实验二：二值图像的信息隐藏

综合评分：

【实验目的】:
掌握 MATLAB 函数编写，能够编写函数实现二值图像的信息隐藏与提取，同时分析分析阈值 R0，R1 以及健壮参数 λ 对实验结果的影响。

【实验内容】:（请将你实验完成的项目涂“■”）

■一、用 MATLAB 函数实现二值图像信息隐藏

■二、用 MATLAB 函数实现二值图像隐藏信息的提取

■三、分析阈值 R0，R1 以及健壮参数 λ 对实验结果的影响

【实验工具及平台】:

■ Windows+Matlab　　□ 其它:（请注明）_____

【实验涉及到的相关算法】:

1、与实验内容选择的项目对应;

2、请使用流程图、伪代码、NS 图或文字方式描述，不要完全贴代码

■一、用 MATLAB 函数实现二值图像信息隐藏

- 步骤一：读取秘密信息和载体图像信息。

  主函数 binaryhide.m。

  使用 fopen 和 imread 函数来读取秘密信息和载体信息,存储到 msg 和 image 矩阵中。

- 步骤二：确定图像块的首地址。

  块的大小取 10*10。

  使用哈希置换法 hashreplacement.m，原本是随机选取像素点，这里希望返回各个块的首地址。可以将图片大小除以 10 输入大小，对于返回的行列标做以下处理：

```
if ( return value ! =1)
    return value = ( return value - 1)× 10+1;
```

  而哈希置换法的主要思想是：

  设每一个输入 i, i 为小于载体总嵌入单位数的一个整数(如载体若有 256×256 像素, 且每一像素点嵌入 1bit 信息, 则 i <256× 256) 。由 i 均能得到一个数 ji , 表示秘 密信息中第 i 个 bit 相应的嵌入载体的索引, 且 ji 不会发生重复。ji 的生成步骤为:

```
v = [ i/X] ;
u = i mod X;
v = ( v + MD5( u, k1 ) ) mod Y;
u = ( u + MD5( v, k2 ) ) mod X;
v = ( v + MD5( u, k3 ) ) mod Y;
```

ji = vX + u; 其中 X, Y 分别为载体图像的行、列像素数量。这个算法就是一个新的伪随机发生器, {ji}才是一个随机序列。

- 步骤三：分析可用的图像块。

函数 available.m。

主要有以下几种情况：

$$\begin{cases} P_1(B_i) > R_1 + 3\lambda \text{ or } P_1(B_i) < R_0 - 3\lambda & (a) \\ R_0 - 3\lambda < P_1(B_i) < R_0 \text{ or } R_1 < P_1(B_i) < R_1 + 3\lambda & (b) \\ R_0 - 3\lambda < P_1(B_i) < R_1 & (c) \\ R_0 < P_1(B_i) < R_1 + 3\lambda & (d) \\ R_0 - \lambda < P_1(B_i) < R_0 \text{ or } R_1 < P_1(B_i) < R_1 + \lambda & (e) \end{cases}$$

需对块进行调整，具体的调整有两个方面:

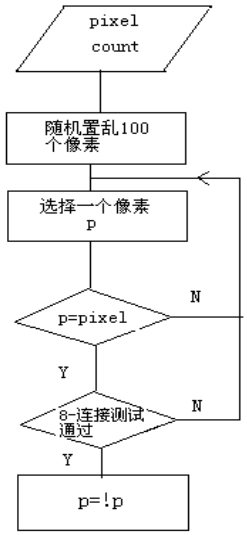
一是将难以调整块改变为 不可用块; 二是将可用块改变为最终隐藏块。

对 P1 (Bi) 进行调整, 最终使其完全落在[0, R0 - 3λ] ∪[R0 - λ , R0] ∪[R1 , R1 + λ] ∪[R1 + 3λ , 1] 区域内。

- 步骤四：修改每块的像素关系，完成信息嵌入。

我们在 10×10 的范围内随机置乱 100 个点。然后根据置乱后的顺序找到第一个与所输入的像素值相同的像素，进而判断它的八连接有没有与之相反的像素，如果有则表明这一点可以是边界，可以修改；否则继续往后寻找。

结合块首地址，我们可以在每一块中任意确定一个像素。



如果要嵌入的信息为 1, P1 (Bi) 的可能取值为以下三个区域：

$$\begin{cases} R_0 < P_1(B_i) < R_1 & (a) \\ R_1 < P_1(B_i) < R_1 + \lambda & (b) \\ R_1 + \lambda < P_1(B_i) < R_1 + 3\lambda & (c) \end{cases}$$

对于(a) , 需要将 R1 - P 1 (Bi) +1 个 0 像素改为 1 像素;

对于(b) , 不需要进行任何修改;

对于(c) ，则需要将 P 1(Bi) - (R1 + λ)+1 个像素变为 0.
如果要嵌入的信息为 0, P1 (Bi) 的可能取值为以下三个区域：

$$\begin{cases} R_0 < P_1(B_i) < R_1 \\ R_0 - \lambda < P_1(B_i) < R_0 \\ R_0 - 3\lambda < P_1(B_i) < R_0 - \lambda \end{cases}$$

对于(d) , 需要将 P1 (Bi) - R0 + 1 个 1 像素改为 0 像素;
对于(e) , 不需要进行任何修改;
对于(f) 则需要将(R0 - λ)-P1(Bi)+1 个 0 像素变为 1 像素。

- 步骤五：信息写回保存。

```
image = round( image) ; % 防止边界扩散后的取整复原
result = image;
imwrite( result, goalfile) ;
```

■二、用 MATLAB 函数实现二值图像隐藏信息的提取

- 步骤一：读取隐蔽载体信息。

主函数 binaryextract. m。
使用 imread 函数来读取含有秘密信息的图片，存储到 stegoimage 矩阵中。

- 步骤二：根据密钥确定图像块的首地址和图像块使用的顺序。

```
[ m, n] = size( stegoimage) ;
m = floor( m/10) ;
n = floor( n /10) ;
temp = zeros( [ m, n] ) ;
[ row, col] = hashreplacement( temp, m* n, m, key, n) ;
% 将 m, n 也作为密钥简化输入
for i = 1：m* n   if row( i) ～ = 1
      row( i) = ( row( i) - 1) * 10 +1;
   end   if col( i) ～ =1
      col( i) = ( col( i) - 1) * 10 + 1;
   end
end
```

- 步骤三：按隐藏时顺序分析图像块 。
- 步骤四：按照每块的像素关系，完成信息提取。
- 步骤五：信息写回保存。

提取算法描述为:

```
if ( P1 ( Bi ) >50% && P 1( Bi ) < R1 +3λ )
   message = 1;
else if ( P1 ( Bi ) <50% && P 1 ( Bi ) > R0 -3λ )
   message = 0
```

■三、分析阈值 R0，R1 以及健壮参数 λ 对实验结果的影响。

（1）通过不同的压缩，比较像素的改变率：

控制 R0,R1 相同，观察不同 λ 对实验的影响：

```
function jpgandlumda(test)
image=imread(test);
image=round(double(image)/255);
[M,N]=size(image);
quality=5:5:100;%定义压缩质量比从 5%到 100%
result=zeros([1 max(size(quality))]);
count=0;
different=0;
for q=quality
    count=count+1;
    imwrite(image,'temp.jpg','jpg','quality',q);%利用 imwrite 函数完成压缩
    comdone=imread('temp.jpg');
    comdone=round(double(comdone)/255);
    for i=1:M
        for j=1:N
            if comdone(i,j)~=image(i,j)
                different=different+1;
            end
        end
    end
result(1,count)=different/(M*N);
different=0;
end
plot(quality,result);
xlabel('jpeg 压缩率');
ylabel('像素改变的百分比例');
title('二值图像在 JPEG 条件下像素改变的状况')
```

（2）通过调节输入，使 λ 等于 2，观察可用块于总块的比例。

【实验分析】:

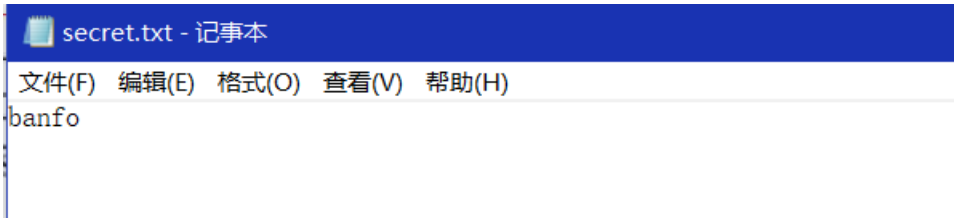
1、 请尽量使用曲线图、表等反映你的实验数据及性能

2、 对照实验数据从理论上解释原因

3、 如无明显必要，请不要大量粘贴实验效果图

■一、用 MATLAB 函数实现二值图像信息隐藏：

需要隐藏的信息：

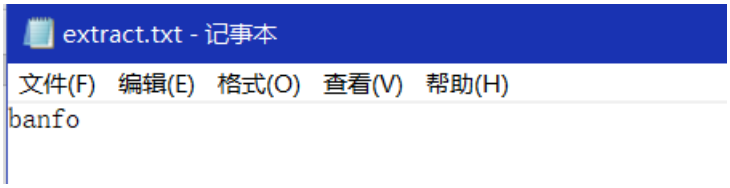


载体图像：

■二、用 MATLAB 函数实现二值图像隐藏信息的提取

含有秘密信息的图像：

以及提取的秘密信息：



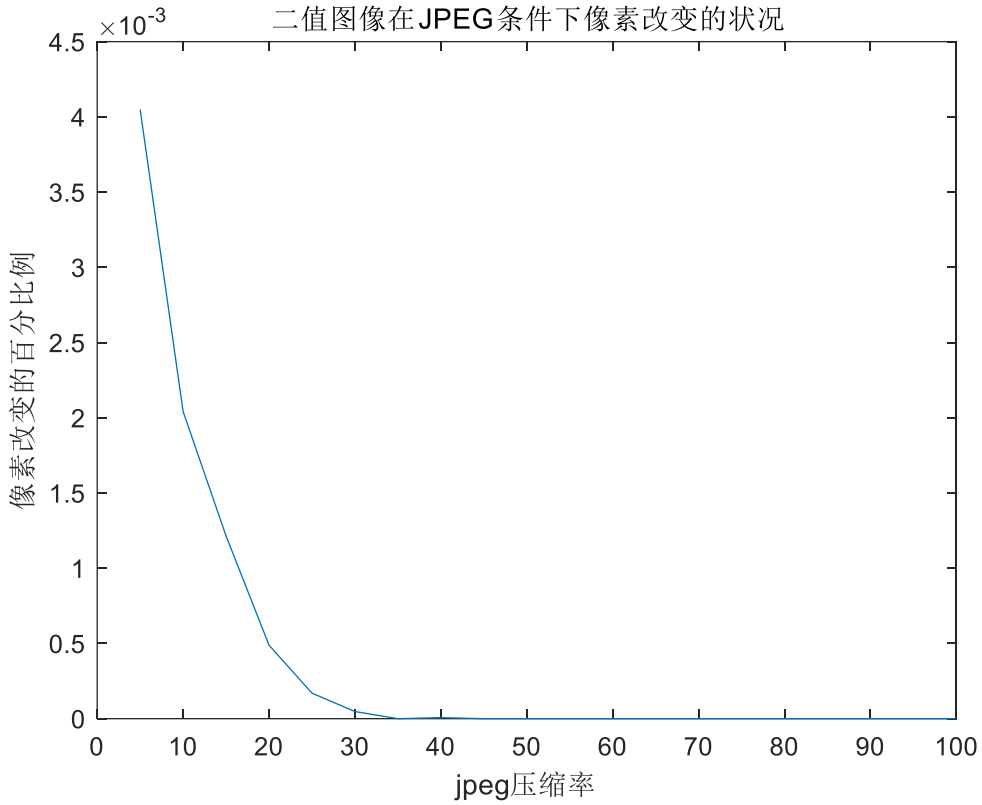
提取成功！

■三、分析阈值 R0，R1 以及健壮参数 λ 对实验结果的影响。

（1）通过不同的压缩，比较像素的改变率：

控制 R0,R1 相同，观察不同 λ 对实验的影响：

运行结果如下图所示：



分析得到，在压缩率大于 45% 时，图像基本上是鲁棒的。即使发生变化，像素的 变化比例均不超过 1%！也就是说，λ 取为 0.5% 就可以达到上述区分的目的。

在 R0，R1 一定的情况下，λ 设置过小是为了使每块图像块的像素都满足条件，要 修改的像素就会多一些，同时，λ 大一点，隐藏块与无用块的区别也大一些，提取出错 的概率就小一些。

（2）通过调节输入，使 λ 等于 2，观察可用块于总块的比例。

第一次 R0,R1 为 47，53

第二次 R0,R1 为 45，55

第三次 R0,R1 为 43，57

适当调整 R0 和 R1 的距离，可以使所分析的图像块的数量大大减小了。比如：
取 R0 =47， R1 =53 时，隐藏 40bits 信息要分析 1303 块才能确定其隐藏的位置；
取 R0 =45， R1 =55 时，隐藏 40bits 信息只用分析 1058 块就确定了隐藏的位置；
取 R0 =43， R1 =57 时，隐藏 40bits 信息只用分析 687 块就确定了隐藏的位置，

从而也就可以认为将隐藏容量扩大了。