**西安邮电大学**

**网络空间安全学院**

**实 验 报 告**

**（2022/ 2023 学年 第 2学期）**

实验名称： 信息隐藏与数字水印（课内实验）

学生姓名： 杨婉云

专业班级： 安全2003

学 号： 26201076（15）

指导教师： 任方

**西安邮电大学网络空间安全学院信息隐藏与数字水印**

**课内实验考核表**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **学生姓名** | **杨婉云** | **班级/学号** | | **安全2003/26201076** | |
| **承担任务实验室（单位）** | **信息安全与信息对抗实验教学中心** | **所在部门** | | **网安院信息安全系** | |
| **实施时间** | **2023年 6月 9 日 — 2023年 6 月 20 日** | | | | |
| **实验内容** | **实验4：DCT域图像隐藏实验**  **1．图像DCT域信息隐藏嵌入算法**  **2．图像DCT域信息隐藏提取算法。**  **实验5：数字水印技术实验**  **1．拼凑水印算法实验**  **2．扩频水印算法实验**  **实验6：信息隐藏与数字水印算法攻击实验**  **1．StirMark软件的使用**  **2. 信息隐藏和数字水印算法健壮性测试实验** | | | | |
| **指导教师（师傅）姓名** | **任方** | **职务或职称** | | **副教授** | |
| **指导教师**  **对学生的评价** | **评价点** | **评价内容** | | | **评价得分** |
| **平时成绩**  **（20分）** | **课堂出勤、课堂纪律及学习态度**  **遵守实验室的规章制度等情况** | | |  |
| **实验能力**  **（30分）** | **文件检索或软硬件设计能力**  **分析、解决实际问题能力** | | |  |
| **实验验收**  **（20分）** | **实验最终完成质量情况**  **实验演示与答辩情况** | | |  |
| **实验报告**  **（30分）** | **语言及文字表达能力**  **实验报告结构及书写格式规范** | | |  |
| **总评成绩** |  | | **五级制** |  | |
| **指导教师评语** | **指导教师签字**  **年 月 日** | | | | |

**实验4：DCT域图像隐藏实验**

**一．实验目的**

1．熟悉DCT域图像信息隐藏的基本原理；

2．用Matlab实现图像DCT域的信息隐藏与信息提取算法。

**二．实验设备**

1．PC机一台；

2．Matlab软件。

**三．实验内容**

1．DCT图像隐写实验。

1）提前准备载体图像（如lena600.bmp）和待隐藏信息（文件不超过0.3KB）；

2）编写DCT图像隐写算法，将待隐藏的信息隐写至载体图像，得到伪装图像；

代码如下：

图像随机像素位置生成函数Randinterval.m

function [row,col] =randinterval(matrix,count,key)

%注：本程序生成一个矩阵中的若干位置，均匀且随机，count表示需要的位置总数，key为随机种子

%最终的位置保存在row和col两个一维矩阵中，分别表示行号和列号

[m,n] = size(matrix) ;

interval1 = floor(m\*n/count) + 1;

interval2 = interval1-2;

if interval2 == 0

error( '载体太小，不能将秘密信息隐藏进去!') ;

end

% 生成随机序列

rand( 'seed', key) ;

a = rand( 1, count) ;

% 初始化

row = zeros( [ 1 count] ) ;

col = zeros( [ 1 count] ) ;

% 计算row 和col

r = 1;

c = 1 ;

row( 1, 1) = r;

col( 1, 1) = c;

for i = 2:count

if a(i)>=0.5

c = c + interval1;

else

c = c + interval2;

end

if c>n

k=mod(c,n);

r=r+floor((c-k)/n);

if r>m

error( '载体太小不能将秘密信息隐藏进去!') ;

end

c=k;

if c==0

c = 1;

end

end

row( 1, i) = r ;

col( 1, i) = c;

end

DCT信息隐藏算法

frr = fopen('msg-hh.txt','r');

[msg,count] = fread(frr,'ubit1') ;

fclose(frr) ;

data0 = imread('lena600.bmp') ;

% 将图像矩阵转为double 型

data = double(data0) /255;

% 对图像分块

T = dctmtx( 8) ;

% 对分块图像做DCT 变换

IMDCT=blkproc(data,[8 8 ],'P1\*x\*P2',T,T') ;

% 产生随机的块选择, 确定图像块的首地址

[ row, col] = size( IMDCT) ;

row = floor(row/8) ;

col = floor(col/8) ;

a = zeros( [ row col] ) ;

[k1,k2] = randinterval(a,count,12) ; %此处引用程序1给出的函数，生成随机像素位置

for i = 1:count %计算随机块的位置

k1(1,i) = (k1(1,i)-1)\*8+1;

k2(1,i) = (k2(1,i)-1)\*8+1;

end

% 信息嵌入

temp=0;

H=0.2; %控制参数，保证嵌入结果的正确性。

for i=1:count

if msg(i,1)==0

if IMDCT(k1(i)+4,k2(i)+1)>IMDCT(k1(i)+3,k2(i)+2)

temp=IMDCT(k1(i)+4,k2(i)+1);

IMDCT(k1(i)+4,k2(i)+1)=IMDCT(k1(i)+3,k2(i)+2);

IMDCT(k1(i)+3,k2(i)+2)=temp;

end

else

if IMDCT(k1(i)+4,k2(i)+1)<IMDCT(k1(i)+3,k2(i)+2)

temp=IMDCT(k1(i)+4,k2(i)+1);

IMDCT(k1(i)+4,k2(i)+1)=IMDCT(k1(i)+3,k2(i)+2);

IMDCT(k1(i)+3,k2(i)+2)=temp;

end

end

if IMDCT(k1(i)+4,k2(i)+1)>IMDCT(k1(i)+3,k2(i)+2)

IMDCT(k1(i)+3,k2(i)+2)=IMDCT(k1(i)+3,k2(i)+2)-H; % 将小系数调整更小

else

IMDCT(k1(i)+4,k2(i)+1)=IMDCT(k1(i)+4,k2(i)+1)-H;

end

end

result = blkproc(IMDCT,[8 8],'P1\*x\*P2',T',T);

imwrite(result,'lenadct.bmp') ;

subplot(2,2,1); imshow(data); title('原始图像') ; % 显示实验结果

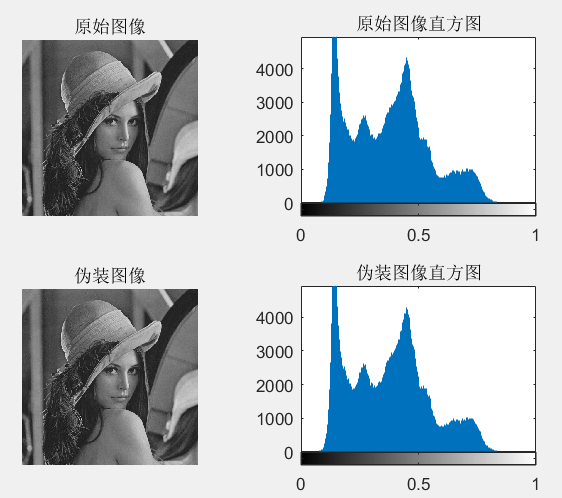
subplot(2,2,2); imhist(data); title('原始图像直方图') ;

subplot(2,2,3);imshow(result); title('伪装图像') ;

subplot(2,2,4);imhist(result); title('伪装图像直方图');

3）在视觉上对载体图像和伪装图像进行比较，进一步用差异图、直方图等方式对二者进行更详细的比较和分析。

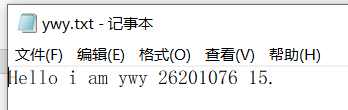
得到输出图像：



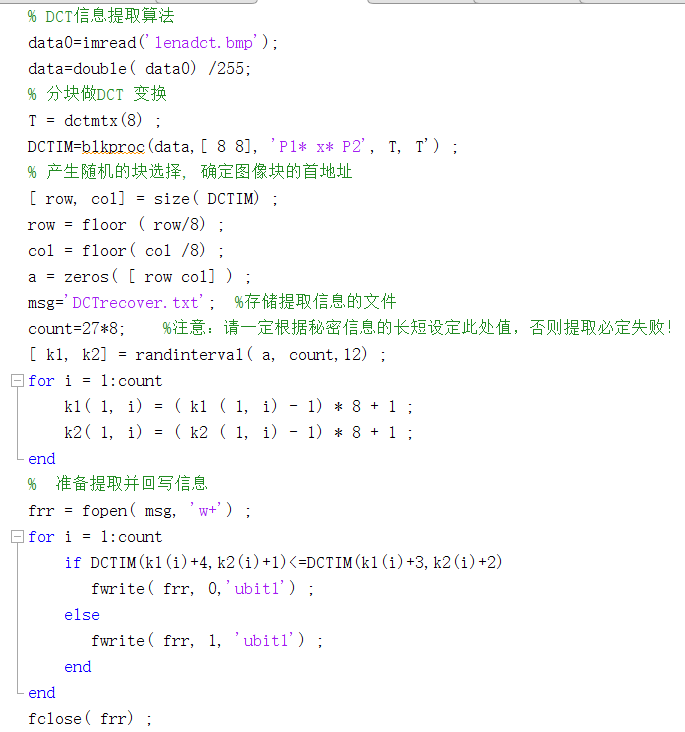
2．DCT图像隐写的提取实验。

1）编写DCT图像隐写的提取算法，提取上面所得到伪装图像中的隐藏信息，将其与隐写之前的信息进行比较；

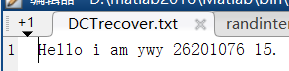
隐藏信息为：



DCT信息提取算法代码如下：



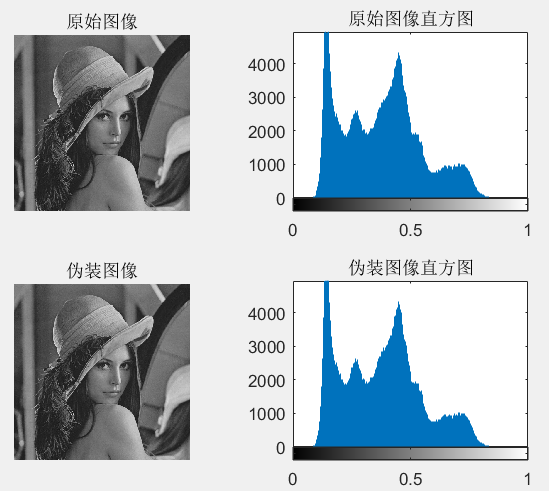
提取出的隐藏信息如下：

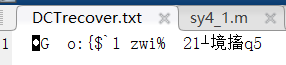


（2）选择不同的控制参数H，比较隐写效果与信息提取的正确性之间的关系，总结H在DCT隐写与提取算法中的作用。

选择H=0.2时，为上述提取结果。

修改H=0.001，隐藏结果及提取结果如下：





可见H越小，信息提取正确性越低。

**四、实验总结及心得体会**

DCT图像信息隐写算法是一种常见的隐写术，它利用频域转换技术，将秘密信息嵌入到图像的高频系数中，这些高频系数通常对人眼不太敏感，因此可以在不引起明显视觉变化的情况下隐藏更多的信息，所以具有较高的隐写容量。与DCT算法相比，LSB算法的隐写容量较低，仅能隐藏少量的秘密信息。该算法将秘密信息直接嵌入到图像的最低有效位中，因此每个像素只能隐藏一个比特的信息。

DCT算法在安全性方面相对较强。它采用频域转换，并将秘密信息嵌入到高频系数中，这些系数通常受到压缩算法和图像处理操作的影响，从而增加了隐写信息的安全性。LSB算法在安全性方面相对较弱。由于秘密信息直接嵌入到图像的最低有效位，攻击者可以通过对图像进行统计分析或采用图像处理技术来检测到隐藏的信息。

DCT算法对一些常见的图像处理操作和压缩算法具有一定的健壮性。高频系数相对于低频系数更容易受到图像处理操作的影响，因此DCT算法对于一些简单的图像处理操作具有较好的健壮性。LSB算法中，由于秘密信息直接嵌入到像素的最低有效位，只有对像素进行较大的修改才能破坏隐藏的信息。因此，LSB算法在一些简单的图像处理操作和无损压缩算法下表现较好。

通过本次实验，我熟悉了DCT域图像信息隐藏的基本原理，明白了H在DCT隐写与提取算法中的作用（H越小，信息提取正确性越低），比较了DCT图像信息隐写算法和的LSB算法的性能如隐写容量、安全性、健壮性等。

**实验5：数字水印技术实验**

**一．实验目的**

1．熟悉数字水印技术的基本原理；

2．熟练使用Matlab实现常见的数字水印算法。

**二．实验设备**

1．PC机一台；

2．Matlab软件。

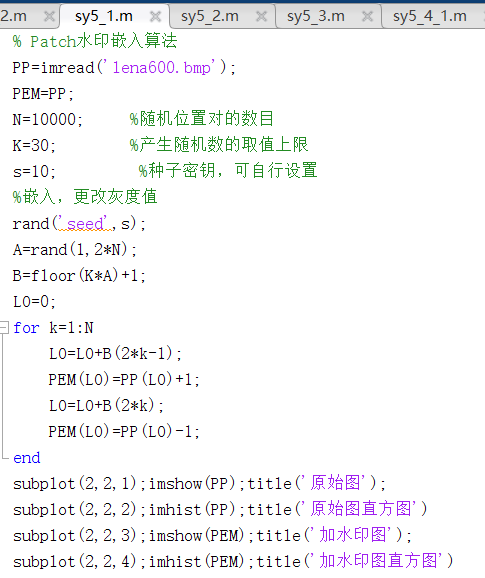
**三．实验内容**

1．拼凑水印算法实验

1）提前准备载体图像lena600.bmp；

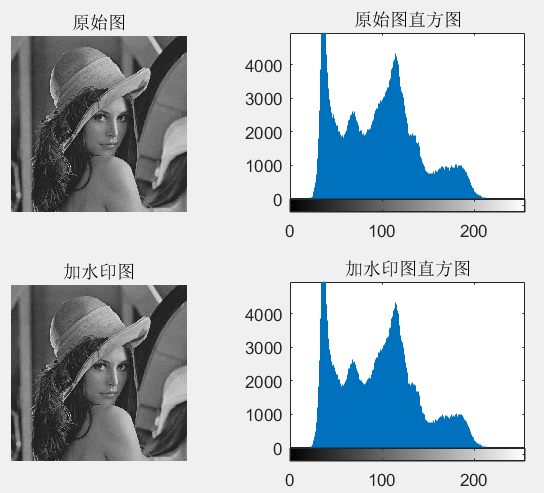
2）编写拼凑水印嵌入程序，输入原始图像及用户随机选择的私钥，输出加水印后的图像；

拼凑水印嵌入算法代码如下：



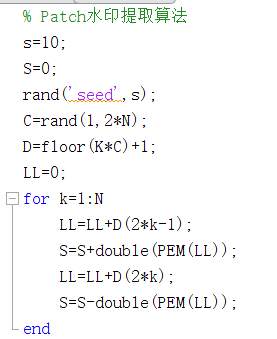
3）在视觉上对原始图像和加水印后的图像进行比较，进一步用差异图、直方图等方式对二者进行更详细的比较和分析；

输出结果如下：



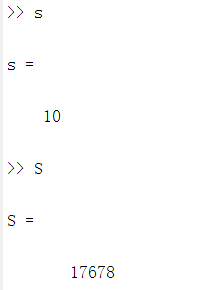
4）编写水印恢复算法，比较输入正确私钥和错误私钥时所恢复的不同水印结果。

拼凑水印恢复算法代码如下：



5）更改用户私钥进行多次重复实验，观察实验结果。尝试分析并总结拼凑算法的本质以及它抵抗攻击的能力。

输出结果：



2．扩频水印算法实验

1）提前准备载体图像（灰度图）和水印图像（二值图）；

2）编写扩频水印嵌入算法，将水印图像嵌入至载体图像中，得到加水印后的图像；

扩频水印嵌入算法代码如下：

cover = imread('lena600.bmp') ;

ste\_cover = cover;

ste\_cover = double( ste\_cover ) ;

N=3; %衡量随机公钥与随机私钥序列的相似度：1/N

k1=6; %片率，偶数，需要被N所整除！

% 判断嵌入消息量是否过大

wat=imread('xiyou2.bmp');

[r,t]=size(wat);

[m,n]=size(ste\_cover ) ;

if r\*t\*k1>m\* n

error('嵌入消息量过大, 请更换图像') ;

end

len\_total=r\*t;

water=double(wat);

%水印转为正负1

for k=1:len\_total

if water(k)==0

water(k)=-1;

end

end

%扩频部分

bi=zeros(k1\*len\_total,1);

for i=1:len\_total

for j=1:k1

bi((i-1)\*k1+j,1)=water(i);

end

end

%伪随机序列生成，充满整个载体

rand('seed',10);

A=rand(m\*n,1);

B=2\*floor(2\*A)-1; %私钥序列

rand('seed',20);

BB=rand(m\*n,1);

BB=2\*floor(2\*BB)-1; %公钥序列

for i=1:m\*n;

if mod(i,N)==0

BB(i,1)=B(i,1);

end

end

%准备嵌入的lsb信息

lsb=zeros(m\*n,1);

for i=1:len\_total\*k1

lsb(i,1)=(bi(i,1)\*B(i,1)+1)/2;

end

for i=len\_total\*k1+1:m\*n

lsb(i,1)=B(i,1);

end

%嵌入水印到LSB

for i=1:m\*n

ste\_cover(i)=bitand(ste\_cover(i),254)+lsb(i,1);

end

ste\_cover=uint8(ste\_cover);

% 显示实验结果

subplot(2,2,1); imshow( cover); title('原始图像') ;

subplot(2,2,2); imhist( cover); title('原始图像直方图') ;

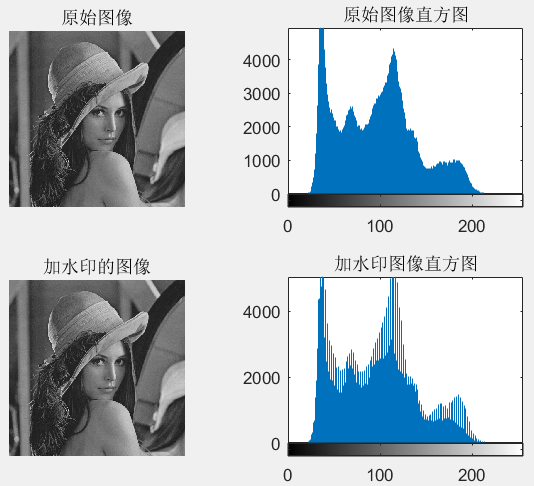
subplot(2,2,3);imshow( ste\_cover); title('加水印的图像') ;

subplot(2,2,4);imhist( ste\_cover); title('加水印图像直方图') ;

imwrite(ste\_cover,'lena\_kuopin.bmp');

3）在视觉上对原始图像和加水印后的图像进行比较，进一步用差异图、直方图等方式对二者进行更详细的比较和分析；

输出结果如下：



4）编写水印提取算法，用私钥和公钥两种算法分别提取水印，查看提取结果并进一步与原始水印图进行比较，分析提取算法的性能。

扩频水印提取算法代码如下：

用私钥提取：

s=zeros(len\_total,1);

for j=1:len\_total

s(j,1)=0;

for i=(j-1)\*k1+1:j\*k1

s(j,1)=s(j,1)+(double(bitand(ste\_cover(i),1))\*2-1)\*B(i,1);

end

s(j,1)=s(j,1)/k1;

end

%最终显示恢复的二值图

www=zeros(r,t);

www=logical(www);

for i=1:len\_total

if s(i,1)==1

www(i)=1;

else

www(i)=0;

end

end

imtool(www);

用公钥提取：

s=zeros(len\_total,1);

for j=1:len\_total

s(j,1)=0;

for i=(j-1)\*k1+1:j\*k1

s(j,1)=s(j,1)+(double(bitand(ste\_cover(i),1))\*2-1)\*BB(i,1);

end

s(j,1)=(s(j,1)\*N)/k1;

end

%最终显示恢复的二值图

www=zeros(r,t);

www=logical(www);

for i=1:len\_total

if s(i,1)==1

www(i)=1;

else

www(i)=0;

end

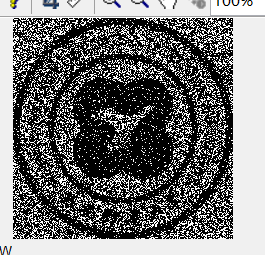
end

imtool(www);

私钥提取结果如下：



公钥提取结果如下：



**四、实验总结及心得体会**

拼凑算法通常用于嵌入二进制信息，例如数字水印或隐藏的标识符。它将信息嵌入到图像或音频的最低有效位（LSB）中，通过修改最不显著的像素或样本来实现。扩频水印算法可以嵌入更多种类的信息，包括文本、图像、音频等。它通过将原始信息与伪随机扩频序列相乘，将信息扩展到更宽的频带上。这种算法利用了信号处理中的频域扩展原理。

对于拼凑算法，恢复的结果度量通常以检测率和误检率为指标。检测率是指成功提取出嵌入的信息的比例，而误检率是指错误地将非嵌入信息识别为嵌入信息的比例。而扩频水印算法的恢复结果度量可以根据具体应用情况选择不同的指标。例如，在图像水印中，可以使用峰值信噪比（PSNR）来度量提取出的水印与原始水印之间的相似度。

拼凑算法对一些简单的图像处理操作（如亮度/对比度调整）具有一定的健壮性，但对于更复杂的图像处理操作和压缩算法可能不太稳定。因为拼凑算法直接修改最低有效位，所以稍微的修改或压缩都可能破坏水印的嵌入和提取。扩频水印算法相对于拼凑算法具有更好的健壮性。由于扩频水印算法将水印信号扩展到较宽的频带上，并采用了伪随机序列，即使在一些图像处理操作和压缩算法的影响下，仍能提取出较好质量的水印信息。

通过本次实验，我熟悉了数字水印技术的基本原理，学会了使用Matlab实现常见的数字水印算法，加深了对拼凑算法和扩频水印算法的理解，对这两种算法的性能如嵌入的信息类型、恢复的结果度量、算法的健壮性有了一定了解。

**实验6：信息隐藏与数字水印算法攻击实验**

**一．实验目的**

1．熟悉StirMark软件的基本功能和使用方法；

2．结合StirMark，用Matlab软件实现常见信息隐藏与数字水印算法的健壮性测试实验。

**二．实验设备**

1．PC机一台；

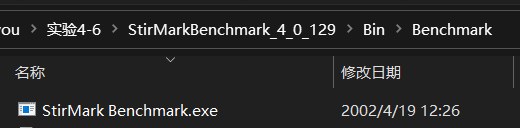
2．Matlab软件；

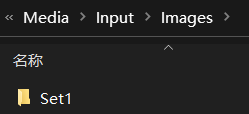
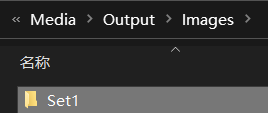
3．StirMark软件。

**三．实验内容**

1．安装并运行StirMark软件。

解压缩包，找到可执行程序Bin\Benchmark\Stirmark Benchmark.exe。输入文件夹为Media\Input\Images\Set1，相应的输出文件夹Media\Output\Images\Set1。

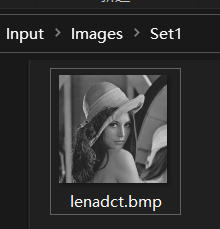
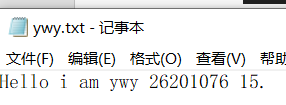


2．利用StirMark和Matlab实现对信息隐藏和数字水印算法的健壮性测试

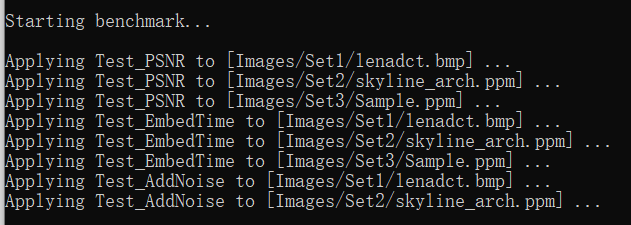
1）选择一个之前做过的信息隐藏或者数字水印算法，运行得到隐藏信息或者加了水印后的图像，将其作为待检测图像；

本次实验中选择使用DCT信息隐藏算法后得到的图像作为待检测图像：隐藏信息为ywy.txt：

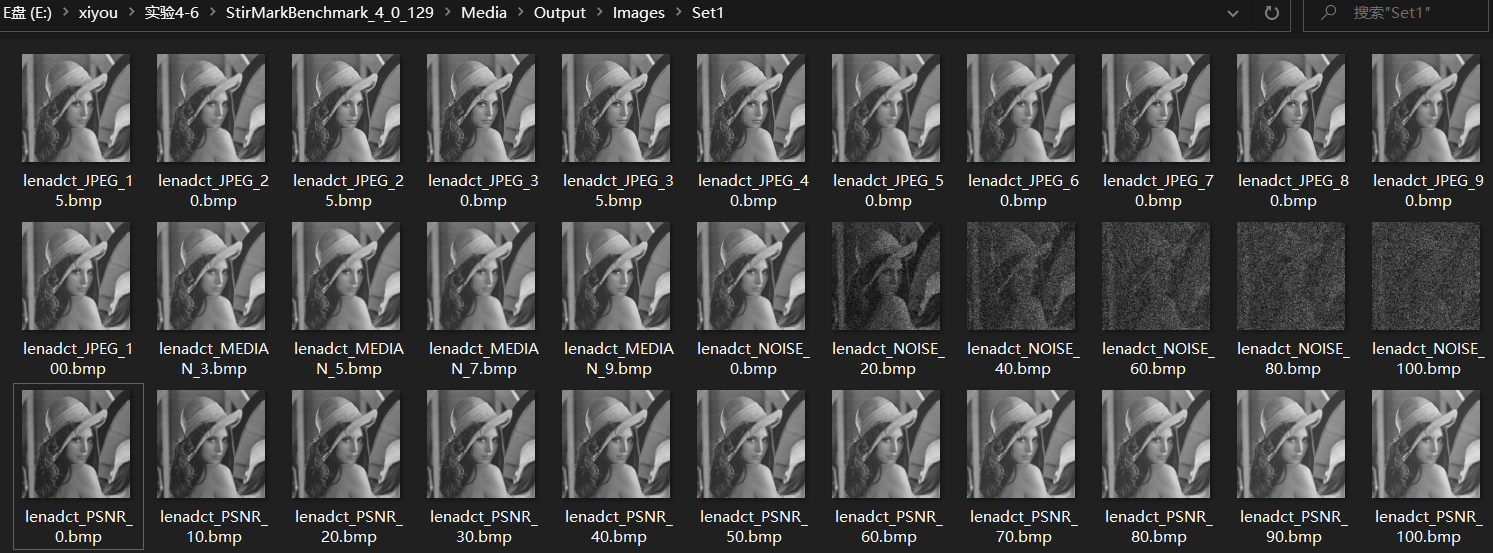
 

2）将待检测图像放在输入文件夹，运行StirMark得到多幅输出图像；

程序运行过程如下：

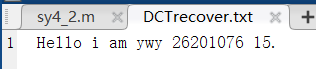


输出图像如下：

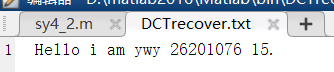


3）选择多幅输出图像执行1）中的隐藏信息提取或者水印提取算法，观察提取结果；

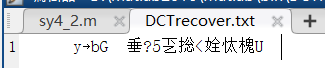
lenadct\_JPEG\_15.bmp



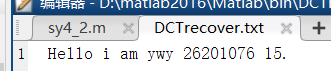
lenadct\_JPEG\_100.bmp



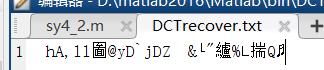
lenadct\_NOISE\_20.bmp



lenadct\_PSNR\_0.bmp



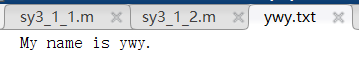
lenadct\_MEDIAN\_9.bmp

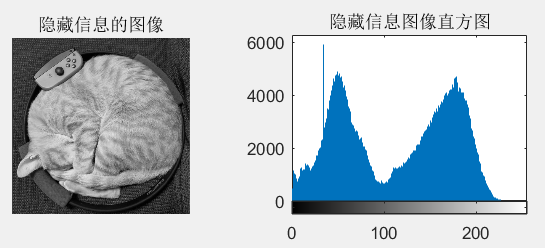
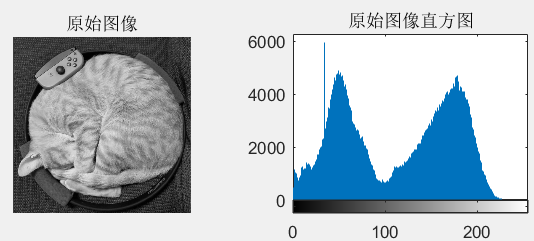


4）分析提取结果，说明该算法能抵抗对图像的哪些类型修改，其健壮性究竟如何。

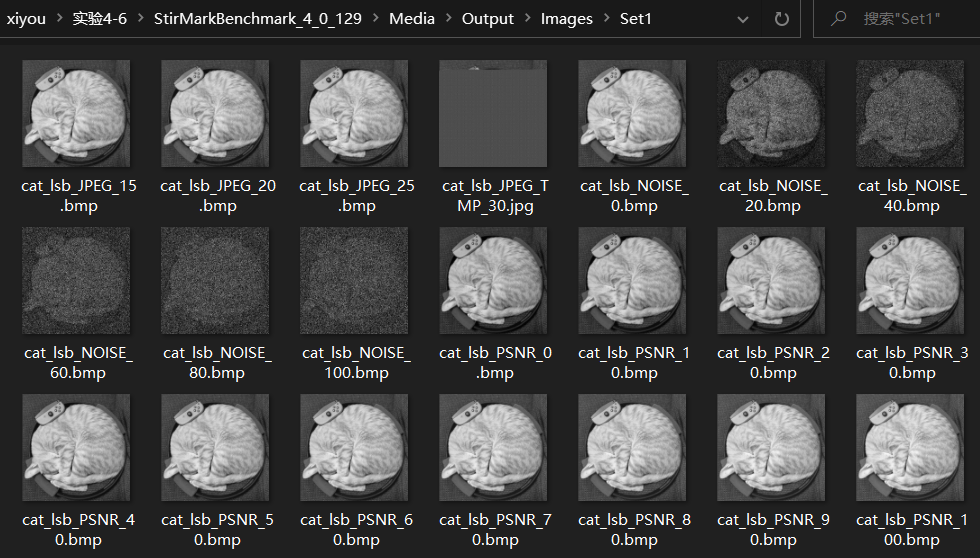
3．选择不同的隐藏或者水印算法（如LSB提取算法）再次实验，分析这些算法的健壮性。

秘密信息、原始图像、嵌入秘密信息后的图像：

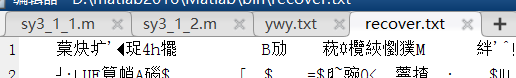
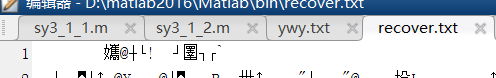




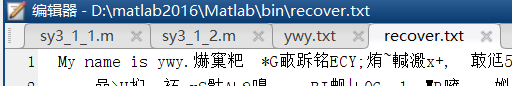
运行StirMark得到多幅输出图像：



对多幅图片使用LSB提取算法：





最终使用cat\_lsb\_PSNR\_40.bmp恢复出了秘密信息。

**四、实验总结及心得体会**

StirMark是最早的也是最常用的水印健壮性测试工具。它通过对给定的一幅加了水印的图像进行测试，生成许多修改后的图像，以此来验证嵌入的水印是否能被检测到。

在本实验中，通过观察测试提取结果，我发现DCT算法能抵抗对图像的PSNR修改，但无法抵抗NOISE修改和MEDIAN修改，它对于一些简单的图像处理操作具有较好的健壮性，在遇到更复杂的操作或压缩算法（如NOISE、MEDIAN）时，DCT算法的健壮性会下降。

而LSB算法能抵抗PSNR修改，对于其他几种修改无法抵抗，可见LSB算法在一些简单的图像处理操作和无损压缩算法下健壮性好，但对于JPEG、NOISE、MEDIAN以及更复杂的图像处理操作健壮性较差。

通过此次实验，我熟悉了StirMark软件的基本功能和使用方法，并结合StirMark，用Matlab软件实现常见信息隐藏与数字水印算法的健壮性测试，对不同算法的健壮性也有了一定了解。