

DOI:10.11784/tdxbz201604045

基于 Hadoop 的视觉词袋模型图像分类算法

侯春萍¹, 张倩楠¹, 王宝亮², 常 鹏², 孙韶伟²

(1. 天津大学电气自动化与信息工程学院, 天津 300072; 2. 天津大学信息与网络中心, 天津 300072)

摘要: 随着互联网的发展和数字图像获取技术的进步, 传统图像分类算法在处理海量数字图像时, 面临耗时过多、文件系统及处理架构落后的问题. 针对这一问题, 利用主流的 Hadoop 开源分布式计算平台, 引入视觉词袋模型实现对图像的表达, 并对模型的图像直方图化过程做出改进, 提出一种自适应的特征分配方法, 最后采用易于并行的随机森林算法作为分类器, 以充分利用 Hadoop 平台强大的分布式计算能力. 实验显示, 基于 Hadoop 平台的图像分类方法在处理大规模数据集时较单机环境能有效减少时间消耗, 同时具有良好的分类效果.

关键词: Hadoop; 图像分类; 视觉词袋; 随机森林; 软分配

中图分类号: TP391

文献标志码: A

文章编号: 0493-2137(2017)06-0643-06

Image Classification Approach of Bag of Visual Words Model Based on Hadoop

Hou Chunping¹, Zhang Qiannan¹, Wang Baoliang², Chang Peng², Sun Shaowei²

(1. School of Electrical and Information Engineering, Tianjin University, Tianjin 300072, China;

2. Information and Network Center, Tianjin University, Tianjin 300072, China)

Abstract: As the Internet grows and technology of acquiring digital images advances rapidly, problems with the conventional image classification methods gradually arise while dealing with massive digital images, such as being time-consuming and lacking timely update of the file system and processing architecture. To combat this problem, an image classification approach is proposed based on Apache Hadoop, the mainstream open-source distributed processing system. Firstly, the bag of visual words (BoVW) model was utilized to achieve simplified image representations. Meanwhile, an improvement was made to the model during the histogram representation period and an adaptive soft assignment algorithm was proposed. Lastly, the easy-paralleled random forest algorithm was employed as the classifier so as to make full use of the advantages of the platform. Experiments show that the proposed method of image classification based on Hadoop could effectively decrease the computing time compared with single-PC method while dealing with mass images, and at the same time gain good classification results.

Keywords: Hadoop; image classification; bag of visual words; random forest; soft assignment

随着计算机技术和图像采集手段的不断发展, 数字图像数据大规模地生成出来. 图像分类技术利用计算机对图像进行自动化分析和归类, 是当前计算机领域的研究热点之一. 然而, 当数字图像的数据量特别大时, 图像分类算法所依赖的传统文件系统和处理架构受到了很大的挑战, 同时, 大规模的图像分类也面临耗时过长的问题.

一般地, 图像分类^[1]方法由图像特征的提取和基

于特征的分类两个部分组成. 首先, 通过对图像的全局或局部视觉特征的提取, 建立对图像的表达, 然后使用机器学习的方法进行训练和分类. 近年来的视觉词袋 (bag of visual words, BoVW) 模型进一步对特征进行聚集及整合, 在中层语义层面上对图像进行表达, 是一种有效的图像表示方法.

经典的 BoVW 模型一般结合支持向量机 (support vector machine, SVM)^[2]分类器进行, 但该分

收稿日期: 2016-04-18; 修回日期: 2016-11-01.

作者简介: 侯春萍 (1957—), 女, 教授, hcp@tju.edu.cn.

通讯作者: 王宝亮, wbl@tju.edu.cn.

基金项目: 国家自然科学基金资助项目 (61571325).

Supported by the National Natural Science Foundation of China (No. 61571325).

类器在海量高维图像数据处理中具有较大局限性。而随机森林(random forest)建立在统计学习理论上,将随机性引入决策树模型建立组合分类器,易于进行并行化扩展,契合大数据处理中通常采用的云计算平台的分布式特点,为图像分类提供了新的思路。但当图像规模过大时,随机森林分类器也面临分类耗时过多的问题。

为了解决以上问题,本文结合云计算这一新兴计算模式,提出一种基于主流的 Hadoop^[3]开源分布式计算平台的大规模图像分类方法。首先并行地实现基于 BoVW 模型的图像表示,并对模型中图像直方图化的过程进行改进。然后,对随机森林分类器进行并行化设计,以充分利用 Hadoop 平台的分布式计算能力,提高对大规模图像分类的效率。

1 本文方法总体框架

作为当前主流的云计算架构之一, Hadoop 由 Apache 基金会开发,旨在以低成本、高效率的方式来处理海量数据。与适用于小数据量任务的 Storm 平台、高实时性任务的 Spark 平台相比, Hadoop 十分适合用于处理大批量、大数据量、非实时性的任务,更加适合复杂的大规模图像分类工作。

Hadoop 下包括一系列的子项目,最为关键的是分布式文件系统(Hadoop distributed file system, HDFS)及 MapReduce 并行计算框架。其中, HDFS 将 Hadoop 组织为由一个名称节点(NameNode)和大量数据节点(DataNode)构成的主从式集群,为 Hadoop 的上层应用提供高吞吐率的数据读写服务,具有可扩展性和高度容错性。其中,名称节点与外部机进行交互,并且存储集群文件系统的元数据,维护文件和目录的树结构;数据节点冗余式地存储数据文件,并负责任务的执行。

MapReduce^[4]作为 Hadoop 的分布式计算框架,借鉴自函数式编程语言 Lisp,以 Map(映射)和 Reduce(归并)过程对数据处理进行抽象化,过程的输入和输出均以键/值对 < Key, Value > 形式表示。开发者自定义 Map 和 Reduce 函数,实现键/值对之间的映射。工作时, MapReduce 框架通常在名称节点上启动 JobTracker 服务,控制整体任务的执行;在数据节点上启动 TaskTracker 服务,监控各节点的任务情况。如图 1 所示,主要分为以下 4 个步骤。

步骤 1 客户端(Client)通过 JobClient 向集群提交 MapReduce 作业及原始数据,原始数据被默认分

块(Split)后冗余地存储在集群的节点上。

步骤 2 JobTracker 节点根据集群机器及负载情况将 Map/Reduce 任务分发到集群机器上, Map/Reduce 任务节点启动 TaskTracker 维护任务的进行。

步骤 3 Map 任务节点以键/值对 < K_1, V_1 > 形式输入块数据,运行产生中间结果键/值对 < K_2, V_2 >, 缓冲后溢写(spill)到本地磁盘。

步骤 4 Reduce 任务节点并行地从 Map 节点取得中间结果,以键/值对 < $K_2, \text{List}\{V_2\}$ > 形式输入,经过归并后将结果键/值对 < K_3, V_3 > 输出到 HDFS 上。

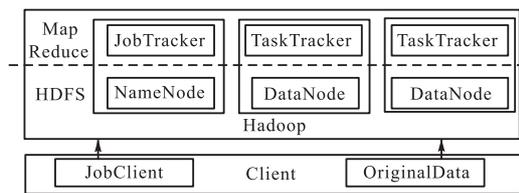


图 1 Hadoop 平台架构

Fig.1 Architecture of Hadoop

在 Hadoop 上进行图像分类主要分为两个阶段。第 1 阶段基于 BoVW 模型进行图像表示,首先利用从训练图像中提取的图像特征建立视觉词典,在此基础上将图像表示为词典的直方图向量形式。第 2 阶段使用随机森林进行并行分类。首先,利用基于 BoVW 模型的训练图像向量训练决策树,生成随机森林;然后,使用随机森林对基于 BoVW 模型的测试图像向量进行分类。整体分类过程在 MapReduce 框架下分布式进行,一共包括并行提取图像特征、并行生成视觉词典、并行生成随机森林、并行分类 4 个 MapReduce 过程,如图 2 所示。

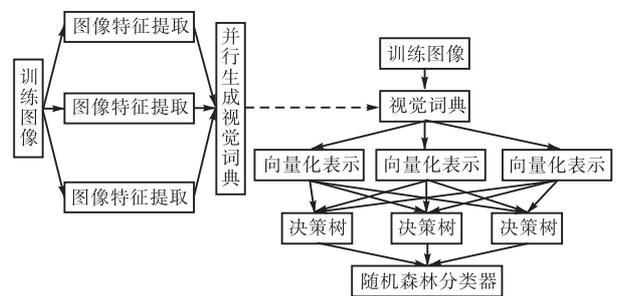


图 2 基于 Hadoop 的图像分类总体框架

Fig.2 Image classification framework based on Hadoop

2 基于 BoVW 模型的图像表示

BoVW 模型借鉴自文本分类领域中表现良好的

词袋 (bag of words) 模型^[5], 将图像看作大量无序的视觉特征的集合, 模型的创建是根据图像特征生成视觉词典的过程, 模型的使用是根据视觉词典将图像表示为向量形式的过程. BoVW 模型避免了直接使用图像特征进行分类所带来的数据杂乱、计算复杂的问题, 同时可以提高图像特征的表现力.

2.1 SIFT 特征的提取

本文选取尺度不变特征变换 (scale-invariant feature transform, SIFT)^[6] 算法进行图像局部特征的提取. 其所提取的 SIFT 特征对图像尺度变换、旋转、亮度变化等保持不变性, 对视角变化、仿射变换也具有一定的稳定性. 算法依次进行图像尺度空间的构建、空间极值点的检测、极值点主方向的分配、描述子的生成等步骤, 最终得到 128 维的 SIFT 特征向量. 本文在 Hadoop 平台上进行海量图像的 SIFT 特征提取, 并行化思路如下所示.

(1) 将训练图像数据集 $\{I(x, y)\}$ 输入 Hadoop 集群, 其中 $I(x, y)$ 表示每一幅图像文件. 当数据集过大时, Hadoop 对其进行分块, 然后存储在 HDFS 上.

(2) Map 节点以键/值对 $\langle \text{Key}, \text{Value} \rangle$ 作为输入, 其中 Key 是图像名, Value 是图像数据 $I(x, y)$.

在各 Map 函数中对图像进行 SIFT 特征提取, 输出结果键/值对 $\langle \text{Key}, \text{Value} \rangle$ 中, 键保持不变, 值是该图像 SIFT 特征的集合 $\{\langle (x_i, y_i), \mathbf{X}_n \rangle\}$, 其中 (x_i, y_i) 是特征点的位置, \mathbf{X}_n 是该处 128 维的 SIFT 向量.

(3) 各 Map 节点产生的 SIFT 特征可以分布式地存储在 HDFS 集群上, 无需进行 Reduce 过程, 因此将 Reduce 函数数目设置为 0.

2.2 BoVW 模型

提取出训练图像的特征后, BoVW 模型将图像特征聚类, 把聚类中心作为视觉单词 (visual words, VW) 组成视觉词典. 在此过程中, 训练图像的大量 SIFT 特征点经过聚类获得有限个视觉单词, 其数量即为词典的大小. 使用词典表示一幅图像时, 将图像的 SIFT 特征匹配为词典的视觉单词, 最后统计视觉单词出现的频次, 将图像表示为词典的直方图向量形式, 如图 3 所示.

聚类过程一般采用 K-means 算法进行, 算法将向量空间中的 n 个特征点根据类内方差和最小的原则分为指定的 K 类, 如式 (1) 所示.

$$\min \sum_{i=1}^K \sum_{x \in C_i} \text{dist}(\mu_i, x_i)^2 \quad (1)$$

式中: C_i 表示第 i 个聚类类别; μ_i 为 C_i 的中心; x_i 为该类别下的数据点.

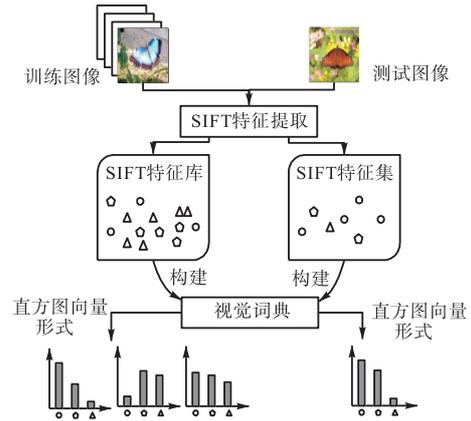


图 3 BoVW 模型的流程
Fig.3 Flow chart of BoVW model

2.3 BoVW 模型并行化

K-means 算法直接有效, 但在串行运算中复杂度较高, 当数据量非常大时, 需要庞大的时间开销. 考虑到算法中待聚类元素均独立计算与中心距离的特点, 使用 MapReduce 框架对其进行并行化. 基本方式为: 对串行算法中的每次迭代都启动一个 MapReduce 任务, 该任务执行待聚类数据与聚类中心距离的计算. 本文使用 Mahout^[7] 中的并行 K-means 聚类方法进行视觉词典的构建, 主要步骤如下.

步骤 1 在每个 Map 节点上, 输入待聚类的 SIFT 特征向量和上一轮 (或初始) 聚类中心, 以 $\langle \text{Key}, \text{Value} \rangle$ 的形式读入, 其中 Key 是行号, Value 是特征向量 \mathbf{X}_n . Map 函数以欧氏距离为标准计算并标示距该向量最近的聚类中心, 输出中间聚类结果 $\langle \text{Key}, \text{Value} \rangle$, 其中 Key 是 \mathbf{X}_n 所属类别的标记 classID, Value 是 \mathbf{X}_n . 伪代码如下.

```
Mapper{
  Map() {
    Min_distance = MAXDISTANCE
    for (i = 0; i < K; i++) {
      if (distance(point, cluster[i]) < min_distance)
      {min_distance = distance (point, cluster[i]);
      currentCluster_ID = i;}
      Key = currentCluster_ID;
      Value_Output = point;
      Emit (Key, Value_Output);
    }
  }
}
```

步骤 2 在执行 Reduce 函数之前, 中间聚类结果在本地进行合并 (Combine), 键值相同的中间结果送至同一个 Reduce 节点. Reduce 输入 $\langle \text{Key}, \text{Value} \rangle$, 其中 Key 是 classID, Value 是键值相同的 \mathbf{X}_n 的集合. Reduce 函数输出 $\langle \text{Key}, \text{Value} \rangle$, 其中 Key 是 classID, Value 是计算产生的均值向量, 是新的聚类

中心. 伪代码如下.

```
Reducer{
  Reduce() {
    Num = 0;
    while (points.hasNext()) {
      currentPoint = points.next();
      Num + = currentPoint.get_Num();
      for (i = 0; i < k; i++)
        {sum[i] + = currentPoint.point[i];}
      mean[i] = sum[i]/Num;
      Key = key;
      Value_Output = mean;
      Emit(Key, Value_Output);
    }
  }
}
```

步骤 3 在主函数中调用上述 Map/Reduce 函数, 比较上一轮聚类中心与本轮中心的距离, 若小于所定阈值, 则聚类完成; 否则使用本轮聚类中心文件启动下一轮 MapReduce 任务, 最终的聚类结果作为视觉词典存储在 HDFS 上.

2.4 算法优化

使用视觉词典表示图像时, 经典的 BoVW 模型采用硬分配^[8]的方式进行 SIFT 特征与视觉单词的匹配, 即通过最近邻查找的方法, 将图像的每个 SIFT 特征对应到一个视觉单词. 实际上, 聚类构建视觉词典的过程存在一定的量化误差, 同时视觉单词具有语义相近或歧义现象^[9]. 在这种情况下对特征进行硬分配, 可能造成误差累积, 为分类带来恶劣影响. 已有的改进方法尝试使用软分配 (soft assignment) 方式, 将特征分配到距其较近的多个单词上, 但是人为设定固定的分配数目, 可能导致过度分配. 因此, 本文提出一种自适应的特征分配方法, 根据 SIFT 特征与视觉单词的距离情况, 基于“词距越小关系越近”的原则, 针对不同的特征点调整软分配的个数, 实现更准确的直方图表示.

设已生成的词典为 $D = \{w_1, w_2, \dots, w_K\}$, 其中 K 是词典的大小, w_i 是第 i 个视觉单词. 计算图像 $I(x, y)$ 的特征向量 X_n 与各个视觉单词的欧氏距离, 按从小到大的顺序为 X_n 建立距离序列, 记作 $d = \{d_1, d_2, \dots, d_M\}$, 其中 M 是序列中维护的距离数量, d_i 是该向量与距其第 i 序近的视觉单词的距离.

在分配过程中, 若图像的某个特征向量与各视觉单词均相对孤立, 则其可能为图像表征和分类带来噪声, 因此将其判定为不可靠特征点, 首先进行判断并剔除. 不可靠特征点的判定依据式 (2).

$$d_1 > \lambda \cdot \text{avg}(\text{dist}(w_{d_i}, w_{d_j})) \quad (2)$$

式中: d_1 为 X_n 与距其最近的单词间的距离; $\text{dist}(w_{d_i}, w_{d_j})$ 为距离序列中任意两个单词的距离; λ 为判别因子, 通常取值 2~4 之间.

对于可靠的特征点, 自适应地为其确定软分配个数, 如式 (3) 所示. 然后, 特征向量被分配到其对应个数的视觉单词上, 其中距离较近的视觉单词被分配较大的权重, 如式 (4) 所示.

$$N = \arg \max_i \{d_i, d_i < \alpha d_1\} \quad i = 1, 2, \dots, M \quad (3)$$

$$\begin{cases} \beta_j = \frac{d_j}{\sum_{j=1}^N d_j} & j = 1, 2, \dots, N \\ d_j = \lg(N+1) - \lg(j) \end{cases} \quad (4)$$

式中: d_i 为距离序列中第 i 个顺序距离; α 为自适应因子, 用以调节分配的个数; β_j 为该特征向量对视觉单词 w_{d_j} 的分配权重.

3 基于随机森林的分类

使用 BoVW 模型获得图像向量后, 采用随机森林进行分类. 随机森林由多个决策树采用随机的方式建立, 通过弱分类器的组合获得高精度的分类器. 随机森林建树时, 采用随机的训练样本子空间和候选属性子集, 因此具有更高的分类准确度和更小的泛化误差上限. 分类预测时, 测试图像向量在每一棵决策树上独立判定, 投票产生分类结果. 由于建树和分类过程的独立性, 随机森林^[10]易于在 MapReduce 框架上进行线性扩展, 十分适合大规模图像分类的场景.

随机森林中的每一棵决策树均由根节点、分支节点、叶节点组成, 从原始训练样本集 S 中随机抽取的训练样本子集 S_{sub} 由根节点输入决策树. 假设训练图像向量 V_n 为 K 维向量, 到达决策树的每个分支节点 N_{node} 处的训练集为 S_{node} . 在每个节点处, 随机选取 V_n 的 m ($m < K$) 个分量组成 N_{node} 处备选分裂属性集合 $F_{\text{node}} = \{f_1, f_2, \dots, f_m\}$, 遍历 F_{node} 的元素对节点进行分裂, 按照基尼不纯度下降最大的原则选出最佳分裂属性, 将该节点分为左节点和右节点. 假设节点 N_{node} 处的集合 S_{node} 包含 G 个分类, 基尼不纯度定义如式 (5) 所示.

$$\text{Gini}(N_{\text{node}}) = 1 - \sum_{i=1}^G p_i^2 \quad (5)$$

式中 p_i 是类别 i 在节点处的占比. 经过分裂后, 设集

合被分为 H 个子集合, 则分裂后的平均基尼不纯度如式(6)所示.

$$\text{GiniE}(H) = \sum_{i=1}^H \frac{|T_i|}{|T|} \cdot \text{Gini}(i) \quad (6)$$

式中: H 为子集合的个数; T_i 为子集合 i 处样本的个数; $\text{Gini}(i)$ 为子集合 i 的基尼不纯度.

本文在 Hadoop 的 MapReduce 框架下实现随机森林的建树过程, 并行化步骤如下.

步骤 1 假设生成有 T 棵决策树的随机森林, 则在原始训练样本集 S 上, 使用 Bootstrap 方法建立 T 个训练样本子集 $\{S_1, S_2, \dots, S_T\}$, 作为每棵树的输入.

步骤 2 依据树的个数, 启动等量的 Map 任务. Map 函数以 $\langle \text{Key}, \text{Value} \rangle$ 作为输入, 其中 Key 是森林标识 RF, Value 是训练样本子集 S_i . Map 函数对每个 N_{node} 进行最佳分裂属性的计算, 递归产生左右分支, 最终返回中间结果键值对 $\langle \text{Key}, \text{Value} \rangle$, 其中 Key 是森林标识 RF, Value 是每棵树的信息 DT_i .

步骤 3 Map 任务全部完成后, Reduce 节点取得 Map 结果并整合, 输出 $\langle \text{Key}, \text{Value} \rangle$, 其中 Key 为森林标识 RF, Value 是随机森林全体, 记作 $\{\text{DT}_1, \text{DT}_2, \dots, \text{DT}_T\}$.

使用随机森林对测试图像进行分类时, 将向量化的测试图像及随机森林作为 Map 输入, 各决策树独立并行判定后, 在 Reduce 过程中投票产生分类结果.

4 实验与分析

本文的分布式实验环境是由 4 台 Dell R730 服务器组成的 Hadoop 集群, 其中 1 台为 Namenode 节点, 3 台为 Datanode 节点. 服务器均配置 Xeon E5-2603 1.6 GHz CPU, 内存 8 GB, 采用 Ubuntu14.04 操作系统, 部署 Hadoop CDH 5.5.0 版本, Java 环境为 Oracle jdk1.7.0. 作为对比的单机实验在相同配置的

服务器上进行, 操作系统为 Ubuntu14.04. 实验使用经典的 Caltech-101 图像数据库, 该数据库包括 101 类视觉物体图像和 1 类背景图像, 共 9 146 幅. 每一类包含图像数量为 80 ~ 130 幅, 大小约为 300 像素 \times 200 像素, 类内图像具有较大差异性, 在图像分类研究中应用广泛.

为验证所提方法的分布式运行性能, 本文从 Caltech-101 每一类中随机选取 30 幅作为训练图像, 一共 3 060 幅; 其余作为测试图像, 一共 6 086 幅. 在分布式集群和单机环境下分别运行本文方法的分布式和单机形式, 并设定视觉词典大小为 1 000, 随机森林建树个数为 200. 实验均进行 10 次, 结果取 10 次运行时间的平均值, 表 1 给出了分布式及单机环境下各个子过程的运行耗时.

表 1 算法加速性能

Tab.1 Acceleration performance of the algorithm

实验环境	视觉词典构造时间/s	训练时间/s	分类时间/s
Hadoop 集群	1 249	758	1 257
单机	3 027	1 964	2 853
加速比	2.42	2.59	2.27

由表 1 可见, 基于 Hadoop 的分类方法在词典构建、随机森林训练、随机森林分类 3 个子过程上均产生了加速效果, 加速比分别为 2.42、2.59、2.27, 较单机执行时间有明显缩短. 不难预见, 若进一步扩大集群规模, 集群上的分类速度可能有更明显的提高.

为测试随机森林的建树个数对本文所提方法分类效果的影响, 表 2 给出了建树个数与分类准确率之间的关系. 该实验在 Caltech-101 上随机选取 bonsai、dog、leopards 等 8 类图像, 在每类图像中分别选取 30 幅用作训练图像, 20 幅作为测试图像, 设置视觉词典大小为 300. 结果显示, 当建树个数过少时, 分类准确率随着建树个数的增加而增大; 当建树个数达到 200 时, 分类器的性能趋于稳定.

表 2 随机森林建树个数对分类性能的影响

Tab.2 Influence of classification accuracy in tree number

%

建树数量	1	2	10	50	100	200	300	500
分类准确率	42.17	43.28	49.20	60.17	68.21	74.93	74.87	75.84

在此基础上, 进一步验证本文所提方法的图像分类性能, 将本文算法与文献[11]中基于高维特征的分布式分类算法、经典 BOVW 分类算法进行比较, 考察 3 种算法在不同数量的训练图像下的分类效果. 考虑到训练图像集的规模及表 2 中随机森林建树个数的规律, 设定建树个数为 200, 视觉词典大小

为 300. 在 8 类图像中随机地选取 5、10、15、20、25、30 幅作为训练图像, 在剩余图像中选择 20 幅作为测试图像, 以平均分类准确率作为分类效果指标, 图 4 显示了 3 种方法的分类性能. 如图 4 所示, 本文提出的基于软分配的 BoVW 算法的性能优于经典 BoVW 算法及文献[11]中的方法.

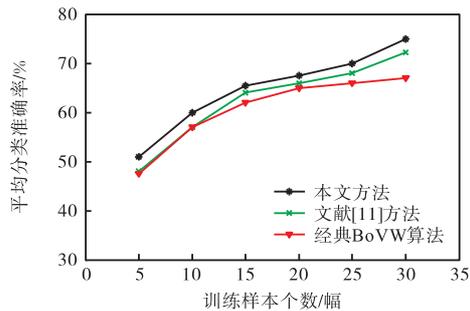


图 4 各算法平均分类准确率对比

Fig.4 Comparison of average classification accuracy

5 结 语

Hadoop 是大数据处理领域的主流计算平台,本文利用 Hadoop 解决海量图像分类中耗时过长的的问题. 首先对 Hadoop 流程进行了简要说明,然后在传统 BoVW 模型的基础上,提出一种自适应的图像特征软分配方法,并将图像特征提取、视觉词典创建和随机森林训练及分类的整体过程基于 Hadoop 进行并行化实现. 实验对比显示,该图像分类方法可以获得良好的分类效果,同时充分利用 Hadoop 在分布式存储和并行运算上的优势,有效减少分类的时间开销.

参考文献:

- [1] Doukim C, Dargham J, Chekima A. State of the art of content-based image classification[C]//*The 2014 International Conference on Computational Science and Technology*. Kota Kinabalu, Malaysia, 2014: 1-6.
- [2] Foody G M, Mathur A. A relative evaluation of multi-class image classification by support vector machines [J]. *IEEE Transactions on Geoscience and Remote Sensing*, 2004, 42(6): 1335-1343.
- [3] Sivaraman E, Manickachezian R. High performance and fault tolerant distributed file system for big data storage and processing using Hadoop[C]//*The 2014 International Conference on Intelligent Computing Applications*. Coimbatore, India, 2014: 32-36.
- [4] Hadoop Architecture Guide[EB/OL]. http://hadoop.apache.org/docs/r1.2.1/hdfs_design.html, 2013-08-04.
- [5] 宋枫溪. 自动文本分类若干基本问题研究[D]. 南京: 南京理工大学计算机系, 2004.
Song Fengxi. Studies on Some Essential Problems in Automatic Text Categorization[D]. Nanjing: Department of Computer Science and Technology, Nanjing University of Science and Technology, 2004(in Chinese).
- [6] Lowe D G. Distinctive image features from scale-invariant keypoints[J]. *International Journal of Computer Vision*, 2004, 60(2): 91-110.
- [7] 牛怡晗, 海沫. Hadoop 平台下 Mahout 聚类算法的比较研究[J]. *计算机科学*, 2015, 42(S1): 465-469.
Niu Yihan, Hai Mo. Comparison research on Mahout clustering algorithms under Hadoop platform[J]. *Computer Science*, 2015, 42(S1): 465-469(in Chinese).
- [8] Chougrad H, Zouaki H, Alheyane O. Soft assignment vs hard assignment coding for bag of visual words. [C]//*The 10th International Conference on Intelligent Systems: Theories and Applications*. Rabat, Morocco, 2015: 1-5.
- [9] van Gemert J C, Veenman C J, Smeulders A W, et al. Visual word ambiguity[J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010, 32(7): 1271-1283.
- [10] van Essen B, Macaraeg C, Gokhale M, et al. Accelerating a random forest classifier: Multi-core, GP-GPU or FPGA?[J]. *IEEE International Symposium on Field-Programmable Custom Computing Machines*, 2012, 282(1): 232-239.
- [11] Liu Qi, Liang Peng, Zhang Haitao, et al. Distributed image classification based on high-order features [C]//*2015 12th IEEE International Conference on Electronic Measurement and Instruments*. Qingdao, China, 2015: 1122-1125.

(责任编辑: 王晓燕)