Scalable Image Retrieval with Multimodal Fusion

Yang Peng, Xiaofeng Zhou, Daisy Zhe Wang
University of Florida
Gainesville, FL
{ypeng, xiaofeng, daisyw}@cise.ufl.edu

Chunsheng Victor Fang*
Awake Networks
Mountain View, CA
vicfcs@gmail.com

Abstract

As the number of images grows rapidly on the Internet, the scalability of image retrieval systems becomes a significant issue. In this paper, we propose two distributed clustering algorithms to scale up the bag-of-visual-words model on millions of images and billions of visual features by leveraging distributed systems. We also introduce a multimodal fusion model to utilize textual data to improve the quality of image retrieval. Our experiments on multimodal datasets demonstrated our fusion approach can achieve high retrieval quality compared to image-only retrieval and text-only retrieval.

Keywords Image Retrieval, Multimodal Fusion, Big Data

Introduction

Image retrieval is the search for desired images from an image dataset according to queries from users. Content-based image retrieval (CBIR), which emerged in 1990s, is a special case of image retrieval, where the queries are images and the search process is based on the visual content of images rather than textual captions or image labels. In the following sections, the term "image retrieval" specifically refers to CBIR, since our focus is to solve the image retrieval problem based on visual content on large-scale datasets.

Huge image datasets of terabytes or even petabytes have been generated from the Internet. For example, ImageNet (Deng et al. 2009), an open image dataset for computer science research, contains over 20 million images. And social networks, such as Facebook and Twitter, can generate over petabytes of images everyday. Comparing all the images in an existing dataset to the query images is not a scalable solution. Thus indexing is a necessary step to handle large-scale image datasets. In order to index images, they should be represented as vectors, similar to the bag-of-words model in information retrieval. With this motivation, the bag-of-visual-words model was designed in the computer vision community (Sivic and Zisserman 2003; Philbin et al. 2007) to represent images in "visual words" vectors. Existing indexing approaches in information retrieval, such as inverted indexing, can be directly applied on the "visual words" vectors.

*This work was done when Chunsheng Victor Fang was affiliated with Pivotal Software Inc.

Since the building process of the bag-of-visual-words model requires a lot of time on large image datasets, we designed two distributed clustering algorithms to scale up the building process of the bag-of-visual-words model by utilizing the state-of-the-art distributed systems. In this paper, we also introduce a multimodal ensemble fusion model to achieve higher retrieval quality by leveraging both images and text. Our experiments on multimodal datasets demonstrated this fusion model can achieve high retrieval quality compared to image-only and text-only retrieval.

Our main contributions focus on two aspects:

- First, we have designed and implemented two distributed clustering algorithms on Hadoop, which can run much faster than Mahout K-Means, to build the bag-of-visual-words model on large image datasets with high efficiency.
- Second, we have designed a multimodal ensemble fusion model to combine image-only retrieval and text-only retrieval to achieve higher retrieval quality compared to single-modality retrieval systems.

Overview For the rest of the paper: (i) we introduce the bag-of-visual-words model and systems used inside our system; (ii) we then explain how to scale up the bag-of-visual-words model with the details of the two distributed clustering algorithms; (iii) we present a multimodal fusion model to utilize both images and text; (iv) experiments on image datasets and multimodal datasets are conducted to demonstrate high performance and retrieval quality of our approaches; (v) finally, we discuss the research work related to large-scale image retrieval systems and multimodal fusion.

Background

The bag-of-visual-words (BoVW) model first appeared in early 2000s (Sivic and Zisserman 2003) and has been widely used in the computer vision community for tasks such as category classification (Li and Perona 2005) and image retrieval (Philbin et al. 2007). BoVW can represent one image as a histogram of independent visual words in vector format. Visual words are generated by applying clustering on local features of images. Then we can use indexing approaches to index the visual words vectors of images. The process to build the bag-of-visual-words model on an image dataset is described in Figure 1.

In the feature extraction step, local features, such as interest points or local patches, are extracted from images. We have chosen SIFT (Scale-Invariant Fast Transform) features,

which are invariant to scale, rotation and illumination, making SIFT (Lowe and David 1999) an ideal candidate for the bag-of-visual-words model.

After feature extraction, a clustering algorithm is used to divide features into different clusters. Researchers (Sivic and Zisserman 2003; Li and Perona 2005; Philbin et al. 2007) have commonly used K-Means clustering for its simplicity and rapid convergence, but previous work (Philbin et al. 2007) pointed out K-Means cannot scale up with a large number of clusters. Even a distributed K-Means, such as Mahout K-Means, fails to scale up with large numbers of clusters. Thus we have implemented two distributed clustering algorithms on Hadoop to overcome this issue.

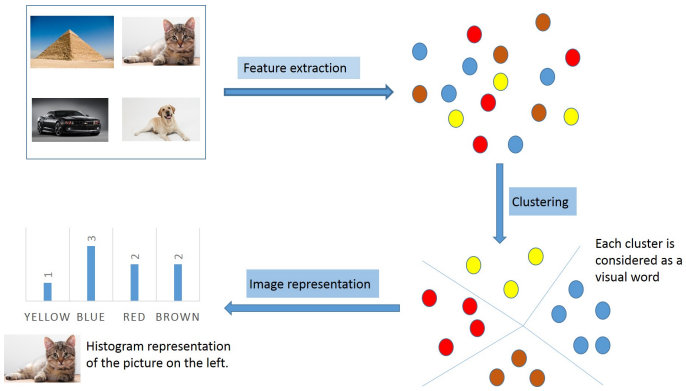


Figure 1: The process of building the BoVW model

After the clustering step, clusters are treated as independent visual words and finally a visual vocabulary is formed with these visual words. Then for a given image, the local features are quantized by assigning the closest visual words to them, to create a histogram of visual words. For example, the cat image is represented as $(1, 3, 2, 2)^T$ in Figure 1.

To handle millions of images and billions of features, state-of-the-art distributed systems were employed for both scalability and stability in our algorithms. All the time-consuming steps, such as feature extraction, vocabulary construction and image representation, are run on Hadoop (Apache Hadoop). Mahout (Apache Mahout), an open-source scalable machine learning library, provides a distributed K-Means implementation on top of Hadoop, which we also utilized in our distributed hierarchical K-Means. Solr (Apache Solr), an information retrieval server based on Lucene (Apache Lucene), is used for indexing and searching.

Scaling up the BoVW Model

To process a large number of images at high speed, the BoVW model is built in parallel on top of Hadoop. After encoding images with visual words, the size of the visual words vectors is significantly smaller than the original image dataset, usually less than 0.1%. A Solr server can then be deployed to handle the indexing and searching quite efficiently without requiring significant resources. In our experiments, the image searching process is very fast, ususally costing less than a few seconds.

Someone may argue the BoVW building process can be conducted offline, so scaling up the building process is not necessary. However, people usually need to run the BoVW building processes many times to tune the vocabulary size, i.e. the number of visual words. And a slow approach may take a few days to finish on large datasets with large numbers of visual words, while a fast approach only costs a few hours in the same scenario, as will be shown in experiments.

Overview

Since a single-node cluster and multi-processing cannot deal with such many images, we employed a Hadoop cluster to provide scalability and stability for our system. The feature extraction and image representation both fit the data-parallel scheme of the Map-Reduce paradigm, hence straight-forward to be parallelized on the Hadoop using Map-Reduce. Lire (Mathias and Chatzichristofis 2008) is used to extract 128-dimensional SIFT features from images.

The bottleneck of the system is the vocabulary construction step, because it involves iterative clustering algorithms to generate visual words from large numbers of local features. As shown in related work (Sivic and Zisserman 2003; Li and Perona 2005; Philbin et al. 2007), K-Means was used as the default clustering algorithm to generate visual words for its fast convergence and good performance. However, the performance of K-Means, even a distributed Mahout K-Means, deteriorates quickly as the number of clusters increases. Thus we have designed and implemented distributed approximate K-Means (d-AKM) and distributed hierarchical K-Means (d-HKM) algorithms on Hadoop to solve this problem. While both d-AKM and d-HKM run much faster than Mahout K-Means, d-AKM has better running time performance than d-HKM for smaller cluster numbers and d-HKM works better for larger cluster numbers, as will be shown in experiments.

Distributed Clustering Algorithms

Since the most time consuming step of each iteration in these three algorithms is the assignment step, where the features are assigned to their corresponding nearest clusters. Let's assume that each HDFS block in Hadoop can hold s features and the Hadoop cluster has sufficient resources, then the time complexity of one iteration of Mahout K-Means (d-KM) on the Hadoop is $O(s \times k)$. The complexities of these three algorithms for one iteration are shown in Table 1.

Table 1: The time complexity of one iteration of Mahout K-Means (d-KM), d-AKM and d-HKM

Algorithm	d-KM	d-AKM	d-HKM
Complexity	$O(s \times k)$	$O(p\%s \times k)$	$O(s \times sqrt(k))$

Distributed Approximate K-Means In the d-AKM, we have applied an approximate process using a randomized k-d tree forest to find the nearest cluster centroid for each feature, as introduced in (Sipla-Anan and Hartley 2008; Muja and Lowe 2009; 2014). The d-AKM is parallelized using Map-Reduce on Hadoop. Let's assume the d-AKM uses at most $p\%k$ comparisons each feature when searching for its closest cluster centroid among k clusters, then the running time complexity for one iteration of d-AKM is reduced to $O(p\%s \times k)$. The time complexity of k-d tree building is $O(k \times logk)$ (Sipla-Anan and Hartley 2008), which is much smaller than $O(p\%s \times k)$, since s is usually much larger than k and $logk$.

Distributed Hierarchical K-Means The d-HKM is shown in Figure 2. At the top layer, a single Mahout K-Means is applied to divide the feature dataset into k_t clusters parallelly on Hadoop. At the bottom layer, for each cluster of the k_t clusters, a single Mahout K-Means is applied to divide this cluster into k_b clusters in parallel. All the bottom-level Mahout K-Means clustering processes run in parallel with the total number of clusters $k = k_t \times k_b$.

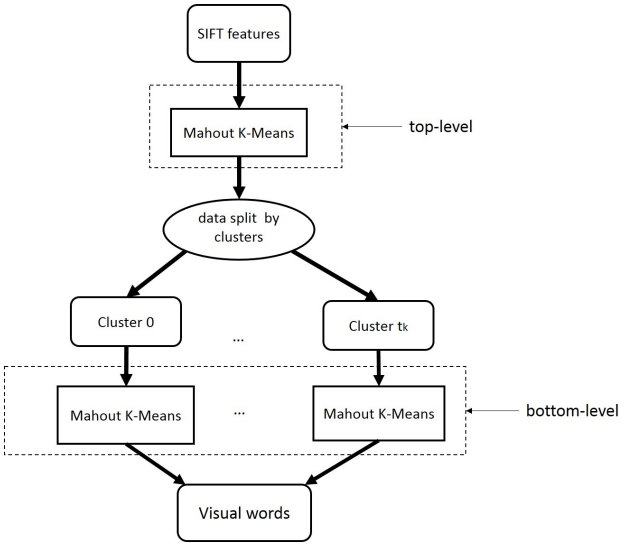


Figure 2: The top-down hierarchical K-Means

At the top level, the running time complexity of one iteration of Mahout K-Means is $O(s \times k_t)$. At the bottom level, for each Mahout K-Means, the time complexity of one iteration is $O(s \times k_b)$. Assuming we have m bottom-level Mahout K-Means clustering running at the same time, the running time complexity of one iteration of all the bottom-level K-Means processes is $O(s \times k_b \times k_t/m) = O(s \times k/m)$. Thus, when k_t, k_b and m are close to each other, the time complexity of one iteration of both the top-level and the bottom-level clustering processes could be $O(s \times sqrt(k))$.

In addition, the number of iterations is also positively related to the number of clusters. The d-AKM usually converges with a similar number of iterations as Mahout K-Means. For the d-HKM, both top-level and bottom-level K-Means converges with smaller numbers of iterations compared to Mahout K-Means. In conclusion, both d-HKM and d-AKM should run much faster than Mahout K-Means. The experimental results comparing Mahout K-Means, d-AKM and d-HKM are explained in Section 5. The code repository is hosted on Github (Peng and Zhou).

Multimodal Fusion

For a query multimodal document, the query image and the title are searched using image retrieval and text retrieval separately, and then a linear rule fusion model is applied to combine the image retrieval and text retrieval results. The system architecture of the multimodal fusion model is shown in Figure 3.

Hadoop and Solr are used to provide storage, distributed computation, indexing and searching services. The BoVW model is built on top of Hadoop and the inverted indexing with tf-idf weighting is provided by Solr. The image retrieval system uses BoVW model, indexing and searching components to represent, index and search images. The text retrieval system uses the indexing and searching services to index and search textual sentences. Then both image retrieval and text retrieval provide similarity scores to the multimodal fusion layer, where a linear rule is employed to combine the similarity scores.

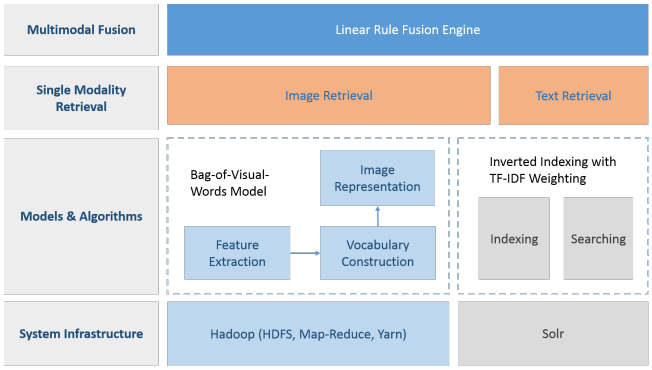


Figure 3: The system architecture of the multimodal fusion model

Linear Rule Fusion

From our observation of multimodal datasets, there exists a complementary relationship between different modalities. Images and text often contain different semantic information. Textual sentences contain more useful information for retrieval in some cases, while images contain more information in other cases. In addition, text retrieval usually has high precision but low recall, while image retrieval has high recall but low precision. Thus we employed a rule-based ensemble fusion approach to combine text retrieval and image retrieval to achieve higher retrieval quality.

For a multimodal document *doc* with one title t and one image i in the datasets, the similarity score of this document returned by the linear rule fusion is:

$$similarity_d = \lambda \times similarity_t + (1 - \lambda) \times similarity_i \quad (1)$$

$$\lambda = \frac{quality_t}{quality_t + quality_i} \quad (2)$$

where $similarity_t$ is the similarity score of the textual sentence in *doc* to the query keywords; $similarity_i$ is the similarity score of the image of the *doc* to the query image; λ is calculated by dividing the retrieval quality of text-only retrieval and image-only retrieval on training queries. Average precision and top-5 accuracy have been used to evaluate the quality of these retrieval systems, as explained in Section 5. The similarity scores of documents are used to sort the documents in descending order to provide a ranked list.

Experiments

The Oxford dataset and ImageNet dataset were used to evaluate the running time performance of our system, especially the distributed clustering algorithms. Experiments on several multimodal datasets also demonstrate multimodal fusion can boost the retrieval quality compared to image-only retrieval and text-only retrieval. The experiments were run on the Pivotal Analytics Workbench (AWB) and Amazon Web Services (AWS).

Datasets

Oxford The Oxford building dataset, provided by University of Oxford (Philbin et al. 2007), contains 5062 images about different landmark buildings in Oxford Campus searched from Flickr.

ImageNet The two training datasets of the ImageNet Large Scale Visual Recognition Challenge 2014 (ILSVRC14) (Russakovsky et al. 2015) were used to provide a large dataset of 185GB with over 1.7 million images and over 230 million features.

Google We have crawled multimodal documents using Google Images with 20 object categories (airplane, cat, dog, etc.) and 14 landmarks (Big Ben, Eiffel Tower, The Taj Mahal, etc.). Each document is composed of one title and one image. For each category/landmark, we have prepared one query for training and one query for testing, with each query containing a few keywords and one image. For each training/testing query, the ground truth results are provided for retrieval quality evaluation.

Twitter Another much larger multimodal dataset has been crawled from Twitter by searching these 20 categories and 14 landmarks using Twitter API. This multimodal twitter dataset contains 200k pairs of textual tweets and images. The training queries and testing queries were also prepared for this dataset.

The specifics of the 4 datasets are shown in Table 2.

Table 2: Dataset specifics

Dataset	image #	image size	feature #	feature size
Oxford	5,062	2.2GB	2,734,105	3.0GB
ImageNet	1,737,734	185.0GB	230,428,057	260.6GB
Google	2,209	1647MB	527,131	831MB
Twitter	200,000	7.2GB	29,308,586	32.6GB

Performance of Mahout K-Means, d-AKM and d-HKM

This section compares the performance of Mahout K-Means (denoted as d-KM in the figures), d-AKM and d-HKM with different cluster numbers. Note *performance* is equivalent to running time in this paper. In all the experiments listed in this paper, the maximum number of comparisons conducted in each iteration for d-AKM is 5% of the number of clusters, and k_t, k_b and m are roughly the same for d-HKM.

The first experiment is to compare the running time of Mahout K-Means, d-HKM and d-AKM on the Oxford dataset with small cluster numbers on AWB, as shown in Figure 4. The running time of d-KM increases almost linearly with the number of clusters, while d-AKM and d-HKM are very flat. d-AKM performs better than d-HKM because d-HKM has very large overhead, due to its two-layer setup and multi-threading mechanism. When the cluster number increases to 10k, the running time of d-KM increases to over 1000 minutes, demonstrating Mahout K-Means cannot scale up with large cluster numbers.

A second experiment to compare d-AKM and d-HKM on the Oxford dataset with larger cluster numbers is shown in Figure 5. The running time of d-AKM increases almost linearly with the number of clusters, while the running time of d-HKM is quite flat as the cluster number increases, since d-HKM has better running time complexity than d-AKM for large cluster numbers.

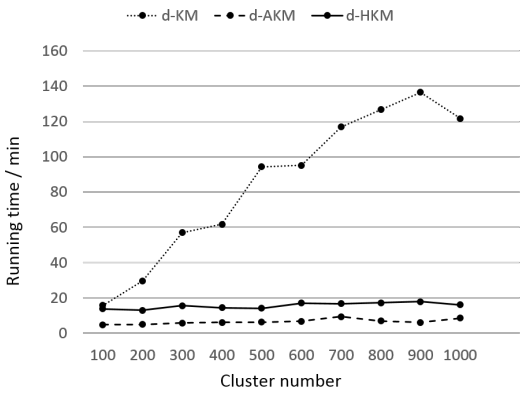


Figure 4: Running time of different algorithms on Oxford dataset

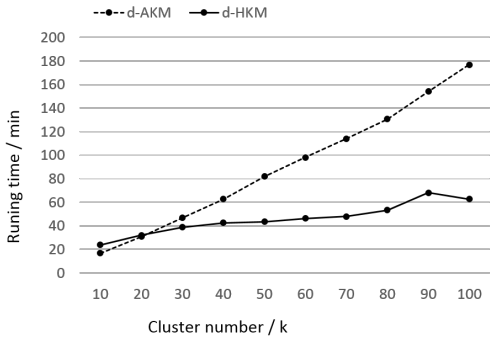


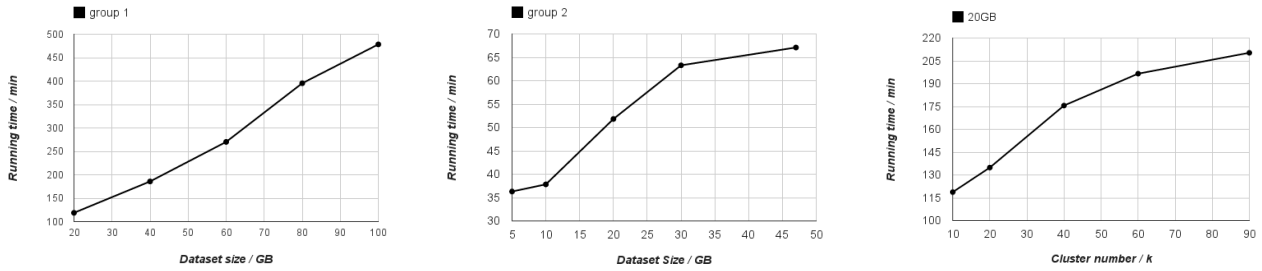
Figure 5: Performance comparison between AKM and HKM with larger cluster numbers. Note: k refers to a thousand in the figure

Performance on Large Datasets

The ImageNet dataset was used for testing the performance of the building process of the BoVW model on large numbers of images. Since d-HKM has better running time complexity than d-AKM and Mahout K-Means with regard to the numbers of clusters, d-HKM was used for vocabulary construction in all the experiments shown in this section.

Different Subsets There are two groups of subsets generated from ImageNet: the first group with 20GB, 40GB, 60GB, 80GB and 100GB; the second group with 5GB, 10GB, 20GB, 30GB and 47GB. The experiments on the first group were run with 10,000 clusters and 300 containers using AWS, as shown in Figure 6(a). The experiments on the second group were run with 2,500 clusters and 2,000 containers on Pivotal AWB, as shown in Figure 6(b).

With sufficient resources, the running time of the building process of BoVW grows sublinearly to the dataset size on Hadoop, as shown in Figure 6(b). But with limited resources, the running time of our approach grows almost linearly to the dataset size on Hadoop, as shown in Figure 6(a). But even with only 300 containers, our approach still



(a) Group 1 with 10k visual words and 300 containers

(b) Group 2 with 2.5k visual words and 2000 containers

(c) 20GB with 300 containers

Figure 6: Experiments on Large Datasets. Note: k refers to a thousand in the figures

can process 100GB image data with 10k visual words in less than 9 hours.

Different Cluster Numbers The number of clusters has significant influence on the running time of the vocabulary construction and image representation steps. Several experiments has been conducted on a 20GB dataset with different cluster numbers from 10k to 90k using 300 containers on AWS as shown in Figure 6(c). The performance of our system is sublinear, very close to $sqrt(k)$, in the number of clusters with d-HKM for vocabulary construction. It can process 20GB with 90k clusters in less than 4 hours, which is quite fast with only 300 containers.

Retrieval Quality

MAP (mean average precision) was used to measure the retrieval quality of the multimodal retrieval system on Google datasets and mean top-5 accuracy was used to measure the quality of the multimodal retrieval system on the Twitter dataset. Top-5 accuracy refers to the accuracy of the top 5 documents of the ranked list returned by any retrieval system for one query. The experimental results are shown in Table 3.

Table 3: Retrieval quality of the multimodal retrieval system on Google and Twitter datasets. MAP is used as retrieval quality measure on the Google dataset and mean top-5 accuracy is used on the Twitter dataset.

dataset	text-only	image-only	multimodal
Google	0.76	0.12	**0.80**
Twitter	0.55	0.11	**0.62**

On both Google and Twitter datasets, multimodal fusion achieves higher retrieval quality compared to image-only and text-only retrieval. The results demonstrate the linear rule fusion can capture the complementary relationship between image retrieval and text retrieval to achieve better retrieval quality than single-modality retrieval systems.

As shown in the table, the image retrieval has very poor retrieval performance due to the imperfect representation of images and lack of sophisticated reranking algorithms. The text retrieval has much better quality than image retrieval because the textual keywords is very informative for text retrieval and the sentences usually contain the keywords. People can improve the quality of image retrieval by applying state-of-the-art reranking algorithms or developing a better representation of images, but it is beyond the scope of this paper. Even in cases image retrieval has better retrieval qualilty, the image retrieval and text retrieval still provide complementary information to each other, and hence our linear rule fusion model can still improve the retrieval quality based on image retrieval and text retrieval.

Related Work

In recent years, some of the research efforts in image retrieval community have been focusing on developing scalable algorithms for image retrieval. For example, in (Perronnin et al. 2010), Perronnin et.al. applied compressed Fisher kernel framework instead of the BoVW model to obtain better retrieval quality, and the compressed Fisher kernel framework was more efficient than the non-compressed version. In (Deng, Berg, and Fei-Fei 2011), Deng et.al. proposed a hierarchical semantic indexing to handle large-scale similarity learning for images. The proposed learning approach was fundamentally parallelizable and as a result scales more easily than previous work, as stated in their paper. These previous work focus on designing new algorithms to improve retrieval quality without spending too much time for the retrieval process, while what we did was to scale up an existing mature BoVW model.

A few projects have used Hadoop as a distributed platform to process image search in parallel. Hadoop was used to parallelize feature extraction, indexing and searching in (Gu and Gao 2012) by Gu and Gao. In (Yin and Liu 2013), Yin and Liu first built a database of image features using SURF (Speeded Up Robust Features) algorithm and LSH (Locality-Sensitive Hashing) and then performed the search on Hadoop in a parallel way. In (Premchaiswadi et al. 2013), Premchaiswadi et.al. proposed a similarity metric between images and performed parallel similarity computation between the query image and existing images using Map-Reduce on Hadoop. Grace et.al. (Grace, Manimegalai, and Kumar 2014) employed Hadoop Map-Reduce to extract features, compute similarity scores and rank the images based on similarity scores on medical datasets. Most of the related work listed above employed Hadoop Map-Reduce to parallelize the search process of finding similar images, while in our projects we used Hadoop as the platform to accelerate the building process of the BoVW model.

In multimedia analysis community (Atrey et al. 2010), people developed two kinds of multimodal fusion ap-

proaches, early fusion and late fusion, for multimedia tasks such as event detection. Early fusion models develop unified features for multimodal data and late fusion models fuse the results at the decision level of each modality. In information retrieval, Rasiwasia et.al. (Rasiwasia et al. 2010) proposed several state-of-the-art approaches to achieve cross-modal information retrieval. The first approach was correlation matching, which aimed to map the different feature spaces for images and text to the same feature space based on correlation analysis of these two spaces. The second approach was semantic matching, which represented images and text with the same semantic concepts using multi-class logistic regression. Their approaches can be categorized as early fusion, while our linear rule fusion model is one of the late fusion models. The experiments in our project reveal certain improvement in the retrieval quality, demonstrating late fusion can be very useful in multimodal information retrieval.

Conclusion

We have introduced our scalable image retrieval system and the multimodal fusion model. Our main contributions focus on two aspects. First, we have designed and implemented two distributed clustering algorithms on Hadoop, which can run much faster than Mahout K-Means, to scale up the BoVW building process. Second, we have designed a multimodal ensemble fusion model to combine image-only retrieval and text-only retrieval to achieve higher retrieval quality.

Our next steps would have two directions: (i) using more sophisticated computer vision models and machine learning techniques to improve retrieval quality, for example deep learning; (ii) constructing a large-scale multimodal dataset for multimodal retrieval and employing sophisticated algorithms to combine the image-only and text-only retrieval results.

Acknowledgements

This work was partially supported by DARPA under FA8750-12-2-0348. We also thank Pivotal for providing funding and Pivotal Analytics Workbench to run experiments.

References

Apache Hadoop. `https://hadoop.apache.org/`. Accessed: 2016-02-11.

Apache Lucene. `http://lucene.apache.org/`. Accessed: 2016-02-11.

Apache Mahout. `https://mahout.apache.org/`. Accessed: 2016-02-11.

Apache Solr. `http://lucene.apache.org/solr/`. Accessed: 2016-02-11.

Atrey, P.; Hossain, M.; Saddik, A.; and Kankanhalli, M. 2010. Multimodal fusion for multimedia analysis: A survey. In *Multimedia Systems*.

Deng, J.; Berg, A.; and Fei-Fei, L. 2011. Hierarchical semantic indexing for large scale image retrieval. In *Computer Vision and Pattern Recognition*.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.and Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *CVPR09*.

Grace, . K.; Manimegalai, R.; and Kumar, S. S. 2014. Medical image retrieval system in grid using hadoop framework. In *International Conference on CSCI*.

Gu, C., and Gao, Y. 2012. A content-based image retrieval system based on hadoop and lucene. In *Second International Conference on Cloud and Green Computing*.

Li, F.-F., and Perona, P. 2005. A bayesian hierarchical model for learning natural scene categories. In *Computer Vision and Pattern Recognition*.

Lowe, D., and David, G. 1999. Object recognition from local scale-invariant features. In *International Conference on Computer Vision*.

Mathias, L., and Chatzichristofis, S. A. 2008. Lire: Lucene image retrieval an extensible java cbir library. In *16th ACM International Conference on Multimedia*.

Muja, M., and Lowe, D. 2009. Fast approximate nearest neighbors with automatic algorithm configuration. In *VISAPP*.

Muja, M., and Lowe, D. G. 2014. Scalable nearest neighbor algorithms for high dimensional data. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Peng, Y., and Zhou, X. ScaDIR: Scalable image retrieval with multimodal fusion. `https://github.com/SunPHM/ScaDIR`.

Perronnin, F.; Yan, L.; Sanchez, J.; and Poirier, H. 2010. Large-scale image retrieval with compressed fisher vectors. In *Computer Vision and Pattern Recognition*.

Philbin, J.; Chum, O.; Isard, M.; Sivic, J.; and Zisserman, A. 2007. Object retrieval with large vocabularies and fast spatial matching. In *Computer Vision and Pattern Recognition*.

Premchaiswadi, W.; Tugnkatsathan, A.; Intarasema, S.; and etc. 2013. Improving performance of content-based image retrieval schemes using hadoop mapreduce. In *International Conference on High Performance Computing and Simulation*.

Rasiwasia, N.; Pereira, J.; Coviello, E.; and etc. 2010. A new approach to cross-modal multimedia retrieval. In *The International Conference on Multimedia*.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; and etc. 2015. Imagenet large scale visual recognition challenge. In *IJCV*.

Sipla-Anan, C., and Hartley, R. 2008. Optimised kd-trees for fast image descriptor matching. In *Computer Vision and Pattern Recognition*.

Sivic, J., and Zisserman, A. 2003. Video google: A text retrieval approach to object matching in videos. In *IEEE 9th International Conference on Computer Vision (ICCV)*.

Yin, D., and Liu, D. 2013. Content-based image retrial based on hadoop. In *Mathematical Problems in Engineering*.