武汉大学国家网络安全学院

信息隐藏实验报告

学　　号　2021302181156

姓　　名　赵伯侯

实验名称　安全隐写对抗技术

指导教师　任延珍

一、 实验名称: 数 字 水 印

二、 实验目的:

1. 实现 W-SVD 数字水印的嵌入算法与检测算法

2. 使用 stirmark 工具攻击嵌入 W-SVD 水印的图片，对攻击后的图片进行水印检测，评价水印算法的鲁棒性

3. 分析不同算法参数情况对水印性能的影响

三、 实验原理:

（一）小波变换

小波变换是将傅里叶变换的基（无限长的三角函数基）换成了有限长的会衰减的小波基。这样不仅能够获取频率，还可以定位到时间了。在小波变换中运用到的小波如下图所示

$$WT(a,\tau)=\frac{1}{\sqrt{a}}\int_{-\infty}^{\infty}f(t)*\psi(\frac{t-\tau}{a})dt$$

图 1: 小波变换函数

在该函数中小波变换有两个变量：尺度 a（scale）和平移量 τ（translation）。尺度 a 控制小波函数的伸缩，平移量 τ 控制小波函数的平移。尺度就对应于频率（反比），平移量 τ 就对应于时间。

图像的小波变换是小波应用于图像处理的基础，且基于二维离散小波变换。图像可以看作是二维的矩阵，每次小波变换后，图像便分解为 4 个大小为原来尺寸 1/4 的子块区域，每一个子块区域包含了相应频带的小波系数，相当于在水平方向和竖直方向上进行隔点采样。进行下一层小波变换时，变换数据集中在 LL 频带上。小波系数的空间分布同原始图像的空间分布具有很好的对应关系:

1.LL 频带是图像内容的缩略图，它是图像数据能量集中的频带。

2.HL 频带存放的是图像水平方向的高频信息，它反映了图像水平方向上的变

化信息和边缘信息。

3.LH 频带存放的是图像竖直方向的高频信息，它反映了图像在竖直方向上的灰度变化信息和图像边缘信息。

4.HH 频带存放的是图像在对角线方向的高频信息，它反映了水平方向和竖直方向上图像灰度的综合变化信息，同时包含了少量的边缘信息。

所以对图像进行一级小波分解和二级小波分解得到的图像的系数分布如下图所示

图 2: 一级小波变换系数分布

图 3: 二级小波变换系数分布

（二）数据归一化

数据的归一化一般是将数据映射到指定的范围，用于去除不同维度数据的量纲以及量纲单位。常见的映射范围有 $[0,1]$ 和 $[-1,1]$ 。

本次实验中采用的归一化方法就是 Min-Max 归一化。也称为离差标准化，是对原始数据的线性变换，使结果值映射到 $[0,1]$ 之间。转换函数为 $xnew = \frac{x-xmin}{xmax-xmin}$

其中 max 为样本数据的最大值，min 为样本数据的最小值。这种归一化方法比较适用在数值比较集中的情况。但是，如果 max 和 min 不稳定，很容易使得归一化结果不稳定，使得后续使用效果也不稳定，实际使用中可以用经验常量值来替代 max 和 min。而且当有新数据加入时，可能导致 max 和 min 的变化，需要重新定义。

（三）奇异值分解生成水印模板

对于任意 M*N 矩阵 A，都可以写成 $A = U \times \sum \times V^T$, 其中 U 和 V 分别是 M*M 和 N*N 的正交矩阵,Σ 是 M*N 的对角矩阵，这种变换成为矩阵 SVD 变换，U 和 V 中分别称作 A 的奇异向量，Σ 是 A 的奇异值。

在本次实验中对图像做小波分解得到其低频系数 LL，然后对 LL 做单值分解 $LL = U \sum V^T$, 用伪随机序列生成两个正交矩阵然后用生成的两个正交矩阵的后 d 列替换掉 U 和 V 的后 d 列，得到 $\widetilde{U}$ 和 $\widetilde{V}$

然后用伪随机数生成器生成一个对角矩阵后乘上 alpha（强度因子）得到新的对角矩阵 $\widetilde{\sum}$

最后能够得到水印模板 watermark=$\widetilde{U} \times \widetilde{\sum} \times \widetilde{V}$

算法的流程如下图所示

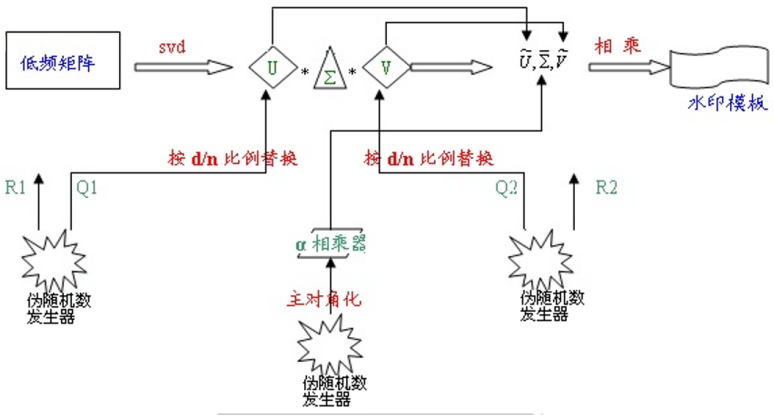

图 4: SVD 生成水印模板流程

（四）数字水印检测

利用原始图像生成一个理论上存在的水印模板（原始水印），从待检测图像中提取可能存在的水印模板（待测水印），计算两者的相关性。当两者高度相关时，待测图像含有水印；反之检测不出水印。

原始水印可以通过首先提取出原图的小波低频系数 A, 然后再提取出将原图进行嵌入水印后提取加有水印的图像的小波系数 B 通过计算可得原始水印为 =A-B，然后提取待检测图像的低频小波系数 C, 计算待测水印 =C-B，即可得到原始水印和

待测水印。

在比较两个水印的相关性时有两种方法：1. 常规检测直接相关性值。借助公式 $d = \frac{\left|\sum_{i=1}^{M}\sum_{j=1}^{N} WijVij\right|}{\sqrt{\sum_{i=1}^{M}\sum_{j=1}^{N} WijVij} * \sqrt{\sum_{i=1}^{M}\sum_{j=1}^{N} WijVij}}$ 可以求出原始水印和待测水印的相关性的值，在该公式中，W 和 V 代表两个水印的矩阵，M 和 N 分别代表其大小。

若两个水印的相关性越强，则得到的 d 越趋近于 1，反之则越小

2.DCT 域相关性值，首先将原始水印和待测水印进行 DCT 域的变换后得到的两个矩阵同样利用上述公式计算 d 的值，得到的 d 的值越趋近于 1 则两个水印的相关性越强。算法的流程如下图所示

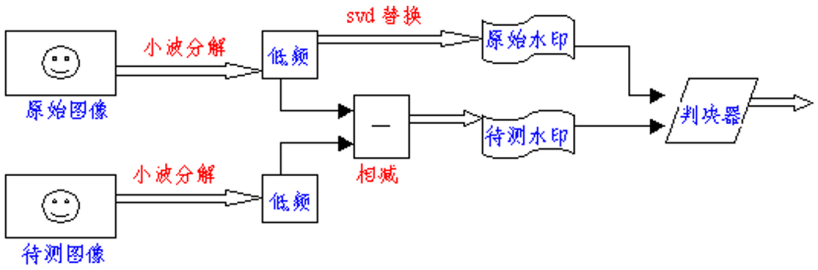

图 5: W-SVD 水印检测原理

四、 实验内容:

（一）W-SVD 水印生成及嵌入

实现对于输入的图像进行 W-SVD 嵌入水印操作的代码如下所示

```
%水印嵌入
%W_SVD_hide("Lena.jpg",'db6',2,0.99,10,0.5)    调用例子
function W_SVD_hide(image_path,wavelet,N,ratio,seed,alpha)

% image_path="Lena.jpg";
% wavelet='db6';      %使用的小波函数
% N=2;                 %小波分解的层数
% ratio=0.99;
% seed=10;
% alpha=0.5;
```

```
11
12 %规定文件的输出路径以及stirmark指定文件夹的路径
13 output_path='lenna-waved.jpg';
14 target_path='E:\matlab_codes\lab_3\stirmark\Media\Input\Images\Set1';
15
16 %取出图像的三个色域，只取R层进行水印嵌入
17 image=imread(image_path);
18 image= double(image)/255;
19 R=image(:,:,1);
20 G=image(:,:,2);
21 B=image(:,:,3);
22
23 %对图像的R层进行小波分解，得到C为各层分解系数,s_real为未补足的各层分解系数长度
24 [C,S_real]=wavedec2(R,N,wavelet);
25
26 %将原始图像补足成正方形
27 [high,width]=size(R);
28 max_size=max(high,width);
29 cover=zeros(max_size);
30 if high <= width
31     cover( 1:high,:) = R;
32 else
33     cover(:,1:width) = R;
34 end
35
36 %补足后的图像小波分解
37 [C,S]=wavedec2(cover,N,wavelet);
38 %提取N级低频成分保存到CA中
39 CA = appcoef2(C,S,wavelet,N);
```

```
40
41 %对低频小波系数归一化处理，对原始数据的线性变换，使结果值映射到[0 - 1]之间
42 CAmin = min(min(CA)) ;
43 CAmax = max(max(CA)) ;
44 CA=(1/(CAmax-CAmin))*(CA-CAmin);
45
46 d=max(size(CA));
47 %对低频系数进行单值分解
48 [U,sigma,V]=svd(CA);
49
50 %计算水印容量,替换的比例
51 np = round(d*ratio);
52
53 %按照种子生成随机矩阵
54 rand('seed', seed)
55
56 %生成两个随机正交矩阵Q_V,Q_U
57 M_V = rand( d, np) - 0.5;        %rand生成的随机矩阵均值为0.5
58 [ Q_V, R_V] = qr( M_V, 0) ;      %将矩阵M分解成正交矩阵Q和上三角矩阵R
59 M_U = rand( d, np) - 0.5;
60 [ Q_U, R_U] = qr( M_U, 0) ;
61
62 %将分解得到的矩阵后面的部分替换
63 V2 = V;
64 U2 = U;
65 V(:, d - np +1: d) = Q_V(:, 1: np) ;
66 U(:, d - np +1: d) = Q_U(:, 1: np) ;
67
68 %生成随机对角矩阵
```

```
sigma_tilda = alpha* flipud(sort( rand( d, 1))) ;

% 生成水印模板
watermark = U* diag( sigma_tilda, 0) * V ;

% 调整系数生成嵌入水印后的图像，将过幅系数与负数修正
CA_tilda = CA + watermark;
over1 = find( CA_tilda > 1) ;
below0 = find( CA_tilda < 0) ;
CA_tilda(over1) = 1;
CA_tilda(below0) = 0;

%系数还原到归一化以前的范围
CA_tilda = ( CAmax-CAmin) * CA_tilda +CAmin;

%加有水印的低频系数
waterCA = CA_tilda;

%只选取有信息的部分
if high <= width
    waterCA = waterCA(1:S_real(1,1),:) ;
else
    waterCA = waterCA(:,1:S_real(1,2) ) ;
end

% 重构带有水印的低频系数矩阵并用其替换C
CA_tilda = reshape( CA_tilda, 1, S( 1, 1) * S( 1,2) ) ;
C( 1, 1:S( 1, 1) * S( 1, 2) ) = CA_tilda;

```

```
%将带有水印的C进行逆小波变换，还原成原图像的格式
watermarkimage = waverec2( C, S, wavelet) ;

% 将前面补上的边缘去掉
if high <= width
    watermarkimage = watermarkimage( 1:high,:) ;
else
    watermarkimage = watermarkimage(:, 1:width) ;
end

%得到嵌入水印后的图像并写入目标位置
watermarkimage_rgb=cat(3,watermarkimage,G,B);
imwrite( watermarkimage_rgb, output_path) ;

%将嵌入水印后的图像复制到stirmark工具对应文件夹中
copyfile(output_path,target_path);

%输出结果
figure(1) ;
subplot( 223) ; imshow( watermark*255) ; title('水印模板') ;
subplot( 221) ; imshow(image) ; title( '原始图像') ;
subplot( 222) ; imshow( watermarkimage_rgb) ; title( '嵌入水印后的图像') ;
```

代码 1: W-SVD 水印生成及嵌入

（二）**W-SVD** 水印的检测

实现对于待检测图像的水印检测操作的代码如下所示

```
%计算原图与待测图片的相关性
%W_SVD_find('Lena.jpg','lenna-waved.jpg','db6',2,0.99,10,0.5)
function [corr_coef,corr_DCTcoef]=W_SVD_find(image_origin_path,image_test_path,
    wavelet,N,ratio,seed,alpha)

% image_origin_path='Lena.jpg';
% image_test_path='lenna-waved.jpg';
% wavelet='db6';      %使用的小波函数
% N=2;                 %小波分解的层数
% ratio=0.99;
% seed=10;
% alpha=0.5;
%提取原图的R层
image_origin=imread(image_origin_path);
image_origin=double(image_origin)/255;
R_origin=image_origin(:,:,1);

%提取测试图像的R层
image_test=imread(image_test_path);
image_test=double(image_test)/255;
R_test=image_test(:,:,1);

%提取出原图带有水印的小波系数
[waterCA_origin]=W_SVD_get_waterCA(image_origin_path,wavelet,N,ratio,seed,alpha
    );

%提取原图的小波系数
```

```
[C_origin,S_origin]=wavedec2(R_origin,N,wavelet);
CA_origin = appcoef2(C_origin,S_origin,wavelet,N);
%提取待测图像的小波系数
[C_test,S_test]=wavedec2(R_test,N,wavelet);
CA_test = appcoef2(C_test,S_test,wavelet,N);

%生成原始水印
watermark_origin=waterCA_origin-CA_origin;
%提取出待测水印
watermark_test=CA_test-CA_origin;

%计算相关性
corr_coef=trace(watermark_origin'*watermark_test)/(norm(watermark_origin,'fro')
    *norm(watermark_test,'fro'));

%取原图和待检测图像的DCT系数
DCTrealwm=dct2(watermark_origin);
DCTtestwm=dct2(watermark_test);

%取出原图和待检测图像32*32块的DCT系数
DCTrealwm=DCTrealwm(1:min(32,max(size(DCTrealwm))),1:min(32,max(size(DCTrealwm)
    )));
DCTtestwm=DCTtestwm(1:min(32,max(size(DCTtestwm))),1:min(32,max(size(DCTtestwm)
    )));

%计算相关性
corr_DCTcoef=trace(DCTrealwm'*DCTtestwm)/(norm(DCTrealwm,'fro')*norm(DCTtestwm,
    'fro'));
```

代码 2: W-SVD 水印检测

该函数调用了获取原图加水印后的低频小波系数的函数 W_SVD_get_waterCA 代码如下所示

```
%取水印嵌入函数的前部分，但不将水印嵌入后的图片写入文件只得到小波系数
function [waterCA]=W_SVD_get_waterCA(image_path,wavelet,N,ratio,seed,alpha)

% image_path="Lena.jpg";
% wavelet='db6';      %使用的小波函数
% N=2;                 %小波分解的层数
% ratio=0.99;
% seed=10;
% alpha=0.5;

image=imread(image_path);
image= double(image)/255;
R=image(:,:,1);

%将原始图像补足成正方形
[high,width]=size(R);
max_size=max(high,width);
cover=zeros(max_size);
if high <= width
    cover( 1:high,:) = R;
else
    cover(:,1:width) = R;
end

%补足后的图像小波分解
[C,S]=wavedec2(cover,N,wavelet);
%提取N级低频成分保存到CA中
```

```
CA = appcoef2(C,S,wavelet,N);

%对低频小波系数归一化处理，对原始数据的线性变换，使结果值映射到[0 - 1]之间
CAmin = min(min(CA)) ;
CAmax = max(max(CA)) ;
CA=(1/(CAmax-CAmin))*(CA-CAmin);

d=max(size(CA));
%对低频系数进行单值分解
[U,sigma,V]=svd(CA);

%计算水印容量,替换的比例
np = round(d*ratio);

%按照种子生成随机矩阵
rand('seed', seed)

%生成两个随机正交矩阵Q_V,Q_U
M_V = rand( d, np) - 0.5;        %rand生成的随机矩阵均值为0.5
[ Q_V, R_V] = qr( M_V, 0) ;      %将矩阵M分解成正交矩阵Q和上三角矩阵R
M_U = rand( d, np) - 0.5;
[ Q_U, R_U] = qr( M_U, 0) ;

%将后面的部分替换
V2 = V;
U2 = U;
V(:, d - np +1: d) = Q_V(:, 1: np) ;
U(:, d - np +1: d) = Q_U(:, 1: np) ;

```

```
%生成随机对角矩阵
sigma_tilda = alpha* flipud(sort( rand( d, 1))) ;

% 生成水印模板
watermark = U* diag( sigma_tilda, 0) * V ;

% 调整系数生成嵌入水印后的图像，将过幅系数与负数修正
CA_tilda = CA + watermark;
over1 = find( CA_tilda > 1) ;
below0 = find( CA_tilda < 0) ;
CA_tilda(over1) = 1;
CA_tilda(below0) = 0;

%系数还原到归一化以前的范围
CA_tilda = ( CAmax-CAmin) * CA_tilda +CAmin;

%加有水印的低频系数
waterCA = CA_tilda;
```

代码 3: 获取原图加水印后的低频小波系数

（三）SC 图绘制

实现在 W-SVD 水印检测过程中绘制 SC 图的代码如下所示

```
%%批量计算各个参数的不同所绘制出的相关性SC图
%原图路径和待测图像路径
image_origin_path='Lena.jpg';
image_test_path='lenna-waved.jpg';

%初始化检测参数
seed_real=10;
N_real=2;
ratio_real=0.99;
wavelet_real='db6';
alpha_real=0.5;

N=N_real;
ratio=ratio_real;
wavelet=wavelet_real;
alpha=alpha_real;

%% 分析种子值的不同
%选定种子值的选取范围
seeds=[2,4,6,8,10,12,14,16,18,20];

%保存检测结果
corr_seed=[];
corr_DCT_seed=[];

%对每一个选定的种子值计算相关性并保存
for index=1:10
```

```
    seed=seeds(index);
    [corr,corr_DCT]=W_SVD_find(image_origin_path,image_test_path,wavelet,N,
        ratio,seed,alpha);
    corr_seed=[corr_seed,corr];
    corr_DCT_seed=[corr_DCT_seed,corr_DCT];
end

%调用函数绘制SC图
disp_xy(seeds,corr_seed,corr_DCT_seed,'关于种子的小波系数阈值分析','种子',0);
%恢复种子值
seed=seed_real;

%% 分析小波变换次数的不同
Ns=[1,2,3,4,5,6,7,8];
corr_N=[];
corr_DCT_N=[];

for index=1:8
    N=Ns(index);
    [corr,corr_DCT]=W_SVD_find(image_origin_path,image_test_path,wavelet,N,
        ratio,seed,alpha);
    corr_N=[corr_N,corr];
    corr_DCT_N=[corr_DCT_N,corr_DCT];
end

disp_xy(Ns,corr_N,corr_DCT_N,'关于小波变换次数的小波系数阈值分析','小波变换次数
    ',0);
N=N_real;

```

```
%% 分析替换比例的不同
ratios=[0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,0.99];
corr_ratio=[];
corr_DCT_ratio=[];

for index=1:10
    ratio=ratios(index);
    [corr,corr_DCT]=W_SVD_find(image_origin_path,image_test_path,wavelet,N,
        ratio,seed,alpha);
    corr_ratio=[corr_ratio,corr];
    corr_DCT_ratio=[corr_DCT_ratio,corr_DCT];
end

disp_xy(ratios,corr_ratio,corr_DCT_ratio,'关于替换比例的小波系数阈值分析','替换
    比例',0);
ratio=ratio_real;

%% 分析使用小波函数种类的不同
wavelets=["haar","db1","db6","db10","db45","coif1","coif2"];
corr_wavelet=[];
corr_DCT_wavelet=[];

for index=1:7
    wavelet=wavelets(1,index);
    [corr,corr_DCT]=W_SVD_find(image_origin_path,image_test_path,wavelet,N,
        ratio,seed,alpha);
    corr_wavelet=[corr_wavelet,corr];
    corr_DCT_wavelet=[corr_DCT_wavelet,corr_DCT];
end
```

```
80
81  disp_xy(wavelets,corr_wavelet,corr_DCT_wavelet,'关于小波函数种类的小波系数阈值
        分析','小波函数种类',1);
82  wavelet=wavelet_real;
83
84  %% 分析生成随机对角矩阵参数的不同
85  alphas=[0,0.1,0.2,0.3,0.4,0.5,0.6];
86  corr_alpha=[];
87  corr_DCT_alpha=[];
88
89  for index=1:7
90      alpha=alphas(index);
91      [corr,corr_DCT]=W_SVD_find(image_origin_path,image_test_path,wavelet,N,
            ratio,seed,alpha);
92      corr_alpha=[corr_alpha,corr];
93      corr_DCT_alpha=[corr_DCT_alpha,corr_DCT];
94  end
95
96  disp_xy(alphas,corr_alpha,corr_DCT_alpha,'关于生成随机对角矩阵参数的小波系数阈
        值分析','生成随机对角矩阵参数',0);
```

代码 4: 绘制每个参数的 SC 图

该函数调用了绘制单个 SC 图的函数的代码如下所示

```
1  %绘制SC图
2  function disp_xy(x,y,y_DCT,top_title,x_label,type)
3  if(type==0)
4      figure;
5      x_max=max(x);
6      x_inc=x(2)-x(1);
7      plot(x,y,'-*b',x,y_DCT,'-or');
```

```
8       axis([0,x_max,0,1])
9       set(gca,'XTick',[0:x_inc:x_max])
10      set(gca,'YTick',[0:0.1:1])
11      legend('相关系数','DCT相关系数');
12      title(top_title)
13      xlabel(x_label)
14      ylabel('相关性')
15  elseif(type==1)
16      figure;
17      X=1:7;
18      plot(X,y,'-*b',X,y_DCT,'-or');
19      xlim([0 7])
20      set(gca,'xtick',1:7)
21      set(gca,'xticklabel',x)
22      xtickangle(45);
23      set(gca,'YTick',[0:0.1:1])
24      legend('相关系数','DCT相关系数');
25      title(top_title)
26      xlabel(x_label)
27      ylabel('相关性')
28  end
```

代码 5: 绘制单个 SC 图

（四）**stirmark** 工具攻击评价算法鲁棒性

实现自动检测攻击后生成的待检测图片的代码如下图所示

```
%对于嵌入水印后的图片攻击后，对于每种攻击方式进行分析
%原图路径
origin_path='Lena.jpg';

%设置每一种攻击方式中的下标
number1=[0,20,40,60,80,100];
number2=[3,5,7,9];
number3=[30,40,50,60,70,80,90,100];

%选定攻击方式,修改n的值和numbers选用的数组
n=2;
numbers=number2;

%选定小波变换参数
seed=10;
N=2;
ratio=0.99;
wavelet='db6';
alpha=0.5;

%定义攻击方式
ways=["NOISE","MEDIAN","JPEG"];
way=char(ways(1,n));

% 攻击后产生的图片的存放位置
file_path =['E:\matlab_codes\lab_3\stirmark\Media\Output\Images\Set1\',way,'\'
    ];
```

```
27
28 %保存选定文件夹所有图片的信息
29 img_path_list = dir(strcat(file_path,'lenna-waved_',way,'_*.bmp'));
30
31 %图像总数量
32 img_num = length(img_path_list);
33
34 if img_num > 0
35     corrs=[];
36     corr_DCTs=[];
37     for j = 1:img_num %逐一读取图像
38         img_name = [file_path,'lenna-waved_',way,'_',num2str(numbers(j)),'.bmp'
              ];
39         [corr,corr_DCT]=W_SVD_find(origin_path,img_name,wavelet,N,ratio,seed,
              alpha);
40         corrs=[corrs,corr];
41         corr_DCTs=[corr_DCTs,corr_DCT];
42     end
43     disp_xy(numbers,corrs,corr_DCTs,'中值滤波攻击下检测出的相关性','中值滤波攻
          击参数',0);
44 end
```

代码 6: 绘制每个参数的 SC 图

五、 实验环境:

1.win11 操作系统

2.MATLAB R2022a

3.Visual Studio Code 1.84.2 (user setup)

4.Texlive

六、 实验步骤:

（一）**W-SVD** 水印生成及嵌入

6.1.1 原图小波分解

在本次实验中使用的原图为 png 格式的图片，取出其 R 色域的信息进行水印的嵌入，首先将水印载体进行一次小波分解得到参数 C 代表各层分解系数和 S_real 代表未补足的各层分解系数长度。

6.1.2 图像补足

为了使后期的实验更加方便，需要将载体转变为正方形的图像即选取载体矩阵行数列数中的较大值作为新的载体矩阵的边长，然后将旧载体矩阵赋值到新矩阵中，余下的位置补 0。

6.1.3 小波分解获取低频系数

将补足为正方形的载体矩阵再次进行小波分解，得到参数 C 代表各层分解系数和 S 代表各层分解系数长度，然后将得到的两个参数传递到函数 appcoef2 中提取小波分解后的 N 级低频成分

在得到图像的小波低频系数后需要对数据进行归一化处理，将其映射到区间 $[0,1]$ 中，最后得到经过处理的小波分解低频系数保存在变量 CA 中。

6.1.4 按照小波低频系数生成水印模板

首先将得到的小波低频系数进行 SVD 变换，分解得到两个正交矩阵 U,V 和一个对角矩阵 Σ, 然后使用事先设定好种子的随机数生成器生成两个随机正交矩阵，利用随机矩阵的后一定比例的列的值替换掉 SVD 变换两个正交矩阵的后一定比例的列，其中一定比例在本次实验中使用变量 radio 保存，由此即可得到两个新的正交矩阵 $\widetilde{U}$ 和 $\widetilde{V}$

然后利用设定好种子的随机数发生器生成一个对角矩阵并且零对角矩阵乘强度因子 alpha，得到一个新的对角矩阵 $\widetilde{\sum}$ 然后将得到的两个正交矩阵与对角矩阵

相乘便能够得到水印模板保存到变量 watermark 中。

6.1.5 获取带有水印的小波低频系数

在得到水印模板后将其与先前得到的原图像的小波低频系数直接相加然后将过辐的系数即超过 1 的系数置为 1，将负系数置为 0。

由于在小波分解后将低频系数进行了归一化处理，所以在此应当逆用公式将其还原得到加有水印的低频系数保存到变量 waterCA 中

因为在图像处理时对图像进行了补足处理，在得到的 waterCA 中会存在有冗余信息，所以需要只保存其原始信息部分。图像在补全前进行过一次小波变换操作得到了参数 S_real，该参数的第一行保存着原始图像小波变换后被分解的行数和列数，因此只保留 waterCA 的 $1:S\_real(1,1),1:S\_real(1,1)$ 部分即可，得到的数据保存到变量 CA_tilde 中。

6.1.6 获得嵌入水印后的图像

在得嵌入有水印的低频系数后首先将其转变为只有一行的格式，然后用其代替小波变换得到的 C 与得到的 S 进行逆小波变换还原图像的格式。

由于在图像处理之初对图像进行了补全操作，所以在此要裁剪掉补全的部分。

最后再将经过处理得到的图像与 G,B 两个色域进行融合操作即可得到嵌入水印后的图像。

（二）W-SVD 水印的检测

6.2.1 计算原始水印和待测水印

在进行相关度分析前首先要对原图进行嵌入水印操作，得到原图嵌入水印后的小波低频系数记为 waterCA_origin，然后得到原图的小波低频系数记为 CA_origin 然后提取出待检测图像的小波低频系数记为 CA_test

计算得到原始水印 watermark_origin 为 waterCA_origin - CA_origin,
待测水印 watermark_test 为 CA_test - CA_origin

6.2.2 计算原始水印和待测水印的相关性

首先利用计算相关性的公式计算出常规检测直接相关性值保存到变量 corr_coef 中

然后将两个水印分别进行 DCT 变换后取 32*32 的块利用公式进行相关性的计算将计算的结果保存在变量 corr_DCTcoef 中

6.2.3 绘制检测结果 SC 图

对小波变换的每一个参数，wavelet（小波函数）、N（小波分解层数）、ratio（替换正交矩阵的比例）、seed（种子）、alpha（强度因子）选取一定的范围作为检测的区间，每一次循环遍历单个参数的所有区间调用计算相关性的函数，得到该参数的多个取值下的相关性的不同，将其结果进行绘制。对每个参数进行一次上述操作即可得到有关所有参数的 SC 图。

（三）分析水印算法鲁棒性

6.3.1 嵌入水印后的图像使用 stirmark 工具生成攻击后的图像

修改 stirmark 工具的配置文件，对已经利用 W-SVD 嵌入水印的图片进行 AddNoise、JPEG、MedianCut 三种攻击方式得到攻击后的图像，然后在 output 文件夹中对不同攻击方式的图片进行分类整理，以便于之后水印的检测。

6.3.2 初始化程序运行参数

首先初始化每种攻击方式的参数，将其保存到各自的 number 数组中，然后定义当前攻击方式 n, 并且利用变量 numbers 表示当前攻击方式的参数数组，然后根据工具方式拼接出攻击方式所在的文件夹路径，然后读取选定的文件夹路径下攻击方式的所有图片

6.3.3 水印检测与绘制结果图

对于读取得到的每一个图片都进行一次水印检测操作，将得到的两个相关性分别保存到数组 corrs 和 corr_DCTs 中。最后使用 corrs 和 corr_DCTs 绘制当前攻击方式下的相关性折线图。

七、 实验结果与分析:

（一）W-SVD 水印嵌入

- 运行 W-SVD 水印嵌入函数得到的结果如下图所示

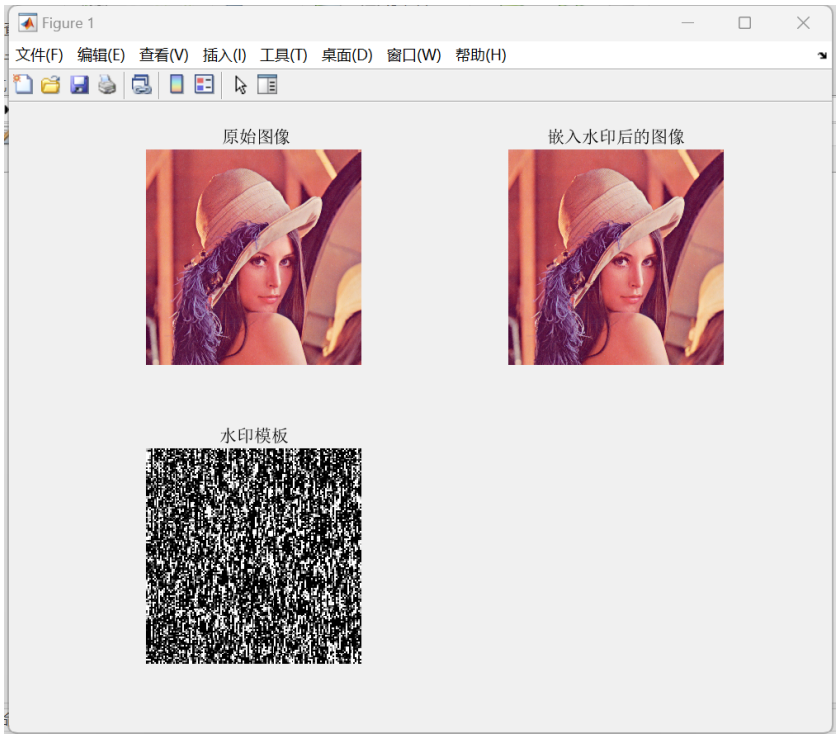

图 6: W-SVD 水印嵌入结果

该函数运行后能够显示出进行 W-SVD 水印嵌入前后的图像还能够得到嵌入的水印模板的图像

（二）**W-SVD** 水印检测

• 检测在 W-SVD 水印嵌入过程中，种子值的不同对水印相关性的影响如下图所示

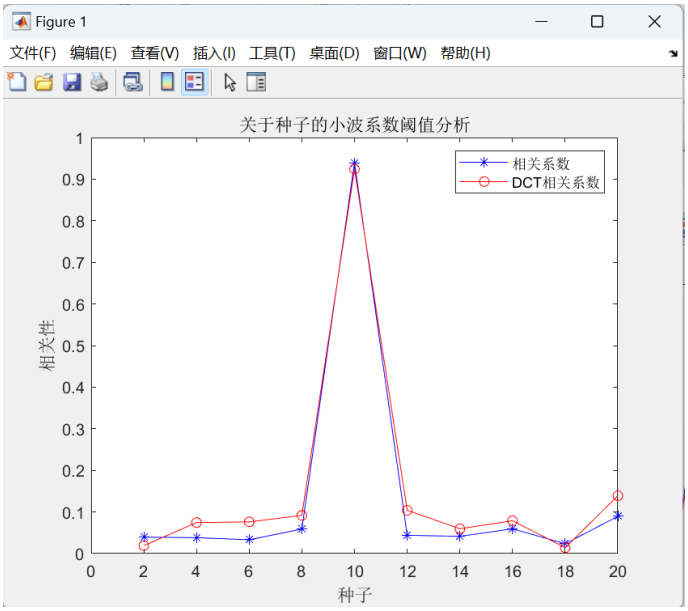

图 7: 种子值-相关性关系

• 检测在 W-SVD 水印嵌入过程中，小波变换次数的不同对水印相关性的影响如下图所示

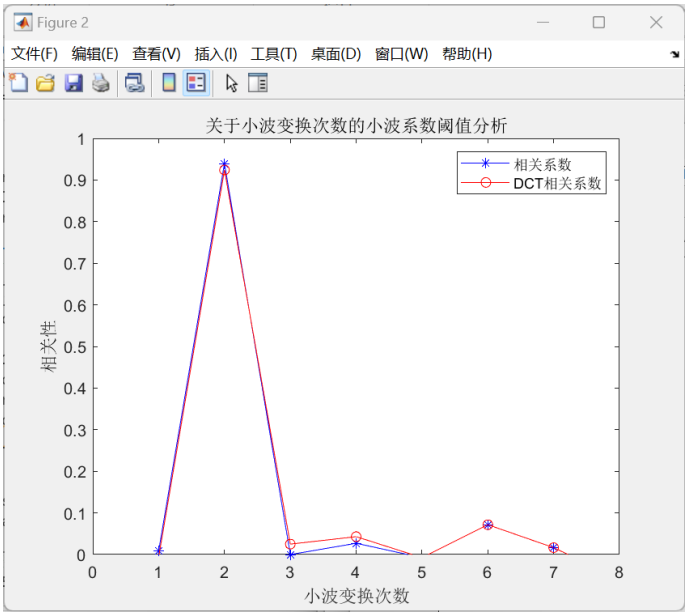

图 8: 小波变换次数-相关性关系

• 检测在 W-SVD 水印嵌入过程中，替换比例的不同对水印相关性的影响如下图所示

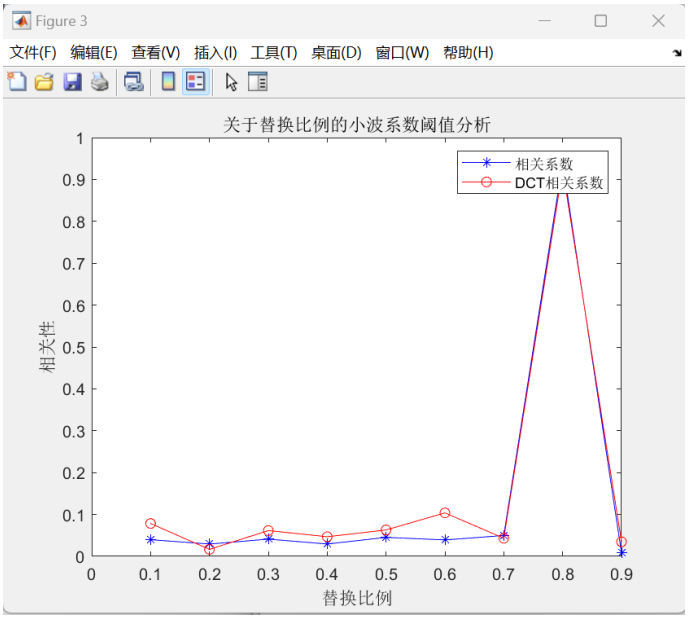

图 9: 替换比例-相关性关系

• 检测在 W-SVD 水印嵌入过程中，小波函数种类的不同对水印相关性的影响如下图所示

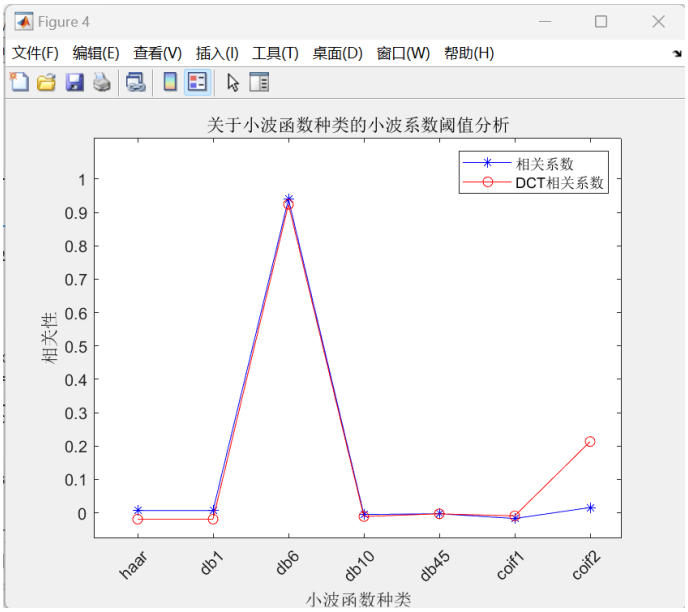

图 10: 小波函数种类-相关性关系

• 检测在 W-SVD 水印嵌入过程中，强度因子的不同对水印相关性的影响如下图所示

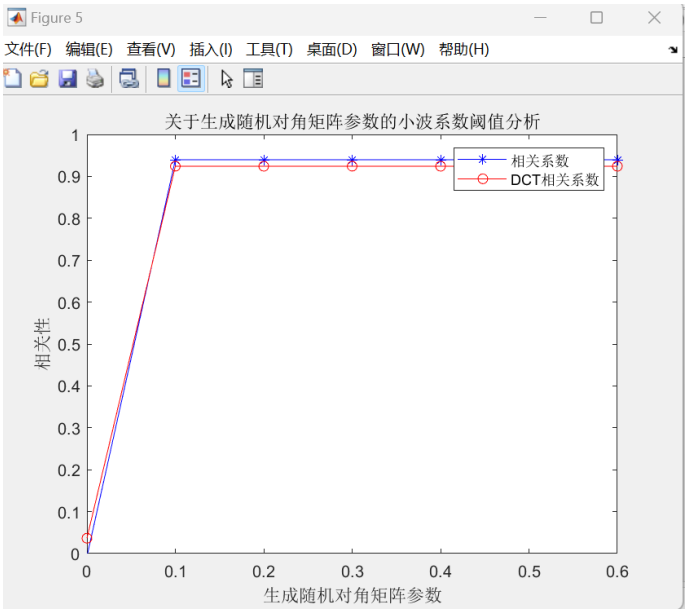

图 11: 强度因子-相关性关系

（三）分析水印算法鲁棒性

• 加噪处理 NOISE

加噪后的图像如下图所示

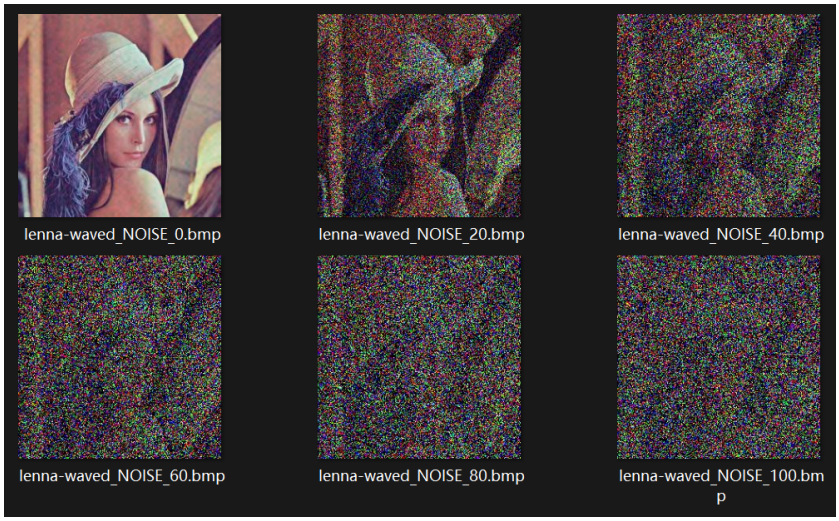

图 12: 加噪处理后的图像

将使用不同参数的加噪处理后的图像进行水印检测后得到的结果如下图所示

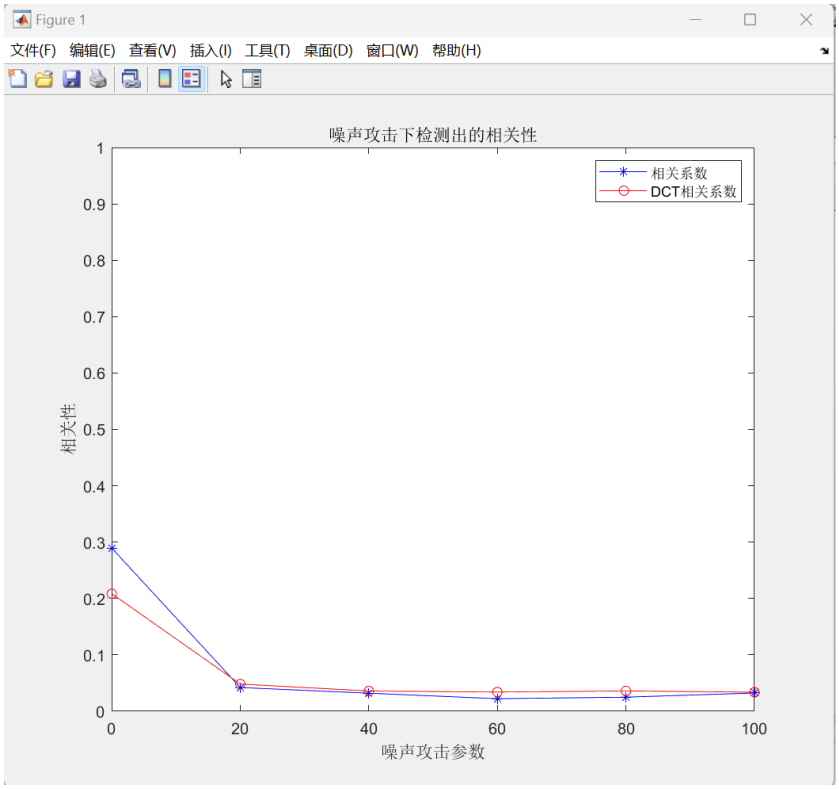

图 13: 加噪处理后的相关性

通过分析相关性的折线图可以看出在加噪信号为 0 的情况下，待检测的图片尚且保存有较高的相关性，能够检测出有水印的嵌入，且直接计算出的相关性高于在 DCT 域计算出的相关性，但在有加噪信号的条件下待检测图片的相关性就骤降至 0.1 以下，不能够有效的分析出是否有水印的嵌入。且两个计算相关性的方法无较大差别

由此可以断定，W-SVD 水印嵌入算法在面对加噪处理时鲁棒性较低。

- JPEG 攻击

JPEG 压缩处理后的图像如下图所示

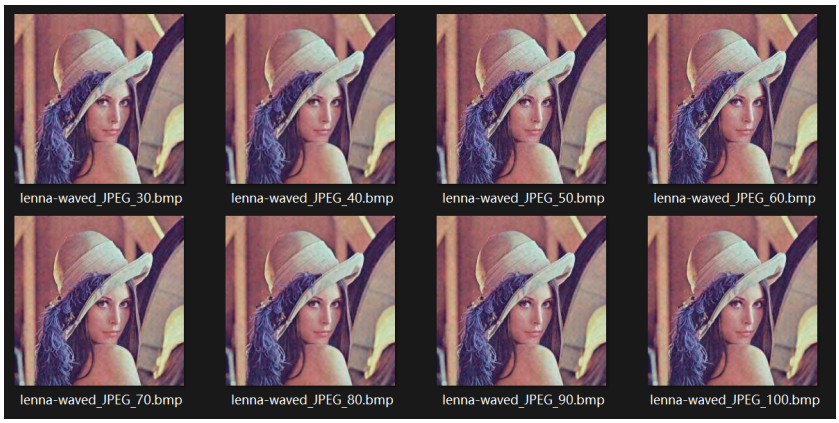

图 14: JPEG 压缩处理后的图像

将使用不同参数的 JEPG 压缩处理后的图像进行水印检测后得到的结果如下图所示

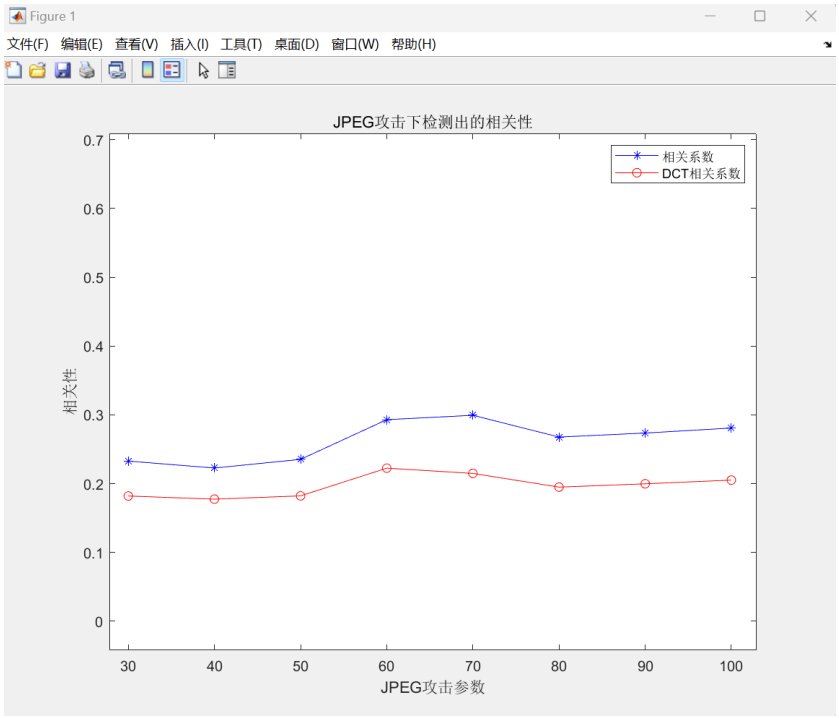

图 15: JPEG 压缩处理后的相关性

通过分析相关性的折线图可以看出在 JPEG 压缩率在 30-50 时，图像的相关性维持在 0.2 左右，仍然有一定的识别能力而且当压缩率到达 60-70 时，图像的相关

性会保持在 0.3 左右，相比于压缩率较低的情况水印的识别能力更强。当压缩率到达 80-100 时，图像的相关性会降低到 0.25 左右，相比于压缩率中等的情况，识别能力较差。且直接计算的相关性始终高于在 DCT 域计算出来的相关性

由此可以断定，W-SVD 水印嵌入算法在面对 JPEG 压缩处理时的鲁棒性较高，相关性仍能够维持在一定的水平，但是压缩率在 60-70 时相比于更高或更低的压缩率，其相关性更高，因此在采取 JPEG 对嵌入水印的图像进行攻击时应当选取较低或者较高的压缩率，不能选择压缩率在 60-70 之间进行压缩。

- 中值滤波 MEDIAN

中值滤波处理后的图像如下图所示

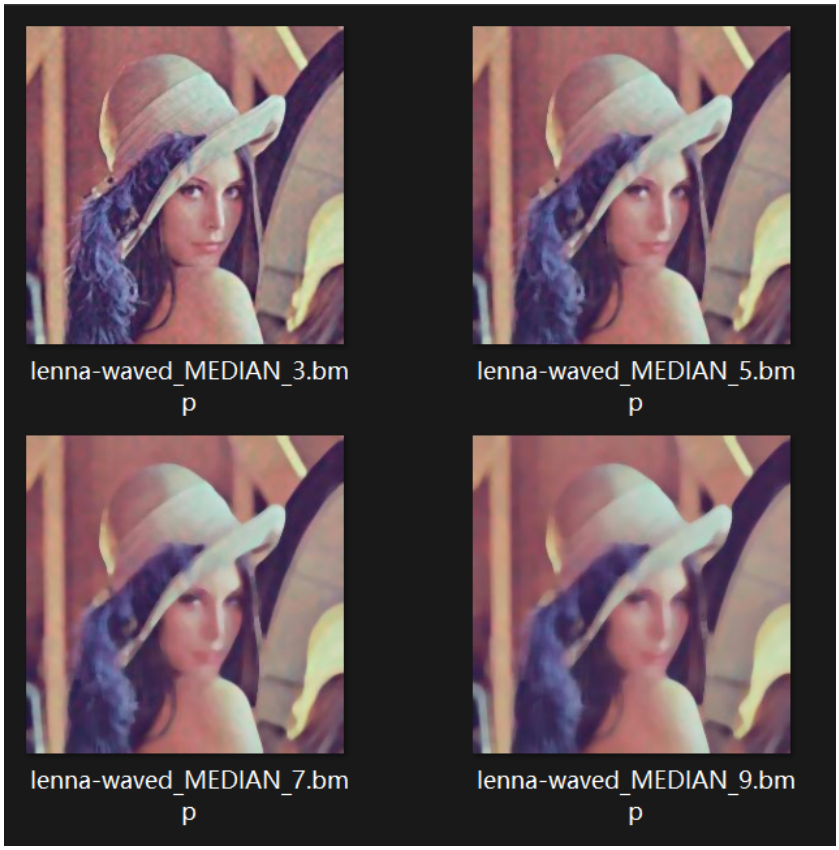

图 16: 中值滤波处理后的图像

将使用不同参数的中值滤波处理后的图像进行水印检测后得到的结果如下图所示

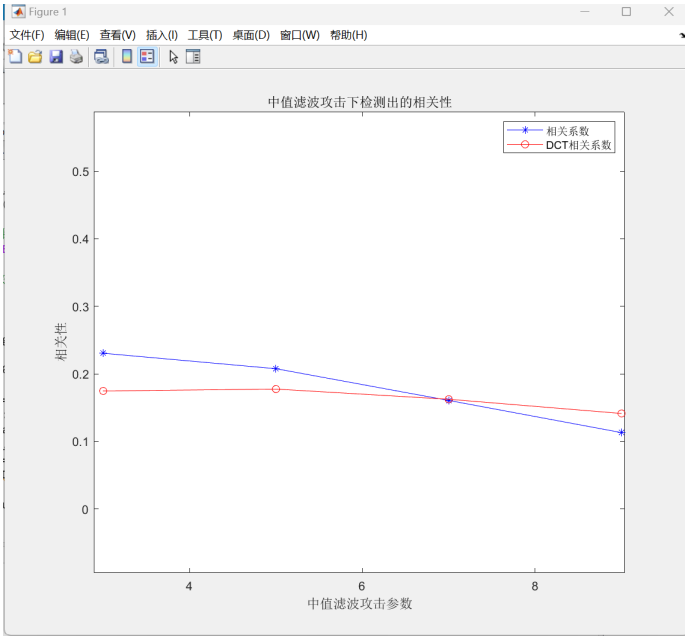

图 17: 中值滤波 MEDIAN 处理后的相关性

通过分析折线图可以得出随着滤波参数核的变大，待检测图像的相关性在缓慢降低。在核为 7 之前，DCT 检测的能力较弱，检测到的相关性相比于直接计算较低。而在核大于 7 的情况下则 DCT 的检测能力较强，检测到的相关性相比于直接计算要高

由此可以断定，W-SVD 水印嵌入算法在面对中值滤波时保持有一定的鲁棒性，且随着中值滤波核的增大，其水印被检测的能力越低。因此在选用中值滤波对 W-SVD 水印嵌入算法进行攻击时，应当尽量选用较大的核。

- 总结

分析三种不同的攻击方式攻击后图片的检测结果可以看出，W-SVD 水印嵌入算法在面对加噪处理时的鲁棒性最低，且近乎无法检测出是否嵌入水印，但是使用该方法对原图的损害较大。W-SVD 水印嵌入算法在面对 JPEG 压缩处理时的鲁棒性最高，能够检测的程度始终维持在一定的水平，并且该种方法对原图的损害较小。W-SVD 水印嵌入算法在面对中值滤波时保持有一定的鲁棒性，但是使用该方法对原图的损害较大。

（四）分析不同算法参数情况对水印性能的影响

7.4.1 水印性能衡量标准

在本次实验中首先将原图嵌入水印后传入 stirmark 中进行 median 攻击，选取使用核为 3 的攻击后的图像进行检测通过比较相关性来衡量当前参数对水印性能的影响。

7.4.2 小波分解层数 N

表 1: 小波分解层数不同对水印性能的影响

小波分解层数	直接相关性	DCT 域相关性
1	0.0615	0.0892
2	0.2305	0.1747
3	0.3131	0.2607
4	0.1270	0.1270
5	0.1684	0.1684

观察小波分解层数的不同得出的结果可以看出，当小波分解的层数是 3 层时，该水印的鲁棒性达到峰值，且当小波分解层数为 1 层时该水印的鲁棒性最弱，在攻击后不能够被检测。由此可见小波分解的层数对水印的鲁棒性有较大影响

7.4.3 小波函数种类 wavelet

表 2: 小波函数种类不同对水印性能的影响

小波函数种类	直接相关性	DCT 域相关性
haar	0.3445	0.3445
db1	0.3445	0.3445
db6	0.3132	0.2608
db10	0.2130	0.1833
coif1	0.3422	0.3188

观察小波分解种类的不同得出的结果可以看出，当小波分解函数为 haar、db1、coif1、db6 时，该水印的鲁棒性没有较大差别，均维持在 0.3 左右，但是小波分解

函数为 db10 时，该水印的鲁棒性较低，只维持在 0.2 左右。由此可见小波分解的函数种类对水印的鲁棒性影响较小

7.4.4 强度因子 alpha

表 3: 强度因子不同对水印性能的影响

强度因子	直接相关性	DCT 域相关性
0	-0.5000	-0.5897
0.1	0.0614	0.0619
0.2	0.1417	0.1239
0.3	0.2084	0.1768
0.4	0.2639	0.2192

观察强度因子的变化得出的结果可以看出，强度因子的值越大，得到的水印的鲁棒性就越强。且强度因子对水印的鲁棒性的影响较大。

7.4.5 种子值 seed

表 4: 种子值不同对水印性能的影响

种子值	直接相关性	DCT 域相关性
2	0.2439	0.1928
5	0.2832	0.2541
10	0.2639	0.2192
20	0.3087	0.2601
30	0.2752	0.2276

观察种子值的不同的变化结果可以看出，所有的种子值得出的结果均在 0.2-0.3 之间且彼此的值相差不大。所以种子值对水印的鲁棒性的影响较小。

7.4.6 替换比例 ratio

表 5: 替换比例不同对水印性能的影响

替换比例	直接相关性	DCT 域相关性
0.6	0.4186	0.4010
0.7	0.3580	0.3484
0.8	0.4105	0.3934
0.9	0.3620	0.3398
0.99	0.3087	0.2601

观察替换比例不同时的结果可以看出，当替换比例为 0.6 和 0.8 时，水印的鲁棒性最强，相关性为 0.4 左右，而当替换比例为 0.7 和 0.9 时水印的鲁棒性相对变弱，替换比例为 0.99 时水印的鲁棒性达到最弱。由此可见替换比例对水印的鲁棒性的影响较大。

八、 总结及心得体会:

1. 在本次实验中学习了对于图像使用 W-SVD 的算法嵌入数字水印以及该种数字水印的检测方法。

2. 通过对 W-SVD 水印嵌入算法的不同方式的攻击得出了 W-SVD 算法在面对不同的攻击方式的鲁棒性。

3. 在本次实验中明白了即使是相同的图像在 matlab 中以矩阵的方式存储时也会因为其拓展名的不同会造成两个图像矩阵的不同。

4. 学习到了在数字水印的嵌入过程中不仅仅要考虑嵌入后能否被检测出来，更要考虑待测图像在面对不同种类的攻击时的鲁棒性。

九、《人工智能背景下的数字水印》学习心得

（一）水印的历史、分类和应用

在该部分下讲述了隐写的产生以及水印这一工具的产生背景，然后介绍了数字水印在数字媒体中的广泛应用。随后介绍了可逆水印出现的原因以及水印在其他方面的拓展应用，了解到了例如 User-Cloud、图像上传、图像取回等与数字水印相关的协议，并且了解到数字水印借助其鲁棒性来解决很多安全问题。强调了水印的鲁棒性的重要性。

（二）基于深度学习模型的水印

在该部分讲解了如何使用深度学习的大致框架框架 END 来训练模型。讲解了 HiDDeN 这一深度学习水印，这一方案虽然能够实现端到端的训练并且具有很好的透明性和应对失真鲁棒性但是在面对真实 JPEG 压缩时的鲁棒性却不强。

借由 HiDDeN 水印提出的挑战问题提出了 MBRS 的解决方案，该方案能够有效的提升 JPEG 压缩的鲁棒性。

介绍了保证嵌入器和提取器之间更好耦合的水印算法 De-END, 通过在嵌入器之前添加一个提取器，两个提取器共享权重的方式来实现两者之间更高质量的耦合

为了保证嵌入器和提取器在参数和结构上完美耦合，提出了可逆网络这一解决思路，在鲁棒性的前提下进一步优化了透明性。

介绍了跨媒介鲁棒水印的概念，这一概念与先前介绍的抗摄屏水印有一定的相关性，这一水印的最大的难点就在于真实的物理过程失真非常复杂且不可导，而且相比于数字处理更难建模。针对这一问题的解决方法有模拟拍摄过程的噪声层，利用可导过程模拟真实物理失真来对水印模型进行训练。该解决方案有 LFM 和 DeNoL 等模型。其他的针对这一问题的解决方法有仅训练提取器的水印方案，介绍了该方案下的具体模型 DTW。

介绍了文档水印、抗翻录音频水印和 3D 模型水印的背景和相关工作。

（三）保护深度学习模型的水印

该部分介绍了深度学习模型当前所面临的知识产权保护的需求，讲解了解决这一需求的白盒水印和黑盒水印以及深度模型水印所面临的各种攻击方式。不仅仅是深度学习模型，图像处理和生成模型同样也需要数字水印的保护。

（四）生成式人工智能鉴别水印

该部分首先提出了大语言模型生成文本的危害以及国家对 AI 生成内容的监督以及其检测方案 GPTzero 但是该种方法检测准确率不高并且受文本内容的影响较大。同样也介绍了基于深度学习的检测方法但是其仍然具有脆弱性且在现实复杂的环境下非常容易被干扰。最后提出可以通过生成内容水印、黑盒水印或者白盒水印来抵抗大语言模型生成文本的危害。

介绍了语音克隆技术的产生以及其风险，急需音色水印来对音色权进行保护，但是如果攻击者能够仿造被攻击者的音色的话会不会能够将音色水印一同仿造呢？

最后介绍了图像概念水印来防止攻击者训练 AI 模仿画家的图像风格来保护画家等创作者的权利。但是图像概念水印如何能够让人们信服并且解决图像风格归属权的问题呢？