

数字内容安全 实验报告



姓 名 项 枫
学 号 2022211570
指导教师 张 茹
学 院 网络空间安全学院

2024 年 5 月 16 日

实验名称 感知哈希算法设计实验 实验日期: 2024 年 5 月 16 日 指导老师 张茹 得分

学院 网络空间安全学院 专业 信息安全 班次 2022211801 姓名 项枫 学号 2022211570

一、实验目的

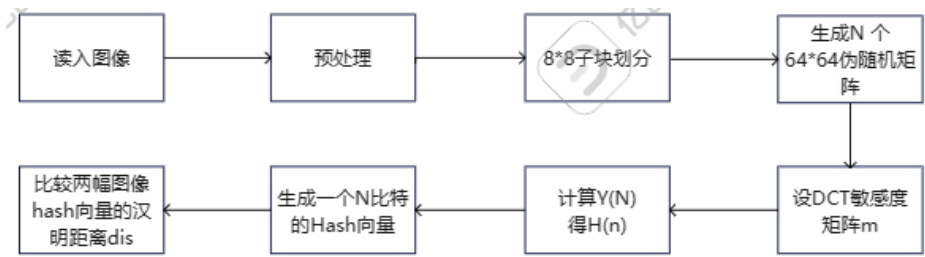
了解鲁棒性内容感知哈希技术的基本特点，设计并实现基于 DCT 的鲁棒性图像感知哈希。了解鲁棒性内容感知哈希技术在数字内容保护中的作用，掌握基于鲁棒性感知哈希的内容保护方法。

二、实验内容

本实验实现一种基于视觉特性的图像感知哈希算法，可通过密钥控制哈希序列，保证安全性。

三、系统整体描述和分功能描述

系统整体描述



分功能描述

- 1) 分功能 1: 预处理
用到的函数: `rgb2gray(I)` , `imresize(I, [64, 64])`
- 2) 分功能 2: 8*8 子块划分
用到的函数: `fun = @dct2`, `Ic = blkproc(I, [8 8], fun)`
- 3) 分功能 3: 生成 N 个 64*64 伪随机矩阵
用到的函数: `randn('state', key)`; `K=fspecial('gaussian')`; `Y=filter2(K, Y)`
- 4) 分功能 4: 设 DCT 敏感度矩阵 m
用到的函数: `M = repmat(m, 8, 8)`
- 5) 分功能 5: 计算 Y(n), 得 H(n)
用到的函数: `I_sum = I_sum + I(i, j) * Y{k}(i, j) * M(i, j)`
- 6) 分功能 6: 生成一个 N 比特的 Hash 向量
用到的函数: `I_Hash(k)`
- 7) 分功能 7: 比较两幅图像 hash 向量的汉明距离 dis
用到的函数:
$$dis = norm((I1_Hash - I2_Hash) / 2 * \sqrt{norm(I1_Hash) * norm(I2_Hash)})$$

四、实验步骤、结果及分析

实验步骤

- 1、读入两幅图像，对图像做预处理：如果读入的是彩色图像，将其转换为灰度图像；在灰度图像中利用差值方式将图像重采样为 64*64 的标准化图表示。
- 2、对标准化图像进行 8*8 子块划分，将标准化图像划分为 $(64*64) / (8*8) = 64$ 个子块，依次对各子块进行二维离散余弦变换，并依次将各分块的 DC 系数，即 (1, 1) 置为 0。
- 3、生成 N 个 64*64 伪随机矩阵：首先通过密钥伪随机生成服从标准正态的 64*64 矩阵，然

后用高斯低通滤波器进行迭代滤波。

4、设 DCT 敏感度矩阵 m

```
m = [71.43 99.01 86.21 60.24 41.67 29.16 20.88 15.24
99.01 68.97 75.76 65.79 50.00 36.90 27.25 20.28
86.21 75.76 44.64 38.61 33.56 27.47 21.74 17.01
60.24 65.79 38.61 26.53 21.98 18.87 15.92 13.16
41.67 50.00 33.56 21.98 16.26 13.14 11.48 9.83
29.16 36.90 27.47 18.87 13.14 10.40 8.64 7.40
20.88 27.25 21.74 15.92 11.48 8.64 6.90 5.78
15.24 20.28 17.01 13.16 9.83 7.40 5.78 4.73]
```

对矩阵 m 进行周期延拓得到大小为 64×64 的矩阵 M ，并将其每个元素作为 I_c 的对应位置频率系数在特征值计算中的权。

5、取第一个伪随机矩阵，计算

$$Y_n = \sum_{i=1}^{s_1} \sum_{j=1}^{s_2} [I_c(i, j) \cdot P_n(i, j) \cdot M(i, j)]$$

如果 $Y(n) < 0$ ，则 $H(n)=0$ ，反之则 $H(n)=1$ 。

6、循环第 5 步，直到将所有的 N 个 64×64 伪随机矩阵都计算完，最终生成一个 N 比特的 Hash 向量。

7、比较两幅图像 hash 向量的汉明距离 dis ，设定一个合适的阈值 τ ，如果 $dis < \tau$ ，则两幅图像内容一致；反之则是内容不同的两幅图像。

8、以 DemoImages 中的图片为测试集，分析实验结果。

上述过程代码如下：

main.m

```
clc;clear;
figure('NumberTitle','off','Name','图片相似度比较');
key = 1;
tau = 0.225;
Hashlen = 1000;

% 获取 DemoImages 中全部图片路径
img_path = dir('D:\DCS-LAB\1\DemoImages');
img_path = img_path(~[img_path.isdir]);
fileList = fullfile({img_path.folder}.', {img_path.name}.');

I1 = imread('D:\DCS-LAB\1\DemoImages\1_1.bmp');
subplot(4,ceil(length(fileList)/4),1);imshow(I1);title('基准图');
for i = 2:length(fileList)
    I2 = imread(fileList{i});
    ds = imgHashSimilar(I1,I2,tau,key,Hashlen);
    disp(fileList{i})
    subplot(4,ceil(length(fileList)/4),i);imshow(I2);title(ds);
end
```

imgHashSimilar.m

```
function[ds] = imgHashSimilar(I1,I2,tau,key,Hashlen)
```

```

% 读取两幅图片，并转为灰度图像
I1 = rgb2gray(I1);
if ndims(I2) == 3
    I2 = rgb2gray(I2);
end
% 对图片重采样为 64*64 的标准化图
I1 = imresize(I1,[64,64]);
I2 = imresize(I2,[64,64]);
% 将图片划分成 8*8 子块，并进行二维离散余弦变换
fun = @dct2;
I1 = blkproc(I1,[8 8],fun);
I2 = blkproc(I2,[8 8],fun);
% 分块的 DC 系数，即 (1,1) 置为 0
I1(1,1) = 0;
I2(1,1) = 0;
% 通过密钥伪随机生成 Hashlen 个服从标准正态 64*64 矩阵
randn('state',key);
N = cell(1,Hashlen);
% 用高斯低通滤波器进行迭代滤波
K = fspecial('gaussian');
Y = cell(1,Hashlen);

for i = 1:Hashlen
    N{i} = randn(64);
    Y{i} = filter2(K,N{i});
end
% DCT 敏感度矩阵 m,周期延拓至 64*64
m = [
71.43 99.01 86.21 60.24 41.67 29.16 20.88 15.24;
99.01 68.97 75.76 65.79 50.00 36.90 27.25 20.28;
86.21 75.76 44.64 38.61 33.56 27.47 21.74 17.01;
60.24 65.79 38.61 26.53 21.98 18.87 15.92 13.16;
41.67 50.00 33.56 21.98 16.26 13.14 11.48 9.83;
29.16 36.90 27.47 18.87 13.14 10.40 8.64 7.40;
20.88 27.25 21.74 15.92 11.48 8.64 6.90 5.78;
15.24 20.28 17.01 13.16 9.83 7.40 5.78 4.73];
% 矩阵 m 进行周期延拓得到大小为 64 × 64 的矩阵 M
M = repmat(m,8,8);
I1_Hash = ones(1,Hashlen);
I2_Hash = ones(1,Hashlen);
% 对 Hashlen 个伪随机矩阵遍历计算
for k = 1:Hashlen
    I1_sum = 0;
    I2_sum = 0;

```

```

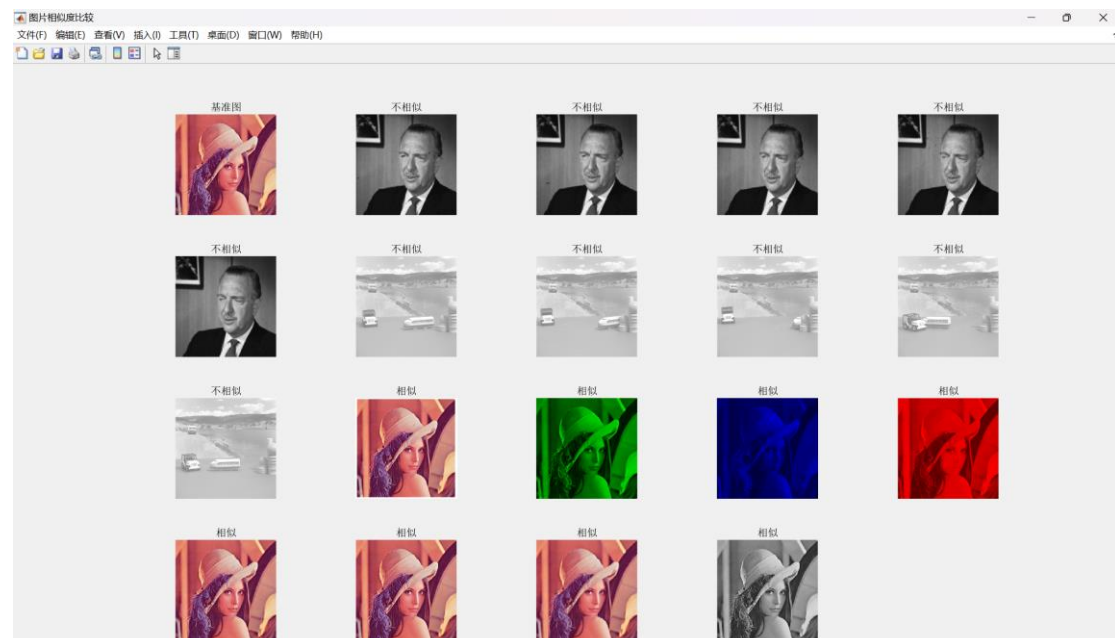
for i = 1:64
    for j = 1:64
        I1_sum = I1_sum + I1(i,j) * Y{k}(i,j) * M(i,j);
        I2_sum = I2_sum + I2(i,j) * Y{k}(i,j) * M(i,j);
    end
end
if I1_sum < 0
    I1_Hash(k) = 0;
end
if I2_sum < 0
    I2_Hash(k) = 0;
end
end
% 汉明距离
dis = norm((I1_Hash-I2_Hash)/(2*sqrt(norm(I1_Hash)*norm(I2_Hash))));
% 与阈值比较
if tau < dis
    ds = '不相似';
else
    ds = '相似';
end
end

```

实验结果及分析

1、实验结果

以 DemoImages 中的图片为测试集，实验结果如下：

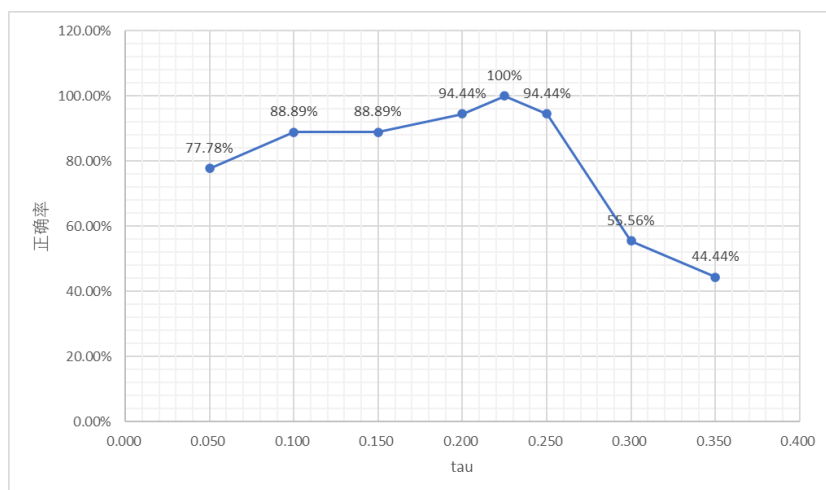


2、分析

(1) 关于 τ 的选取， τ 是一个经验值，其选取依据是根据测试集的准确率决定，给出选取过程。

控制 $N=1000$ ， $\text{key}=1$ 不变，不同 τ ，正确率结果如下：

N=1000 key=1			
tau	正确	不正确	正确率
0.050	14	4	77.78%
0.100	16	2	88.89%
0.150	16	2	88.89%
0.200	17	1	94.44%
0.225	18	0	100%
0.250	17	1	94.44%
0.300	10	8	55.56%
0.350	8	10	44.44%



故 tau 选取 0.225。

(2) 关于 N 的选取, N 值越大 Hash 精度越高, 与不同图像 Hash 碰撞的概率就越小, 但鲁棒性会降低, 因而需设定合适的 N 值, 以满足 Hash 在精度和鲁棒性之间的折衷, 给出选取过程。

当 Hash 的位数决定了其能够表示不同图形的个数, 即最大为 2^n 个。通过实验可以测得在 tau 和 key 不变的情况下, Hash 的位数越大图像感知越精密, 图片相似度判断越精准, 但意味着计算机容错降低, 并且算出的汉明距离变小, 从而更需要 tau 训练出更加精准判别的阈值, 提高了训练成本。因此我认为折衷方案应根据图片数量 N 取较大 2^n 中 n 的数值。

五、实验中遇到的问题及改正的方法

遇到的最大问题是不会书写 matlab 代码, 学习了一点基础知识后, 对 matlab 代码有了初步了解, 最后结合老师所给实验文档写出该代码。