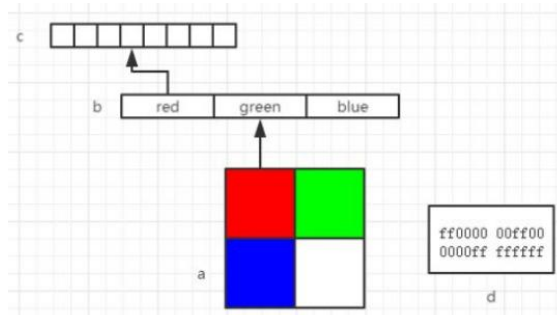


课程名称	保密技术基础 A		实验课时	4
实验项目名称和序号	LSB 图像信息隐藏实验	1	同组者姓名	无
实验目的	1. 掌握对图像的基本操作。 2. 能够用 LSB 算法对图像进行信息隐藏 3. 能够用 LSB 提取算法提取隐藏进图像的信息 4. 能够反映 jpeg 压缩率与误码率之间的关系			
实验环境	Win 10 + MATLAB R2018b			
实验内容和原理	<p><b>实验内容：</b></p> <ol style="list-style-type: none"> <li>1. 用 MATLAB 函数实现 LSB 信息隐藏和提取</li> <li>2. 分析了 LSB 算法的抗攻击能力</li> <li>3. 能随机选择嵌入位（考虑安全性因素）</li> </ol> <p><b>实验原理：</b></p> <p>信息隐藏技术是信息安全领域的一个研究热点，其原理是利用载体中存在的冗余信息来隐藏秘密对象。它通过隐藏通信过程的存在来获得通信的安全。基于 LSB(Least Significant Bits) 的图像信息隐藏技术是信息隐藏中的重要技术手段。它具有隐藏容量大，对原始载体变动小的优点，得到了广泛应用。</p> <p>LSB (Least Significant Bit) 最低有效位，指二进制中最低位数值。LSB 信息隐藏，其核心思想是用秘密信息代替 LSB，即二进制数据的最低位。这是数字密写中常用的方法，并且可以相对容易地应用到图像和音频中。这种方法的优势是相当数量的信息能够被隐藏，同时也几乎不会对载体造成什么可察觉的影响。</p> <p>图像由像素点构成，对于彩色图像来说，每个像素点由红，绿，蓝三种颜色构成，这样的图像称为 RGB 图像，每种颜色为该图像的一个分量，对于每个像素点来说颜色由三个分量共同控制。</p>  <p>如上图所示，a 是一个 <math>2 * 2</math> 的小图像，总共有 4 个像素，每个像素呢，由三种颜色构成 (b)，而每种颜色呢，由 8 位构成 (c)，然后根据小图像的颜色，我们把颜色值写出来，为了方便书写，用 16 进制表示，如 d 图所示，四组数字，分别对应着小图像的四个像素。</p>			

	<p>嵌入：选择一个载体元素的子集<math>\{j_1, j_2, \dots, j_{L(m)}\}</math>, 其中共有 <math>L(m)</math> 个元素, 用以信息隐藏信息的 <math>L(m)</math> 个 Bit。然后在这个子集上执行替换操作, 把 <math>c_{ji}</math> 的最低比特用 <math>m</math> 替换。</p> <p>提取：找到嵌入信息的伪装元素的子集<math>\{j_1, j_2, \dots, j_{L(m)}\}</math>, 从这些伪装对象中抽取他们的最低比特位, 排列组合后组成秘密信息。</p> <p>为了进一步增加隐秘性, 可使用伪随机置换, 即由伪随机数发生器产生秘密消息控制被替换的载体位置。接收端只要知道随机数发生器的种子, 就能再生伪随机序列, 确定隐写的位置。</p> <p>本次实验主要涉及三个算法, LSB 信息嵌入, LSB 信息读取, JPEG 压缩率与误码率关系。</p>
实验步骤	<p><b>算法一：LSB 信息嵌入</b></p> <ol style="list-style-type: none"> <li>1. 读取一副 <math>256 \times 256</math> 大小的图片, 判断是否为 RGB 图像。若为 RGB 图像, 则读取图像的一层信息 (如 R 层)。</li> <li>2. 以二进制形式读取要嵌入到图片里的消息。并读取消息的长度 (嵌入消息的长度不能超过图像位数)。</li> <li>3. 产生与消息长度一致的一串随机数 (不能相同)。</li> <li>4. 按照产生的随机数的序列依次将图片层的最后一位改为消息的信息。即用消息替换图片的最后一位信息。</li> <li>5. 嵌入完成后, 如果为 RGB 则将该层返回原图像。然后将数据信息写回图像。LSB 就完成了。</li> </ol>
方 法	<p><b>代码实现</b></p> <ol style="list-style-type: none"> <li>1. 读取信息隐藏载体图片 (test.jpg) 的功能, 并判断图片是否为 RGB 图像, 并在判断后读取该 RGB 图像的 R 层。</li> </ol>
关键代码	<pre> image=imread('test.jpg'); %读取指定文件 image_size=size(image); %获取图片的分量数  %-----判断是否为RGB----- if numel(image_size)&gt;2     disp('the picture is RGB!') end;  %-----提取图片R层----- Hiding_image=image; imageR=Hiding_image(:, :, 1); </pre> <ol style="list-style-type: none"> <li>2. 读取隐藏的信息, 并且按无符号二进制位读取信息, 获取消息的二进制位信息和信息长度。</li> </ol>

```
%-----读取隐藏信息-----
message=fopen('hiding_message.txt','r');
[msg,message_len]=fread(message,'ubit1');
%按位以二进制形式读取文本内容与长度
```

3. 利用函数产生随机行列数据，用以隐藏指定的信息位。这里随机位的产生调用了 `randinterval` 函数，这个函数的实现是基于随机序列进行间隔控制，选择消息隐藏位。返回值是两个一维矩阵，分别是行与列。

```
%文件名: randinterval.m
%函数功能: 本函数将利用随机序列进行间隔控制，选择消息隐藏位
%matrix 载体矩阵
%count 为要嵌入的信息的数量（要选择的像素数量）
%key 为密钥
%row 为伪随机输出的像素行标
%col 为伪随机输出的像素列标
function [row,col]=randinterval(matrix,count,key)
%计算间隔的位数
[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
        r=r+1;
        if r>m
            error('载体太小不能将秘密信息隐藏进去!');
        end
        c=mod(c,n);
        if c==0
            c=1;
        end
    end
    row(1,i)=r;
    col(1,i)=c;
end
```

- 按照产生的随机数的序列依次将图片层的最后一位改为消息的信息。在隐藏的过程中，需要注意的是修改最低有效位，用 mod 运算即可完成。到此为止，信息隐藏已经完成。

```
%-----LSB隐藏-----
[m,n]=size(imageR);
[row,col]=randinterval(imageR,message_len,2019);
message_num=1;
%message_num为秘密信息的位计数器
for i=1:message_len
    imageR(row(i),col(i))= imageR(row(i),col(i))-mod(
imageR(row(i),col(i)),2)+msg(message_num,1);
    if(message_num==message_len)
        break;
    end
    message_num=message_num+1;
end
```

- 修改的 R 层写入原图片，写入新的图片 Hiding\_image.tif。将两个图片同时显示进行对比。

```
%-----返回原图像-----
Hiding_image(:, :, 1)=imageR;
%返回R层
Hiding_image=uint8(Hiding_image);
imwrite(Hiding_image, 'Hiding_image.tif');
%将含有隐藏信息的图片保存
subplot(121);imshow(image);title('Hiding-Before');
subplot(122);imshow(Hiding_image);title('Hiding-After');
```

### 算法二：读取 LSB 隐藏的信息

- 读取已经隐藏信息的图像。如果为 RGB 图像，则读取图像的一层（该层为嵌入信息的那层）。
- 用与 LSB 算法中相同的随机数种子产生相同的一串随机数。随机数串的长度由 LSB 中获得（长度不得大于图像大小）。
- 按照产生的随机数序列依次读取图像的相应点最后一位的信息。并将其以二进制形式写到文件中。
- 查看文件，即获取的信息，与嵌入的信息进行比较。

### 代码实现

- 读取已经隐藏信息的图像，判断是否为 RGB 图像并判断提取出隐藏信息的图层（R 层）。

```
%-----提取隐藏信息层-----
image_extract=imread('Hiding_image.tif');
imageR_extract=image_extract(:, :, 1);
```

- 用与 LSB 算法中相同的随机数种子产生相同的一串随机数，用于之后隐藏信息的提取。用同样的随机种子产生行列数据，这样获取的随机序列便于隐藏的相同，所以这里的随机数种子起到了密钥的作用。

```
%-----生成相同的随机序列-----
[row_extract,col_extract]=randinterval(imageR_extract,message_len,2019);
```

- 按照产生的随机数序列依次读取图像的相应点最后一位的信息。并将其以二进制形式写到文件（extract\_message.txt）中。

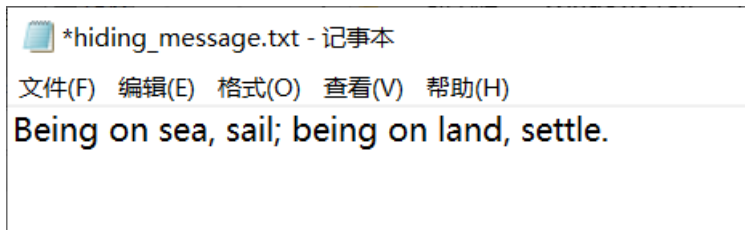
```
%-----提取隐藏信息-----
t=1;
frr=fopen('extract_message.txt','w');
%以写入方式打开只写文件
for i=1:message_len
    if bitand(imageR_extract(row_extract(i),col_extract(i)),1)==1
        fwrite(frr,1,'ubit1');
    else
        fwrite(frr,0,'ubit1');
    end;
    if t==message_len
        break;
    end;
    t=t+1;
end;
fclose(frr);
```

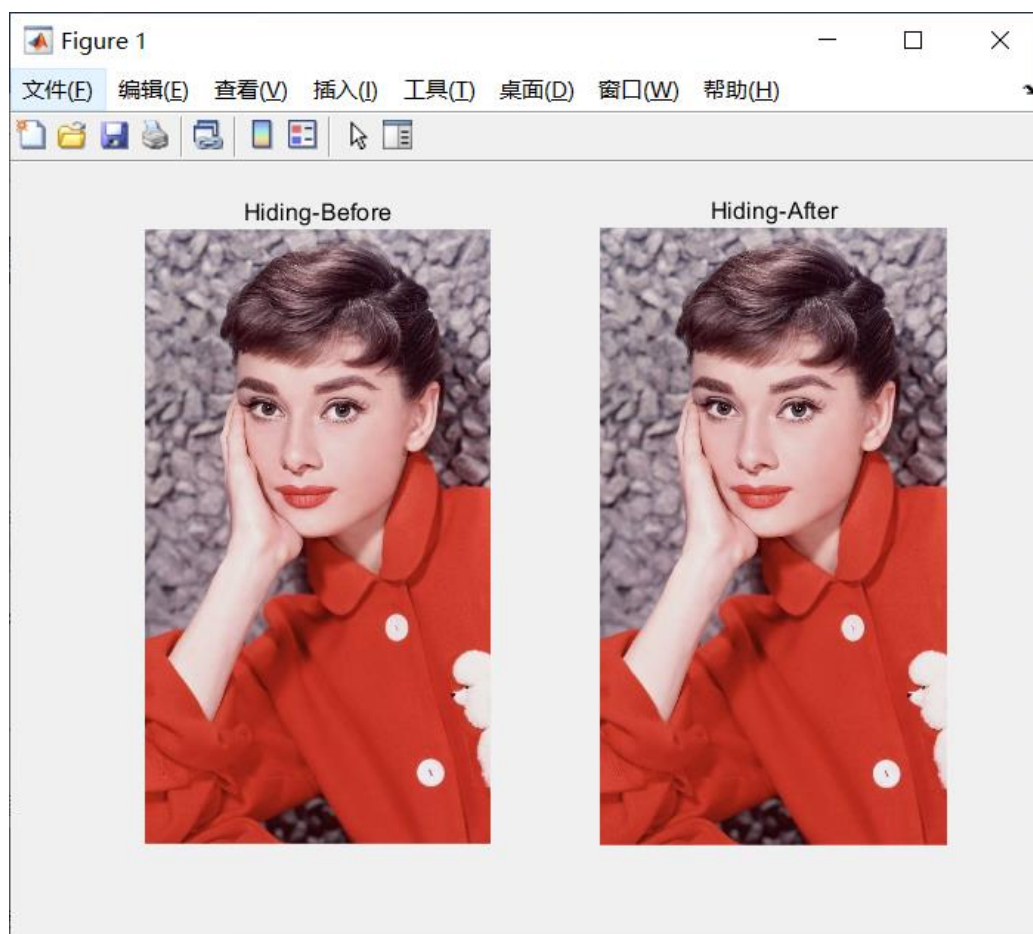
### 算法三：JPEG 压缩率与误码率之间的关系

- 读取已经隐藏信息的图像。
- 使用 imwrite 函数对图像进行压缩，设定压缩比例。
- 如果为 RGB 图像，则读取嵌有信息的一层。按照读取 LSB 隐藏信息算法的步骤，读取 信息，不写入文件。
- 读取原文件，即隐藏的信息，以二进制读取。并取得消息长度。
- 比较取得的信息和原信息的每一位，记录不相等位数的个数。
- 用不相等个数除以总长度即可得到误码率。
- 改变压缩率。得到一组误码率关于压缩率的函数。

### 代码实现

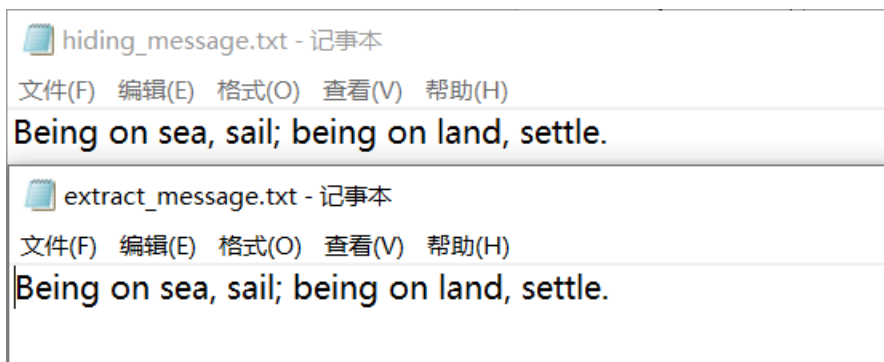
```
%-----计算误码率-----
[m,n]=size(R_imageR);
p=1;
[row,col]=randinterval(R_imageR,message_len,2019);
for i=1:message_len
    if bitand(R_imageR(row(i),col(i)),1)==1
        result(p,1)=1;
    else
        result(p,1)=0;
    end
    if p==message_len
        break;
    end
    p=p+1;
end
wbit=find(result~=msg);
%寻找不相等的位置
wbitn=size(wbit,1);
```

	<pre> %统计不相等的个数 ber(compressibility/5)=wbitn/message_len; end %-----绘图----- compressibility=5:5:100; %每隔5绘制一个点 plot(compressibility,ber,'*',compressibility,ber); </pre> <p>以上代码（JPEG_ber）首先对隐藏信息后的图片读取后进行压缩，这里压缩率（compressibility）取样 20 组。在按照算法二同样的步骤进行信息读取，写入 result 一维矩阵，与原信息进行比较，记录不同的位数。误码率=错误位数/信息总位数。</p> <p>最后采用 Plot 对变量 compressibility 和 ber 绘图，得出函数图像。</p>
测试记录 分 析 结 论	<p>算法一：LSB 信息嵌入测试</p> <div data-bbox="502 1386 1251 1615">  </div> <p style="text-align: center;"><b>隐藏信息</b></p> <p>在运行了 LSB 信息嵌入程序（LSB_embed）后，得到了进行信息隐藏后的图片，通过对两张图片的对比，未发现人眼可辨的明显差别，故 LSB 算法的隐藏是较为成功的，也体现出其隐藏特性。</p>



### 算法二：读取 LSB 隐藏的信息测试

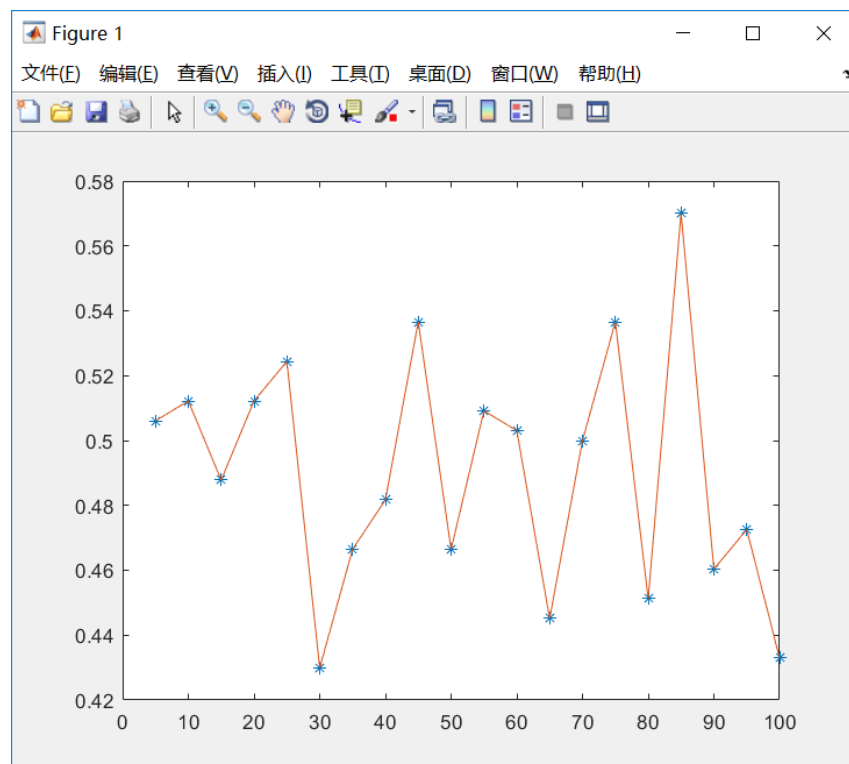
在运行了 LSB 隐藏信息提取程序 (LSB\_extract) 后，对比隐秘信息文本与提取文本，两者完全一致，说明 LSB 隐藏信息的提取是成功的，至此，我们完成了整个 LSB 算法流程，即完成了基于 LSB 的图像信息隐藏与提取。



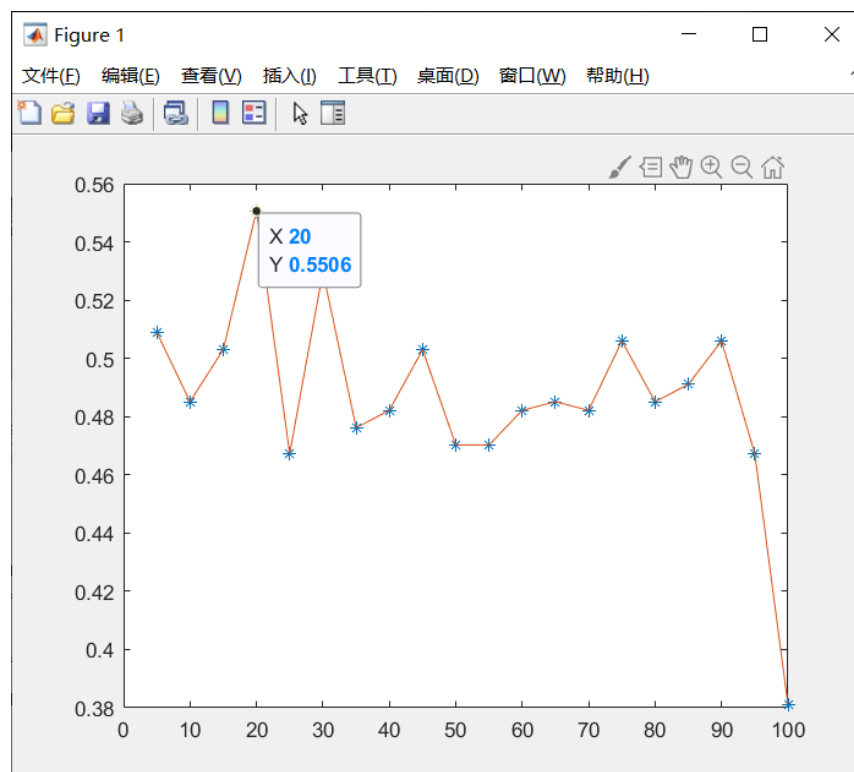
### 算法三：JPEG 压缩率与误码率之间的关系测试

运行 JPEG 压缩及计算误码率程序 ( )，对不通压缩率下面的同一张图片进行采样分析 (共 20 组)，并将实验数据绘制成图像，通过下图可知，LSB 信息隐藏对于压缩

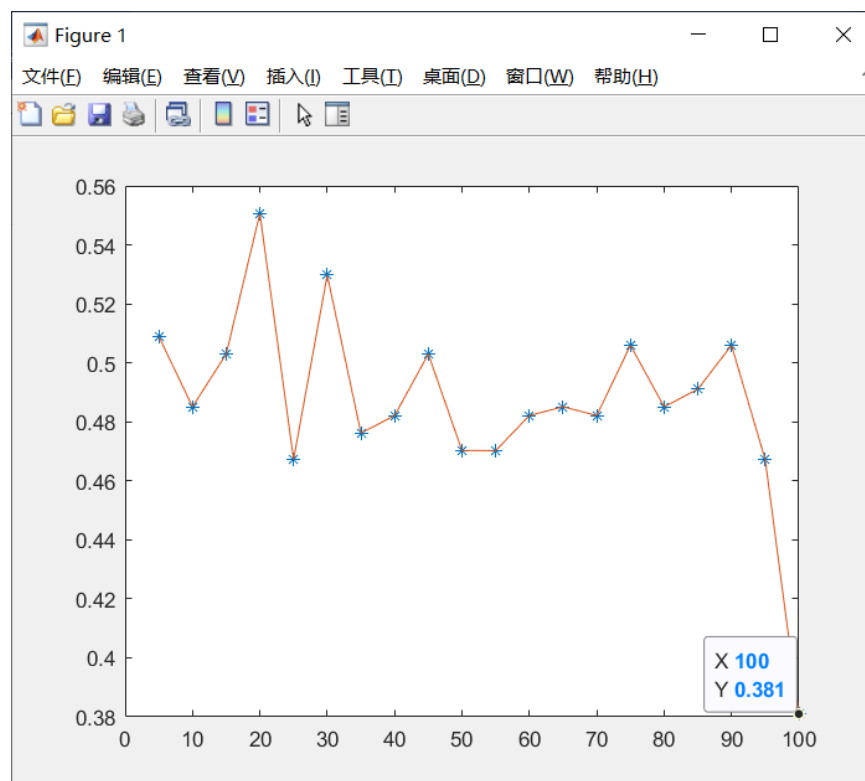
的抵抗力较弱，且其稳定性不足。



在不同程度的压缩率下，误码率呈现较大范围上下波动，在压缩率为 20% 时，误码率为 55.06%，在压缩率为 20% 时，误码率为 55.06%。







基于以上分析可知，基于 LSB 的图像信息隐藏抗压缩能力差，鲁棒性较差。

## 小 结

经过本次实验，了解了图像相关的基本知识，如灰度图像、RGB 图像、图像在计算机中的存储等，学习了 matlab 这一工具，并使用其编写脚本及函数，熟悉了基于 LSB 图像信息隐藏方法的整个过程——隐藏和提取，并通过 JPEG 与误码率之间关系的比较对 LSB 信息隐藏方法进行了性能的评估。结合本次实验，对 LSB 图像信息隐藏归纳如下。

### LSB 算法优点：

1. 算法简单，易于实现，计算速度也快。
2. 在基础算法上能够很快的进行改进，并在脆弱性水印中应用广泛。
3. 由于能在最低有效位（一般是最后两位）进行嵌入，故对于 256 色（8 位）的 RGB 图像，在 3 层图像中均可插入 1/8 到 1/4 的消息，总的来说，容量还是足够大的。

### LSB 算法缺点：

1. 嵌入消息较大时，所花时间较长。
2. 只能处理简单的流格式的文件。
3. 为了满足水印的不可见性，允许嵌入的水印强度较低，对空域的各种操作较为敏感。

	<p>4. 基本的 LSB 算法抗 JPEG 压缩能力弱。</p> <p>5. 鲁棒性差。</p> <p style="text-align: center;"><b>参考文献</b></p> <p>[1]csdn_moming. 利用 LSB 算法隐藏文字信息的 MATLAB 实现 [EB/OL].<a href="https://blog.csdn.net/csdn_moming/article/details/50936687">https://blog.csdn.net/csdn_moming/article/details/50936687</a>, 2016-03-20.</p> <p>[2]XA 小白. 简单的图像信息隐藏 [EB/OL].<a href="https://blog.csdn.net/qq_38253732/article/details/82916433">https://blog.csdn.net/qq_38253732/article/details/82916433</a>, 2018-10-02.</p> <p>[3]松子茶. LSB 图像信息隐藏 [EB/OL].<a href="https://blog.csdn.net/songzitea/article/details/18953829">https://blog.csdn.net/songzitea/article/details/18953829</a>, 2014-02-06.</p> <p>[4]cheeseandcake. LSB 图像信息隐藏实验 [EB/OL].<a href="https://blog.csdn.net/cheeseandcake/article/details/52997903">https://blog.csdn.net/cheeseandcake/article/details/52997903</a>, 2016-11-01.</p> <p>[5]kross. 计算机只认识 0 和 1 如何表示图像和影视等众多应用? [EB/OL].<a href="https://www.zhihu.com/question/36269548">https://www.zhihu.com/question/36269548</a>, 2015-10-08.</p>
以下由实验教师填写	
记 事 评 议	
成绩评定	<p>平时成绩_____ 实验报告成绩_____ 综合成绩 _____</p> <p style="text-align: right;">指导教师签名:</p>