MATLAB学习笔记 隐写技术-将消息/图像隐藏在图像中

原创

bashendixie5 已于 2022-04-09 11:02:35 修改 2121 收藏 6
分类专栏： matlab # 图像处理 文章标签： matlab 隐写技术 图像处理 位平面切片 位平面压缩
于 2022-04-04 11:17:16 首次发布

matlab 同时被 2 个专栏收录

19 篇文章 1 订阅
订阅专栏

图像处理

90 篇文章 2 订阅
订阅专栏

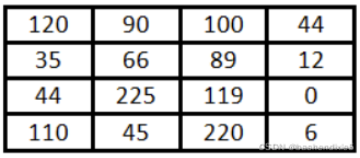
一、隐写技术

1、概述

隐写术是一种将秘密信息隐藏在另一个文档或图像中并由不同的人提取的技术。我们将了解如何在原始图像的修改不被察觉的情况下将文本消息或图像隐藏在图像中。

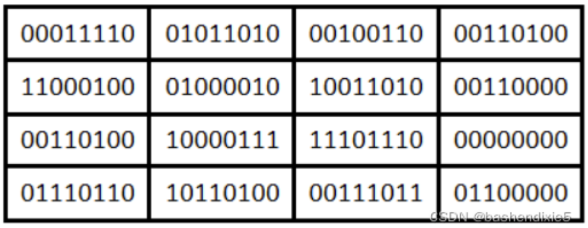
2、流程简述

（1）有这样一个矩阵



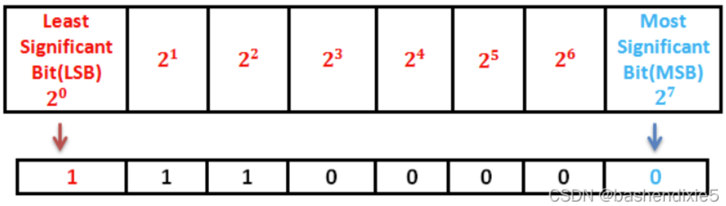
120	90	100	44
35	66	89	12
44	225	119	0
110	45	220	6

（2）使用"de2bi"方法转为二进制格式



00011110	01011010	00100110	00110100
11000100	01000010	10011010	00110000
00110100	10000111	11101110	00000000
01110110	10110100	00111011	01100000

de2bi方法默认采用right MSB顺序。

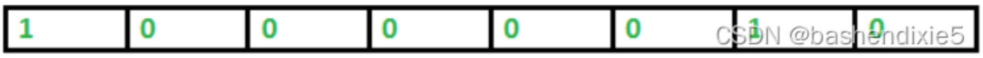


Least Significant Bit(LSB) 2^0	2^1	2^2	2^3	2^4	2^5	2^6	Most Significant Bit(MSB) 2^7
1	1	1	0	0	0	0	0

表示十进制的7

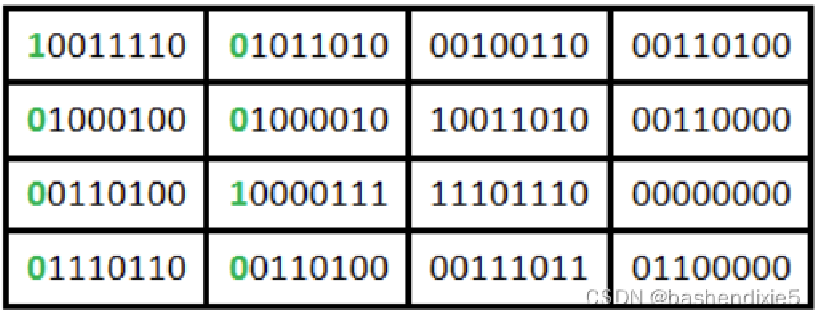
（3）'A'和'h'对应的数值分别为65和104

65的二进制表示



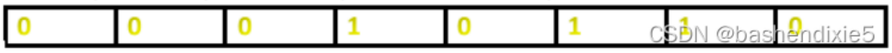
1	0	0	0	0	0	1	0

替换矩阵中部分LSB位。

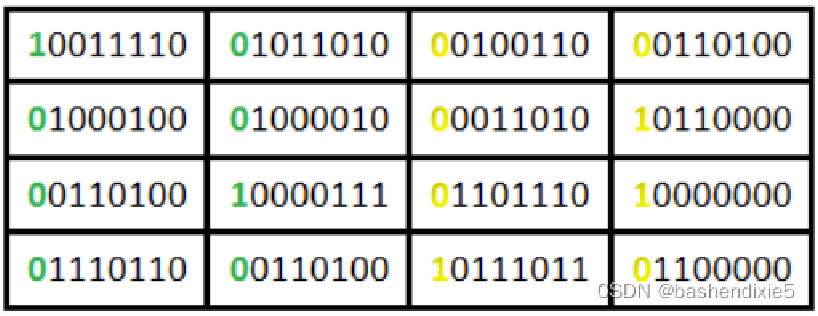


10011110	01011010	00100110	00110100
01000100	01000010	10011010	00110000
00110100	10000111	11101110	00000000
01110110	00110100	00111011	01100000

（4）同样处理h，104

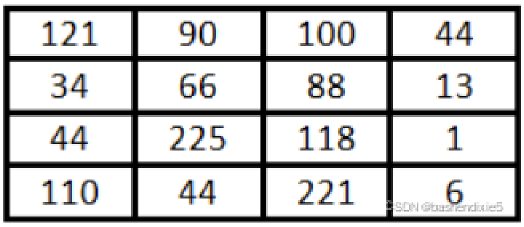


0	0	0	1	0	1	1	0



10011110	01011010	00100110	00110100
01000100	01000010	00011010	10110000
00110100	10000111	01101110	10000000
01110110	00110100	10111011	01100000

（5）最终的矩阵变为



121	90	100	44
34	66	88	13
44	225	118	1
110	44	221	6

二、matlab示例 隐藏文字消息

1、加密代码

要加密的字符串为“this is a matlab demo, if this lenght small”，加密后的图片保存为Encrypt1.png。注意消息的长度和图像的大小的匹配，不能太短也不能太长。

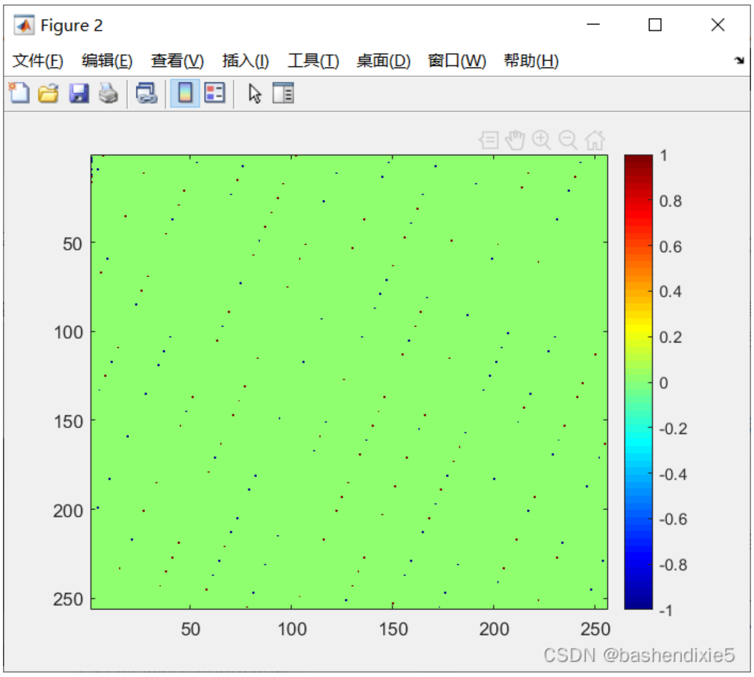
```
%ENCRYPTION

clear all
%READ THE INPUT IMAGE
A = imread('图片/cameraman.tif');

%
Output = A;

%MESSAGE TO BE HIDDEN INSIDE THE IMAGE
Str = 'this is a matlab demo, if this lenght small';

%NUMBER OF CHARACTERS IN THE MESSAGE
numLetters = numel(Str);

%REPRESENT THE LENGTH OF THE MESSAGE IN BINARY FORMAT
Bi_Total = de2bi(numLetters,8);
Bi_Msg = de2bi(double(Str),8)';

Encrypt_msg = Bi_Msg(:)';

%FIND THE LENGTH OF THE MESSAGE IN THE BINARY FORMAT
num_En_msg = numel(Encrypt_msg);


%DEFINE THE INTERVAL BETWEEN EACH VALUE
dt=floor(numel(A)/num_En_msg);

%BINARY VALUE OF THE INTERVAL
Bi_dt = de2bi(dt,8);

% FIRST 8 BITS - INTERVAL
% SECOND 8 BITS - TOTAL LENGTH OF THE BINARY VALUES
Info_data = [Bi_dt,Bi_Total];
Output(1:16) = bitset(A(1:16),1,Info_data);

%REPLACE THE LEAST SIGNIFICANT BIT IN THE IMAGE
Start_pt =17;
End_pt = (num_En_msg*dt)+16;
Output(Start_pt:dt:End_pt) = bitset(A(Start_pt:dt:End_pt),1,Encrypt_msg);


figure,subplot(121),imshow(A);title('Original Image');
subplot(122),imshow(Output);title('Image with the hidden message');

%SHOW THE DIFFERENCE BETWEEN THE ORIGINAL AND THE ENCRYPTED IMAGE
figure,imagesc(double(A)-double(Output));colormap(jet);colorbar;

%WRITE THE ENCRYPTED IMAGE IN THE PNG FORMAT
imwrite(Output,'Encrypt1.png');

display('Message is hidden inside the image!');
```

通过将消息转换为二进制格式并将其放置在图像像素的最低有效位中，将消息隐藏在图像中。加密图像保存为“PNG”图像文件，以便进行无损压缩。使用无损压缩技术，像素值不会改变，因此嵌入在 LSB 中的信息不会丢失。

原始图像和加密图像之间的最大绝对值差为 1。因为我们只处理图像的 LSB 位。

左：原始图像；右：隐藏消息的图像

我们可以将秘密消息的位以特定的模式或以规则的间隔放置，而不是替换彼此相邻的像素。下图表明这些位以特定的间隔值放置。



2、解密代码

```
%DECRYPT THE MESSAGE
clear all
    clc
    %READ THE ENCRYPTED IMAGE
    A = imread('Encrypt1.png');

    %FIRST 8 BITS REPRESENT THE INTERVAL
    Dt = bitget(A(1:8),1);
    Dt = double(bi2de(Dt));


    %LENGTH OF THE TEXT MESSAGE
    Len = bitget(A(9:16),1);
    Len = double(bi2de(Len));

    %FETCH THE LEAST SIGNIFICANT BIT FROM THE ENCRYPTD IMAGE
    Start_pt = 17;
    End_pt = (Len*Dt*8)+16;
    Decrypt_msg = bitget(A(Start_pt:Dt:End_pt),1);

    %CONVERT THE IMAGE INTO DECIMAL
    Decrypt_msg = reshape(Decrypt_msg,8,[])';
    Dec_Char   = bi2de(Decrypt_msg)';

    display('Text Message inside the image:');
    display(char(Dec_Char));
```

运行代码，会输出，解密成功。

```
Text Message inside the image:
this is a matlab demo, if this lenght small
```

三、matlab示例 隐藏图片

1、图片示例

用于隐藏图片的图片

待隐藏的图片，注意待隐藏的图片应小于大图的10%

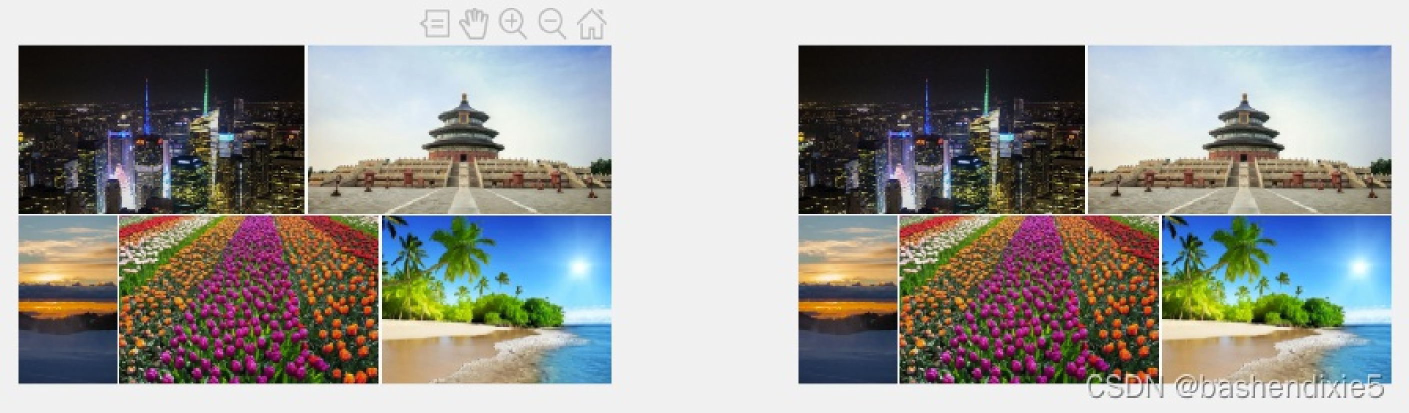
图片违规！

1、加密

参考代码

```
%ENCRYPTION
    clear all

    %READ THE COVER IMAGE
    I = imread('123.png');

    %READ THE IMAGE TO HIDE
    J = imread('a12345.png');

    %PREALLOCATE THE OUTPUT IMAGE
    Output = zeros(size(I));
    %REPRESENT THE NUMBER OF ROWS AND COLUMNS OF THE SECRET IMAGE
    %IN BINARY FORMAT
    Jsize = de2bi([size(J,1),size(J,2)]);

    %RED,GREEN AND BLUE COMPONENTS OF THE SECRET IMAGE
    Red_Ch = J(:,:,1);
    Green_Ch = J(:,:,2);
    Blue_Ch = J(:,:,3);

    %CONVERT THE RED COMPONENT OF THE SECRET IMAGE INTO BINARY FORMAT
    Encrypt_Red = de2bi(Red_Ch(:),8)';
    Encrypt_Red = Encrypt_Red(:)';

    %CONVERT THE GREEN COMPONENT OF THE SECRET IMAGE INTO BINARY FORMAT
    Encrypt_Green = de2bi(Green_Ch(:),8)';
    Encrypt_Green = Encrypt_Green(:)';

    %CONVERT THE BLUE COMPONENT OF THE SECRET IMAGE INTO BINARY FORMAT
    Encrypt_Blue = de2bi(Blue_Ch(:),8)';
Encrypt_Blue = de2bi(Blue_Ch(:),8)';
Encrypt_Blue = Encrypt_Blue(:)';

%CALCULATE THE INTERVAL AND CONVERT IT INTO BINARY FORMAT
dt=floor((size(I,1)*size(I,2))/numel(Encrypt_Red));
Bi_dt = de2bi(dt,8);

Info_data = [Bi_dt,Jsize(1,:),Jsize(2,:)];

%RED,GREEN AND BLUE COMPONENT OF THE COVER IMAGE
Red_mat = I(:,:,1);
Green_mat = I(:,:,2);
Blue_mat = I(:,:,3);

% INTERVAL,NUMBER OF COLUMNS, NUMBER OF ROWS
Red_mat(1:24) = bitset(Red_mat(1:24),1,Info_data);
Green_mat(1:24) = bitset(Green_mat(1:24),1,Info_data);
Blue_mat(1:24) = bitset(Blue_mat(1:24),1,Info_data);

%%REPLACE THE LEAST SIGNIFICANT BIT OF THE COVER IMAGE
%WITH THE BINARY FORMAT OF THE SECRET IMAGE
EndValue = numel(Encrypt_Red)*dt+24;
Red_mat(25:dt:EndValue) = bitset(Red_mat(25:dt:EndValue),1,Encrypt_Red);
Green_mat(25:dt:EndValue) = bitset(Green_mat(25:dt:EndValue),1,Encrypt_Green);
Blue_mat(25:dt:EndValue) = bitset(Blue_mat(25:dt:EndValue),1,Encrypt_Blue);

%ENCRYPTED IMAGE
Output(:,:,1) = Red_mat;
Output(:,:,2) = Green_mat;
Output(:,:,3) = Blue_mat;
Output = uint8(Output);

figure,subplot(121),imshow(I); subplot(122),imshow(Output);

%DIFFERENCE BETWEEN THE ORIGINAL AND THE ENCRYPTED IMAGE
figure,imagesc(double(I)-double(Output));colormap(jet);colorbar;

%WRITE THE ENCRYPTED IMAGE IN THE PNG FORMAT
imwrite(Output,'Encrypt_Image2.png');
```

原始图像和加密后的图像



隐藏的图像在大图中的分布。

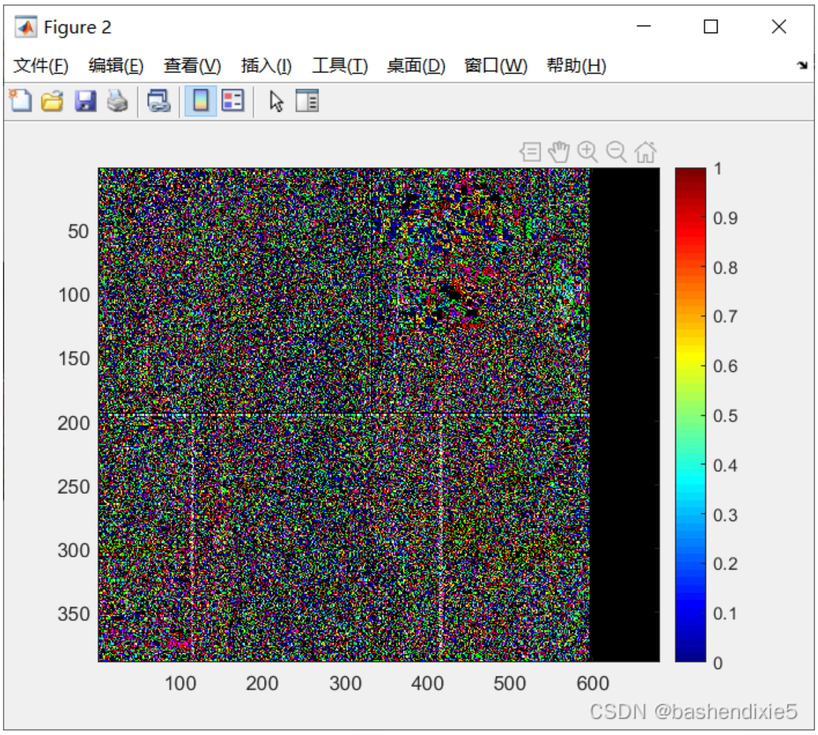


Figure 2
文件(F) 编辑(E) 查看(V) 插入(I) 工具(T) 桌面(D) 窗口(W) 帮助(H)
50
100
150
200
250
300
350
100
200
300
400
500
600
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
CSDN @bashendixie5

2、解密

```
%DECRYPT IMAGE
    clear all
    %READ THE ENCRYPTED IMAGE
    A = imread('Encrypt_Image2.png');
    %RED,GREEN AND BLUE COMPONENTS
    Red_ch = A(:,:,1);
    Green_ch = A(:,:,2);
    Blue_ch = A(:,:,3);
    %INTERVAL
    Dt = bitget(Red_ch(1:8),1);
    Dt = double(bi2de(Dt));
    %NUMBER OF COLUMNS
    Ncols = bitget(Red_ch(9:16),1);
    Ncols = double(bi2de(Ncols));
    %NUMBER OF ROWS
    Nrows = bitget(Red_ch(17:24),1);
    Nrows = double(bi2de(Nrows));
    %FETCH THE LEAST SIGNIFICANT BIT FROM THE ENCRYPTED IMAGE
    EndPoint = (Dt*Nrows*Ncols*8)+24;
    Decrypt_Red = bitget(Red_ch(25:Dt:EndPoint),1);
    Decrypt_Green = bitget(Green_ch(25:Dt:EndPoint),1);
    Decrypt_Blue = bitget(Blue_ch(25:Dt:EndPoint),1);
    Decrypt_Red = reshape(Decrypt_Red,8,[])';
    Decrypt_Red = bi2de(Decrypt_Red);
    Decrypt_Red = reshape(Decrypt_Red,Ncols,Nrows);
    Decrypt_Green = reshape(Decrypt_Green,8,[])';
    Decrypt_Green = bi2de(Decrypt_Green);
    Decrypt_Green = reshape(Decrypt_Green,Ncols,Nrows);
    Decrypt_Blue = reshape(Decrypt_Blue,8,[])';
    Decrypt_Blue = bi2de(Decrypt_Blue);
    Decrypt_Blue = reshape(Decrypt_Blue,Ncols,Nrows);
    Decrypted_Image = zeros([Ncols,Nrows,3]);
    Decrypted_Image(:,:,1)=Decrypt_Red;
    Decrypted_Image(:,:,2)=Decrypt_Green;
    Decrypted_Image(:,:,3)=Decrypt_Blue;
    Decrypted_Image = uint8(Decrypted_Image);
    imwrite(Decrypted_Image,'Decrypted_Image2.png');
    figure,imshow(Decrypted_Image);title('Secret Image');
```

解密出来的秘密图像。