图像搜索原理和实践

徐长茂

收钱吧

目录

- ✓ 什么是图像搜索
- ✓ 图像数学表示
- ✓ 图像特征提取
- ✓ 相似性度量
- ✓ 向量检索
- ✓ 工程实践
- ✓ 总结和问答

收钱吧

什么是图像搜索

图像搜索是指根据图像内容搜索出相似的图像

内容相似

类别相同

收钱吧

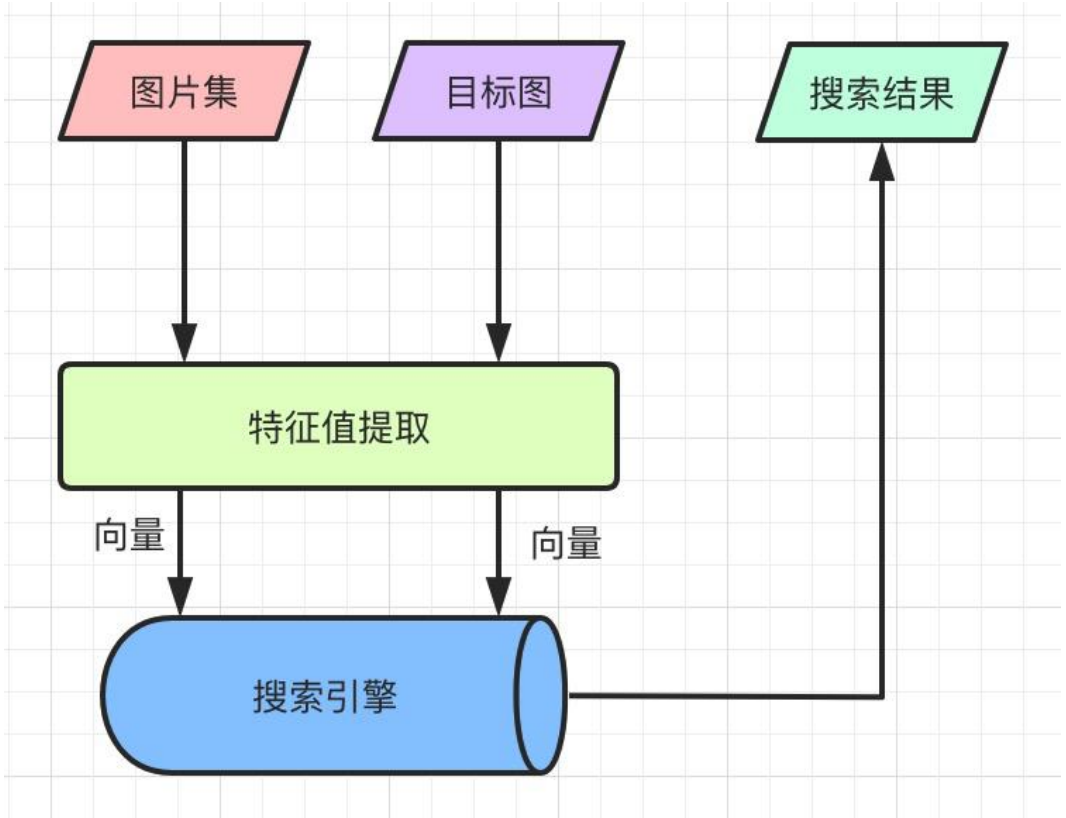
图像搜索应用

- ✓ 拍照购物 – 京东、淘宝
- ✓ 拍照识物 – 形色
- ✓ 图片查重 – 视觉中国
- ✓ 搜来源 – 百度谷歌识图

收钱吧

图像搜索过程

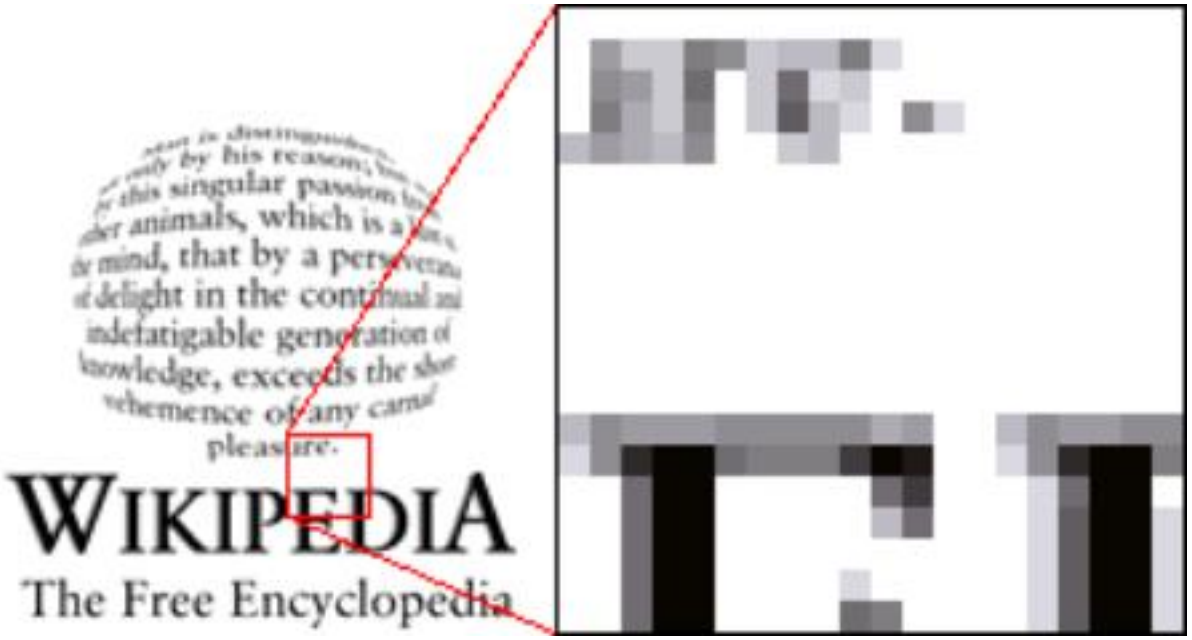
- ✓ 按照某种算法计算图像特征值
- ✓ 构建特征值库
- ✓ 计算目标图像特征值
- ✓ 在特征值库中检索
- ✓ 返回搜索结果

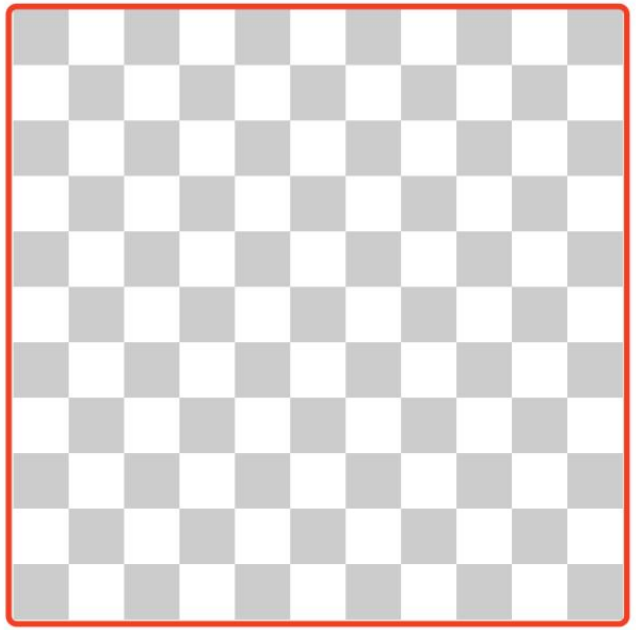


收钱吧

图像数学表示

图像 – 像素点的集合





收钱吧

图像数学表示

图像可以用 *二维矩阵* 表示，每个像素点对应的就是矩阵中的一个元素

二值图像

二值图像的像素点只有黑白两种情况，因此每个像素点可以由 0 和 1 来表示

RGB图像

红、绿、蓝作为三原色可以调和成任意的颜色，每个像素点包含 RGB 共三个通道的基本信息

[156,22,45] [255,0,0] [0, 156, 32] [14,2,90]

[12,251,88] [78,12,3] [94, 90, 87] [134,0,2]

[240,33,44] [5,66,77] [1,28,167] [11, 11, 11]

[0,0,0] [4,4,4] [50,50,50] [100, 10, 10]

4x4 RGB图像矩阵表示示例

收钱吧

图像特征提取

图像特征

- ✓ 抽象的数值、向量或者符号
- ✓ 可描述图像内容、特点
- ✓ 可对比计算或者度量

Lenna

收钱吧

图像特征提取

图像哈希表示

图像通过一系列的变换和处理最终得到的一组哈希值 (fingerprint)

哈希算法

- 均值哈希 – aHash 计算速度较快、精确度较低
- 感知哈希 – pHash 精确度较高、计算速度较慢
- 差异值哈希 – dHash

计算步骤

缩小尺寸 -> 灰度化 -> 变换 -> 计算均值 -> 计算哈希值

灰度化心理学公式

Gray = 0.30R + 0.59G + 0.11B

收钱吧

图像特征提取

感知哈希计算过程

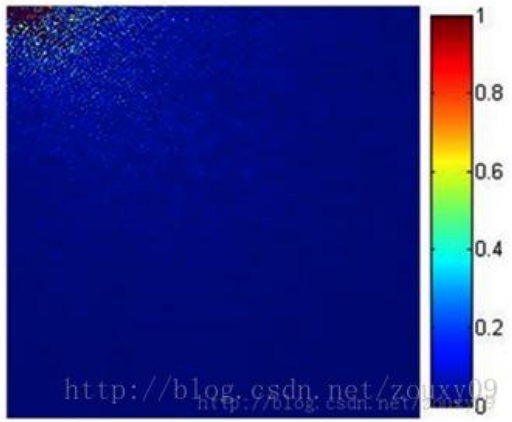
- 缩小尺寸 – 图像采样，忽略细节，保留结构、明暗等基本信息
- 简化色彩 – 将图片转化成灰度图像，进一步简化计算量
- DCT变换
- 缩小DCT – 保留左上角的8*8的矩阵，这部分呈现了图片中的低频内容
- 计算平均值
- 计算hash值 1110000101000001111110011110101100100101110110010001011100000001

```
// 缩小尺寸，灰度化简化色彩
int[][] grayMatrix = getGrayPixel(imagePath, 32, 32);

// 计算DCT
grayMatrix = DCT(grayMatrix, 32);

// 缩小DCT，计算平均值
int[][] newMatrix = new int[8][8];
double average = 0;
for (int i = 0; i < 8; i++) {
    for (int j = 0; j < 8; j++) {
        newMatrix[i][j] = grayMatrix[i][j];
        average += grayMatrix[i][j];
    }
}
average /= HASH_LEN * 1.0;

// 计算hash值
char[] chArr = new char[HASH_LEN];
for (int i = 0; i < 8; i++) {
    for (int j = 0; j < 8; j++) {
        if (newMatrix[i][j] < average) {
            chArr[i * 8 + j] = '0';
        } else {
            chArr[i * 8 + j] = '1';
        }
    }
}
```

原图

低频成分


高频成分



http://blog.csdn.net/zouxy09

收钱吧

图像特征提取

SIFT算法

SIFT(Scale-invariant feature transform)是一种提取局部特征的算法

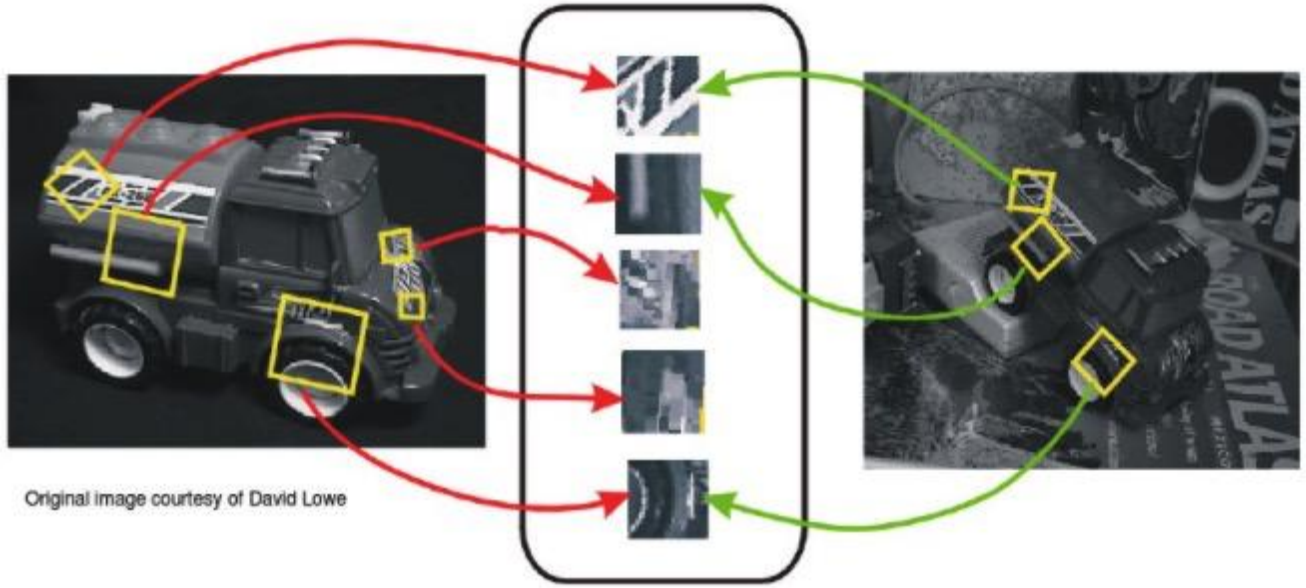
在尺度空间寻找极值点，提取位置，尺度，旋转不变量

局部特征

边缘、角，区域 亮点 暗点

计算步骤

- 尺度空间的极值检测
- 特征点定位
- 特征方向赋值
- 特征点描述

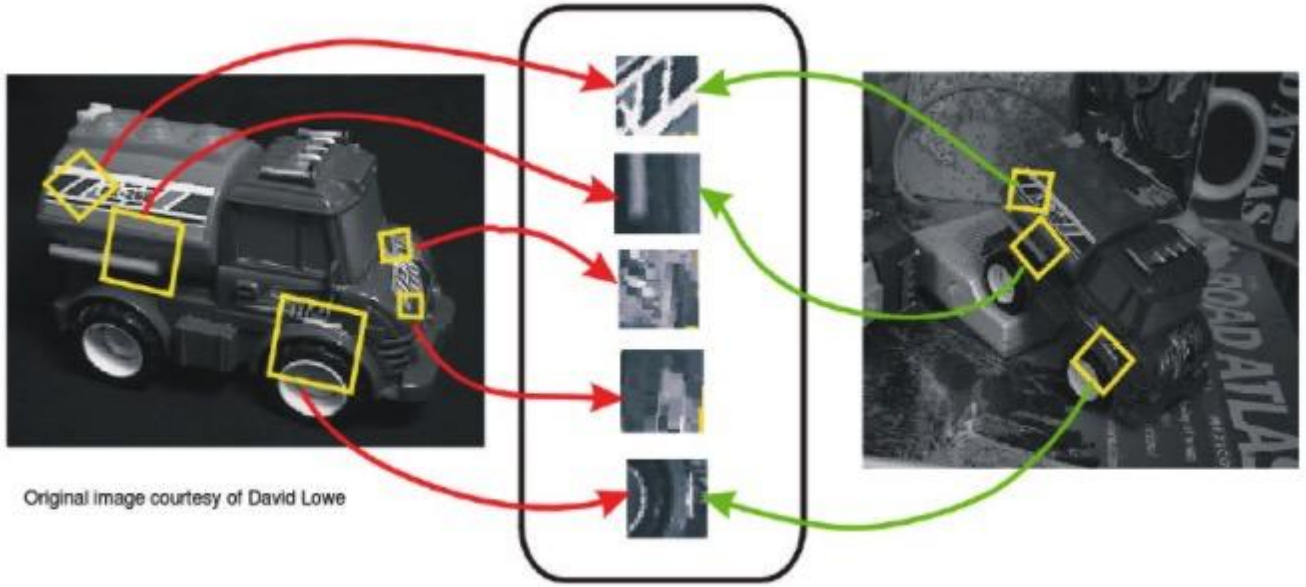


收钱吧

图像特征提取

SIFT算法优点

- 对旋转、尺度缩放、亮度变化保持不变性，对视角变化、仿射变换、噪声也保持一定程度的稳定性。
- 独特性(Distinctiveness)好，信息量丰富，适用于在海量特征数据库中进行快速、准确的匹配.
- 多量性，即使少数的几个物体也可以产生大量SIFT特征向量。 K * 128维向量



收钱吧

图像特征提取

SIFT算法向量融合

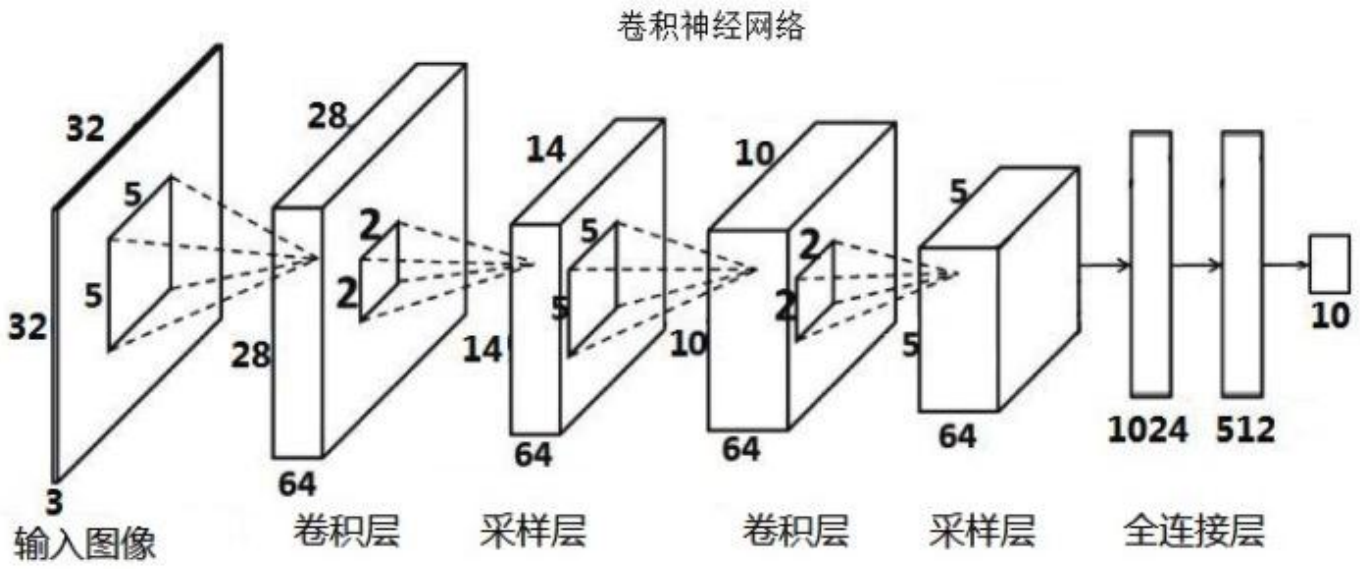
- ✓ 一幅图像提取出来的特征点有多个，且每一个特征点都是一个局部向量
- ✓ 为了进行相似性计算，需要先将这一系列特征点融合编码为一个全局特征
- ✓ 融合编码相关的算法
  - BOW – 训练图像集预处理，K-means对训练集SIFT特征进行聚类，构造图像码本
  - VLAD
  - Fisher vector

收钱吧

图像特征提取

CNN特征

- 基于 CNN 卷积神经网络提取图像特征
- 通过 CNN 提取出来的图像特征也是一个多维向量



收钱吧

相似性度量

如何根据图像特征判断相似?

图1 64位感知哈希值

1110101000010000001101010100011010011000000010100001001101010100

图2 64位感知哈希值

1100100111100100001000101000100000100000000000000000000000000000

图1 – 16维向量

(0.5479437,0.8057077,0.31508246,0.6889049,0.80902284,0.31425726,0.7628163,0.762055,0.5371946,0.56051695,0.43620002,0.33228478,0.6553062,0.74335515,0.51085174,0.20000218)

图2 – 16维向量

(0.43774754,0.8853849,0.28815767,0.44375873,0.8846527,0.29914922,0.60492235,1.174769,0.4345614,0.67761606,0.75630516,0.19378087,0.76082563,0.9290161,0.9209442,0.3088285)

图 1

图 2

收钱吧

相似性度量

相似性度量(Similarity Measurement)，通常采用的方法就是计算样本间的“距离”(Distance)或者相似度

常见计算方法

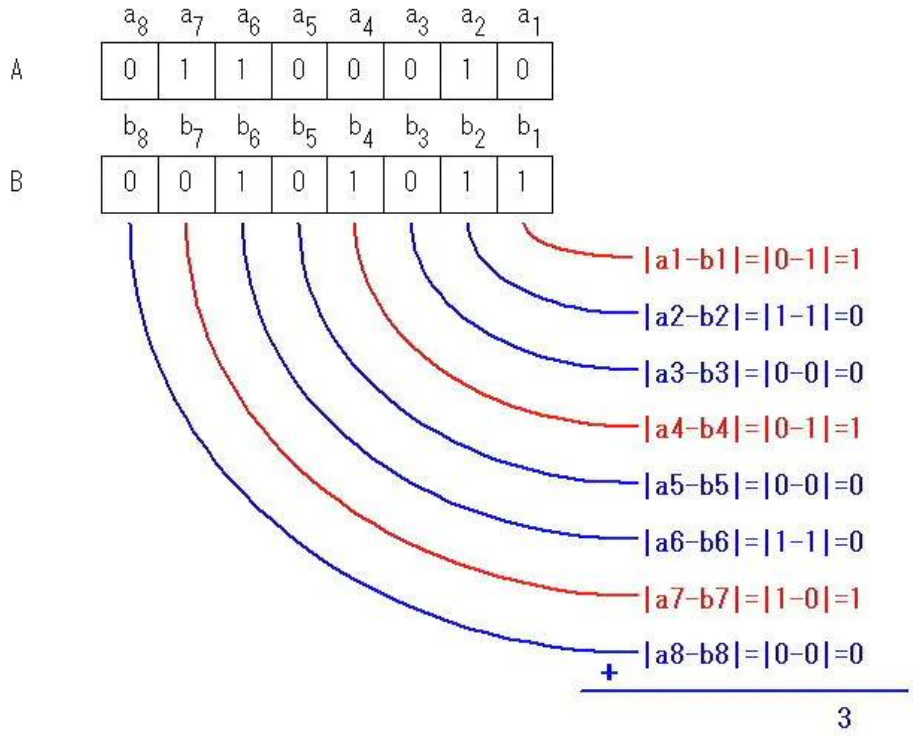
- 汉明距离
- 欧氏距离
- 标准化欧氏距离
- 夹角余弦
- 曼哈顿距离
- 切比雪夫距离

收钱吧

相似性度量

汉明距离(Hamming Distance)

- 在信息论中，两个等长字符串之间的Hamming distance是两个字符串对应位置的不同字符的个数
- 两个字符串，汉明距越小越相似



收钱吧

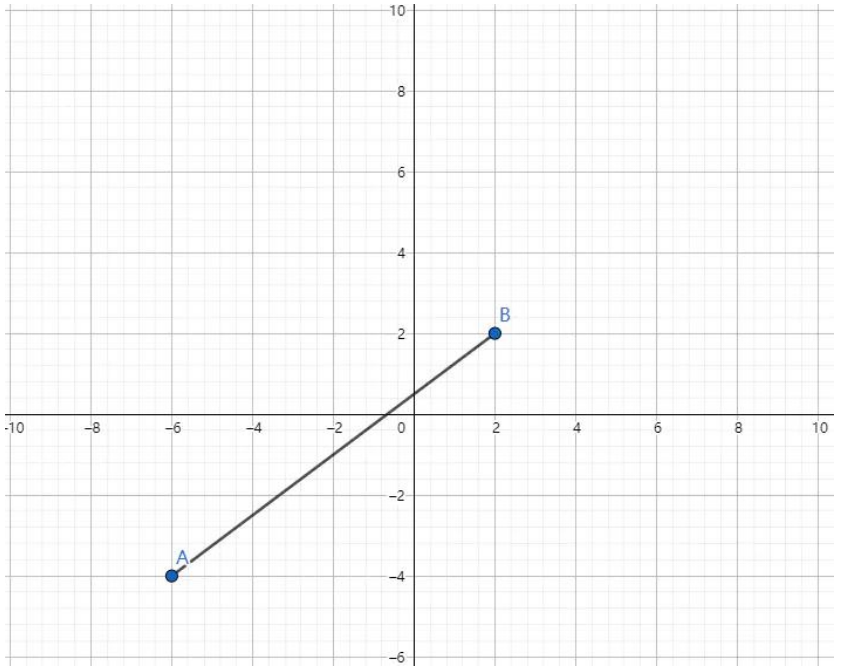
相似性度量

欧氏距离(Euclidean Distance)

- 欧氏距离（也称欧几里得度量）指在m维空间中两个点之间的真实距离，或者向量的自然长度（即该点到原点的距离）
- 欧氏距离越小，两个向量相似度越大
- 欧氏距离需要保证各维度指标在相同的刻度级别, 否则需要对数据进行标准化

计算公式

$$dist(A,B) = \sqrt{\sum_{i=1}^{n}(A_i - B_i)^2}$$

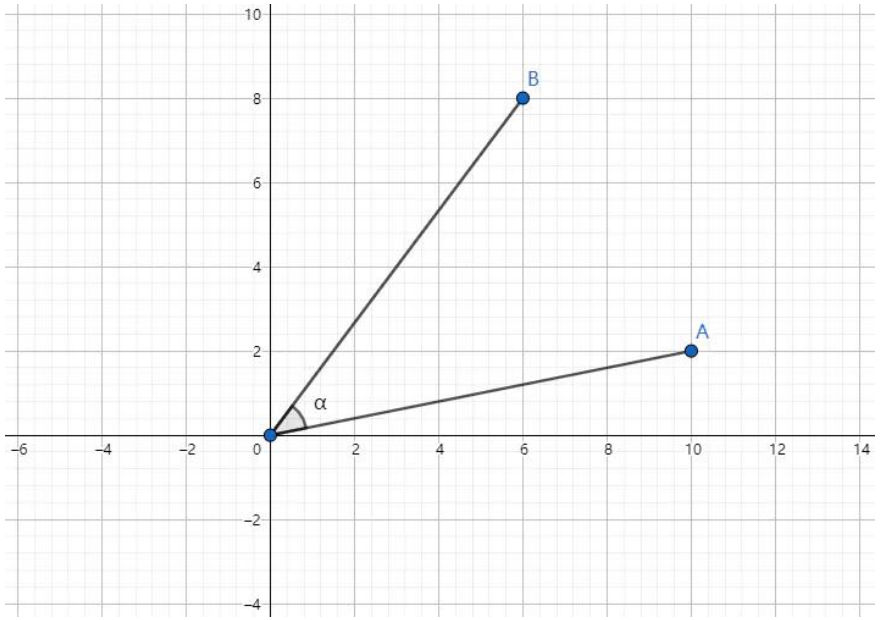


收钱吧

相似性度量

余弦相似度(Cosine Similarity)

- 余弦相似度，又称为余弦相似性，是通过计算两个向量的夹角余弦值来评估他们的相似度
- 角度越小越相似
- **具体应用：数学之美 系列 12 – 余弦定理和新闻的分类**



收钱吧

向量检索

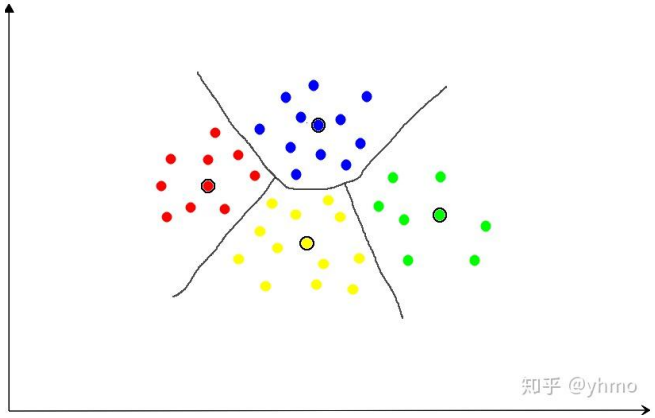
- ✓ 假设已有N张图片特征向量，如何在N张图片中查找与目标图最相似的图片?
- ✓ 根据相似度度量方法，分别目标图片与其他图片特征向量欧式距离，返回距离最小的即最相似
- ✓ 向量检索两个基本参数: 一个是n，意思是拿n条目标向量去数据库中做检索。另一个是k，意思是查找离目标向量最近的前k个向量，我们一般称为top–k

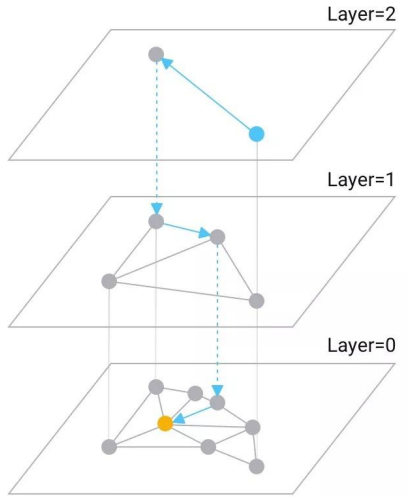
收钱吧

向量检索

向量检索方法

- ✓ 存在最近邻检索 (Nearest Neighbor Search，NN) – 精度高, 查找速度慢
- ✓ 近似最近邻检索 (Approximate Nearest Neighbor Search， ANN) – 以精度换时延
- ✓ 空间切分 – 向量聚簇 KD树 LSH, 高维向量检索效率低
- ✓ 近邻图 (Proximity Graph) – HNSW NSG 亿级向量毫秒级查询





知乎 @pills101

收钱吧

向量检索

向量检索库

- Fasis – Facebook开源的一个高性能的高维向量相似度检索和聚类库, C++编写
- ***Milvus*** – 集成了 Faiss、NMSLIB、Annoy 等广泛应用的向量索引库，上手简单。号称十亿高维向量秒级查询
- PostgreSQL + cube/imgsmlr插件
- ElasticSearch + 插件

工程实践 – PG在图像搜索中的应用

背景

- ✓ 2020年6月，内控部门发现有BD使用其他商户真实性审核资料图片进行大量虚假开户
- ✓ 为了防止此类事件再次发生，需要自动识别商户入网上传图片与存量图片是否相似

方案选型

- 感知哈希 – 两张图片内容有一定差异时识别效果差
- 小波变换 – 识别效果可接受
- 最终方案 PostgreSQL + imgsmlr插件 + Haar小波变换提取特征值
- 迭代方案 CNN + Milvus

工程实践 – PG在图像搜索中的应用

Haar小波变换提取特征值

- imgsmlr插件地址 https://github.com/postgrespro/imgsmlr
- ImgSmlr — is a PostgreSQL extension which implements similar images searching functionality.
- ImgSmlr method is based on ***Haar wavelet transform***. The goal of ImgSmlr is not to provide most advanced state of art similar images searching methods. ImgSmlr was written as sample extension which illustrate how PostgreSQL extendability could cover such untypical tasks for RDBMS as similar images search.

计算步骤

- Decompress image.
- Make image black&white.
- Resize image to 64x64 pixels.
- Apply Haar wavelet transform to the image.
- Pattern converted into signature
- Shuffle signature for less sensitivity to image shift.

收 收钱吧

工程实践 – PG在图像搜索中的应用

- PG + imgsmlr插件图像搜索demo
  - ✓ 插件编译和安装
  - ✓ 创建 imgsmlr_demo库
  - ✓ 创建 imgsmlr扩展
  - ✓ 创建 image表
  - ✓ 创建特征值pat表
  - ✓ 查询相似图

工程实践 – PG在图像搜索中的应用

优化背景

- pg 9.4版本 — 阿里云pg版本 9.4 支持imgsmlr插件，其他版本不支持
- 阿里云服务器 CPU 4核, 内存 8G, 磁盘 2T SSD
- 图像特征值，16维向量, gist索引
- 阿里云pg参数未调整，也不能自行调整

单表

数据量	浮点数精度	执行时间
100万	6位	618ms
200万	6位	1259ms
1000万	6位	8000ms
1200万	3位	6800ms
1200万	4位	7301ms

分区表

数据量	分区数	浮点数精度	执行时间
1200万	64	6位	5680ms

收钱吧

工程实践 – PG在图像搜索中的应用

优化之道

- 源码编译安装PG 12.3版本
- 图片特征值建立分区表
- PG dblink并行查询
- 调整内存参数，数据预热到内存，减小磁盘IO

工程实践 – PG在图像搜索中的应用

测试条件

- 服务器配置 cpu 8核 内存16G 磁盘 2T SSD
- 数据库参数 pg shared_buffer = 12GB, max_worker_processes = 16, max_parallel_workers_per_gather = 8, max_parallel_workers = 8
- 使用dblink + 并行查询
- pg_prewarm插件进行数据预热
- 每个分区返回十条记录

测试结果

行数	数据总量	分区数	分区表数据量	索引数总据量	分区索引数据量	浮点数精度	数据预热	分区查询时间	总查询时间
1000万	-	64	-	-	-	6	无	-	<500ms
2000万	-	64	-	-	-	6	无	-	<600ms
3000万	-	64	-	-	-	6	无	-	<1000ms
5000万	13GB	64	200MB	6.4GB	100MB	6	是	< 150ms	< 1000ms
1亿	26GB	64	400MB	12.8GB	200MB	6	是	< 300ms	< 2000ms

总结

- dblink建立连接的时间消耗对查询总时间有一定的影响
- 数据预热需要将image_sig表中数据和索引idx_img_sig全部预热到内存，减少分区查询磁盘随机IO
- 图片特征值表建立32分区表，cpu核数也提高到32核。这样即能降低dblink连接时间，又能够充分利用cpu

结论

- 查询时间与数据总量线性相关
- ***理论上单机配置32核计算型cpu + 32GB内存，一秒即可在一亿张图片中搜索到相似图片***

收钱吧

工程实践 – PG在图像搜索中的应用

优化结果

- 中等配置服务器 CPU16核 16G内存
- 图片特征值16分区表
- 千万级图片100ms返回结果
- 一亿图片1000ms以内返回结果
- git链接 https://git.wosai-inc.com/OSP/imgsmlr

总结

- ✓ 开源的向量检索库简化了相关应用开发的入门门槛
- ✓ 以上原理不止于搜图，也可以用于计算广告、电商商品推荐等等场景

收钱吧


收钱吧