

综合实训指导书

题目 基于内容的图像检索 web 系统
综合实训实验指导书

版本 5.0

院系 信息学院

编写时间 2015.4-2015.5

编写教师 翟瑞芳 彭辉 罗俊 高俊祥 王建勇

修订时间 2023.6

一、实训目的

基于内容的图像检索 Web 系统 (Content Based Image Retrieval, 以下简称 CBIR), 是计算视觉领域中关注大规模数字图像内容检索的研究分支。典型的 CBIR 系统, 允许用户输入一张图像, 在图像数据库 (或本地机、或网络) 中查找具有相同或相似内容的其它图片。本实训的基本功能要求是实现基于视觉特征的图像检索 Web 平台。具体包括:

- (1) 实现基于颜色信息的图像检索, 可通过颜色直方图、颜色矩、颜色一致性矢量等方法来实现。
- (2) 实现基于纹理特征的图像检索, 可从四个方面进行: 统计法、结构法、模型法、频谱法。
- (3) 实现基于形状特征的图像检索, 可分别从图像的边缘信息和区域信息来实现。
- (4) 实现基于综合信息的图像检索。
- (5) 实现在 Web 平台上的展示。

二、实训内容

该项目的实训内容主要包括:

1. 能够利用颜色特征实时从图像库中检索出和输入图像相同或相似的图像, 按照相似度大小排列, 并显示在 Web 网页中。
2. 能够利用纹理特征实时从图像库中检索出和输入图像相同或相似的图像, 按照相似度大小排列, 显示在 Web 网页中。
3. 能够利用形状特征实时从图像库中检索出和输入图像相同或相似的图像, 按照相似度大小排列, 并显示在 Web 网页中。

说明:

1. 除上述基本要求外, 鼓励使用新特征和新方法 (如 HoG 特征, LBP 特征, Harr 特征, SVM, BP 神经网络、深度卷积神经网络等) 完成。
2. 实验对象也可以扩展到基于音频、视频等内容的检索。

三、主要仪器及试材

1. 系统开发工具

Web 系统开发工具可以选择 VScode, Java 的 Idea 等, 由学生结合实际情况自行选择。

使用 vue 开发框架进行开发，组件库配合 vue 使用 element-plus（vue3 使用），element-ui（vue2 使用）组件库。

Vue 文档: <https://cn.vuejs.org/>

Element-plus 文档: <https://element-plus.org/zh-CN/>

Element-ui 文档: <https://element.eleme.cn/>

Java 开发软件 IntelliJ IDEA 下载地址: <https://www.jetbrains.com/zh-cn/idea/download/>

Maven 下载地址: <https://maven.apache.org/download.cgi>

2. 数据库开发工具

待检索图像数据库可以选择 MySQL 等开发工具，由学生结合实际情况自行选择。

MySQL 下载网址: <https://www.mysql.com/downloads/>

3. 图像数据库

检索用图像数据，由学生到以下网址下载: <https://www.cs.toronto.edu/~kriz/cifar.html>

四、实验方法与步骤

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。

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 1 基于内容的图像检索框架

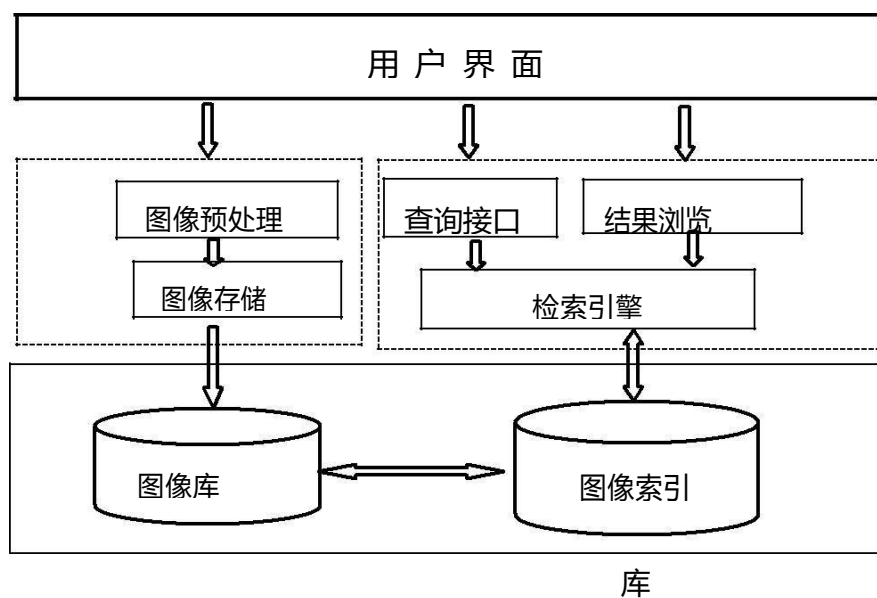


图 2 基于内容的图像检索系统逻辑模型

CBIR 系统典型示例界面如图 3 所示。该示例界面主要包括：待检索图像，检索出的 12 幅图像显示区域、树形控件、检索菜单参数设置等。用户可以自行进行界面设置并完善。

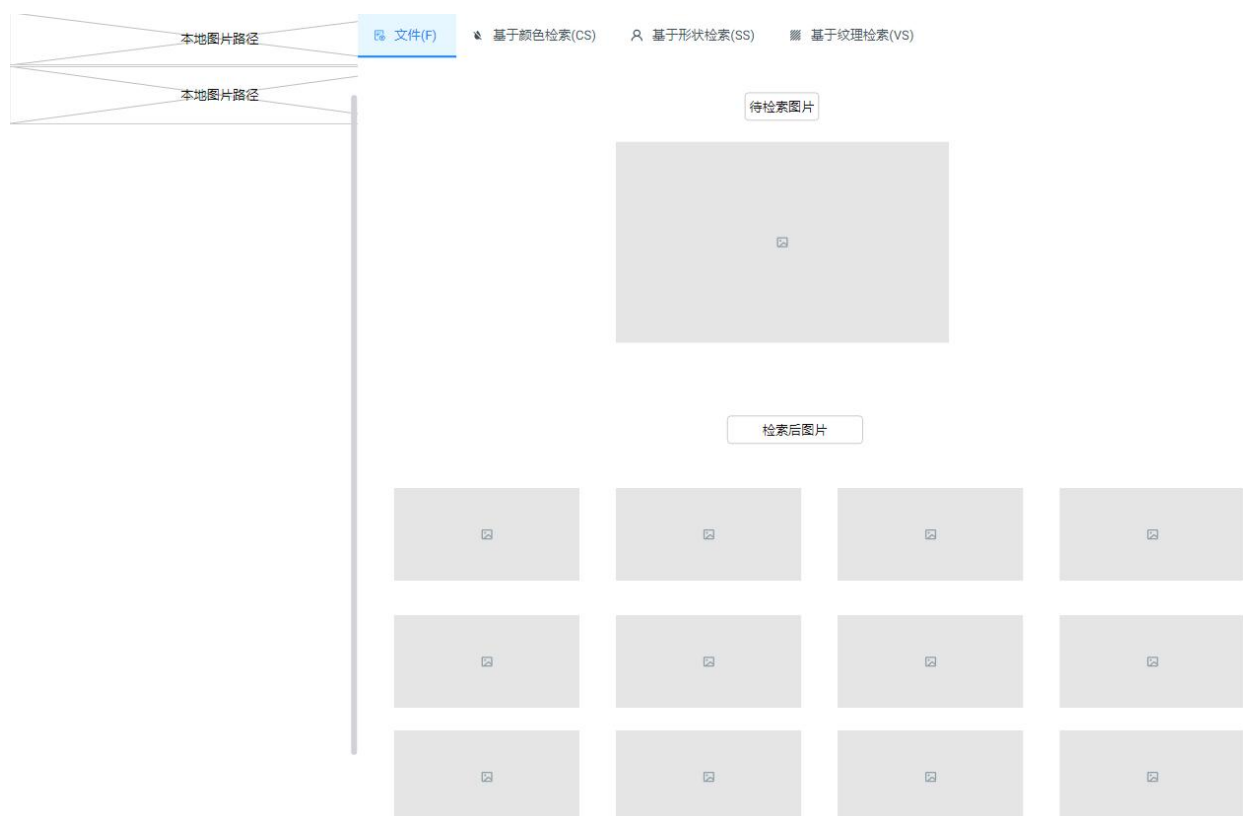


图 3 CBIR Web 系统实现示例界面

3. 数据库构建与实现

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。

3.1 图像数据库

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 4 所示。

```
1 D:\Users\gao\Desktop\图库3\国旗1.jpg
2 D:\Users\gao\Desktop\图库3\国旗10.jpg
3 D:\Users\gao\Desktop\图库3\国旗11.jpg
4 D:\Users\gao\Desktop\图库3\国旗12.jpg
5 D:\Users\gao\Desktop\图库3\国旗13.jpg
6 D:\Users\gao\Desktop\图库3\国旗14.jpg
7 D:\Users\gao\Desktop\图库3\国旗15.jpg
8 D:\Users\gao\Desktop\图库3\国旗16.jpg
9 D:\Users\gao\Desktop\图库3\国旗17.jpg
10 D:\Users\gao\Desktop\图库3\国旗18.jpg
11 D:\Users\gao\Desktop\图库3\国旗19.jpg
12 D:\Users\gao\Desktop\图库3\国旗2.jpg
```

图 4 图库中每幅图像的路径

3.2 图像索引库的建立与实现

3.2.1 数据组织说明

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。因此本实训项目建议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。分别利用 4.2、4.3 和 4.4 实现图像颜色、纹理和特征值的提取及 txt 文件的生成, 生成结果文件分别如图 5 所示。此步骤需要用户先完成图像特征提取后, 再来构建图像特征索引库。

```
File imagepath = new File("src\\dataImage\\path.txt");           //将图库中每个图片的路径存入文件
File filedata = new File("src\\dataImage\\yansezhifangtu.txt");   //直方图相交法生成的文件
File colorJuData = new File("src\\dataImage\\colorjuData.txt");   //HSV颜色矩生成的文件
File shapeNchangeData=new File("src\\dataImage\\shapeNchangeData.txt"); //形状不变矩生成的文件
File shapeEdgeHistogramData=new File("src\\dataImage\\shapeHistogramData.txt");//边缘直方图生成的文件
```

图 5 生成的特征文件

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。

Yansezhifangtu.txt 保存的是直方图相交法的特征值。

colorjuData.txt 保存的是 HSV 中心距法的特征值。示例文件如图 6 所示。

ShapeNchangeData.txt 保存的是基于形状的不变矩法特征值。

shapeHistogramData.txt 保存的是基于形状边缘直方图法特征值。

greymatrixData.txt 保存的是灰度矩阵法的特征值。

veinMCMDData.txt 保存的是颜色共生矩阵 MCM 法的特征值。

3.2.2 图像特征数据库设计说明

分别以颜色、纹理、形状特征为例，说明创建各特征表时的工作过程。以颜色特征为例，说明创建该表时需要包括的字段如下（实际处理过程中，用户选择的特征描述方法不同，所得到的颜色特征值的数量也不同，由用户酌情处理）：

```
140.07664174992442 0.3467426400067435 0.6429748173779333 162.2150951162401 12.736368992622666 12.736368992622666 127.54530885820743 5.033709672290581
173.7378648939632 0.8145889141712894 0.8225019223376112 96.12055970140412 9.804109327287417 9.804109327287417 -98.77380747262866 -4.622531353188779
26.247190893924277 0.45796010326835823 0.9144313725490323 69.55919972312643 8.340215807946844 8.340215807946844 103.42537239666264 4.69399220960243
291.8555653606824 0.9902722564699401 0.8541820887654697 138.72220612312537 11.778039145932798 11.778039145932798 -161.09312140545123 -5.441170467550878
236.4306417074404 0.7416474118886852 0.9812005347593685 161.6527943107917 12.714275217675278 12.714275217675278 -138.3939077717603 -5.172561434685015
93.80384058806398 0.6688245177813548 0.6519407763105286 140.4497121042615 11.851148134432439 11.851148134432439 153.72006774924948 5.356858673599743
91.15464168028896 0.9306338060570397 0.83867647059882281 93.15004915339368 9.651427311718908 9.651427311718908 95.35662776251128 4.5717969998142545
91.15648961408891 0.6059264464109506 0.8628283388827512 109.51787992644229 10.465079069287642 10.465079069287642 92.32587187943109 4.51968123483657
205.13920576893167 0.6614344311342606 0.9879298642334184 139.39141153788666 11.80641399993607 11.80641399993607 -109.06613757760418 -4.777822131582941
123.58797233142072 0.28776436964055513 0.9042947178871567 113.91349028982178 10.67302629481544 10.67302629481544 42.10202819796157 3.4788390730006324
132.6684274056519 0.9788912772223024 0.8102100769035579 119.65902994565219 10.938913563913871 10.938913563913871 124.56195813540568 4.994152606335339
```

图 6 colorjudgetata.txt 中数据

表 1 颜色特征表设计

颜色特征表(Color)			
字段名称	字段描述	字段类型	备注
ID	数据库中的编号	int	Primary Key
Name	图像名称	string	NOT NULL
Color	颜色特征值	float	NOT NULL
Path	储存图像的位置	string	NOT NULL

表名：Color

元组：ID ,Name,Color,Check,Path

ID：数据库中的编号

Name：图像名称

Color：图像特征值

Path：储存图像的位置

以纹理特征为例，说明创建该表时需要包括的字段如下（实际处理过程中，用户选择的特征描述方法不同，所得到的纹理特征值的数量也不同，由用户酌情处理）：

表 2 纹理特征表设计

纹理特征表(Texture)			
字段名称	字段描述	字段类型	备注
ID	数据库中的编号	int	Primary Key
Name	图像名称	string	NOT NULL
Exp1	纹理特征值	float	NOT NULL
Exp2	纹理特征值	float	NOT NULL
Exp3	纹理特征值	float	NOT NULL
Exp4	纹理特征值	float	NOT NULL
Stadv1	纹理特征值	float	NOT NULL
Stadv2	纹理特征值	float	NOT NULL
Stadv3	纹理特征值	float	NOT NULL
Stadv4	纹理特征值	float	NOT NULL
Path	储存图像的位置	string	NOT NULL

表名: Texture

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

ID: 数据库里的编号

Name: 图像名称

Exp1,Exp2,Exp3,Exp4,Stadv1,Stadv2,Stadv3,Stadv4: 图像特征值

Path: 储存图像的位置

表 3 形状特征表设计

形状特征表(shape)			
字段名称	字段描述	字段类型	备注
ID	数据库中的编号	int	Primary Key
Name	图像名称	string	NOT NULL
shape0	形状特征值	float	NOT NULL
shape1	形状特征值	float	NOT NULL
shape2	形状特征值	float	NOT NULL
shape3	形状特征值	float	NOT NULL
shape4	形状特征值	float	NOT NULL
shape5	形状特征值	float	NOT NULL
shape6	形状特征值	float	NOT NULL
shape7	形状特征值	float	NOT NULL
Path	储存图像的位置	string	NOT NULL

表名：Shape

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

ID：数据库里的编号

Name：图像名称

Shape1,Shape2,Shpae3,Shape4,Shape5,Shape6,Shape7：图像特征值

Path：储存图像的位置

需要实现的功能包括：数据库连接，向数据库里添加记录，和从数据库里读取记录；当有检索命令激发时，需要读取相关记录，并对特征值进行比较，获得符合要求的图像；同时，按相似度逆序序列存放到数组中。

4. 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。

4.1 图像预处理

图像预处理中, 用户可以结合图像实际情况酌情选择是否必需。

主要的参考步骤如下:

- (1) 统一图像大小, 使得每一幅待检索图像大小一致;
- (2) 图像降噪, 避免图像噪声带来的影响。可使用数字图像处理课程中所学各种方法, 如高斯模板、中值滤波等来实现。
- (3) 图像增强, 主要用来增强对比度等。可使用数字图像处理课程中所学各种方法, 如直方图均衡化等方法实现。

4.2 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。

下面以中心矩为例, 说明实现纹理特征的提取基本步骤。

算法步骤：

- (1) 图像灰度化 $\text{Gray}=0.30*R+0.59*G+0.11*B$;
- (2) 将图像从 RGB 颜色空间转至 HSI 颜色空间;
- (3) 以 HSI 空间的 Hue 分量为例, 如果记 $\text{Hue}(p_i)$ 为图像 P 的第 i 个像素的 Hue 值, 则其前三阶中心矩分别为:

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

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

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

- (4) 将特征值存放在图像特征数据库中。

4.3 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。

下面以灰度共生矩阵为例, 说明实现纹理特征的提取基本步骤。

算法步骤：

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

4.4 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。

以形状不变矩法和边缘方向直方图法为例说明形状特征提取的基本步骤。

4.4.1 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。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 shape 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。

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；

Sobel 算子：

x 方向：

	i-1	i	i+1
j-1	-1	0	1
j	-2	0	2
j+1	-1	0	1

y 方向：

	i-1	i	i+1
j-1	-1	-2	-1
j	0	0	0
j+1	1	2	1

(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 G_1 和 G_2 两部分， G_1 包含所有小于等于 t_1 的像素， G_2 包含所有大于 t_1 像素；

②分别求出 G_1 和 G_2 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;

(6) 将特征值存如图像特征数据库中。

4.4.2 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) 用高斯滤波对图像进行平滑滤波, 用于消除噪声干扰;

(3) 用 sobel 算子对图像进行锐化求梯度;

(4) 对锐化后的图像求边缘方向角, 角度范围在 0 度 — 360 度;

(5) 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存入图像特征数据库中。

5. 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。

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。

下面分别以图像的直方图 and 中心矩为例来说明图像特征匹配的相似性度量函数计算方法。

1、基于直方图的图像特征匹配

设 $(h_{Q1}, h_{Q2}, \dots, h_{Qi}, \dots, h_{QN})$, $(h_{I1}, h_{I2}, \dots, h_{Ii}, \dots, h_{IN})$ 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)、一般欧氏距离度量函数

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

- (2)、加权距离函数

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

- (3)、相交法度量函数

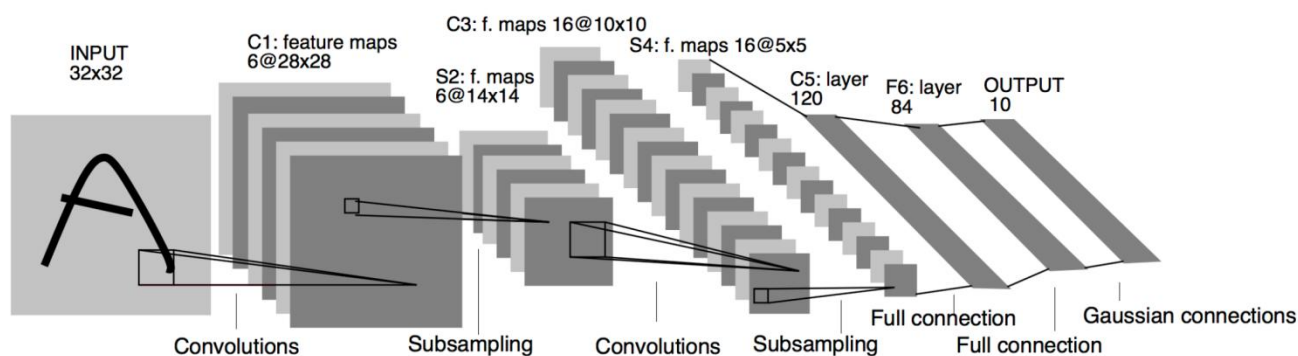
$$D_I(Q, I) = \frac{\sum_{i=1}^N \min(h_i^Q, h_i^I)}{\sum_{i=1}^N h_i^Q}$$

2、中心矩法的度量函数

设 $M_{HiQ}, M_{SiQ}, M_{ViQ}$ 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, $M_{HiI}, M_{SiI}, M_{ViI}$ 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

$$D(Q, I) = \sqrt{w_H \sum_{i=1}^3 (M_{Hi}^Q - M_{Hi}^I)^2 + w_S \sum_{i=1}^3 (M_{Si}^Q - M_{Si}^I)^2 + w_V \sum_{i=1}^3 (M_{Vi}^Q - M_{Vi}^I)^2}$$

除此之外, 鼓励大家使用新的方法进行特征提取和匹配。如可以使用各种深度神经网络进行图像特征的提取。基于深度学习的特征提取方法首先需要设计一个深度神经网络, 深度神经网络的设计方法有很多, 以简单的 LeNet 为例。



LeNet 为卷积层块和全连接层块两个部分。

卷积层块里的基本单位是卷积层后接最大池化层: 卷积层用来识别图像里的空间模式, 如线条和物体局部, 之后的最大池化层则用来降低卷积层对位置的敏感性。卷积层块由两个这样的基本单位复堆叠构成。在卷积层块中, 每个卷积层都使用的窗口, 并在输出上使用sigmoid 激活函数。第一个卷积层输出通道数为 6, 第二个卷积层输出通道数则增加到 16。这是因为第二个卷积层比第一个卷积层的输入的高和宽要小, 所以增加输出通道使两个卷积层的参数尺寸类似。卷积层块的两个最大池化层的窗口形状均为, 且步幅为 2。由于池化窗口与步幅形状相同, 池化窗口在输入上每次滑动所覆盖的区域互不重叠。卷积层块的输出形状为(批量大小, 通道, 高, 宽)。当卷积层块的输出传入全连接层块时, 全连接层块会将小批量中每个样本变平 (flatten)。也就是说, 全连接层的输入形状将变成二维, 其中第一维是小批量中的样本, 第二维是每个样本变平后的向量表示, 且向量长度为通道、高和宽的乘积。全连接层块含 3 个全连接层。它们的输出个数分别是 120、84 和 10, 其中 10 为输出的类别个数, 若要进行图像特

征的提取，则最后一个全连接层的输出个数可以是用户定一个要提取的特征数量，若要提取 10 个特征，则输出个数就是 10，若要提取 128 维特征，则输出个数就是 128。

设计好深度神经网络后，可以使用多种方法对该网络进行训练，比如

(1) 基于分类问题的训练方法，可以将该神经网络作为一个分类器训练，则最后一个全连接层的输出个数是类别数，在训练足够多次后，摘除最后一层全连接层，就得到了提取特征的深度神经网络，提取的特征的维数等于倒数第二层全连接层的输出个数。

(2) 基于距离的训练方法，这种训练方法的理念是“同类图片的距离越近越好，不同类图片的距离越远越好”，基于这个理念，定义三元组损失函数如下：

$$L = \sum_{i=1}^N [D(f(x_i^a), f(x_i^p)) - D(f(x_i^a) - f(x_i^n)) + \alpha]$$

D 是一个距离度量函数，可以使用欧氏距离、加权距离等距离函数。 x_i^a, x_i^p 属于同一类别， x_i^a, x_i^n 属于不同类别。当中括号中的式子的值大于等于零时，取了+号之后是本身，此时 loss 大于零，会更新参数；当中括号中的式子的值小于零时，取了+号之后是零，这时就没办法产生梯度，也就无法更新参数。训练时则需要选取这样的三元组 (x_i^a, x_i^p, x_i^n)

五、注意事项

1. 部署项目时，注意项目端口号的使用
2. 学习在 Java 代码中调用 python 代码

六、附录