




数字水印图像攻击测试实验MATLAB实现

原创

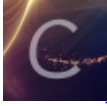
尼克选手  于 2020-05-14 18:44:47 发布  5074  收藏 85

分类专栏: [图像处理](#) [数字水印](#) 文章标签: [计算机视觉](#) [matlab](#)

版权声明: 本文为博主原创文章, 遵循 [CC 4.0 BY-SA](#) 版权协议, 转载请附上原文出处链接和本声明。

本文链接: https://blog.csdn.net/weixin_44132627/article/details/106125803

版权



[图像处理](#) 同时被 2 个专栏收录

7 篇文章 7 订阅

订阅专栏



[数字水印](#)

4 篇文章 5 订阅

订阅专栏

继上次介绍了三种基本的图像数字水印算法以及代码后, 其中大部分实验者都需要进行水印的抗攻击实验来测量算法的性能。上次文中也提到后期会介绍图像攻击算法, 今天就介绍几个简单的图像攻击。

一、展示攻击代码

为了服务上次文章提到的数字水印攻击实验, 先将代码展示如下吧, 具体的使用接下来说明。

```

%% 进行攻击测试 %%
disp('1-->加入白噪声');
disp('2-->高斯低通滤波');
disp('3-->剪切图像');
disp('4-->旋转攻击');
disp('5-->直接检测')
begin=input('请选择攻击 (1-5) : ')
switch begin
    % 加入白噪声
    case 1
        Aimage1=wimage;
        Wnoise=20*randn(size(Aimage1));
        Aimage1=Aimage1+Wnoise;
        subplot(2,3,4),imshow(Aimage1,[]),title('加入白噪声后的图像');
        att=Aimage1;
        %imwrite(att,'whitenoiseimage.bmp');
        % 高斯低通滤波
    case 2
        Aimage2=wimage;
        H=fspecial('gaussian',[4,4],0.5);
        Aimage2=imfilter(Aimage2,H);
        subplot(2,3,4),imshow(Aimage2,[]),title('高斯低通滤波后的图像');
        att=Aimage2;
        %imwrite(att,'gaussianimage.bmp')
        % 剪切攻击
    case 3
        Aimage3=wimage;
        Aimage3(1:128,1:128)=256;
        subplot(2,3,4),imshow(Aimage3,[]),title('剪切后的图像');
        att=Aimage3;
        %imwrite(att,'cutpartimage.bmp');
        % 旋转攻击
    case 4
        Aimage4=wimage;
        Aimage4=imrotate(Aimage4,0.55,'bilinear','crop');
        Aimage_4=mat2gray(Aimage4);
        subplot(2,3,4),imshow(Aimage_4,[]),title('旋转10 度后的图像');
        att=Aimage_4;
        imwrite(att,'rotatedimage.bmp');
        % 没有受到攻击
    case 5
        subplot(2,3,4),imshow(wimage,[]),title('直接提取的图像');
        att=wimage;
        %imwrite(att,'directimage.bmp');
end

```

代码说明：从上面代码可以看出，我未将代码封装为函数，目的也是为了让使用者使用方便。那么，在数字水印实验中怎能试用呢？很简单，将这段代码复制粘贴插入到完成水印嵌入的步骤后面（使用时注意变量名的衔接和更改），通过这段代码进行水印的攻击之后再**提取**，完成水印攻击测试。如果您习惯将水印的嵌入、攻击、提取分别封装函数，您当然也可将此段代码封装为独立的函数。

二、代码效果测试

为了直观的说明代码的运行效果，我将上述代码稍加更改新建了一个脚本进行运行测试，代码如下：

```

clc
clear all;

I = imread('lena256.bmp');

Aimage1=I;
Wnoise=uint8(20*randn(size(Aimage1)));
%Wnoise=20*randn(size(Aimage1),'like',wimage);
Aimage1=Aimage1+Wnoise;
subplot(2,3,1),imshow(Aimage1,[]),title('加入白噪声后的图象');

% 高斯低通滤波
Aimage2=I;
H=fspecial('gaussian',[4,4],0.5);
Aimage2=imfilter(Aimage2,H);
subplot(2,3,2),imshow(Aimage2,[]),title('高斯低通滤波后的图象');

% 剪切攻击
Aimage3=I;
image_size=size(I);
dimension = numel(image_size);
if dimension == 3
    Aimage3_r = Aimage3(:,:,1);
    Aimage3_r(1:128,1:128)=256;
    Aimage3_g = Aimage3(:,:,2);
    Aimage3_g(1:128,1:128)=256;
    Aimage3_b = Aimage3(:,:,3);
    Aimage3_b(1:128,1:128)=256;

    Aimage3(:,:,1) = Aimage3_r;
    Aimage3(:,:,2) = Aimage3_g;
    Aimage3(:,:,3) = Aimage3_b;
end
% 如果是灰度图像
if dimension == 2
    Aimage3(1:128,1:128)=256;
end
subplot(2,3,3),imshow(Aimage3,[]),title('剪切后的图象');

% 旋转攻击
Aimage4=I;
Aimage4=imrotate(Aimage4,10,'bilinear','crop');
Aimage_4=mat2gray(Aimage4);
subplot(2,3,4),imshow(Aimage_4,[]),title('旋转10 度后的图象');

% 没有受到攻击
subplot(2,3,5),imshow(I,[]),title('无攻击图像');

```

其运行结果如下图：

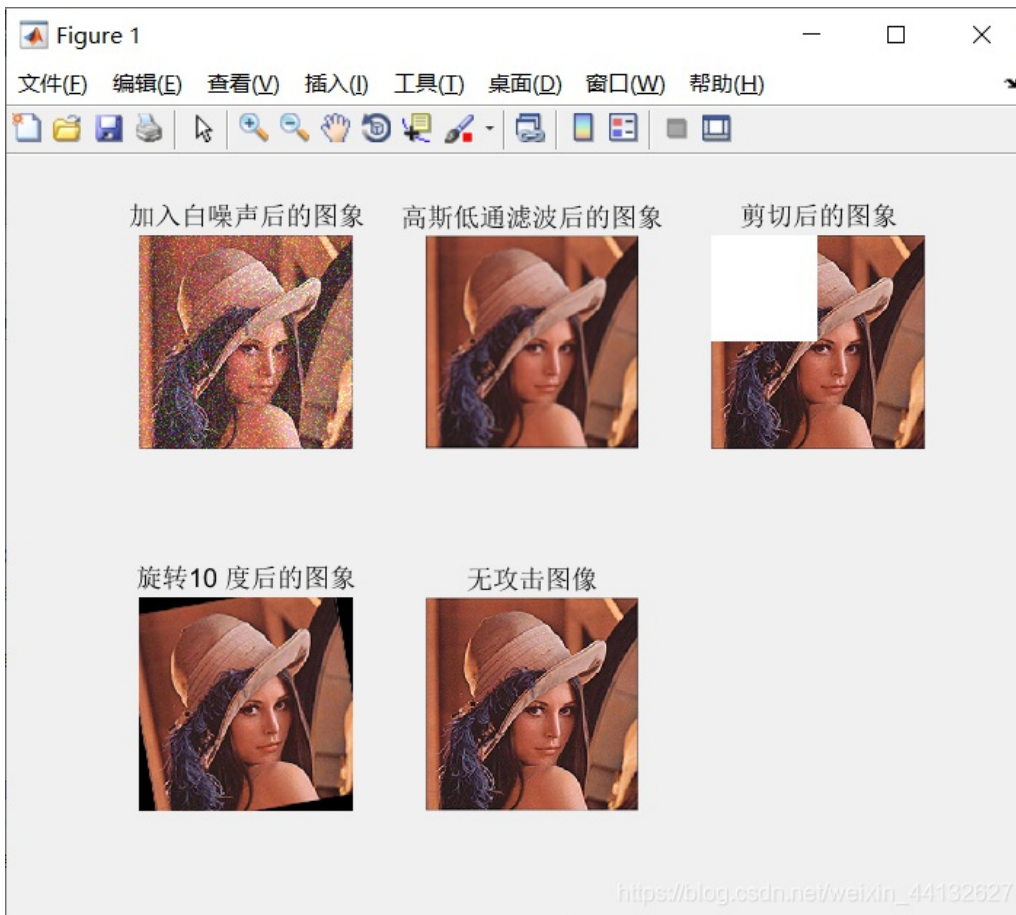


图1 图像攻击结果

这只是几种简单攻击测试，还有很多攻击可以提供我们选择，在噪声中也可以加入椒盐噪声等等，可根据自己需求进行图像的攻击测试。