

基于深度学习的 SIFT 图像检索算法^①



苏勇刚¹, 高茂庭²

¹(常州工业职业技术学院, 常州 213164)

²(上海海事大学, 上海 201306)

摘要: 深度学习作为一个新的机器学习方向, 被应用到计算机视觉领域上成效显著. 为了解决分布式的尺度不变特征转换 (Scale-Invariant Feature Transform, SIFT) 算法效率低和图像特征提取粗糙问题, 提出一种基于深度学习的 SIFT 图像检索算法. 算法思想: 在 Spark 平台上, 利用深度卷积神经网络 (Convolutional Neural Network, CNN) 模型进行 SIFT 特征抽取, 再利用支持向量机 (Support Vector Machine, SVM) 对图像库进行无监督聚类, 然后再利用自适应的图像特征度量来对检索结果进行重排序, 以改善用户体验. 在 Corel 图像集上的实验结果显示, 与传统 SIFT 算法相比, 基于深度学习的 SIFT 图像检索算法的查准率和查全率大约提升了 30 个百分点, 检索效率得到了提高, 检索结果图像排序也得到了优化.

关键词: 卷积神经网络 (CNN); 深度学习; 图像检索; 重排序

引用格式: 苏勇刚, 高茂庭. 基于深度学习的 SIFT 图像检索算法. 计算机系统应用, 2020, 29(9): 164–170. <http://www.c-s-a.org.cn/1003-3254/7628.html>

SIFT Image Retrieval Algorithm Based on Deep Learning

SU Yong-Gang¹, GAO Mao-Ting²

¹(Changzhou Institute of Industry Technology, Changzhou 213164, China)

²(Shanghai Maritime University, Shanghai 201306, China)

Abstract: Deep learning is a new field in machine learning research, and to apply it to computer vision achieves effective result. To solve the problem that the traditional Scale-Invariant Feature Transform algorithm (SIFT) has low efficiency and extracts image features roughly, A SIFT image retrieval algorithm based on deep learning is proposed. The algorithm idea is that on the Spark platform, a deep Convolutional Neural Network (CNN) model is used for SIFT feature extraction, and Support Vector Machine (SVM) is utilized for unsupervised clustering of image library, then the adaptive image feature measures are used to re-sort the search results to improve the user experience. The experiment results on the Corel image set show that compared with the traditional SIFT algorithm, the precision and recall rate of the SIFT image retrieval algorithm based on deep learning is increased by about 30 percentage points and the retrieval efficiency is improved, the resulting image order is also optimized.

Key words: Convolutional Neural Network (CNN); deep learning; image retrieval; resort

随着大数据时代的到来, 庞大数据集的图像检索已经广泛应用于计算机视觉^[1]和人工智能^[2]等先进领域. 如何从这个大数据库的数字资源中快速检索到用户需要的图像信息和提高图像检索效率, 就成为了计

算机视觉一个亟待解决的问题.

在图像库中, 查找具有包含指定内容或特征图像的这个过程, 定义为图像检索. 其中图像之间的相似度值高低对一个图像检索算法的性能起着至关重要的作

① 基金项目: 国家自然科学基金 (61202022)

Foundation item: National Natural Science Foundation of China (61202022)

收稿时间: 2019-12-06; 修改时间: 2020-01-02, 2020-02-25, 2020-03-24; 采用时间: 2020-04-14; csa 在线出版时间: 2020-09-04

用。目前的图像检索方法分为早期的监督图像检索和无监督图像检索方法。监督图像检索方法中使用的图像是人工标记的,而无监督图像检索所使用的是基于深度特征的。在计算机视觉领域,起初的图像检索是根据图像的低级特征(人工特征)来检索,比如 SIFT^[3]、Bow^[4] 和 VLAD^[5], 作为特征提取的主流算法,取得了一定的成效,但其效果仍然不能令人满意。传统图像检索效果不佳的原因是低级特征无法表达图像的内容,所以图像检索的深度特征逐渐取代低级特征,目前神经网络提取深度特征能力得到广大科研人员的认可。

随着机器学习的迅猛发展,图像检索领域运用了卷积神经网络模型,代表性的模型有 Alex Net^[6] 和 VGG-Net^[7] 等。CNN 模型大多数都被用来确定每一目标的位置和类别。比如 CNN 学习 SIFT 特征^[8] 和深度特征,利用学习来的图像特征来做目标检测,成功应用于艺术品的图像分类。但是, CNN 算法存在一些弊端,例如边缘和位置信息容易被忽视。对此,文献[9]提出了对卷积层的卷积特征提取与加权的解决思路,从而使得包含边缘和位置信息的元素被赋予更大的权重。同时 CNN 算法还存在不能适用于不同尺寸的图像问题,文献[10]对此改进 CNN 算法即在卷积层和全连接层中加了 SPP (Spatial Pyramid Pooling) 层,这样在不同规格尺寸的图像下进行学习以及生成多种尺度大小的特征。其实最主要的问题在于,当深度学习生成的高维图像特征较多时,就会造成维灾难问题。同时社交媒体时代的快速发展,网络图像数量的爆炸式增长大,给大规模图像检索带了巨大挑战。若使用常规的检索算法,检索效率会受到极大限制。

针对传统图像检索的检索效率低、图像内容无法准确表达和高维图像特征的维灾难等问题和借鉴深度学习网络的优点,提出一种基于深度学习的 SIFT 图像检索算法,对 CNN 的卷积层的选择框构造图像金字塔、池化层融合了 SIFT 算法,来保证图像的位置特征不丢失,再利用 Spark^[11] 大数据平台的 SVM 对图像库进行无监督聚类,然后再利用自适应的图像特征度量来对检索结果进行重排序,以改善用户体验。

1 基本概念

1.1 CNN

卷积神经网络 (CNN) 包含以下几种层:

卷积层 (convolutional layer), 由多个卷积单元构

成,但只能提取低级的特征。

矩阵卷积: 计算图像的特征,其中有两种方法: 全卷积和有效值卷积。

全卷层的计算如式 (1) 所示为:

$$z(u, v) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} X_{i,j} \cdot K_{u-i, v-j} \quad (1)$$

其中, $X_{i,j}$ 是图像特征, $K_{u-i, v-j}$ 卷积核, $z(u, v)$ 是图像卷积特征。

假设 X 是 $m \times m$ 阶, K 是 $n \times n$ 阶矩阵, K_{rot} 是由 K 旋转 180° 得到, 有效值卷积的计算如式 (2)、式 (3) 为:

$$z(u, v) = \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} X_{i+u, j+v} \cdot K_{rot i, j} \cdot X(i, j) \quad (2)$$

$$X(i, j) = \begin{cases} 1, 0 \leq i, j \leq n \\ 0, \text{others} \end{cases} \quad (3)$$

其中, $X(i, j)$ 为 1 表示 $X_{i,j}$ 有效卷积特征, 若为 0 则表示无效卷积特征。

池化层 (pooling layer), 在上面的几层特征提取和运算后, 会产生维度很大的特征, 将特征切成几个区域, 取其最大值或平均值, 得到新的、维度较小的特征, 其计算如式 (4):

$$\begin{cases} \text{new_height} = (\text{input_height} - \text{filter_height}) / S + 1 \\ \text{new_width} = (\text{input_width} - \text{filter_width}) / S + 1 \end{cases} \quad (4)$$

其中, input_height , input_width 分别是池化特征矩阵的高度和宽度, filter_height , filter_width 分别是自定义的过滤矩阵的高度和宽度, new_height , new_width 分别是新特征矩阵的高度和宽度。

注意: 池化层的输出深度与输入的深度相同。另外池化操作是分别应用到每一个深度切片层, 如图 1 所示。

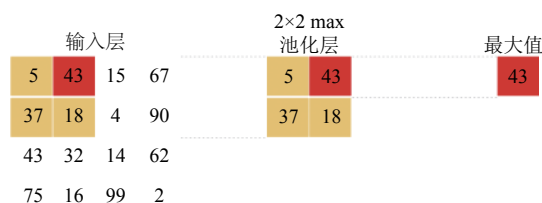


图 1 池化过程演示

全连接层 (fully-connected layer), 把全部的局部特征联系在一起, 然后生成全局特征, 最终用来计算最后每一类的得分, 如图 2 所示。

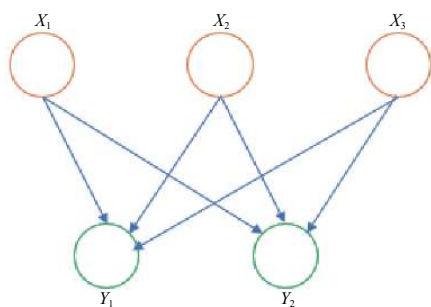


图2 全连接示意

$$(X_1, X_2, X_3) * \begin{pmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \\ W_{31} & W_{32} \end{pmatrix} = (Y_1, Y_2)$$

其中, (X_1, X_2, X_3) 是输入神经元, 而 (Y_1, Y_2) 是输出神经

元, 两层之间的连接就是卷积核: $\begin{pmatrix} W_{11} & W_{12} \\ W_{21} & W_{22} \\ W_{31} & W_{32} \end{pmatrix}$.

1.2 相似性度量

相似性度量极大关乎一个算法的性能, 其方法一般分为特征度量、CNN 特征度量和距离度量.

1.2.1 图像特征度量

图像特征度量是指通过图像的颜色、位置和形状等特征来衡量它们的相似性.

颜色特征: 即图像区域的平均色代表. 为了方便颜色相似度的计算, 将色度坐标 (色调 h , 饱和度 s , 亮度 v) 统一转换为柱坐标系下的欧式空间坐标 (c_1, c_2, c_3) 表示, 如式 (5) 所示.

$$c_1 = s * \cos(h), c_2 = s * \sin(h), c_3 = v \quad (5)$$

位置特征: 图像的主目标位置坐标 (\bar{x}, \bar{y}) , 用其重心坐标 (x, y) 的归一化表示, 如式 (6) 所示.

$$(\bar{x}, \bar{y}) = \left(\frac{x}{W}, \frac{y}{H} \right) \quad (6)$$

式 (6) 中, W 和 H 分别代表图像的宽度和高度.

形状特征: 主要包括形状大小 ρ 和偏心率 e . e 是图像最适椭圆的短、长轴之比; ρ 是目标面积占图像总面积的百分比.

综合考虑颜色、位置和形状等多个特征, 图像 i 的特征用矢量表示为 $F_i = (c_{i1}, c_{i2}, c_{i3}, \bar{x}_i, \bar{y}_i, \rho_i, e_i)$. 采用高斯函数计算图像相似度, 在计算图像 i 和图像 j 之间的特征相似度中分别需要用到颜色特征方差 σ_1 、位置特征方差 σ_2 、形状特征方差 σ_3 .

颜色相似度计算如式 (7) 所示.

$$D_1 = \exp \left(\frac{(c_{i1} - c_{j1})^2 + (c_{i2} - c_{j2})^2 + (c_{i3} - c_{j3})^2}{3\sigma_1^2} \right) \quad (7)$$

位置相似度计算如式 (8) 所示.

$$D_2 = \exp \left(-\frac{(\bar{x}_i - \bar{x}_j)^2 + (\bar{y}_i - \bar{y}_j)^2}{2\sigma_2^2} \right) \quad (8)$$

形状相似度计算如式 (9) 所示.

$$D_3 = \exp \left(-\frac{w_\rho(\rho_i - \rho_j)^2 + w_e(e_i - e_j)^2}{\sigma_3^2} \right) \quad (9)$$

式 (9) 中, w_ρ 和 w_e 分别为形状大小和偏心率的加权系数, $w_\rho + w_e = 1$.

上述图像归一化的图像特征 D 越接近 1, 两个图像越相似, D 越接近 0, 两个图像越不相似.

综合多个特征的图像相似度是各特征间相似度的加权平均, 设 w_1 、 w_2 和 w_3 分别表示颜色特征、位置特征和形状特征的加权系数, $w_1 + w_2 + w_3 = 1$, 两幅图像 i 与 j 之间的多特征相似度计算如式 (10) 所示.

$$D_{\text{object}}(i, j) = w_1 D_1 + w_2 D_2 + w_3 D_3 \quad (10)$$

利用用户对返回结果集的反馈, 提出一种自适应的特征权值调整, 依据返回结果集的主特征来增强该特征, 在每次检索时, 每次检索后加强结果图像的主特征的加权系数, 从而可以使特征相关程度较大的图像排在较前的位置, 以改善与用户交互体验. 即当用户检索完, 若所选择图像和查询图像间相似度最高的特征为特征 i , 则增大特征 i 的加权系数来加强该特征, 同时减小其它特征 $j(j \neq i)$ 的加权系数以减弱其它特征, 权值调整具体计算如式 (11) 所示.

$$w'_i = \frac{w_i + \frac{1}{N}}{1 + \frac{1}{N}}, w'_j = \frac{w_j}{1 + \frac{1}{N}}, j \neq i \quad (11)$$

式 (11) 中, N 为图像库中图像的总数.

而传统的距离度量方法因为要满足距离的条件, 所以会在原来的模型上加以另外的限制条件, 增加了模型的复杂度. 同时对于很多传统的距离度量方法, 因为有距离的概念, 在样本扰动一点, 其距离相差不大, 难以区分不同类的图像^[12].

文献 [13] 图像检索领域的双线性相似性度量 (Bilinear Similary Measure, BSM), 克服了距离度量的缺陷. 它是由成对的相似性函数 S 是算法学习得到的, S 函数公式如式 (12) 所示.

$$S_w(p_i, p_j) = p_i^T W p_j, W \in R^{d \times d} \quad (12)$$

其中, p_i, p_j 是两个样本图片的特征向量, W 是本文要学习的矩阵。

2 基于深度学习的 SIFT 图像检索算法

SIFT 算法主要应用于图像检索工作, 大致流程: 先按照某种规则生成尺度空间, 在尺度空间检测图像位置来剔除尺度和旋转变化的兴趣点, 然后选取稳定的兴趣点作为关键点同时也为分配一个方向或多个方向, 最后利用关键点的邻域向量来度量图像的相似程度。SIFT 算法优势在于图像缩放、旋转和亮度变化保持不变性。

2.1 SIFT 算法

SIFT 算法的处理过程一般分为以下几步:

第 1 步. 生成尺度空间。

第 2 步. 在尺度空间检测并精确定位极值点。

第 3 步. 设定关键点方向参数, 并生成其的描述子。

第 4 步. 最后通过 SIFT 特征向量来计算图片之间的相似度。

下面以一个实例 SIFT 算法, 设有两幅大小不一的红花 a 和红花 b 图像, 如图 3 所示。

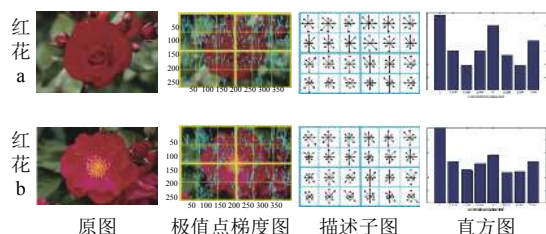


图 3 SIFT 算法的图解过程

开始先进行 SIFT 特征提取, 在尺度空间中检测出描述子, 然后将不同方向的描述子映射相对应方向的局部子直方图, 最后计算直方图的相似度。若相似度系数越接近 0, 则表示两幅图像越相似。

根据上述 SIFT 算法的图像相似度的计算过程, 分析了 SIFT 算法的缺陷: (1) 大数据时代下的图像库数量爆炸式增长, SIFT 算法的检索效率也随之急剧下降; (2) SIFT 特征比较粗糙和单一。为此, 本文提出基于深度学习的 SIFT 图像检索算法 (SIFT Image Retrieval Algorithm based on Deep Learning)。

2.2 本文算法

本文算法通过 CNN 和 SIFT 特征相似度度量来提

高相似度的精度, 并采用基于内存计算的并行计算框架 Spark 和 SVM 算法对原始图像库进行分类来缩小检索范围, 从而大大提升算法的效率并自适应改善图像检索结果的排序, 其处理过程总共 5 步, 流程图如图 4。



图 4 本文算法流程图

第 1 步: 统一提取原始图像库的 CNN、SIFT 特征;

第 2 步: 通过 Spark 的 MLlib 库中的 SVM 来训练图像特征这些数据分类, 同时计算出每个图像库类别的均值 CNN、SIFT 特征;

$$ave(C, N) = \frac{1}{n} \sum_{i=1}^n (C, N) \quad (13)$$

其中, C, N 分别是图像 CNN 特征、SIFT 特征。

第 3 步: 建立训练、测试数据集并训练优化 CNN 网络 (如图 5 所示): 依次是卷积层、池化层、激活层、全连接层、全连接层、分类层; 其中, 为便于建模, 对图像进行分块 (如图 6 所示), 输入图像为 32×32 , 70% 数据为训练集, 15% 数据校验集, 15% 数据测试集, 通过动量法动态调整学习率的训练、校验和测试, 卷积层、池化层、激活层和全连接层的参数都得到优化调整, 最终本文神经网络实现了分类的最优化效果。本文的目标函数如式 (14) 所示:

$$F_{i,j} = F_{i,j} - \alpha \frac{\partial}{\partial F_{i,j}} P(a, b) \quad (14)$$

其中, $F_{i,j}$ 是图像特征, a 是学习率, $P(a, b)$ 为均方差损失函数, 用以训练本文神经网络模型。

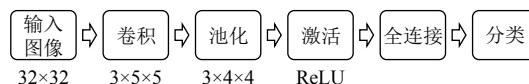


图 5 本文卷积神经网络框架

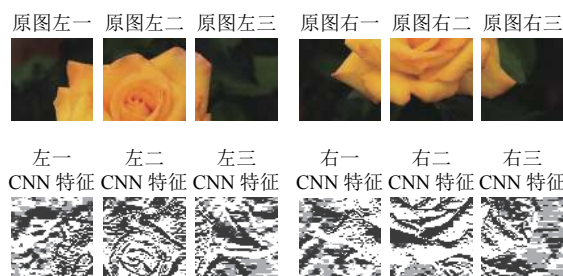


图 6 图像分块的原图和 CNN 特征

第4步:检索图像时,先设定图像相似度度量的阈值,然后比较待查询图像特征与每一类库索引特征的相似性度量来确定类别库,再从类库中产生一些符合查询条件的候选结果,结果排序则是根据图像 SIFT 和 CNN 特征相似度的值。

$$R_i = \begin{cases} 1, & S_{i,j} \leq t \\ 0, & \text{others} \end{cases} \quad (15)$$

其中, t 为阈值, $S_{i,j}$ 为图像 i, j 之间的相似度量, R 为 1 时,表示相似; R 为 0,则表示不相似。

第5步:在用户从结果集中挑选出最满意的图像之后,再按式(8)对图像各特征的加权系数进行调整,以使用户下一次更好的体验。

在本文算法检索过程中,先对原始图像库图像特征处理并建立训练数据,再利用改进的 CNN 网络对图像特征库进行训练,然后在 Spark 平台下由 SVM 算法进行分类,将图像特征库训练成若干类图像特征库。选出代表类特征库的索引。检索的过程中,最先把图像提取特征与各类库特征索引比较来确实图像所在大概某个或某几个类库,这样大大缩小了查找范围,有效提高图像检索效率,如图7所示。

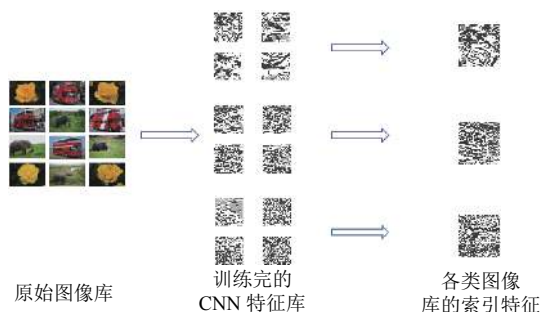


图7 基于 Spark 平台训练图像库

本文算法中选用的是 Spark 平台, Spark 计算效率高,主要是因为采用了基于内存计算的并行计算框架,可以处理大数据以及自带一些经典的机器学习算法,弥补了随着图像库数据增长导致检索效率下降的劣势。

本文算法检索时,先将原始的图像库进行图像 CNN、SIFT 特征提取,然后利用 Spark 平台 MLlib 库中的 SVM 算法对图像特征进行分类。分类之后,取每个类图像库的均值图像特征作为索引,用户检索时,需提交要查询的图像,交由图像特征提取机制进行特征提取,递交给搜索机制,让其根据特征相似度度量的返回查

询结果,同时图像各视觉特征的加权系数会随用户的检索行为而改变(即图像相似性度量的原理),从而达到自适应的效果。若提交的图片不在标准库中,则算法利用离线方式对图像 CNN 特征进行学习,优化结果集;当再次提交时,系统就会返回用户满意的结果。本文算法流程如图8所示。

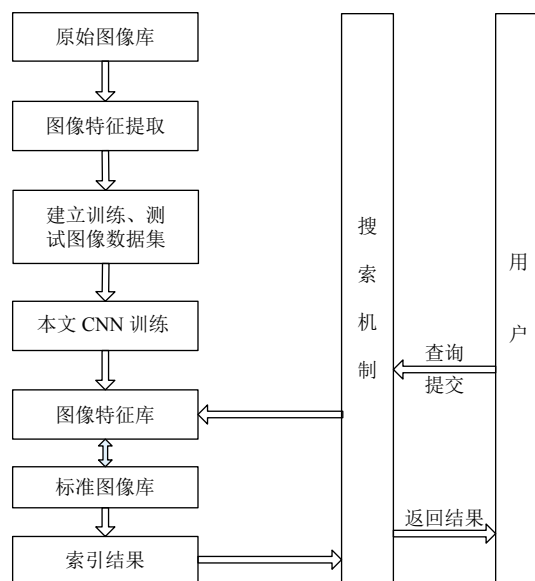


图8 本文算法流程图

3 实验过程与分析

实验硬件环境: 3.5 GHz 主频的 CPU、内存 8 GB, 软件开发环境: 开发工具 PyCharm、开发语言 Python, 并基于 Spark 平台对原图像库进行分类; 实验数据: 10 000 幅 corel 图像库^[14]中的图像, 总共分为 10 个类别, 每类 1000 幅, 分别为花、巴士、水果、大象、建筑、骏马、恐龙、人脸、天空和雪山。

3.1 实验设计

3 个对照实验, 主要验证本文算法比传统 SIFT 算法的性能更佳并且对用户更加友好。实验 1 比较算法查准率; 实验 2 验证在检索海量数据集时本文算法的时间复杂度比传统 SIFT 算法低; 实验 3 验证本文算法检索出的图像结果集排序更合理。

3.2 图像检索性能评价性能

为了评价本文算法的性能, 采用图像检索领域最基本的评价指标: 查全率和均值查准率 mAP (mean Average Precision)。

$$\begin{cases} recall = \frac{relevant\ Correctly\ Retrieved}{all\ Relevant} = \frac{A}{A+B} \\ mAP = \frac{1}{m} \sum_{i=1}^m \frac{A}{R} \end{cases} \quad (16)$$

式(16)中, A 为检索返回图像相关的数量, B 为目标图像相关, 但未检索到的数量; m 为检索的次数, R 为与目标图像相关的数量。

3.3 实验结果与分析

实验 1. 是本文算法与传统 SIFT 算法、VLAD 算法和 BOW 算法对图像库中 3 类(花、巴士、水果-本文挑选了 3 个具有代表性的类别)的查准率, 见表 1。

表 1 传统算法与本文算法对某类图像库的查准率对比

图像类别	传统SIFT	VLAD算法	BOW算法	本文算法
花	0.5747	0.6432	0.7468	0.8606
巴士	0.5897	0.6596	0.7545	0.8987
牛	0.5536	0.6378	0.7269	0.9053
平均值	0.5727	0.6469	0.7427	0.8882

从表 1 和表 2 可知, 与传统 SIFT 算法、VLAD 算法和 BOW 算法相比, 本文算法对各类图像的平均查准率和平均查全率均得到提高, 尤其对于传统的 SIFT 算法, 查准率提高了约 30 个百分点及查全率提高了约 20 个百分点。因为本文算法提取的图像特征提取更加精准, 所以本文图像的查准率和查全率优势明显。

表 2 传统算法与本文算法对某类图像库的查全率对比

图像类别	传统SIFT	VLAD算法	BOW算法	本文算法
花	0.3654	0.1356	0.2564	0.5756
巴士	0.3859	0.1564	0.2789	0.5854
牛	0.3497	0.1268	0.2493	0.5565
平均值	0.3670	0.1396	0.2615	0.5725

实验 2. 本文算法与传统算法、VLAD 算法和 BOW 算法在不同数量级的图像库时的时间复杂度进行对比, 结果详见图 9。

从图 9 可以看出, 随着图像数目规模增大, 传统 SIFT 算法检索耗时呈指数增长, 而 BOW 算法、VLAD 算法和本文算法检索耗时增长相对平缓, 但从中可以得出本文算法的检索效率较高。因为本文算法借助大数据 Spark 平台的高效计算和 SVM 的高效分类, 所以图像库数量激增到 5000 的时候, 本文算法优势比较明显。

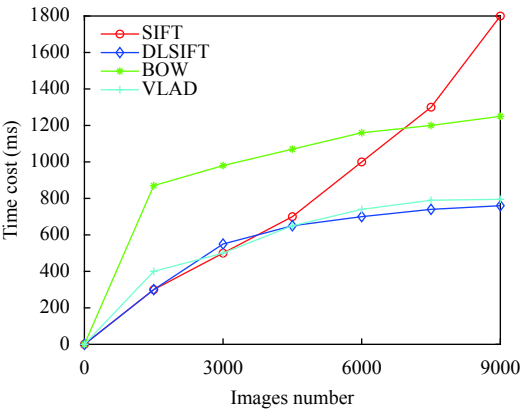


图 9 本文算法与传统 SIFT 算法运行时间对比

实验 3. 本文算法与传统 SIFT 算法的检索结果集图像排序的对比, 结果详见图 10。

图 10 为检索黄色花朵图像时的返回结果集, 图 10(c) 的传统 SIFT 算法页面检索结果中图像的欧式距离较小的排在首面, 唯一的排序因素造成排序结果较不理想, 即会影响到用户的友好体验; 由于本文采用了图像的自适应特征度量, 所以图 10(b) 的本文算法检索图像排序则依据图像 CNN 学习特征和 SIFT 特征的相似度大小来排序, 从而检索图像的结果集排序更加合理, 最终用户的体验得到改善。



图 10 两种算法检索结果排序

4 结语

本文提出了一种基于深度学习的 SIFT 图像检索算法, 该算法适用于大容量的图像数据库检索, 也方便用户对检索结果集筛选, 然后利用 Spark 平台 MLlib 库中的 SVM 算法对图像特征进行分类. 分类之后, 取每个类图像库的均值图像 CNN 特征作为索引, 缩小图像检索范围, 再利用特征度量自适应地重构加权系数, 不但适用于大规模数据集时的图像的检索, 而且检索结果集图像排序更符合用户要求. 实验表明本文算法有效地解决了数据集的暴增带来的图像检索效率低下和检索结果图像集的不合理图像先后次序等问题, 从而极大地改善用户体验.

参考文献

- 1 胡二雷, 冯瑞. 基于深度学习的图像检索系统. 计算机系统应用, 2017, 26(3): 8–19. [doi: [10.15888/j.cnki.csa.005692](https://doi.org/10.15888/j.cnki.csa.005692)]
- 2 Poularakis S, Katsavounidis I. Initialization of dynamic time warping using tree-based fast Nearest Neighbor. Pattern Recognition Letters, 2016, 79: 31–37. [doi: [10.1016/j.patrec.2016.04.016](https://doi.org/10.1016/j.patrec.2016.04.016)]
- 3 Inoue H, Iwahori Y, Kijirikul B, *et al.* SVM based defect classification of electronic board using bag of keypoints. Proceedings of 30th International Technical Conference on Circuits Systems, Computers and Communications. Seoul, Republic of Korea. 2015.31–34.
- 4 韦秋含, 梁海华, 张新鹏. 基于动态 BoW 模型的密文 JPEG 图像检索. 应用科学学报, 2018, 36(4): 628–634. [doi: [10.3969/j.issn.0255-8297.2018.04.006](https://doi.org/10.3969/j.issn.0255-8297.2018.04.006)]
- 5 颜文, 金炜, 符冉迪. 结合 VLAD 特征和稀疏表示的图像检索. 电信科学, 2016, 32(12): 80–85.
- 6 Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. Proceedings of the 25th International Conference on Neural Information Processing Systems. Lake Tahoe, NV, USA. 2012.1097–1105.
- 7 Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv: 1409.1556, 2014.
- 8 朱建清, 林露馨, 沈飞, 等. 采用 SIFT 和 VLAD 特征编码的布匹检索算法. 信号处理, 2019, 35(10): 1725–1731.
- 9 袁晖, 廖开阳, 郑元林, 等. 基于 CNN 特征加权和区域整合的图像检索. 计算机工程与科学, 2019, 41(1): 113–121. [doi: [10.3969/j.issn.1007-130X.2019.01.015](https://doi.org/10.3969/j.issn.1007-130X.2019.01.015)]
- 10 He KM, Zhang XY, Ren SQ, *et al.* Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2015, 37(9): 1904–1916. [doi: [10.1109/TPAMI.2015.2389824](https://doi.org/10.1109/TPAMI.2015.2389824)]
- 11 Zaharia M, Chowdhury M, Franklin MJ, *et al.* Spark: Cluster computing with working sets. Proceedings of the 2nd USENIX Conference on Hot Topics in Cloud Computing. Berkeley, CA, USA. 2010: 1765–1773.
- 12 Weinberger KQ, Saul LK. Fast solvers and efficient implementations for distance metric learning. Proceedings of the 25th International Conference on Machine Learning. New York, NY, USA. 2008.1160–1167.
- 13 Chechik G, Sharma V, Shalit U, *et al.* Large scale online learning of image similarity through ranking. Proceedings of the 4th Iberian Conference on Pattern Recognition and Image Analysis. Póvoa de Varzim, Portugal. 2009.11–14.
- 14 Wang J Z's Research Group. The Pennsylvania State University. Test image database. <http://qikan.cqvip.com/Qikan/Article/Detail?id=27238118>. (2005-10-08).