1=[0 1 0;1 0 1;0 1 0]/4; %4×4 领域模板
J1=imfilter(J,H1); %领域平均
subplot(1,4,1);
imshow(I); %显示图像 I
title('原图像');
subplot(1,4,2);
imshow(I1);
title('加噪声后图像');
subplot(1,4,3);
imshow(J1);
title('不加门限平滑后图像');
%加门限后滤波
T = 2*sum(I1(:))/500^2;
% T= 0;

```

```

im_T = zeros(500,500);
for i = 1:500
    for j = 1:500
        if abs(I1(i,j) - J(i,j))>T
            im_T(i,j) = J(i,j);
        else
            im_T(i,j) = I1(i,j);
        end
    end
end
end
subplot(1,4,4);
imshow(im_T);
title('加门限后');

```

问题五 matlab 代码:

```

I=imread('D:\Desktop\计算机视觉\平时作业\img\lena.jpg');
figure(1);
subplot(1,3,1);imshow(I);title('原始图像');
L=fspecial('laplacian');
L1=[0 -1 0;-1 5 -1;0 -1 0];
L2=[0 -2 0;-2 9 -2;0 -2 0];
LP1=imfilter(I,L1,'replicate');%  $\alpha=1$  时的拉普拉斯算子
LP2=imfilter(I,L2,'replicate');%  $\alpha=2$  时的拉普拉斯算子
subplot(1,3,2);imshow(LP1);title('Laplacian 算子  $\alpha=1$  锐化图像');
subplot(1,3,3);imshow(LP2);title('Laplacian 算子  $\alpha=2$  锐化图像');

```

问题六 matlab 代码:

```

clear;
close all;
Img=imread('D:\Desktop\计算机视觉\平时作业\img\iris-Na.tif');
% Img=imread('D:\Desktop\计算机视觉\平时作业\img\bridge-RS.jpg');
% Img=imread('D:\Desktop\计算机视觉\平时作业\img\road-SAR.png');
I = rgb2gray(Img);
figure(1)
%Roberts 梯度法锐化
subplot(2,3,1);imshow(Img);title('原始图像');
S=edge(I,'Roberts');
subplot(2,3,2);imshow(S);title('Roberts 梯度法锐化图像');
%Prewitt 算子锐化

```

```
S=edge(I,'Prewitt');
subplot(2,3,3);imshow(S);title('Prewitt 算子锐化图像');
%Sobel 算子锐化
S=edge(I,'Sobel');
subplot(2,3,4);imshow(S);title('Sobel 算子锐化图像');
%Laplacian 算子锐化
S=edge(I,'log');
subplot(2,3,5);imshow(S);title('Laplacian 算子锐化图像');
%Canny 算子锐化
Cimg = edge(I,'canny');
subplot(2,3,6);imshow(Cimg);title('Canny 算子锐化图像');
```