实验一：鲁棒图像感知哈希实验

1. 实验类别

设计型实验：设计一种鲁棒性图像感知哈希算法，并给出实验结果。

2. 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。

3. 实验条件

(1) WindowsXP 或 WindowsVista 等操作系统；

(2) Matlab 7.x 以上版本软件；

(3) 图像库。

4. 实验原理

4.1 原理

数字图像的真实性完整性认证技术是多媒体安全保护的重要研究课题。图像完整性认证是不允许图像有任何丝毫的改变，在要求很严格或是非常机密的情况下就需要用到完整性认证。基于密码学意义上的认证方法均属于图像完整性认证。图像内容认证允许图像经受一些不损害图像质量的操作，只要使代表图像内容的特性得到认证即可。数字图像占据了数字化作品的很大一部分，除了数字图像本身的诸多优点外，数字图像还有如下特点：数据量大，在存储和传输的过程中会经受 JPEG 有损压缩、添加噪声、滤波等不会损耗图像内容质量的处理操作。这就要求我们在选择认证方法时不能选择完整性认证，而要选择内容性认证。在信息安全的发展过程中，基于图像内容认证的方法主要有数字水印和数字哈希两

种方法。

图像内容认证的两种方法里，数字水印的技术已日渐成熟，而数字哈希技术是一个新兴的课题，因此研究数字哈希技术更具有挑战性和实际意义。

数字图像哈希是将图像数据映射到一个简短的固定长度的比特流中。根据人类视觉系统反应，使得不同内容的图像具有不同的哈希值，内容相似的图像具有相似的哈希值。

现有的许多图像哈希方案都用以下三个步骤来产生哈希序列：

(1)产生一个关键依赖型的特征向量；

(2)量化该特征向量；

(3)压缩已经量化了的向量。

其中最具有挑战性的一步就是第一步，产生的特征向量既要代表图像的内容，又要适应一些轻微的误差失真，如普通的过滤操作(如平均和均值滤波)、几何失真(如旋转，平移和尺度变换)、JPEG 压缩和非线形亮度变化。其方法主要有：基于图像统计学特性的方法（灰度直方图、小波统计特性）、基于图像关系的方法（DCT 系数对、小波系数对）、基于原始图像特征表述的方法（SVD 值）、基于低层图像特征提取的方法（特征点）等等。

4.2 典型应用

图像哈希的典型应用是相似图像搜索。如 Google 和 Tineye 相似图像搜索主要利用的算法是感知哈希算法，它的作用是对每张图片生成一个“指纹”字符串，然后比较不同图片的指纹，结果越接近，就说明图片越相似。典型算法有：

（1） ahash 算法

- **缩小尺寸：**将图片缩小到 8x8 的尺寸，总共 64 个像素。去除图片的细节，只保留结构、明暗等基本信息，摒弃不同尺寸、比例带来的图片差异。
- **简化色彩：**将缩小后的图片，转为 64 级灰度。也就是说，所有像素点总共只有 64 种颜色。
- **计算平均值：**计算所有 64 个像素的灰度平均值。
- **比较像素的灰度：**将每个像素的灰度，与平均值进行比较。大于或等于平均值，记为 1；小于平均值，记为 0。

- **计算哈希值：**将上一步的比较结果，组合在一起，这就是这张图片的指纹。
- **计算“汉明距离”**，如果不相同的数据位不超过某个阈值，就说明两张图片很相似，距离为 0 说明完全相同；如果大于某个阈值，就说明这是两张不同的图片。

（2） phash 算法

- **图片缩放：**缩小尺寸到 32×32
- **图像灰度化：**转化为 256 灰度图
- **对图像进行 DCT 变换**
- **缩小 DCT：**DCT 是 32*32，保留左上角的 8*8，代表图片的低频区域
- **计算平均值：**计算缩小 DCT 后的所有像素点的平均值
- **与均值比较：**将每个像素的 DCT 值，与平均值进行比较。大于或等于平均值，记为 1；小于平均值，记为 0。
- **计算哈希值：**将上一步的比较结果，组合在一起，这就是这张图片的指纹。
- **用汉明距离比较：**如果不相同的数据位不超过某个阈值，就说明两张图片很相似，距离为 0 说明完全相同；如果大于某个阈值，就说明这是两张不同的图片。

（3） dhash 算法

- **缩小图片：**收缩到 8*9 的大小，共有 72 个像素点
- **转化为灰度图：**转化为 256 灰度图
- **计算差异值：**dHash 算法工作在相邻像素之间，这样每行 9 个像素之间产生了 8 个不同的差异，一共 8 行，则产生了 64 个差异值
- **获得指纹：**如果左边的像素比右边的更亮，则记录为 1，否则为 0
- **用汉明距离比较：**如果不相同的数据位不超过某个阈值，就说明两张图片很相似，距离为 0 说明完全相同；如果大于某个阈值，就说明这是两张不同的图片。

（4） whash 算法

- **缩小图片：**缩小尺寸到 32×32

- **转化为灰度图：**转化为 256 灰度图
- **对图像进行 DWT 变换,取低频信息**
- **计算平均值：**计算低频区域的所有像素点的平均值
- **与均值比较：**将低频区域的每个像素点的 DWT 值，与平均值进行比较。大于或等于平均值，记为 1；小于平均值，记为 0。
- **计算哈希值：**将上一步的比较结果，组合在一起，这就是这张图片的指纹。
- **用汉明距离比较：**如果不相同的数据位不超过某个阈值，就说明两张图片很相似，距离为 0 说明完全相同；如果大于某个阈值，就说明这是两张不同的图片。

4.3 算法实现

Python 的 imagehash 库（https://github.com/JohannesBuchner/imagehash）实现了上述四种算法，具体过程如下：

(1) 安装 imagehash 库，pip install imagehash

(2) 调用相应的哈希函数

```
import imagehash
from PIL import Image      #PIL 是 python 的图像处理库
path1= 'C:\\DemoImages\\1.bmp'
path2= 'C:\\DemoImages\\lena_contrast.tiff'
hash1 = imagehash.average_hash(Image.open(path1))      #ahash 算法
hash2 = imagehash.average_hash(Image.open(path2))
print(hash1- hash2)

hash1 = imagehash.phash(Image.open(path1))      #phash 算法
hash2 = imagehash.phash(Image.open(path2))
print(hash1- hash2)

hash1 = imagehash.dhash(Image.open(path1))      #dhash 算法
hash2 = imagehash.dhash(Image.open(path2))
print(hash1- hash2)
```

```
hash1=imagehash.whash(Image.open(path1))     #whash 算法
hash2=imagehash.whash(Image.open(path2))
print(hash1- hash2)
```

5. 实验要求

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，具体过程如下：

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（rgb2gray）；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（imresize）
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（dct2）

   fun = @dct2

   I_c = blkproc(I,[8 8],fun)

   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、的 64*64 矩阵（randn('state',key)，randn），然后用高斯低通滤波器进行迭代滤波（K=fspecial('gaussian'); Y=filter2(K,Y);）
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

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

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 tau，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。汉明距离为

dis=norm((hashVec1 - hashVec2)/(2*sqrt(norm(hashVec1)*norm(hashVec2))));

8. 以 DemoImages 中的图片为测试集，分析实验结果。
9. 关于实验报告

（1）关于 tau 的选取，tau 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。

（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。