浙江大学

课程名称:	多媒体安全
姓　　名:	
学　　院:	计算机学院
专　　业:	计算机科学与技术
学　　号:	
指导老师:	黄劲
完成时间:	2023 年 5 月 7 日

实验二：E_SIMPLE_8/D_SIMPLE_8 系统测试

一、实验目的

1. 理解 E_SIMPLE_8/D_SIMPLE_8 系统的基本原理，掌握简单的多位信息水印技术。

二、实验内容与要求

1. 实现 E_SIMPLE_8/D_SIMPLE_8 系统。
2. 设计一张水印，嵌入强度 $\alpha = \sqrt{8}$，使用该水印测试 E_SIMPLE_8/D_SIMPLE_8 系统应用于不同封面时的检测准确率，计算 False Positive/Negative Rate 和解码的准确率。要求封面数量不少于 40 张。False Positive/Negative Rate 的计算可以采取不同的原则。其中一种可以使用的原则是，预先设定一个固定的阈值，8 个检测值（detect value）中有 4 个超过了阈值，就认为存在水印，否则认为不存在水印。(也可以使用其他合理的原则，需要在报告中说明使用的是哪种原则)。准确率的计算，则是对确实添加了水印的图片，计算解码出来的信息的错误率。
3. 设计不少于 40 张不同的水印, 使用固定的嵌入强度 $\alpha = \sqrt{8}$, 测试 E_SIMPLE_8/D_SIMPLE_8 系统应用于同一封面时的检测准确率，计算 False Positive/Negative Rate。
4. 分析信息长度增加对检测准确率的影响。

三、实验环境

语言版本：MATLAB R2020b

四、实验过程

4.1 E_SIMPLE_8/D_SIMPLE_8 系统实现

此部分主要是实现 E_SIMPLE_8 水印生成和 D_SIMPLE_8 水印检测。

为实现 E_SIMPLE_8 水印生成，我们需要采用一些随机分布函数随机出 8 种水印模式。由于提供的测试集中，最大大小的图片为 5312*5312，我们使用 randn() 函数生成均值为 0、方差为 1 的 5312*5312 的 double 类型矩阵，作为 1 种水印模式。为保证 8bit 信息水印仍然是一个均值为 0、方差为 1 的矩阵，我们把每一个存储的水印模式除以 8，在需要植入时，把 8 个水印模式简单相加即可得到 1 的方差。

定义函数 E_SIMPLE_8() 根据随机数种子 seed 返回水印模式 watermark（8 页），实现如下：

```
function [watermark] = E_SIMPLE_8(seed)

    siz = 5312;
    watermark = zeros([siz, siz, 8]);
    rng(seed);
    for i =1:8
        watermark(:, :, i) = randn(siz, siz) / 8;
    end
end
```

生成的水印需要植入封面。首先将水印模式裁剪到图片大小，然后采取逐元素简单相加方式即可得到含水印的图片。在植入的信息以外，另需要一个水印强度参数 α 控制水印的鲁棒性。

定义函数 embed() 根据封面 cover，水印模式 watermark，植入信息 infm，水印强度 alpha 返回含水印的封面 markedWork，实现如下:

```
function [markedWork] = embed(cover, watermark, infm, alpha)

    [sizeX, sizeY] = size(cover);
    markedWork = cover;
    for i = 1:8
        mki = watermark(:, :, i);
        mki = mki(1:sizeX, 1:sizeY);
        markedWork = markedWork + alpha * infm(i) * mki;
    end
end
```

对于 D_SIMPLE_8 检测系统，我们采取 D_CC 检测方法实现。D_CC 的检测原理是计算图片向量与水印向量的相关系数，如有明显的绝对值，则认为水印存在。定量的检测函数是:

$$z_{cc}(c, w) = c' * w'$$

其中 c' 和 w' 分别是 c 和 w 归一化后的向量。

如果 z_{cc} 的绝对值大于人为设定的检测阈值 τ_{cc}，则认为图片中存在水印，并根据 z_{cc} 的符号判定是正或负水印。否则，认为未植入水印。

定义函数 D_CC() 根据待检测图像 img 和水印模式 watermark 返回水印判定值,定义函数 detect() 根据判定值 result 检测水印是否存在，取阈值 $\tau = 0.004$，分别实现如下:

```
function [result] = D_CC(img, watermark)
```

```
2
3       [sizeX, sizeY] = size(img);
4       watermark = watermark(1:sizeX, 1:sizeY);
5       result = correlation(img, watermark);
6   end
```

```
1   function [result] = detect(result)
2
3       tau = 0.004;
4       for i = 1:8
5           if result(i) > tau
6               result(i) = 1;
7           elseif result(i) < -tau
8               result(i) = -1;
9           else
10              result(i) = 0;
11          end
12      end
13
14      result = (sum(abs(result)) >= 4);
15  end
```

D_SIMPLE_8 解码器的实现如下:

```
1   function [infm] = D_SIMPLE_8(cover, watermark)
2
3       [sizeX, sizeY] = size(cover);
4       infm = zeros([1, 8]);
5       for i = 1:8
6           mki = watermark(:, :, i);
7           mki = mki(1:sizeX, 1:sizeY);
8           infm(i) = D_CC(cover, mki);
9       end
10  end
```

以上已经实现 E_SIMPLE_8/D_SIMPLE_8 系统的大部分工作，只需人为给出 τ_{cc} 等其他参数

以及判定逻辑即可。

对于有无水印的判定原则: 在 8 个水印模式中检测出 4 个以上超过阈值则认为图片有水印, 否则认为无水印。

4.2 相同水印应用于不同封面的检测

设定随机数种子 seed 为 19260817, 取给定数据集中文件名字典序最小的 40 张图片作为测试封面, 植入随机 8bit 信息 infm (检测阈值 $\tau = 0.004$)。

```
watermark = E_SIMPLE_8(19260817);
infm = (randi(2, [1, 8]) - 1) * 2 - 1;
fp = 0;
fn = 0;
er = 0;
for i = 1:40
    image = imread(fullfile('data', files(i).name));
    image = cast(image, "double");

    r1 = D_SIMPLE_8(image, watermark);

    markedImage = embed(image, watermark, infm, sqrt(8));
    r2 = D_SIMPLE_8(markedImage, watermark);

    if detect(r1) == 1
        fp = fp + 1;
    end
    if detect(r2) == 0
        fn = fn + 1;
    end

    flag = 0;
    for j = 1:8
        if sign(r2(j)) ~= sign(infm(j))
            flag = 1;
        end
    end
```

```
    er = er + flag;
end

fp = fp / 40
fn = fn / 40
er = er / 40
```

所得检测结果为: 准确率 95%, 假阳性率 5%, 假阴性率 2.5%。

4.3 不同水印应用于相同封面的检测

选取数据集中的 lena512.bmp 作为测试封面, 水印种子为 1 到 40, 植入随机 8bit 信息 infm (检测阈值 $\tau = 0.004$)。

```
image = imread(fullfile('data', 'lena512.bmp'));
image = cast(image, "double");
infm = (randi(2, [1, 8]) - 1) * 2 - 1;
fp = 0;
fn = 0;
er = 0;
for i = 1:40
    watermark = E_SIMPLE_8(i);

    r1 = D_SIMPLE_8(image, watermark);

    markedImage = embed(image, watermark, infm, sqrt(8));
    r2 = D_SIMPLE_8(markedImage, watermark);

    if detect(r1) == 1
        fp = fp + 1;
    end
    if detect(r2) == 0
        fn = fn + 1;
    end

    flag = 0;
```

```
    for j = 1:8
        if sign(r2(j)) ~= sign(infm(j))
            flag = 1;
        end
    end
    er = er + flag;
end

fp = fp / 40
fn = fn / 40
er = er / 40
```

所得检测结果为: 准确率 100%, 假阳性率 0, 假阴性率 0。

4.4 信息长度对检测准确率的影响

仍然选取数据集中的 lena512.bmp 作为测试封面, 让信息长度从 1B 到 1kB 逐渐增加, 每 8bit 为一组进行解码计算错误率, 每 8B 计算一次。

```
image = imread(fullfile('data', 'lena512.bmp'));
image = cast(image, "double");
infm = (randi(2, [1, 8]) - 1) * 2 - 1;
ers = [];
for i = 1:1024
    er = 0;

    watermark = E_SIMPLE_8(i);
    image = embed(image, watermark, infm, sqrt(8));

    if mod(i, 64) == 0
        for j = 1:i
            watermark = E_SIMPLE_8(j);
            res = D_SIMPLE_8(image, watermark);
            for k = 1:8
                if sign(res(k)) ~= sign(infm(k))
                    er = er + 1;
```

```
                end
            end
        end
        ers = [ers, er / i / 8]
    end
end
```

得到的错误率在 832B 之前全部为 0，在 896B 是 $0.1375 * 10^{-3}$，在 950B 是 $0.5208 * 10^{-3}$，在 1024B 是 $0.4883 * 10^{-3}$。

可以看出，随着信息长度增大，检测的准确率逐渐降低。

五、实验分析与结论

5.1 检测准确率分析

由于在本次实验中使用了 D_CC 进行解码，检测准确率比实验一要相对高。

在多 bit 信息中，水印强度和检测能力高度挂钩。由于需要在同一张水印图片中插入多个水印模式而又要保持一定的均值和方差，为检测出某一码字所需的水印强度也必然需要增加。

由于本次实验所需的信息是 8bit，因此每个水印模式在整个水印中的强度约为 $\frac{1}{\sqrt{8}}$，因此我们也设定 $\alpha = \sqrt{8}$，以达成适当的检测效果。

同时，鉴于 D_CC 解码的灵敏性，τ 值有必要取很小（如同书上要求对未植入水印的图片解码的信息随机，即 $\tau = 0$）。因此，根据设定的 τ 值，得到的准确率、假阳性和假阴性率相差会较大。

5.2 信息长度增加对检测准确率的影响

根据数学上的分析，我们可以知道，信息长度越增加，检测准确率越低；这是因为生成的许多水印模式不能保证低线性相关。因此，在信息效率和水印鲁棒性之间存在一定的取舍。

但实验中，我们得到的检测错误率还是很低。分析原因，应该是：D_CC 解码太过灵敏，加上水印向量空间太大，使得植入很多 bit 后错误率也很低。1024 张水印远不能在 512*512 的浮点数向量空间中得到很大的相关性。但可以想到，如果水印信息长度继续增加，检测准确率会急剧下降。

由于验证此事的运行时间太长，此处不做过多的定量分析。

六、实验感想

采取 D_CC 进行解码的灵敏度太大，使我难以分析消息长度增加后的实验结果，从而只能通过理论上的分析来占据篇幅。可能在最后采取 D_LC 错误率会高一些，但是似乎也难以保证是因为信息变

长才导致的。

通过本实验，我理解了多信息位水印和检测的方法及原理，有助于后续对课程的学习。