

实验五 图像的位平面实验

学号：2110688 姓名：史文天 专业：信息安全

实验要求

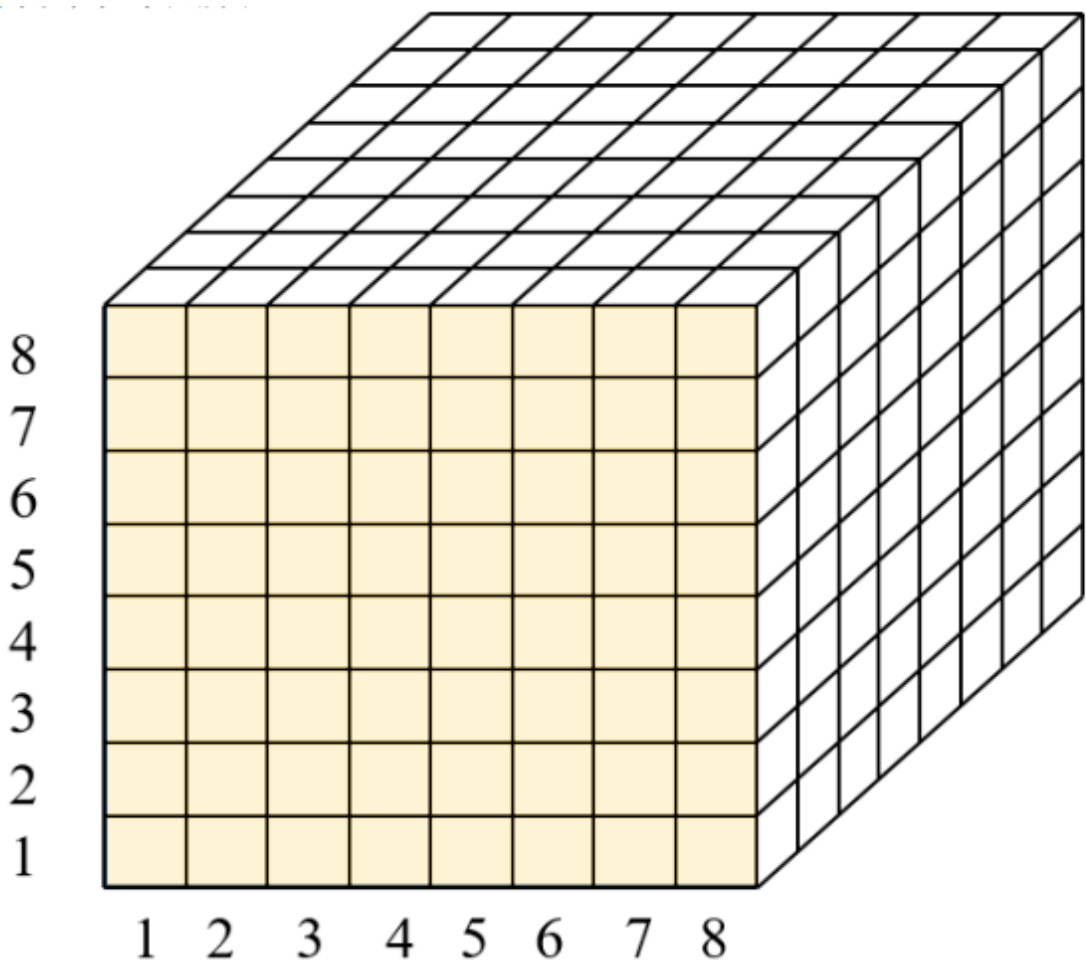
- 1、实现对1-8任意位平面的提取并显示；
- 2、实现对1~n低位平面的图像显示和8~（n+1）高位平面的图像显示；
- 3、实现去掉1~n位平面后的图像的显示。

实验原理

图像是目前互联网上传播较多的媒体文件之一。图像信息隐藏是信息隐藏领域研究时间最长，研究成果最多的载体类型之一。

主要考察像素的各个比特对图像的贡献，保留贡献大的比特位，贡献微弱的则可用于隐藏秘密信息的比特。

图像中的各个像素由八个比特组成，值为0-255，各个像素相同的位形成一个平面，称为“位平面”，如下图所示。



不同位平面的重要程度不同，实验表明，低位平面的重要性较低。

原图像展示



1-8任意位平面的提取并显示

代码部分

将“Lena.bmp”图片读入，并将其每个像素点的二进制表示的第1 到第n 位提取出来，然后将这些位重新组合成一张新的图片并显示出来。

其中，k 是自己输入的参数，表示要提取的二进制位数；a 和b 分别表示图片的高度和宽度；y 和z 是用来存储提取出来的二进制位的数组；x是临时变量，用来存储每个像素点的二进制表示的第1 到第n 位；bitget 函数用来提取二进制位；bitset 函数用来重新组合二进制位；imshow 函数用来显示图片；title 函数用来设置图片的标题，其中num2str 函数用来将数字转换成字符串。

最终，这段代码会显示出一张新的图片，其中每个像素点的二进制表示的第1 到第n 位与原图片不同。即，接收了自己的输入以后，作为要显示的位平面，之后，通过bitget 函数提取相应的位平面。

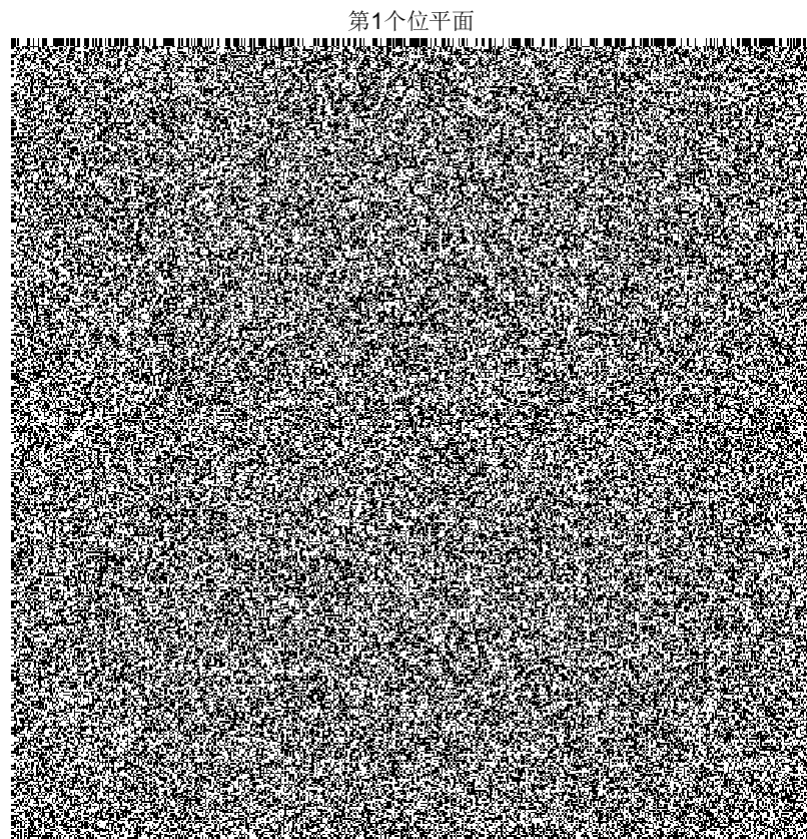
```
% Clear Memory and Command window
clc;
clear all;
close all;
img = imread("Lena.bmp");

k = input("请输入要显示的位平面：");

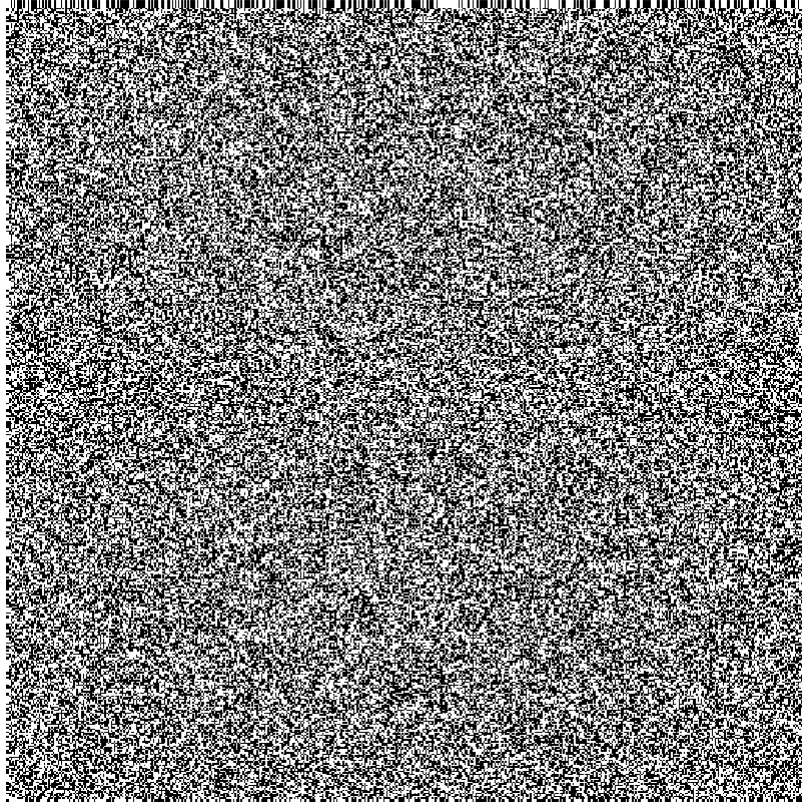
[m,n]=size(img);
c=zeros(m,n);
```

```
for i=1:m
    for j=1:n
        c(i,j)=bitget(img(i,j),k);
    end
end
figure;
imshow(c, []);
title(['第', num2str(k), '个位平面']);
```

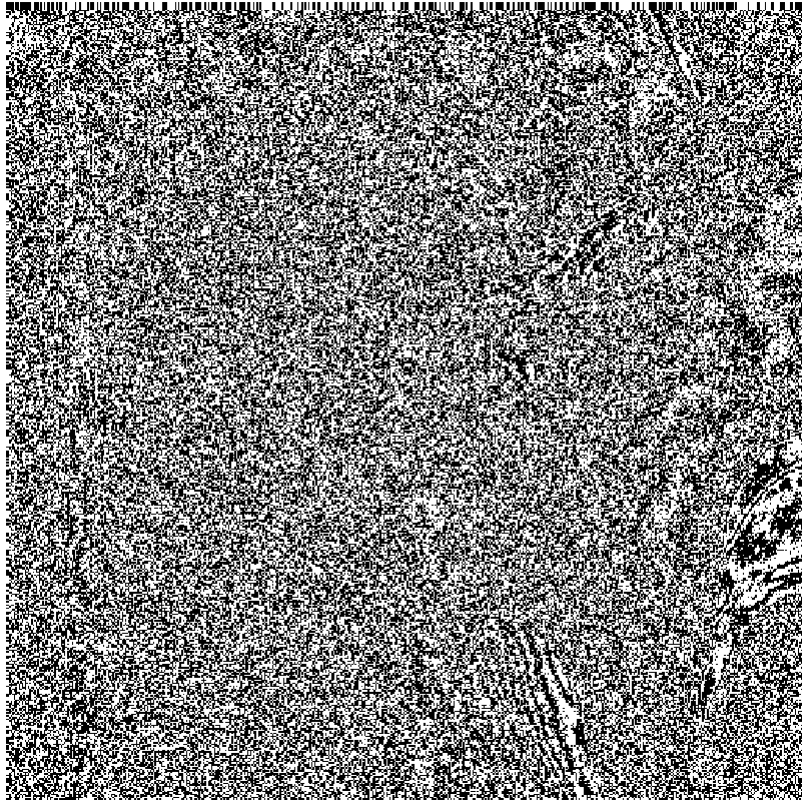
结果分析



第2个位平面



第3个位平面



第4个位平面



第5个位平面



第6个位平面



第7个位平面



第8个位平面



通过观察位平面图，我们可以发现，越低位的位平面图越不规则，越高位的位平面图越能体现出原图的特征。尤其是最低位的位平面图，看上去非常像是一个随机的像素分布图。这是由于二进制数中，越低位的bit 越容易发生改变，越高位的bit 越难改变。从神经网络的角度来看，高位平面保存了相对高维、整体的特征。

这个发现对于隐写术的应用非常有价值。在嵌入秘密信息时，我们可以选择将秘密信息嵌入到高位平面图中，以保证信息的安全性和图像质量的平衡。同时，我们也可以根据具体情况选择合适的位平面图，以达到最好的安全性和图像质量平衡。

1~n低位平面的图像显示和8~ (n+1) 高位平面的图像显示

代码部分

首先将“Lena.bmp” 图片读入，并将其每个像素点的二进制表示的第1 到第k 位提取出来，然后将这些位重新组合成一张新的图片并显示出来。同时，还将原图片的第k+1 到第8 位提取出来，重新组合成另一张新的图片并显示出来。

其中，k 是用户输入的参数，表示要提取的二进制位数；a 和b 分别表示图片的高度和宽度；y 和z 是用来存储提取出来的二进制位的数组；x是临时变量，用来存储每个像素点的二进制表示的第1 到第k 位或第k+1 到第8 位；bitget 函数用来提取二进制位；bitset 函数用来重新组合二进制位；imshow 函数用来显示图片；title 函数用来设置图片的标题，其中num2str 函数用来将数字转换成字符串。

最终，这段代码会显示出两张新的图片，其中一张图片的每个像素点的二进制表示的第1 到第k 位与原图片相同，而另一张图片的每个像素点的二进制表示的第k+1 到第8 位与原图片不同。

```
% Clear Memory and Command window
clc;
clear all;
close all;
img = imread("Lena.bmp");
k = 1
```

```

[a,b]=size(img);

y=zeros(a,b);
z=zeros(a,b);

for n=1:k
    for i=1:a
        for j=1:b
            x(i,j)=bitget(img(i,j),n);
        end
    end
    for i=1:a
        for j=1:b
            y(i,j)=bitset(y(i,j),n,x(i,j));
        end
    end
end

for n=k+1:8
    for i=1:a
        for j=1:b
            x(i,j)=bitget(img(i,j),n);
        end
    end
    for i=1:a
        for j=1:b
            z(i,j)=bitset(z(i,j),n,x(i,j));
        end
    end
end

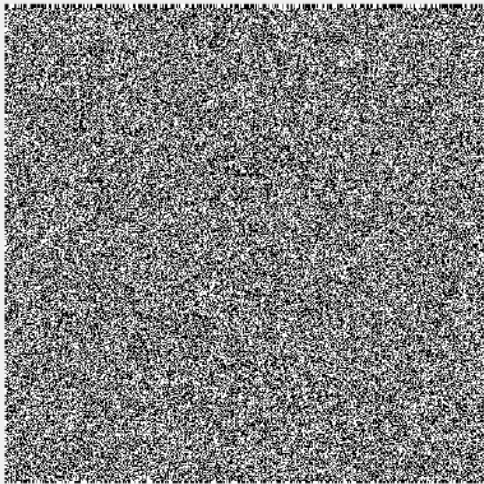
figure;
imshow(y,[]);
title(['第1-',num2str(k),'个位平面']);

figure;
imshow(z,[]);
title(['第',num2str(k+1),'-8个位平面']);

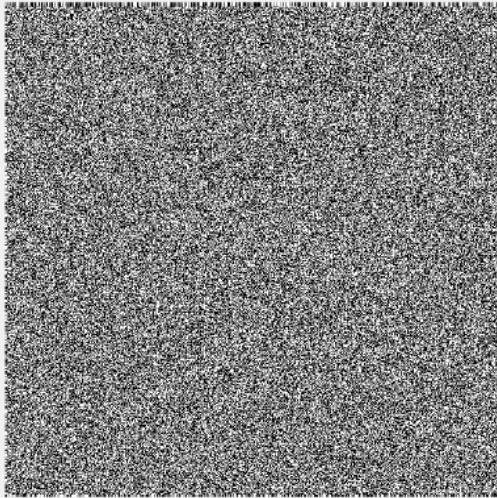
```


结果分析

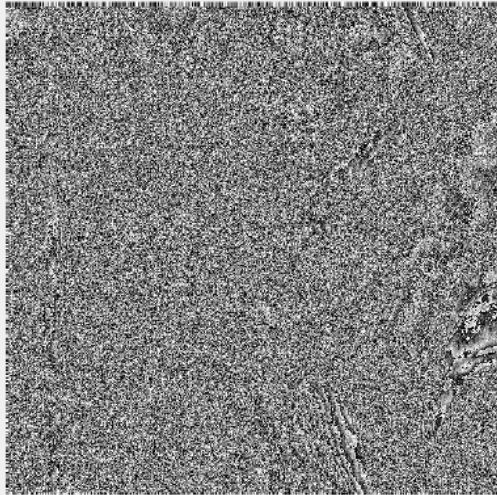
第1-1个位平面



第1-2个位平面



第1-3个位平面



第2-8个位平面



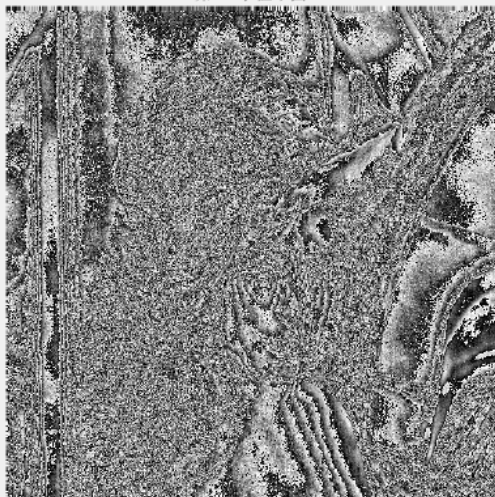
第3-8个位平面



第4-8个位平面



第1-4个位平面



第1-5个位平面



第1-6个位平面



第5-8个位平面

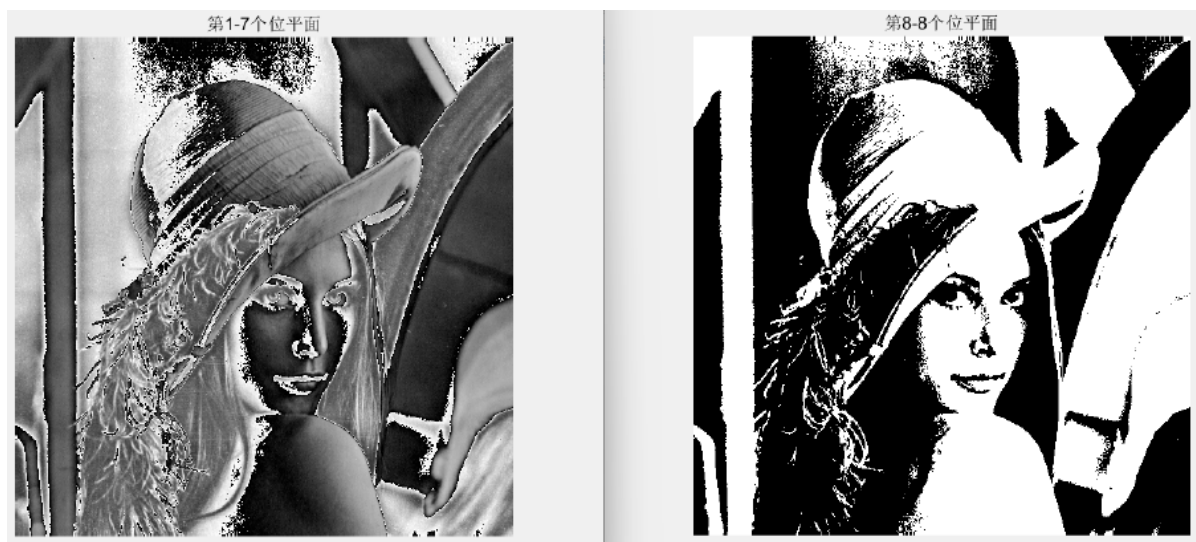


第6-8个位平面



第7-8个位平面





去掉1~n位平面后的图像的显示

代码部分

```
% Clear Memory and Command window
clc;
clear all;
close all;
img = imread("Lena.bmp");
%k = input("请输入n的值: ");
for k=1:7
    [a,b]=size(img);

    for n=1:k
        for i=1:a
            for j=1:b
                img(i,j)=bitset(img(i,j),n,0);
            end
        end
    end

    figure;
    imshow(img, []);
    title(['去除前', num2str(k), '个位平面']);
end
```

结果分析

去除前1个位平面



去除前2个位平面



去除前3个位平面



去除前4个位平面



去除前5个位平面



去除前6个位平面



去除前7个位平面



实验总结

通过本次的实验，我们更加清晰直观的了解到了一张灰度图像具有的8个位平面各自的作用效果。由于处于低位的位平面含有的图像信息较少，并且对原图的影响相对较少，所以我们可以将少量的子密钥图像存放到低位平面中，或者可以直接将其他图像隐藏到较低的位平面中。