实验二 传统哈希

1. 实验目的：感受传统哈希对输入的敏感性以及其在多媒体领域应用的局限性

二、实验内容：

1）将数字图像空域像素点的值进行微调

2）对数字图像进行传统哈希处理，计算摘要值

3）观察分析原图像与改动后图像摘要值的变化情况

1. 实验过程
2. 使用imread读取图像

\*方法：message=imread(file\_name);

\*说明：图片文件的数据即一个二维数组，这个二维数组存储着一张灰度图片各个像素点的灰度值。

1. 对图像某些像素点进行细微的调整

\* 方法：

%原值

message(1,1); %160

message(340,340); %190

message(50,20); %114

message(20,50); %93

%改变初始阵列，观察哈希值

message\_change\_1(1,1)=159;

message\_change\_2(340,340)=191;

message\_change\_3(50,20)=93;

message\_change\_3(20,50)=114;

\*说明：新设三个原图像矩阵的拷贝，调整二维数组特定位上的数值大小，即可改变图片中该像素点的灰度值，在此次试验中只对很少位的灰度值做细微调整。

1. 使用sha-1算法计算哈希摘要

\*方法：

%使用SHA-1算法

algs='SHA-1';

h1=hash(message,algs)

h2=hash(message\_change\_1,algs)

h3=hash(message\_change\_3,algs)

\*说明：另外使用hash函数生成摘要值，algs为所使用的算法，hash函数展示说明如下：

function h = hash(inp,meth) %输入算法和待操作字符串

inp=inp(:); %把输入矩阵中的数值按列的顺序排为一列

% convert strings and logicals into uint8 format

if ischar(inp) || islogical(inp)

inp=uint8(inp);

else % convert everything else into uint8 format without loss of data

inp=typecast(inp,'uint8'); %转换数据类型

end

% verify hash method, with some syntactical forgiveness:

meth=upper(meth);

switch meth %允许使用的算法

case 'SHA1'

meth='SHA-1';

case 'SHA256'

meth='SHA-256';

case 'SHA384'

meth='SHA-384';

case 'SHA512'

meth='SHA-512';

otherwise

end

algs={'MD2','MD5','SHA-1','SHA-256','SHA-384','SHA-512'};

if isempty(strmatch(meth,algs,'exact'))

error(['Hash algorithm must be ' ...

'MD2, MD5, SHA-1, SHA-256, SHA-384, or SHA-512']);

end

x=java.security.MessageDigest.getInstance(meth); %java接口，使用Java 自带的加密类

x.update(inp); %对输入序列进行处理

h=typecast(x.digest,'uint8'); %转换数据类型

h=dec2hex(h)';

if(size(h,1))==1 % remote possibility: all hash bytes 128, so pad:

h=[repmat('0',[1 size(h,2)]);h]; %B = repmat(A,[m,n])，将矩阵 A 复制 m×n 块，即把 A 作为 B 的元素，B 由 m×n 个 A 平铺而成。size（h，2）：获取矩阵h的列数。将摘要值平铺到一行

end

h=lower(h(:)'); %将hash值转换为小写字符串

clear x

Return

1. 测试数据



figure 4-1 原图与微调后的各图片

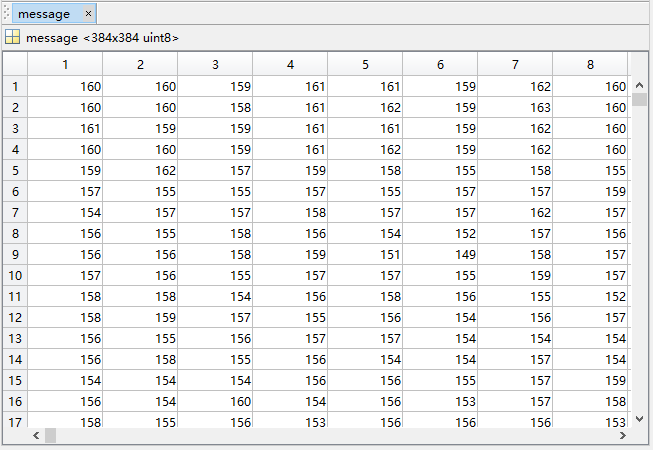


figure 4-3 观察图像以二维矩阵存储



figure 4-2 变换前原图中(0,0) (340,340) (50,20) (20,50) 四个位置处的灰度值

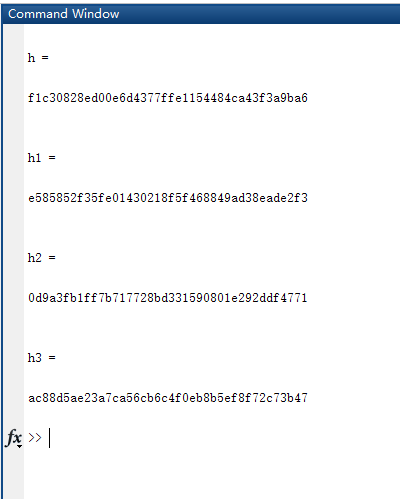


figure 4-4 各图片生成的哈希摘要

1. 实验结果

由该实验可得，传统哈希对输入十分敏感，虽然语义内容相同，但只要像素大小有微小变动得到的摘要值完全不同，无法满足多媒体内容的感知鲁棒性。

1. 附录（实验代码）

Project02.m

%Name: GuoYunting

%Course: 数字内容安全

%aim: 1）将数字图像空域像素点的值进行微调

% 2）对数字图像进行传统哈希处理，计算摘要值

% 3）观察分析原图像与改动后图像摘要值的变化情况

%

%Name: GuoYunting

%Course: 数字内容安全

%Project: 传统哈希

clc

clear all

close all

%% save start time

start\_time=cputime;

%% read in the cover object

file\_name='lena.bmp';

% 384x384像素

message=imread(file\_name);

message\_change\_1=message;%改变(1,1)

message\_change\_2=message;%改变(340,340)

message\_change\_3=message;%改变(50,20)&(20,50)

%原值

message(1,1); %160

message(340,340); %190

message(50,20); %114

message(20,50); %93

%改变初始阵列，观察哈希值

message\_change\_1(1,1)=159;

message\_change\_2(340,340)=191;

message\_change\_3(50,20)=93;

message\_change\_3(20,50)=114;

%输出图像

figure(1);

subplot(2,2,1);

imshow(message),title('原图');

subplot(2,2,2);

imshow(message\_change\_1),title('改变(1,1)');

subplot(2,2,3);

imshow(message\_change\_2),title('改变(340,340)');

subplot(2,2,4);

imshow(message\_change\_3),title('改变(50,20)&(20,50)');

%使用SHA-1算法

algs='SHA-1';

h=hash(message,algs)

h1=hash(message\_change\_1,algs)

h2=hash(message\_change\_2,algs)

h3=hash(message\_change\_3,algs)

hash.m

function h = hash(inp,meth)

inp=inp(:);

% convert strings and logicals into uint8 format

if ischar(inp) || islogical(inp)

inp=uint8(inp);

else % convert everything else into uint8 format without loss of data

inp=typecast(inp,'uint8');

end

% verify hash method, with some syntactical forgiveness:

meth=upper(meth);

switch meth

case 'SHA1'

meth='SHA-1';

case 'SHA256'

meth='SHA-256';

case 'SHA384'

meth='SHA-384';

case 'SHA512'

meth='SHA-512';

otherwise

end

algs={'MD2','MD5','SHA-1','SHA-256','SHA-384','SHA-512'};

if isempty(strmatch(meth,algs,'exact'))

error(['Hash algorithm must be ' ...

'MD2, MD5, SHA-1, SHA-256, SHA-384, or SHA-512']);

end

% create hash ，使用java接口

x=java.security.MessageDigest.getInstance(meth); %Java 自带的加密类

x.update(inp);

h=typecast(x.digest,'uint8'); %转换数据类型

h=dec2hex(h)';

if(size(h,1))==1 % remote possibility: all hash bytes 128, so pad:

h=[repmat('0',[1 size(h,2)]);h]; %B = repmat(A,[m,n])，将矩阵 A 复制 m×n 块，即把 A 作为 B 的元素，B 由 m×n 个 A 平铺而成。size（h，2）：获取矩阵h的列数。将摘要值平铺到一行

end

h=lower(h(:)'); %将hash值转换为小写字符串

clear x

return