



华中农业大学
HUAZHONG AGRICULTURAL UNIVERSITY

基于内容的图像检索系统设计与实现 实训报告

姓 名： 刘 开

班 级： 计科 1403

学 号： 201331000305

指导教师： 翟瑞芳 彭辉 周雄辉 高俊祥

中国·武汉
二〇一七 年 七 月 九 日

目录

1. 系统设计	3
1.1、主要开发工具	3
1.2、实验方法与步骤	3
1.2.1 图像检索系统工作流程	3
1.2.2 图像检索系统框架	3
1.2.3 系统设计	5
2. 所采用算法思想	8
2.1 图像预处理	8
2.2.1 中值滤波	8
2.1.2 直方图均衡化	8
2.2 图像颜色特征	8
2.2.1 HSI 中心矩法	9
2.2.2 直方图相交法	10
2.3 图像纹理特征	10
2.3.1 灰度共生矩阵法	11
2.4 基于形状特征的检索	12
2.4.1 形状不变矩法	12
2.4.2 边缘方向直方图法	14
2.5 图像特征匹配	14
3.详细实现过程	15
3.1 图像特征数据库设计说明	15
3.2 JAVA 包与类的设计	18
4.实验结果分析	22
5.实训总结和心得	26
6.参考文献	27

1.系统设计

1.1、主要开发工具

1. 系统开发工具：

使用 Java 语言进行开发；

2. 数据库开发工具：

使用 mysql 数据库，可视化工具为 navicat for mysql；

3. 图像数据库：

检索采用的图像数据为如下网址中的部分数据【1】；

http://www.vision.caltech.edu/Image_Datasets/Caltech101/

1.2、实验方法与步骤

1.2.1 图像检索系统工作流程

基于内容的图像检索技术是对输入的图像进行分析并分类统一建模，提取其颜色、形状、纹理、轮廓和空间位置等特征，建立特征索引，存储于特征数据库中。检索时，用户提交查询的源图像，通过用户接口设置查询条件，可以采用一种或几种的特征组合来表示，然后在图像数据库中提取出查询到的所需关联图像，按照相似度从大到小的顺序，反馈给用户。用户可根据自己的满意程度，选择是否修改查询条件，继续查询，以达到满意的查询结果。

1.2.2 图像检索系统框架

基于内容的图像检索系统框架如图 1 所示。系统的核心是图像特征数据库。图像特征既可以从图像本身提取得到，又可以通过用户交互获得，并用于计算图像

之间的相似度计算。系统框架应主要包含以下几个基本功能模块：检索方法设置、检索结果浏览、数据库管理维护等。其逻辑结构如图 2 所示，图 3 为图像检索过程与结果输出。

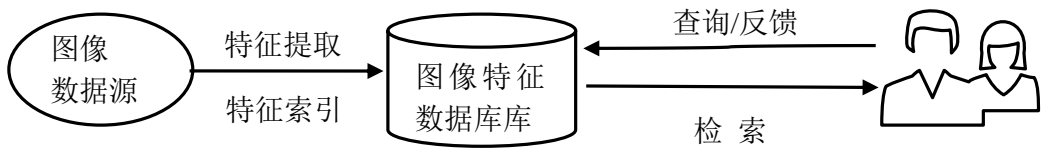


图 1 基于内容的图像检索框架

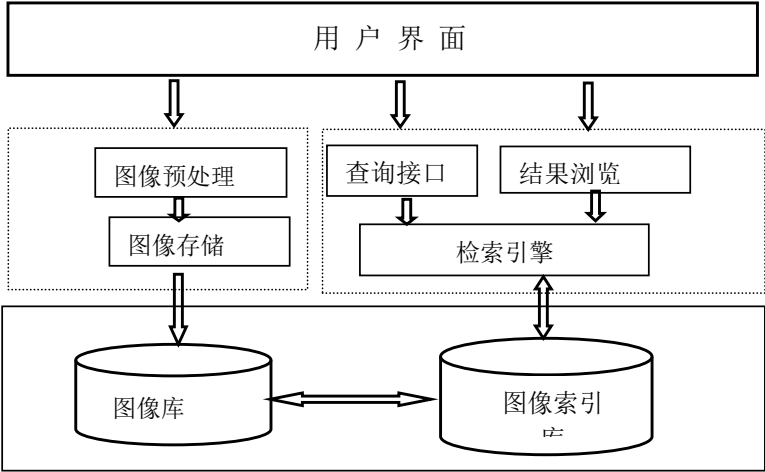


图 2 基于内容的图像检索系统逻辑模型

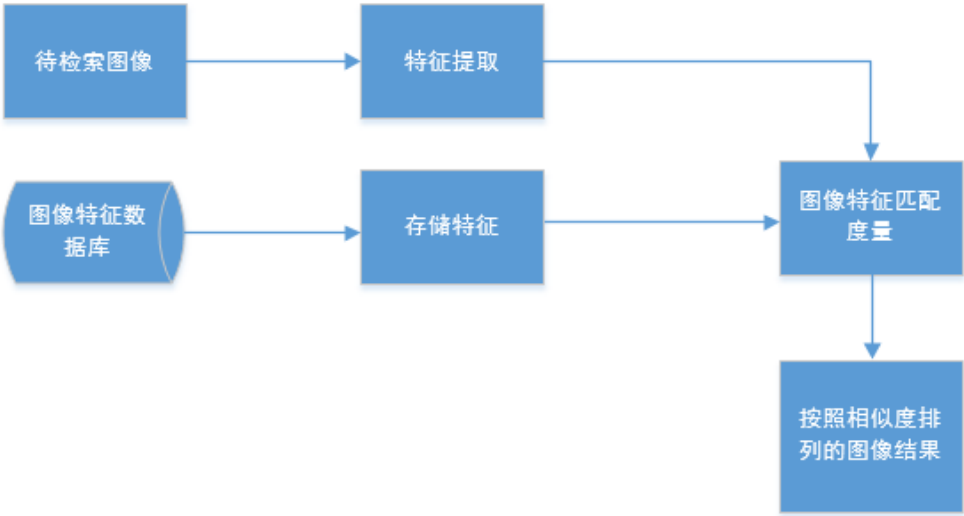


图 3 图像检索过程与结果输出

系统软件名称为“基于内容的图像检索 1.0”，主界面如图 4 所示。

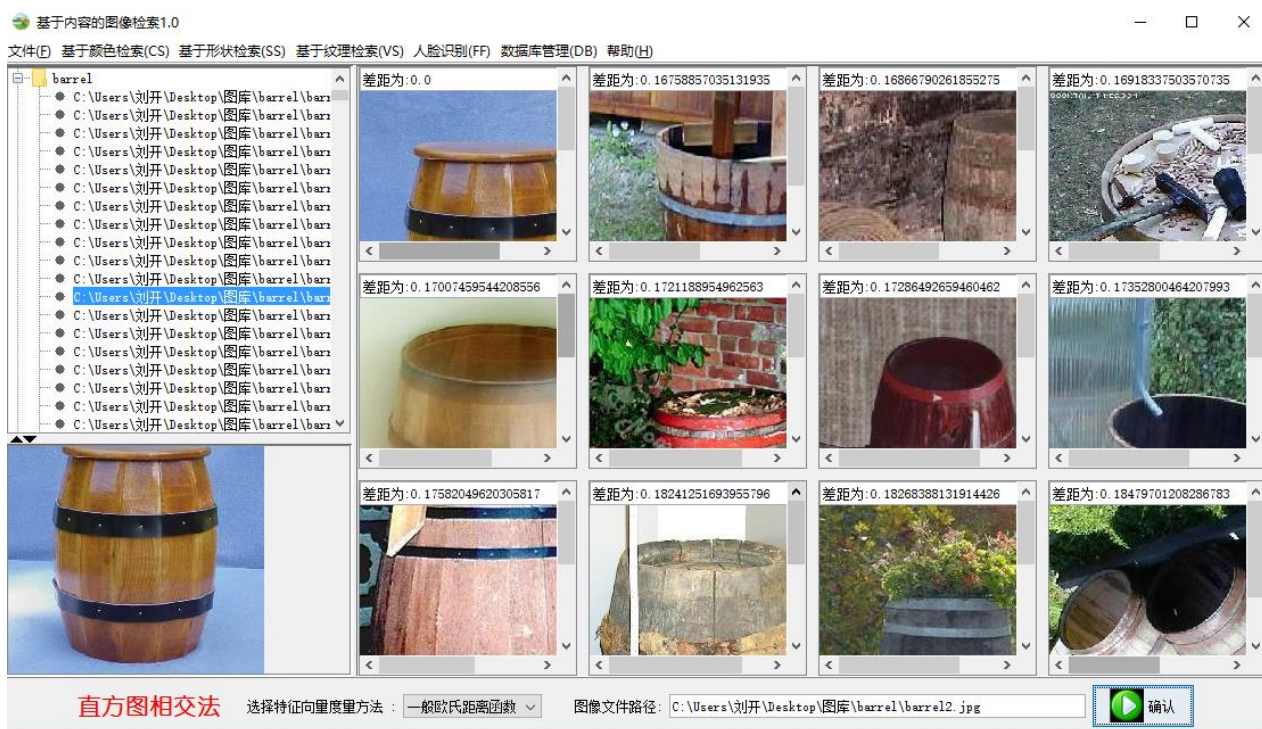


图 4 系统主界面

1.2.3 系统设计

菜单设计

1. “文件”菜单中可以打开图像文件进行检索，点击退出可以退出系统；
2. “基于颜色检索(CS)”菜单可选择“HSI 中心矩法”与“直方图相交法”两种；
3. “基于形状检索(SS)”菜单可选择“形状不变矩法”与“边缘方向直方图法”两种；
4. “基于纹理检索(VS)”菜单使用“灰度共生矩阵法”进行检索；
5. “人脸识别(FF)”菜单调用照相机拍照后识别（未实现）；
6. “数据库管理(DB)”菜单功能有“建立新数据库”、“增加图像”、“删除图像”、“图像统计”、“图像显示”功能；
7. “帮助”菜单有“使用说明”与“关于”两项；

系统使用方法设计

- 1.如果没有数据库需要先点击数据库管理菜单进行新建，选择文件夹导入；
- 2.界面左上方放一个树形控件显示的是所有图像特征文件，点击可进行打开（关闭）控件，点击路径选择文件，也可通过打开菜单选择需要进行检索的图片；
- 3.选择检索方法后点击“确定”按钮进行检索，会在窗口右部显示结果，最多显示12张图片，按照相似度大小排列，图像上方为对应的差距，点击其中一张图片可进行保存，选择路径以及图片格式后保存图片；
- 4.数据库管理中可进行插入新的图片，选择图片文件，输入图片标识符进行增加操作，点击删除数据后，可在树形控件中选择需要删除的数据，仅代表图片从数据库中删除，图片本身并没有被删除。点击统计菜单，显示图片总数目；
- 5.点击图片显示菜单，使用另一个界面显示所有图片，一行四张，部分过程中间使用进度条提示。

数据库设计

图像数据库包括两个部分：图像库和特征库。图像库是图像的集合，存储数字化的图像信息，具体在系统实现中，可以存放图像的路径和文件名称来表示；特征库存储图像中提取出的特征表达的相关信息，并为提取出的各个特征简历索引，且和图像间存在一一对应的关系。

图像数据库设计

图像是图像元灰度值的纪录，以行列数据矩阵表示，一般信息量比较大。直接读取图像的信息存入数据库中，不但增加了数据库的容量，而且增加了计算机的负担。若直接根据图像的路径名称建立图像地址库，可有效提高计算机的计算效率。当需要提取图像时，再根据图像的地址，对图像进行检索，平时，只需对图像的基本信息进行管理。数据库的基本功能包括：增加删除图像、图像统计、图像的显示等。Path.txt 示例如图 5 所示。

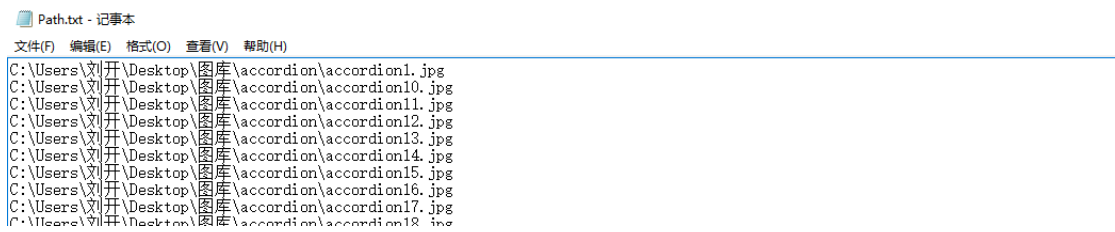


图 5 Path.txt 部分数据

图像索引库的建立

数据组织说明如下：

由于图库中有大量的图像，每一张图像中有许多像素点，每个像素点又有许多特征值，如果都将其放入内存中，是不可能实现的。因此本实训项目将提取出的特征值存放到 txt 文件中，每打开应用系统，就可以直接从外存中读取文件，构建图像特征索引库。本实训项目比如将直方图相交法得到的数据放入“C:\Users\刘开\Desktop\文件”文件夹下，先建立好一个名为文件的文件夹，将对应数据放入其中。其中 Path.txt 主要存的是图库中每张图像的路径，匹配时程序可以根据这个路径找到图库中图像所在的路径。

图像特征的提取与表达

图像特征的提取与表达是基于内容的图像检索技术的基础。广义而言，图像的特征包括基于文本的特征(关键字、注释等)和视觉特征(如色彩、纹理、形状、对象表面等)两类。视觉特征又分为通用的视觉特征和领域相关的视觉特征。前者描述所有图像共有的特征，与图像的具体类型或内容无关，主要包括色彩、纹理和形状；后者建立在对所描述图像内容的某些先验知识(或假设)基础上，与具体的应用紧密有关，如人的面部特征或指纹特征等。

图像特征匹配

在基于内容的图像检索中，两幅图像是否相似是指图像的特征向量是否相似。常用的图像相似性测度通常采用几何模型，将图像特征看作是向量空间中的点，通过计算两个点之间的接近程度来衡量图像特征之间的相似度。基于内容的图像检索算法主要有最邻近查询算法和区间查询算法，它们都依赖于距离函数或者相似性度量。

2. 所采用算法思想

2.1 图像预处理

图像预处理中，可使用中值滤波图像降噪，使用直方图均衡化图像增强，主要用来增强对比度等。

2.2.1 中值滤波

对待处理的当前像素，选择一个模板，该模板为其邻近的若干个像素组成，对模板的像素由小到大进行排序，再用模板的中值来替代原像素的值的方法，权系数矩阵模板为公式（1），计算 g 的过程如公式（2）所示。

$$\text{mid} \begin{matrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{matrix} \quad (1)$$

$$g = \text{median}[f(x-1,y-1) + f(x,y-1) + f(x+1,y-1) + f(x-1,y) + f(x,y) + f(x+1,y) + f(x-1,y+1) + f(x,y+1) + f(x+1,y+1)] \quad (2)$$

代码实现为包 `mainAlgorithm` 中的类 `MedianFilter.Java`;

2.1.2 直方图均衡化

“中心思想”是把原始图像的灰度直方图从比较集中的某个灰度区间变成在全部灰度范围内的均匀分布。直方图均衡化就是对图像进行非线性拉伸，重新分配图像像素值，使一定灰度范围内的像素数量大致相同。直方图均衡化就是把给定图像的直方图分布改变成“均匀”分布直方图分布。

代码实现为包 `mainAlgorithm` 中的类 `HistogramEqualizationr.Java`;

2.2 图像颜色特征

颜色特征是在图像检索中应用最为广泛的视觉特征，主要原因在于颜色往往和图像中所包含的物体或场景十分相关。此外，颜色特征对图像本身的尺寸、方向、视角

的依赖性较小，从而具有较高的检索性。利用颜色特征进行检索的方法主要基于直方图，如：直方图相交法，比例直方图法，距离法，参考颜色表法和聚类算法，累计直方图法，此外，还有 HSI 中心矩法，图 6 为颜色特征的图像检索关键技术路径。

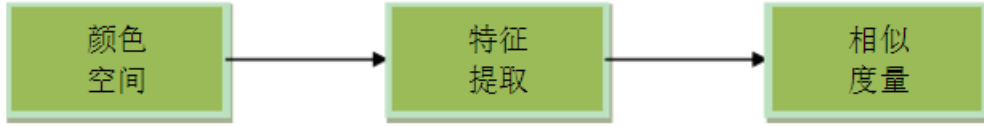


图 6 颜色特征的图像检索关键技术路径

2.2.1 HSI 中心矩法

算法步骤如下：

- (1) 将图像从 RGB 颜色空间转至 HSI 颜色空间；转化公式采用公式 (3)、(4)、(5)，HSI 反映了人的视觉系统感知彩色的方式，以色调、饱和度和亮度三种基本特征量来感知颜色。

$$\theta = \cos^{-1} \left\{ \frac{[(R-G)+(R-B)]/2}{\sqrt{(R-G)^2+(R-B)*(G-B)}} \right\} \quad (3)$$

$$H = \begin{cases} \theta & B \leq G \\ 360 - \theta & B > G \end{cases} \quad (4)$$

$$S = 1 - \frac{3 * \min(R, G, B)}{R + G + B} \quad (5)$$

- (2) 以 HSI 空间的 Hue 分量为例，如果记 Hue(p_i) 为图像 P 的第 i 个像素的 Hue 值，则其前三阶中心矩计算公式分别为 (6)、(7)、(8)。

$$M_1 = \frac{1}{N} \sum_{i=1}^N \text{Hue}(p_i) \quad (6)$$

$$M_2 = \left[\frac{1}{N} \sum_{i=1}^N ((\text{Hue}(p_i) - M_1)^2) \right]^{1/2} \quad (7)$$

$$M_3 = \left[\frac{1}{N} \sum_{i=1}^N ((\text{Hue}(p_i) - M_1)^3) \right]^{1/3} \quad (8)$$

- (3) 将特征值存放在图像特征数据库中。

代码实现为包 mainAlgorithm 中的类 ColorHSICenterMoment.Java;

2.2.2 直方图相交法

为利用图像的特征描述图像，可借助特征的统计直方图。图像特征的统计直方图实际是一个 1-D 的离散函数，即公式（9）。

$$H(k) = \frac{n_k}{N} \quad k = 0, 1, \dots, L - 1 \quad (9)$$

k 代表图像的特征取值， L 是特征可取值个数， n_k 是图像中具有特征值为 k 的像素的个数， N 是图像像素的总数，一个示例如下图：其中有 8 个直方条，对应图像中的 8 种灰度像素在总像素中的比例。

令 $H_Q(k)$ 与 $H_D(k)$ 分别为两幅图像某一特征的统计直方图，则两图像之间的匹配值 $P(Q, D)$ 可借助直方图相交来实现，即公式（10）。

$$P(Q, D) = \frac{\sum_{k=0}^{L-1} \min[H_Q(k), H_D(k)]}{\sum_{k=0}^{L-1} H_Q(k)} \quad (10)$$

将每个图像得到的直方图相交法的特征值保存到文件中。

代码实现为包 `mainAlgorithm` 中的类 `HistogramIntersecting.Java`;

2.3 图像纹理特征

纹理特征是一种不依赖于颜色或亮度的反映图像中同质现象的视觉特征。纹理特征包含了物体表面结构组织排列的重要信息以及它们与周围环境的联系。用户可通过提交包含有某种纹理的图像来查找含有相似纹理的其他图像。从人类的感知经验出发，纹理特征主要有粗糙性、方向性和对比度。纹理分析方法，大致可分为两类：（1）统计方法。用于分析像木纹、沙地、草坪等细致而不规则的物体，并根据关于像素间灰度的统计性质对纹理规定特征及特征与参数间的关系。（2）结构方法。适于像布料的印刷图案或砖瓦等一类元素组成的纹理及其排列比较规则的图案，然后根据纹理基元及其排列规则来描述纹理的结构及特征、特征与参数间的关系。

2.3.1 灰度共生矩阵法

定义：是对整张图像所有像素点的灰度值进行统计的结果。它通过计算图像中一定距离和一定方向的两点灰度之间的相关性，来反映图像在方向、间隔、变化幅度及快慢上的综合信息。

取图像 ($N \times N$) 中任意一点 (x, y) 及偏离它的另一点 $(x+a, y+b)$ ，设该点对的灰度值为 (g_1, g_2) 。令点 (x, y) 在整个画面上移动，则会得到各种 (g_1, g_2) 值，设灰度值的级数为 k ，则 (g_1, g_2) 的组合共有 k 的平方种。对于整个画面，统计出每一种 (g_1, g_2) 值出现的次数，然后排列成一个方阵，再用 (g_1, g_2) 出现的总次数将它们归一化为出现的概率 $P(g_1, g_2)$ 这样的方阵称为灰度共生矩阵。距离差分值 (a, b) 取不同的数值组合，可以得到不同情况下的联合概率矩阵。 (a, b) 取值要根据纹理周期分布的特性来选择，对于较细的纹理，选取 $(1, 0)$ 、 $(1, 1)$ 、 $(2, 0)$ 等小的差分值。

当 $a=1, b=0$ 时，像素对是水平的，即 0 度扫描；

当 $a=0, b=1$ 时，像素对是垂直的，即 90 度扫描；

当 $a=1, b=1$ 时，像素对是右对角线的，即 45 度扫描；

当 $a=-1, b=1$ 时，像素对是左对角线，即 135 度扫描。

图 7 显示了如何求解灰度共生矩阵，以 $(1, 1)$ 点为例，GLCM $(1, 1)$ 值为 1 说明只有一对灰度为 1 的像素水平相邻。GLCM $(1, 2)$ 值为 2，是因为有两对灰度为 1 和 2 的像素水平相邻。

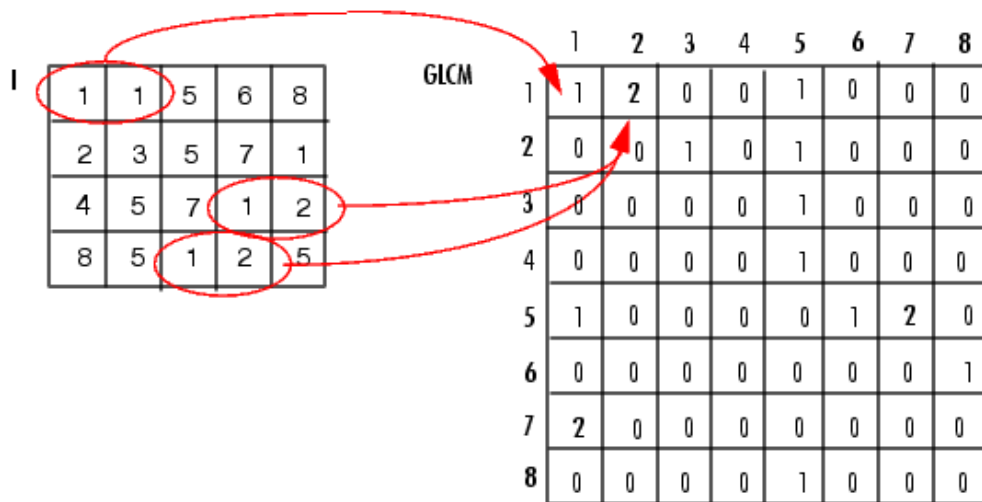


图 7 求解灰度共生矩阵例子

算法步骤:

- (1) 图像灰度化 $\text{Gray}=0.30*\text{R}+0.59*\text{G}+0.11*\text{B}$;
- (2) 降低图像灰度等级, 把图像 256 个灰度级按顺序分为 8 个区间;
- (3) 然后获得图像 0 度, 45 度, 90 度, 135 度四个方向的灰度共生矩阵;
- (4) 求每个灰度共生矩阵的纹理一致性, 纹理对比度, 纹理熵, 纹理相关性;
- (5) 求每个灰度共生矩阵的四个值的期望和标准差, 得到 8 个特征值;
- (6) 将特征值存放在图像特征数据库中。

代码实现为包 mainAlgorithm 中的类 GLCM.Java;

2.4 基于形状特征的检索

形状是描述图像内容的本质特征, 在实际检索中, 很多查询可能并不针对图像的颜色, 因为同一物体可能有各种不同的颜色, 但其形状总是相似的。如检索某辆汽车的图像, 汽车可以是红的、绿的等, 但形状决不会像飞机的外形。另外, 对于图形来说, 形状是它唯一重要的特征。目前用于图像检索的形状描述方法主要有两类: 基于边缘和基于区域的形状方法。基于边缘的形状特征提取是利用图像的边缘信息, 在边缘检测的基础上, 用面积、周长、偏心率、角点、链码、兴趣点、傅里叶描述子、矩描述子等特征来描述物体的形状, 适用于图像边缘较清晰、容易获取的图像。

2.4.1 形状不变矩法

作为一组关于形状统计值, 矩不变量的表示形式有多种, 如 Hu 矩, 具有对图像的旋转、平移和尺度变化的不变性。j 基本思想就是用图像的 Hu 不变矩 $u_1 \sim u_7$ 和离心率 e 作为图像的形状特征索引, 使用适当的相似性距离定义, 计算出两幅图像的相似性距离。当距离足够小时, 就认为两幅图像是相似的。

Hu 不变矩

不变矩(Invariant Moments)是一处高度浓缩的图像特征, 具有平移、灰度、尺度、旋转不变性。

一幅 $M \times N$ 的数字图像 $f(i, j)$ ， $f(i, j)$ 为图像在坐标点 (i, j) 处的灰度值。其 $p+q$ 阶几何矩 m_{pq} 和中心矩 μ_{pq} 为公式 (11) 与 (12) 所示。

$$m_{pq} = \sum_{i=1}^M \sum_{j=1}^N i^p j^q f(i, j) \quad (11)$$

$$\mu_{pq} = \sum_{i=1}^M \sum_{j=1}^N (i - \bar{i})^p (j - \bar{j})^q f(i, j) \text{ 其中 } \bar{i} = m_{10}/m_{00}, \bar{j} = m_{01}/m_{00} \quad (12)$$

为了消除图像比例变化带来的影响，定义规格化中心矩为公式 (13)。

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{pq}^\rho} \text{ 其中 } \rho = \frac{p+q}{2} + 1 \quad (13)$$

利用二阶和三阶规格中心矩可以导出下面 7 个不变矩组 (M1, M7)，如公式 (14) 至 (20)，它们在图像平移、旋转和比例变化时保持不变。

$$M_1 = \eta_{20} + \eta_{02} \quad (14)$$

$$M_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{02}^2 \quad (15)$$

$$M_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \quad (16)$$

$$M_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \quad (17)$$

$$M_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})((\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2) + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})(3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2) \quad (18)$$

$$M_6 = (\eta_{20} - \eta_{02})((\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2) + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \quad (19)$$

$$M_7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})((\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2) + (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})(3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2) \quad (20)$$

算法步骤：

(1) 图像灰度化 $\text{Gray} = 0.30 * R + 0.59 * G + 0.11 * B$;

(2) 用中值滤波算法对图像进行平滑滤波，用于消除噪声干扰，其中需要用到排序算法，将 3×3 窗口的像素值进行排序取中间值；

(3) 用 sobel 算子对图像进行锐化；

(4) 对锐化后的图像用迭代阈值法进行二值化；

① 选择一个初始阈值 t_1 ，根据 t_1 将图像分割为 G1 和 G2 两部分，G1 包含所有小于等于 t_1 的像素，G2 包含所有大于 t_1 像素；

② 分别求出 G1 和 G2 的平均灰度值 u_1 和 u_2 ；

③ 计算新的阈值 $t_2 = (u_1 + u_2) / 2$ ；

④ 如果 $|t_1 - t_2| \leq t_0$ (t_0 为预先指定的很小的正数)，即迭代过程中前后两次阈值很接近时，中止迭代，否则 $t_1 = t_2$ ，重复上述过程。

(5) 计算图像的 Hu 不变矩和离心率作为特征值，并对其进行归一化；

将特征值存如图像特征数据库中。

二值化：就是把一副彩色图像处理成一副黑白图像，一般是作为后续复杂图像处理操作的预处理。

代码实现为包 `mainAlgorithm` 中的主要为类 `HuMoment.Java`、`SobelAndBinarization.Java`；

2.4.2 边缘方向直方图法

边缘方向直方图是描述图像形状的一种方法。图像中目标的形状是由边缘勾勒出来的。边缘是图像灰度变化的地方，对应目标或目标和背景的边界，所以对边缘方向进行统计得到的直方图的形状将在一定程度上反映图像中的形状信息。如果图像中有比较规则的目标，则边缘方向直方图会有明显的周期性；否则，边缘方向直方图会显得比较随机。

算法步骤：

- (1) 把一副彩色 RGB 的图像转换为灰度图像；
- (2) 使用中值滤波对图像进行平滑滤波，用于消除噪声干扰；
- (3) 进行锐化求梯度；
- (4) 对锐化后的图像求边缘方向角，角度范围在 0 度 — 360 度；
- (5) 每 10 度划分为一段，共 36 段，统计数目，求取边缘方向角直方图，并将得到的边缘方向直方图进行归一化和平滑；
- (6) 将边缘方向直方图信息输出到文件中。

代码实现为包 `mainAlgorithm` 中的主要为类 `EdgeHistogram Java`、`EdgeDetector.Java`；

2.5 图像特征匹配

算法步骤：

- (1) 按照前文方法，计算出待检索图像的特征；
- (2) 利用距离度量函数计算待检索图像特征和图像特征库间的距离；
- (3) 对计算出的距离值按照相似性大小进行高低排序；
- (4) 对检索出的结果图像进行输出显示。

✧ 基于直方图的图像特征匹配

设 $(h_1^Q, h_2^Q, L, h_i^Q, L, h_N^Q)$, $(h_1^I, h_2^I, L, h_i^I, L, h_N^I)$ 分别为图像的归一化直方图, 则可以按照以下三种方式分别计算距离度量函数:

(1) 一般欧氏距离函数为公式 (21)

$$D_R(Q, I) = \sqrt{\sum_{i=1}^N (h_i^Q - h_i^I)^2} \quad (21)$$

(2) 加权距离函数为公式 (22)

$$D_W(Q, I) = \sqrt{\sum_{i=1}^N (h_i^Q - h_i^I)^2} \text{ 其中 } W_t = \begin{cases} h_i^Q, h_i^I > 0 \\ 1 \text{ if } h_i^Q \text{ 或 } h_i^I = 0 \end{cases} \quad (22)$$

(3) 相交法度量函数为公式 (23)

$$D_I(Q, I) = \sum_{i=1}^N \min(h_i^Q, h_i^I) / \sum_{i=1}^N h_i^Q \quad (23)$$

✧ 中心矩法的度量函数

设 $M_{Hi}^Q, M_{Si}^Q, M_{Vi}^Q$ 分别为图像Q的H-色调, S-饱和度, V-亮度的 $i(i \leq 3)$ 阶中心矩, $M_{Hi}^I, M_{Si}^I, M_{Vi}^I$ 分别为图像I的H-色调, S-饱和度, V-亮度的 $i(i \leq 3)$ 阶中心矩, 则距离度量函数为公式 (24)。

$$D(Q, I) = \sqrt{W_H \sum_{i=1}^3 (M_{Hi}^Q - M_{Hi}^I)^2 + W_S \sum_{i=1}^3 (M_{Si}^Q - M_{Si}^I)^2 + W_V \sum_{i=1}^3 (M_{Vi}^Q - M_{Vi}^I)^2} \quad (24)$$

3. 详细实现过程

3.1 图像特征数据库设计说明

建立图像特征数据库 image_feature, 以颜色、纹理、形状特征为基础建立表 color、texture、shape, 表的详细设计如下:

表 1 颜色特征表设计

名	类型	长度	小数点	是不是 null	是不是主键
ID	int	10	0	不是	是
Name	varchar	255	0	不是	
H1	float	100	30	不是	
H2	float	100	30	不是	
H3	float	100	30	不是	
S1	float	100	30	不是	
S2	float	100	30	不是	
S3	float	100	30	不是	
I1	float	100	30	不是	
I2	float	100	30	不是	
I3	float	100	30	不是	
Path	varchar	255	0	不是	

表名: color

元组: ID , Name, H1, H2, H3, S1, S2, S3, I1, I2, I3, Path

ID: 数据库中的编号

Name: 图像名称

H1, H2, H3, S1, S2, S3, I1, I2, I3: 分别为 H、S、I 三个分量的三阶中心矩

Path: 储存图像的位置

表 2 纹理特征表设计

名	类型	长度	小数点	是不是 null	是不是主键
ID	int	10	0	不是	是
Name	varchar	255	0	不是	
Exp1	float	100	30	不是	
Exp2	float	100	30	不是	
Exp3	float	100	30	不是	
Exp4	float	100	30	不是	
Stadv1	float	100	30	不是	
Stadv2	float	100	30	不是	
Stadv3	float	100	30	不是	
Stadv4	float	100	30	不是	
Path	varchar	255	0	不是	

表名: texture

元组: ID, Name, Exp1, Exp2, Exp3, Exp4, Stadv1, Stadv2, Stadv3, Stadv4, Path

ID: 数据库中的编号

Name: 图像名称

Exp1, Exp2, Exp3, Exp4, Stadv1, Stadv2, Stadv3, Stadv4: 分别为灰度共生矩阵的四个值的期望和标准差

Path: 储存图像的位置

表 3 形状特征表设计

名	类型	长度	小数点	是不是 null	是不是主键
ID	int	10	0	不是	是
Name	varchar	255	0	不是	
Shape0	float	100	30	不是	
Shape1	float	100	30	不是	
Shape2	float	100	30	不是	
Shape3	float	100	30	不是	
Shape4	float	100	30	不是	
Shape5	float	100	30	不是	
Shape6	float	100	30	不是	
Shape7	float	100	30	不是	
Path	varchar	255	0	不是	

表名: shape

元组: ID , Name, Shape1, Shape0, Shape2, Shpae3, Shape4, Shape5, Shape6, Shape7, Path

ID: 数据库里的编号

Name: 图像名称

Shape0, Shape1, Shape2, Shpae3, Shape4, Shape5, Shape6, Shape7: 分别为图像的 7 个 Hu 不变矩和离心率

Path: 储存图像的位置

3.2 Java 包与类的设计

database 包

类 JDBCUtil.java, 作用为连接数据库、关闭流等;

filePretreatment 包

类 FileNewName.java 用于预处理, 将一类图像的名字统一化; 类 FileGetPath.java 可以得到一个文件夹之下的所有文件的路径, 主要用于输出 Path.txt 文件;

mainAlgorithm 包

主要的算法实现包, 有如下类:

类 Gray.java 用于图片灰度化处理, 主要代码如下:

```
int rgb = reader.getRGB(i, j);
int grey = (int) ((0.3 * ((rgb & 0xff0000) >> 16) + 0.59 * ((rgb & 0xff00) >> 8)) + 0.11 * ((rgb & 0xff)));
rgb = 255 << 24 | grey << 16 | grey << 8 | grey;
grayImage.setRGB(i, j, rgb);
```

类 MedianFilter.java 用于实现中值滤波, 其中重载了方法, 可以进行灰度化+中值滤波。

类 HistogramEqualization.java 用于直方图均衡化, 构造函数调用 Gray 类进行灰度化, 可以将均衡化的图片进行输出;

类 HistogramIntersecting.java 用于实现直方图相交法, 选择是否需要输出文件, 将一类图像的名字统一化; 类 FileGetPath.java 可以得到一个文件夹之下的所有文件的路径, 主要用于输出 Path.txt 文件;

类 ColorHSICenterMoment.java 用于图像颜色特征提取中 HIS 中心距方法的实现, 共产生 9 个特征值, 返回一个数组, 计算方法为 2.2.1 中介绍的公式与步骤;

类 GLCM.java 用于图像纹理特征的提取中灰度共生矩阵方法的实现, 共产生 8 个特征值, 返回一个数组, 计算方法为 2.3.1 中介绍的公式与步骤;

类 EdgeDetector.java 用于得到边缘图像, 共产生 8 个特征值, 使用了 canny 算法, 返回边缘图像等信息;

类 SobelAndBinarization.java 用于实现 sobel 算子对图像进行锐化, 以及使用迭代阈值法进行二值化, 计算方法为 2.4.1 中介绍的公式与步骤, 可以调用 Gray.java 进行灰度化处理, 调用类 MedianFilter.java 进行中值滤波;

类 HuMoment.java 用于实现计算 7 种 Hu 不变矩以及离心率, 共产生 8 个特征值, 返回一个数组, 计算方法为 2.4.1 中介绍的公式, 先调用类 EdgeDetector.java 或者 SobelAndBinarization.java 后再调用该类, 具体如下:

```
// 灰度化, 中值滤波, soble, 二值化
SobelAndBinarization s = new SobelAndBinarization();
```

```
s.sobel(srcPath);
s.getBinaryImg();
//得到不变矩

double[] hum = HuMoment.huMoment(s.getImg());
```

类 EdgeHistogram.java 用于实现边缘方向直方图，重载了方法可以传入边缘图像等信息，先调用类 EdgeDetector.java 后再调用该类，具体如下：

```
// 得到边缘直方图
EdgeDetector edgeDetector = new EdgeDetector();
edgeDetector.setSourceImage(srcPath); // 设置边缘处理的参数
edgeDetector.setThreshold(128);
edgeDetector.setWidGaussianKernel(5);
edgeDetector.process();
EdgeHistogram eh = new EdgeHistogram();
//得到归一化后的边缘方向直方图
double[] result = eh.edgeHistogram(edgeDetector.getFileName(),
    edgeDetector.getName(), edgeDetector.getEdgeImage(), true);
```

match 包

类 DataBaseAndFileOperating.java 用于初始化、操作数据库以及生成必要的文件，在点击“数据库管理(DB)”菜单的“建立新数据库”、“增加图像”、“删除图像”、“图像统计”时进行调用。初始化数据库时先要删除数据库，使用 truncate 为了使 ID 可以从 1 开始，具体如下：

```
// 连接数据库，获得连接
Connection conn = (Connection) JDBCUtil.getMysqlConn();
// 使用truncate清空原有表，为了使ID能够从1开始
conn.prepareStatement("truncate table texture").executeUpdate();
conn.prepareStatement("truncate table color").executeUpdate();
conn.prepareStatement("truncate table shape").executeUpdate();
```

然后读取“Path.txt 文件”调用 mainAlgorithm 包中的有关类得到特征值以及特征文件，输出文件，特征值插入到相关的表中；addData(String name, String path) 往数据库添加数据、deleteData(String path) 删除数据库中某条数据、int count() 用于统计图像数目；

类 FeatureMeasure.java 用于图像特征匹配算法，实现欧几里得距离度量、相交法度量、中心距法度量、夹角余弦衡量几种，传入数组，返回度量结果；

类 HistogramIntersectingMathch.java 用于直方图相交法匹配，传入需要检索图片的路径，调用相关方法以及数据库，得到衡量的差距，然后排序，返回一个 Map<String, Double>，大小不超过 12，键为路径，值为衡量的差距；

类 HSI CenterMomentMatch.java 用于 HIS 中心矩法的匹配，具体实现参考类 HistogramIntersectingMathch.java 的实现过程；

类 ShapeMatch.java 用于形状不变矩法的匹配，具体实现参考类 HistogramIntersectingMathch.java 的实现过程；

类 TextureMatch.java 用于纹理特征中灰度共生矩阵法的匹配，具体实现参考类 HistogramIntersectingMathch.java 的实现过程；

ui 包

类 MainFrame.java 用于系统主界面的实现，使用了 JTree 组件显示树形结构，使用了 JSplitPane 组件实现创建一个水平面板和一个垂直的分割面板，使用 JLabel 显示图片，JTextField 显示差距，创建了一个 3*4 的 JPanel 显示检索结果，菜单参照 1.2.2 的系统菜单设计，滑动还可以观察图片的其他部分，点击图片还可以选择路径进行图片的保存。点击“数据库管理”的增加、删除、创建新数据库操作会更新树形结构；

增加数据关键代码如下：

```
// 往树形中添加
DefaultMutableTreeNode dmu = new DefaultMutableTreeNode(name);
dmu.add(new DefaultMutableTreeNode(f.getPath()));
root.add(dmu);
// 实现显示新节点（自动展开父节点）
TreeNode[] nodes = model.getPathToRoot(dmu);
TreePath path = new TreePath(nodes);
treeFileList.scrollPathToVisible(path);
treeFileList.updateUI();
```

删除数据关键代码如下：

```
// 获取JTree对应的TreeModel对象
model = (DefaultTreeModel) treeFileList.getModel();
model.removeNodeFromParent(note);// 删除指定节点
```

创建新数据库关键代码如下：

```
// 先删除“文件”与“文件2”文件夹里原有的文件
String s = "C:\\Users\\刘开\\Desktop\\文件";
deleteAllFilesOfDir(new File(s));
// 然后创建一个空文件夹
File f = new File(s);
// 如果文件夹不存在则创建
if (!f.exists() && !f.isDirectory()) {
    f.mkdir();
}
//移除树形控件除root外的使用节点
root.removeAllChildren();
model.reload();
//重新建立树形结构
treeList();// 得到所有文件路径的树形结构
```

/实现显示新节点（自动展开父节点）

```
TreeNode[] nodes =model.getPathToRoot(root);  
TreePath path = new TreePath(nodes);  
treeFileList.scrollPathToVisible(path);  
treeFileList.updateUI();
```

类FrameShow. java用于图片显示界面的实现，在主界面中点击“图片显示”菜单会调用该界面，共有4列，传入总的图片个数，通过计算显示所有图片信息；

类UseFrame. java用于实现使用说明界面。显示使用步骤；

类InfiniteProgressPanel. java用于实现一个一个转圈的进度条，在检索、显示等处进行调用；

4.实验结果分析

下面是使用五种方法对同一张图的检索效果，度量方法选择功能没有实现，每种方法仅使用了一种度量方法，“形状不变矩法”与“边缘方向直方图法”使用了一般的欧式距灰度共生矩阵法离法、“HIS 中心矩法”使用中心矩度量、“直方图相交法”使用相交法度量、“灰度共生矩阵法”由于特征值差别较大使用夹角余弦进行度量。

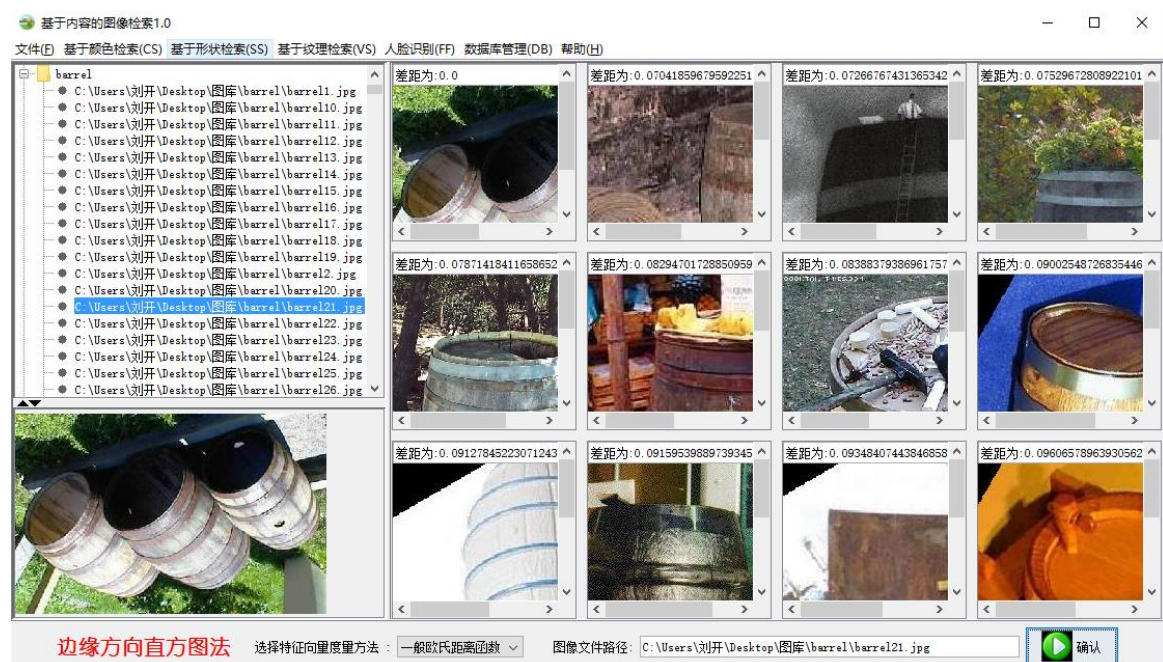


图 8 使用“边缘方向直方图法”进行检索



图 9 使用“HIS 中心矩法”进行检索



图 10 使用“直方图相交法”进行检索

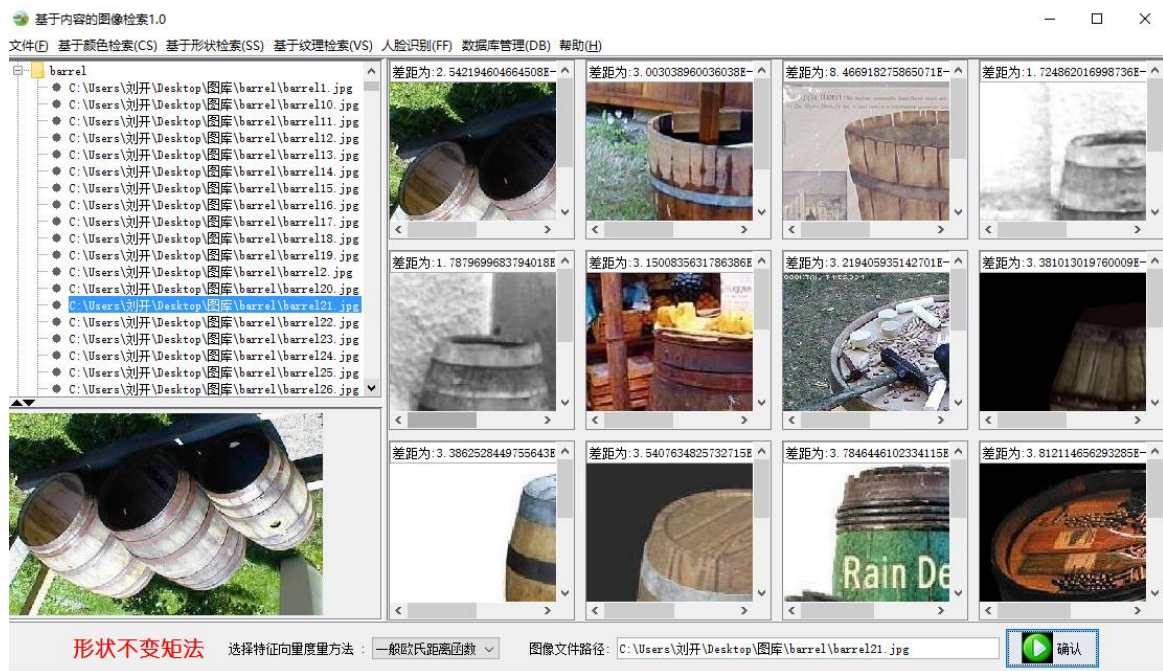


图 11 使用“形状不变矩法”进行检索

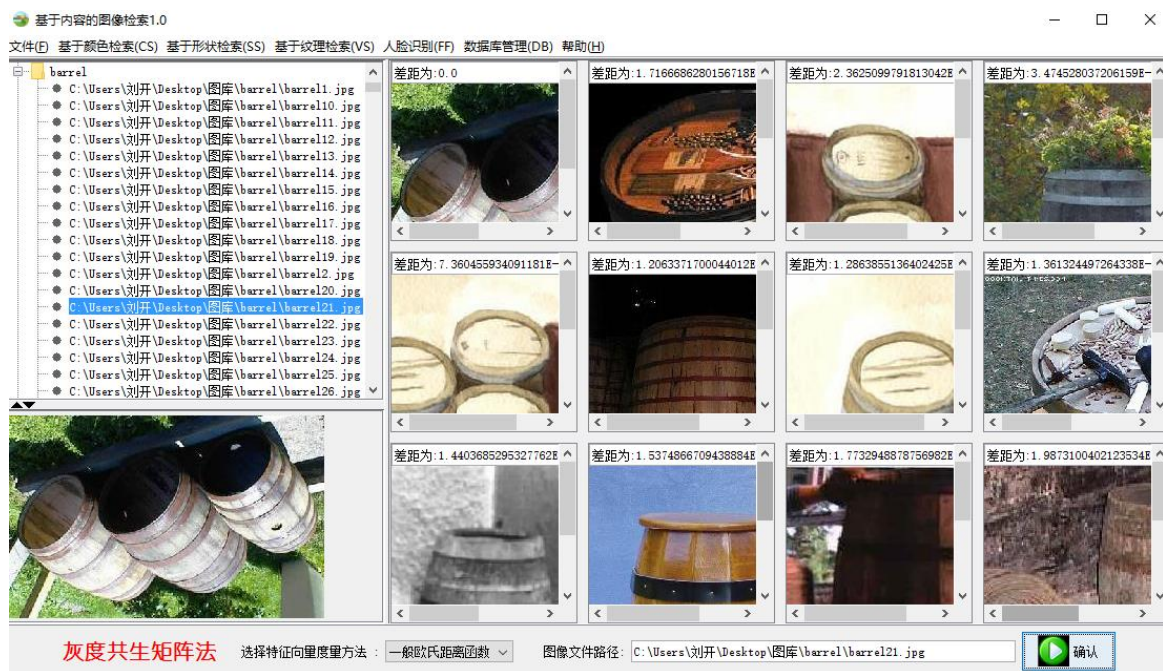


图 12 使用“灰度共生矩阵法”进行检索

从结果中可以看出可以检索出和测试图一样的图，差距为 0，其他的图存在一些不同，可同时使用几种方法同时进行检索，还可以找到最合适原图的检索方法，后期还需要继续的改进系统。

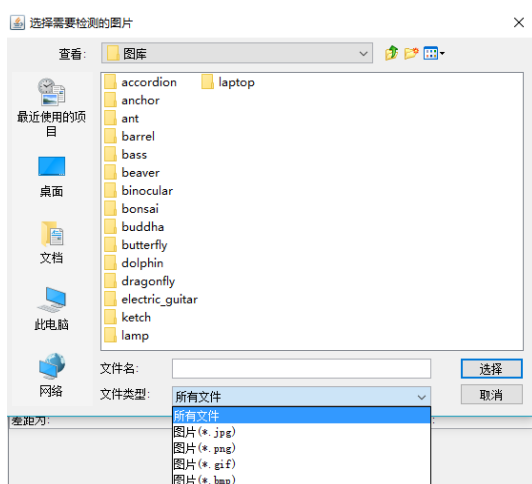


图 13 选择打开文件对话框

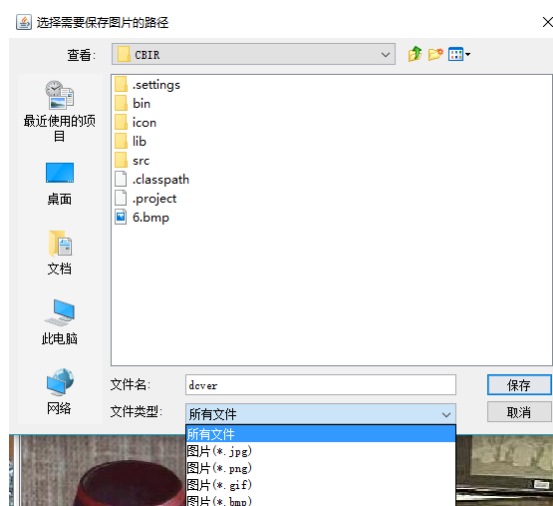


图 14 保存图片文件对话框

图 13 与图 14 可以打开和保存文件，可选择.jpg、.gif、.bmp、.png 四种图片格式。在“数据库管理”菜单点击图片显示会打开图片显示界面，传入的参数为图片的个数，图 15 为图片显示结果，滑动可观察使用图片。

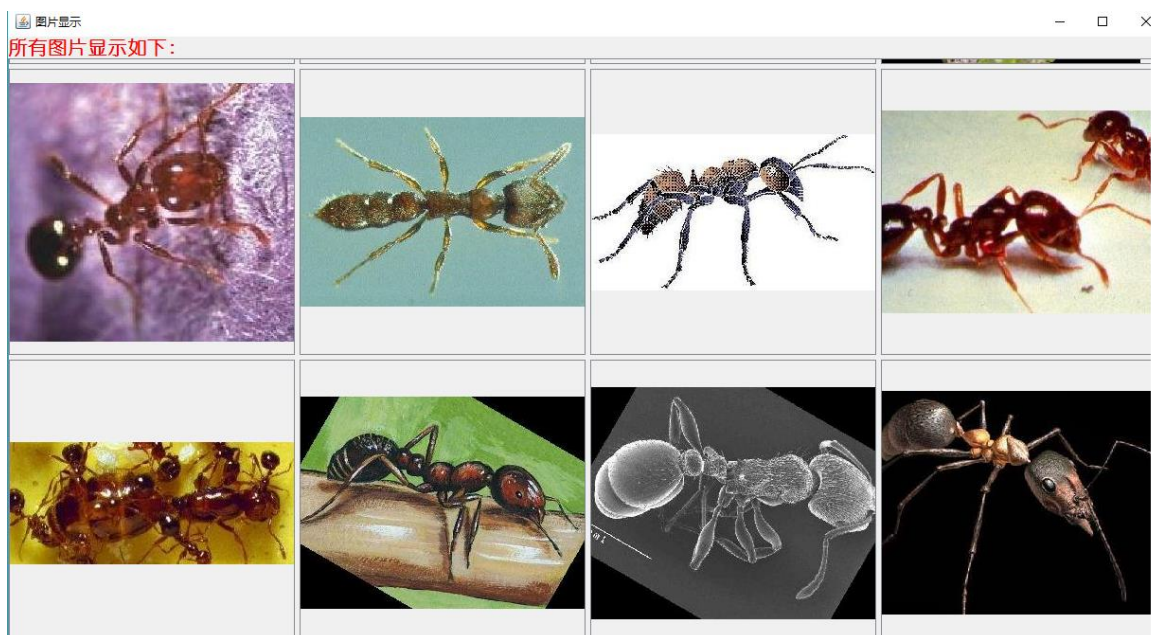


图 15 图片显示界面

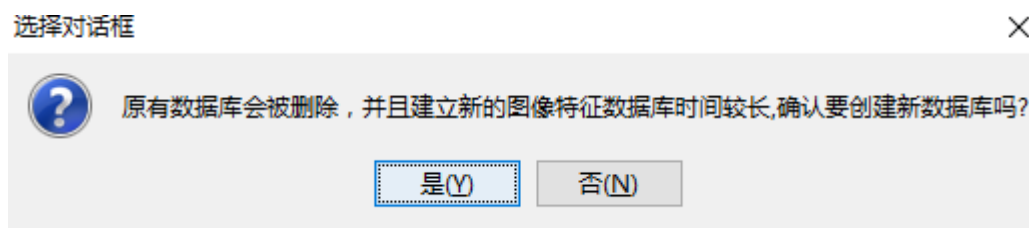


图 16 是否建立新的数据库选择对话框



图 17 图片统计结果显示

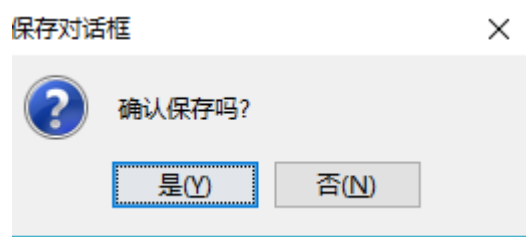


图 18 图片保存提示对话框



图 19 警告对话框

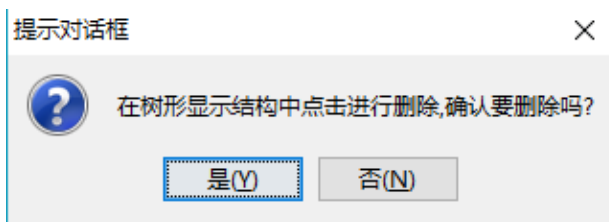


图 20 删除图片提示对话框

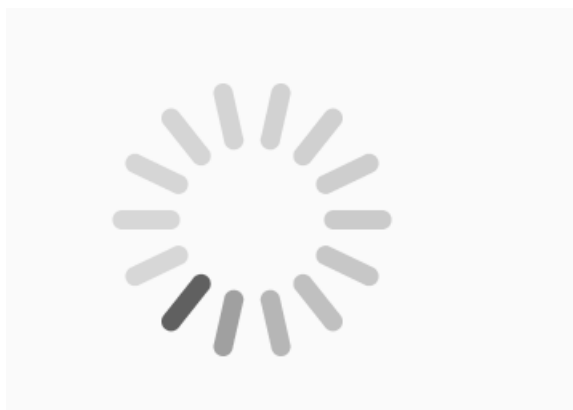


图 21 显示一个转圈的进度条提示正在运行

图 16 至图 21 为系统软件的一些提示功能，通过这些提示可有效的防止用户的无意之举造成错误，增加了系统的可用性。

5.实训总结和心得

通过本次实训，很大的提高了学习能力有了，在设计过程中使我了解到很多的知识，也提高了自己独立思考问题的能力。由于在课程实训之前没有学习过“数字图像处理”这门课程，过程中遇到了很多困难，但经过不断的查找相关资料，还是弄懂了一部分，对一些图像特征提取方法有了很深的理解，比如“HIS 中心矩法”、

“直方图相交法”、“边缘方向直方图法”等。实训系统使用 Java 语言开发，加深了对有关 Java 知识的理解，提高了自己对 Java 的应用能力。期间我们也遇到了很多的困难，非常感谢老师以及同学们的帮助。在设计实现一个系统时，首先需要有一个条理清晰的思路，这样的话才会有很高的效率。遇到困难时通过寻找课本知识，或者在网上查询资料，网上相关的解释有很多，一开始的时候有很多不懂的地方，这些都需要认真的去学习，需要有耐心。系统实现过程遇到了很多的困难，遇到 Bug 应该冷静下来再找，这样效率会高很多。

在实训系统基本完成之际，我发现自己还是有很多不足的地方。对于系统实现方面，系统设计的并不够完美，比如数据库新建会覆盖原有数据库。另外，还有部分需求的功能并没有实现，比如打开摄像头拍照等功能，后期还需要继续的完善，而且编写的其他内容避免不了存在一些错误，主要还是由于自身能力有限。希望能够在以后的日子里通过自身的努力，能够使能力有所提高，可以使设计更加完善。

总之，衷心的感谢在实训过程中所有给予我帮助的人。

6.参考文献

- 【1】L. Fei-Fei, R. Fergus and P. Perona. Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories. IEEE. CVPR 2004, Workshop on Generative-Model Based Vision. 2004
- 【2】数字图像处理 第二版 何东健 （西安电子科技大学出版社）
- 【3】疯狂 Java 讲义 第三版 李刚 （电子工业出版社）
- 【4】博客（图像噪声的抑制）<http://blog.csdn.net/luoweifu/article/details/8150196>
- 【5】博客（RGB 和 HIS 的转换）<http://blog.csdn.net/jacke121/article/details/54730822>
- 【6】博客（灰度共生矩阵）<http://www.cnblogs.com/8335IT/p/5648445.html>
- 【7】博客（图像的矩特征）<http://www.cnblogs.com/ronny/p/3985810.html>
- 【8】博客（java 边缘检测）<http://blog.csdn.net/epleone/article/details/38173931>