实验四：隐写分析

综合评分：

【实验目的】:
利用卡方分析，RS 分析依赖数理统计来分析判断载体对象（图像）是否嵌入了隐藏信息。其中卡方隐写分析主要针对图像所有像素点的 LSB 全嵌入情况；
RS 隐写分析主要是针对采用伪随机 LSB 嵌入算法进行攻击的一种方法。RS 方法不但能检测出图像是否隐藏信息，而且还能比较准确的估算出隐藏的信息长度。

【实验内容】:（请将你实验完成的项目涂“■”）
■一、实现针对 LSB 隐写的卡方隐写分析算法，并分析其性能。
■二、实现针对 LSB 隐写的 RS 隐写分析算法，并分析其性能。

【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、实现针对 LSB 隐写的卡方隐写分析算法，并分析其性能。

原理：

设图像中灰度值为 j 的象素数为 hj，其中 0≤j≤255。如果载体图像未经隐写，h2i 和 h2i+1 的值会相差很大。秘密信息在嵌入之前往往经过加密，可以看作是 0、1 随机分布的比特流，而且值为 0 与 1 的可能性都是 1/2。如果秘密信息完全替代载体图像的最低位，那么 h2i 和 h2i+1 的值会比较接近，可以根据这个性质判断图像是否经过隐写。

步骤分析：

① 首先输入带有隐藏信息的图片，我们将图像分为 20 块，计算每一块的隐写概率 P。

② 在计算每一块时，令 $h_{2i}^{*} = \frac{h_{2i} + h_{2i+1}}{2}$　$q = \frac{h_{2i} - h_{2i+1}}{2}$，之后计算 r=q*q/h。最后利用累计密度函数计算出隐写概率 P。

③ 用图像显示 20 块中每一块的 r 和 P.

代码：

```
function [r,P]=x2(input)
cover=imread(input);
cover=double(cover);
cover=uint8(cover);
S=[];
P_value=[];
interval=384/20;
 for j=0:19
```

```
    count=imhist(cover(:,floor(j* interval)+1:floor((j+1)* interval)));
    h_length=size(count);
    p_num=floor(h_length/2);
    r=0;
    k=0;
    for i = 1:p_num
        h=(count(2*i-1,1)+count(2*i,1))/2;
        q=(count(2*i-1,1)-count(2*i,1))/2;
        if h ~= 0
            r=r+q*q/h;
            k=k+1;
        end
    end
    P=1-chi2cdf(r,k-1);
    S=[S r];
    P_value=[P_value P];
end
x_label=1:1:20
figure,
subplot(211),plot(x_label,S,'LineWidth',2),title('X^2 统计');
subplot(212),plot(x_label,P_value,'LineWidth',2),title('p 值');
```

■二、实现针对 LSB 隐写的 RS 隐写分析算法，并分析其性能。

原理：

任何经过 LSB 隐写的图像，其最低比特位 0，1 分布满足随机性，即 0，1 的取值概率均为 1/2，而未经过隐写的图像不存在此特性。

步骤分析：

1.首先输入含有隐秘信息的图像，将图像变为一维数组，之后再 4 个像素为一组分为若干组，

2.用函数 f（）统计每一组的空间相关性记为 summ(1)，再将该组通过掩模数组 M（1，0，0，1）进行非负和非正翻转，再计算空间相关性分别记为 summ(2)，summ(3)。

3.若 summ(2) > summ(1)，则 Rm+1.

若 summ(2) < summ(1)，则 Sm+1.

若 summ(3) > summ(1)，则 R_m+1.

若 summ(3) < summ(1)，则 S_m+1.

将这四个值存储到数组 R 中。

4.接下来对原总体图像做全正反转（此时 M=(1,1,1,1)），再进行 1，2，3 步，得到新的 Rm，Sm，R_m，S_m 再存储到数组 R 中。

5. 将 R 数组中的 8 个数除以 bufsize,

- 结合如下方程：

$$2(d_1+d_0)x^2+d_{-0}-d_{-1}-d_1-3d_0)x+d_0-d_{-0}=0$$

其中：

$$d_0=R_M(p/2)-S_M(p/2),d_1=R_M(1-p/2)-S_M(1-p/2)$$

$$d_{-0}=R_{-M}(p/2)-S_{-M}(p/2),d_{-0}=R_{-M}(1-p/2)-S_{-M}(1-p/2)$$

解上述方程，取绝对值较小的 x，计算嵌入率 p 为：$p = x/(x-1/2)$

代码：

```
%主函数:
function rate=rs(input)
cover=imread(input);
cover=double(cover);
cover=cover(:)';
[m,n] = size(cover);
cover1=cover;
k = n/4;

R = zeros(2, 4); %用来存储两组 RS 的四个值
M = [1;0;0;1]; %用来实现随即翻转
bloc = zeros(4, 1);
for j = 1 : k        %做非负和非正翻转，再计算相关性
    bloc=cover1((j-1)*4+1:j*4);  %每四个像素放在一起
    summ(1) = f(bloc);
    summ(2) = f(posturn(bloc,M));
    summ(3) = f(negturn(bloc,M));
    if summ(2) > summ(1)  %Rm
        R(1,1) = R(1,1) + 1;
    end
    if summ(2) < summ(1)  %Sm
        R(1,2) = R(1,2) + 1;
    end
    if summ(3) > summ(1)  %R_m
        R(1,3) = R(1,3) + 1;
    end
    if summ(3) < summ(1)  %S_m
        R(1,4) = R(1,4) + 1;
    end
end

cover2 = posturn(cover1, ones(n, 1));  %先对载体全做正翻转，接着做非负和非正翻转，再计算相关性
for j = 1 : k
    bloc=cover2((j-1)*4+1:j*4);
    summ(1) = f(bloc);
    summ(2) = f(posturn(bloc,M));
    summ(3) = f(negturn(bloc,M));
    if summ(2) > summ(1)  %Rm
        R(2,1) = R(2,1) + 1;
```

```
    end
    if summ(2) < summ(1)  %Sm
        R(2,2) = R(2,2) + 1;
    end
    if summ(3) > summ(1)  %R_m
        R(2,3) = R(2,3) + 1;
    end
    if summ(3) < summ(1)  %S_m
        R(2,4) = R(2,4) + 1;
    end
end
R = R/k;
dpz = R(1,1) - R(1,2); dpo = R(2,1) - R(2,2);
dnz = R(1,3) - R(1,4); dno = R(2,3) - R(2,4);
C = [2 * (dpo + dpz), (dnz - dno - dpo - 3 * dpz), (dpz - dnz)];
r = roots(C);
p = r./(r - 0.5);
rate=p(2);

end


%其他函数:
% 计算相关复杂度
function y = f(x)
n = length(x);
y = sum(abs(x(1:n-1) - x(2:n)));
end

%非负翻转,F1 变换和 F0 变换
function y = posturn(x, M)
M=M';
y = x + (1 - 2 * mod(x, 2)) .* M;
end

% 非正翻转，F-1 变换和 F0 变换
function y = negturn(x, M)
M=M';
y = x + (2 * mod(x, 2) - 1) .* M;
end
```

【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、实现针对 LSB 隐写的卡方隐写分析算法，并分析其性能。

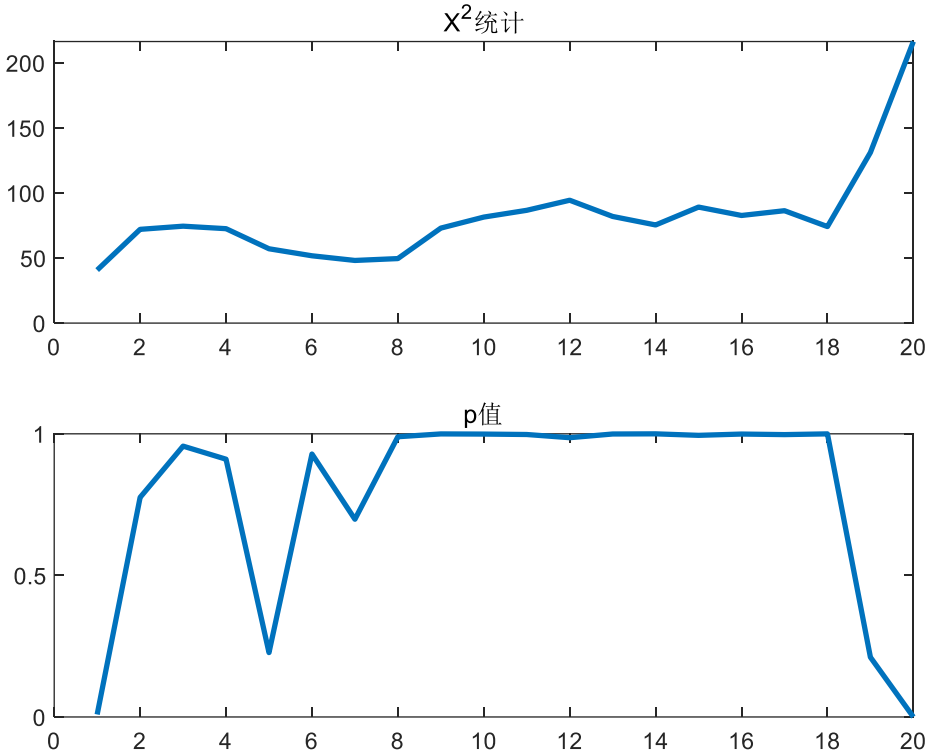
利用实验一的 lsb 顺序隐藏函数，得到含有隐藏信息的图片 girlgray.bmp。

其中隐藏信息和 girlgray.bmp 如下图所示：

运行函数 x2，得到 20 块中每一块的 r 和 P：

```
>> [r,P]=x2('girl1sb.bmp')
```

得到下图：



可以看出如果 p 接近于 1，则说明载体图像中含有秘密信息。

这里需要注意的是如果图像过大而且没有分块的话，相当于把所有的 X2 加在一起，即使每一部分都很小，最后加起来 X2 值也会很大，最后导致 P 的值为 0，所以把图像分块，分别计算每一块的 X2 和 P，这样就能得到很好的实验效果。

■二、实现针对 LSB 隐写的 RS 隐写分析算法，并分析其性能。

利用实验一的 lsb 随机隐藏函数，得到含有隐藏信息的图片 rand.bmp。

其中隐藏信息和 rand.bmp 如下图所示：

其中隐秘信息为 72920 个：

由于图像使用的 384*384，可以算出真实隐写率为 72920/384/384=0.4945

运行 rs 函数，得到隐写率为 0.4882：

结果还是比较准确的。