Histopathological Lung Cancer Image Mining with Kd-tree and MapReduce

Fuyong Xing[1,2], Manish Sapkota[1,2], Lin Yang[2]

[1] Dept. of Electrical and Computer Engineering,
[2] Dept. of Biomedical Engineering, University of Florida, FL 32611, USA

Abstract. Automatic computer-aided image analysis can significantly improve the objectivity and reproducibility of evaluation on histopathological lung cancer specimens, and the content-based image retrieval (CBIR) methods can assist doctors for comparative search of similar cases from the database and hence provide decision support for diagnosis and disease detection. However, it is challenging to efficiently perform CBIR due to the increasing data size. In this paper, we present a distributed content-based image retrieval framework for nearest neighbor searching on lung cancer dataset. Given a query image, we first extract the features with an efficient and effective locality-constrained linear coding algorithm. Next, the parallel CBIR is conducted by searching the nearest neighbors in the image signature space with Kd-tree in a MapReduce framework. We have demonstrated the significant speed improvement with 200 queries conducted in a database with approximate 1000 images in comparison with running a single machine.

1 Introduction

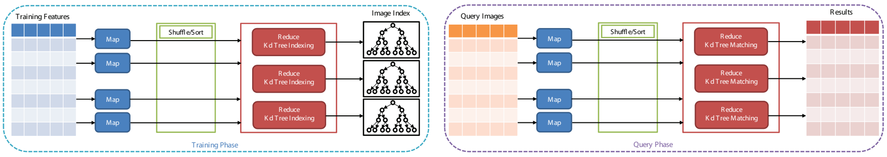
Lung cancer is one of the most frequent cancers, and is the most common cause of death from cancer worldwide, with 70% to 80% of the cases presenting with locally advanced or metastatic disease [2]. Like breast cancer in females, lung cancer is the leading cancer site in males, comprising 17% of the total new cancer cases and 23% of the total cancer deaths [13]. The prognosis of lung cancer is still poor, with five-year survival rates of approximately 10% in most countries. Lung cancer can be classified as small cell lung cancer (SCLC, 10% - 15%) and non-small cell lung cancer (NSCLC, 85% - 90%). Two major types of NSCLC are adenocarcinoma (including bronchioalveolar carcinoma) representing about 40% and squamous cell carcinoma representing about 25-30% [2].

Accurate lung cancer diagnosis is critical to personalize lung cancer treatment. The accurate staging and subtyping can help clinicians to determine personalized patient treatment, allow for reasonable prognostication, and facilitate comparisons between patient groups in clinical studies. For example, current investigations into early detection and adjuvant chemotherapy heavily rely on the proper staging of patients' cancer types. Automatic computer-aid image evaluation of lung cancer biopsies is essential to clinical practice, since it can provide decision support for diagnosis and disease detection, and significantly increase the survival rates of the patients. Meanwhile, it can greatly improve the objectivity and reproducibility of the evaluation. Given a new image, it will be very helpful for doctors to quickly and reliably search and retrieve previous cases exhibiting similar contents with comparable morphometric measures such that they can provide personalized diagnosis at the bedside. Moreover, the major subtypes of diseases such as adenocarcinoma and squamous cell carcinoma can be easily classified by applying majority voting to the retrieved results. This content-based image retrieval (CBIR) technique [18] has attracted a great deal of research interest and achieved great success in medical applications.

In order to effectively describe images for image retrieval, Zheng *et al.* [33] have extracted and combined four types of image features including color histogram, image texture, Fourier coefficients, and wavelet coefficients from networked microscopic pathology images. Based on the support vector machine (SVM) classifier with a combination of multiple types of image features, Rahman *et al.* [19] have refined the retrieval by filtering the irrelevant images; meanwhile, online relevance feedback is introduced to improve the performance. However, the concatenation of multiple types of image features might result in high-dimensional feature vectors, which can decrease run-time computational efficiency. In order to reduce the dimensionality, feature coding [12] is applied to feature descriptors. Based on the bag-of-features (BoF) framework [5], Foncubierta-Rodriguez *et al.* [8,9] have successfully applied voting-based feature coding to medical image retrieval. In [10], a texton histogram is generated using voting-based coding for image signature and applied to image retrieval on tissue microarray specimens.

Due to the gradually increasing patient data, scalable and real-time (sub)image retrieval techniques have recently been proposed. In [23], images are first classified into predefined classes such that region of interest (ROI) correspondences are only considered between images of the same class. Next, the algorithm computes approximate global registration for images within that class. Finally, the ROI is online refined in a target image using the previous registration as a guide. This framework is generic and can be extended to other modalities. A similar strategy with a combination of offline registration and online searching is presented in [22] for ROI searching on 3D medical images. Yang *et al.* [29] have proposed a parallel CBIR system for prostate cancer images, which performs a hierarchical searching with a newly developed annular color histogram on a demand-driven master-worker parallelization platform. Recently, Hashing methods [4,27], which can compress high-dimensional image descriptors into tens of bits for quick searching, have invoked much research interest. A supervised hashing algorithm with kernels [31] is applied to breast cancer image retrieval, whilst anchor graph hashing is reported in [16] for scalable mammogram retrieval.

In this paper, we present a distributed CBIR system in a MapReduce framework for histopathological lung cancer images, as shown in Figure 1. Instead of using voting-based feature coding, we employ a sparse representation-based coding method for image signature, which is fast and effective. Thereafter, we perform nearest neighbor searching using the independent Kd-tree parallel imple-



(a)

(b)

Fig. 1. (a) Training phase: Construct independent Kd-tree indexing for the training signatures distributed in all the machines in the cluster. (b) Query phase: Multiple queries on Kd-tree residing in different machines to get the nearest matches.

mentation. The framework is implemented with Hadoop to enable high throughput content-based image queries in a large-scale lung cancer image dataset. The MapReduce, which is a parallel programming platform, is exploited to distribute the image dataset onto a large number of machines for scalable image retrieval. In the experiments, we have built a distributed independent Kd-tree from the same database and applied hundreds of image queries to the the parallel implementation framework.

2 Methodology

2.1 Sparse Reconstruction-based Feature Coding

Although scale-invariant feature transform (SIFT) [17] is widely used as image signatures, it has been proved that SIFT-based feature coding is more robust and achieves better representation [12]. Many feature coding approaches have been proposed, and hard voting-based coding is very popular in the BoF framework [5], but it simply assigns each feature to its closet codeword such that it may be insufficient to represent images with large variations on appearance. Sparse reconstruction-based coding methods have been recently reported to enhance the hard coding. It uses a linear combination of codewords to approximate features with smaller description error, and the sparsity constrain can improve both the expressive and discriminative powers of the image descriptors. In this paper, we represent an efficient and effective sparse feature coding approach for image description.

Given the SIFT descriptors, we first use K-means to generate the codewords $D = [d_1,\ ...,\ d_K] \in R^{P \times K}$ as the reference dictionary. For a feature descriptor $y \in R^{P \times 1}$, sparse reconstruction-based feature coding attempts to represent y with a linear combination of a few codewords with a sparse constraint:

$$\min_{\alpha} ||y - D\alpha||_2^2 + \lambda\phi(\alpha),\ s.t.\ 1^T\alpha = 1, \tag{1}$$

where α denotes the sparse codes, $\phi(\alpha)$ represents the sparse constraint, λ controls its sparsity, and the affine constraint $1^T\alpha = 1$ ensures the shift invariance.

Usually $\phi(\alpha)$ is chosen to be the l_0 or l_1 norm, and the corresponding optimization problems can be solved with orthogonal matching pursuit [25] or LARS [7]. However, these algorithms do not guarantee the local smooth sparsity and may ignore the correlations between codes [30]. Fortunately, a locality-constrained linear coding (LLC) algorithm [26] has been proposed to ensure that similar feature descriptor will be represented with similar codes:

$$\min_{\alpha} ||y - D\alpha||_2^2 + \lambda||b \odot \alpha||_2^2, \ s.t.\ 1^T \alpha = 1, \tag{2}$$

where $\odot$ denotes the element-wise multiplication and $b \in R^{K \times 1}$ represents the distance between the descriptor y and the codewords $\{d_i\}$. For the feature descriptor x, LLC actually prefers using a few of y's neighboring codewords for optimization to guarantee the local smooth sparsity for better representation. Therefore, we can simply select the M $(M < K)$ nearest neighbors of y to form a local dictionary $\tilde{D}$ and solve a much smaller least square-style problem to obtain approximate codes $\tilde{\alpha}$:

$$\min_{\tilde{\alpha}} ||y - \tilde{D}\tilde{\alpha}||_2^2, \ s.t.\ 1^T \tilde{\alpha} = 1. \tag{3}$$

Compared with traditional sparse coding methods, solving (3) is much faster such that the run-time cost will be significantly improved. In order to enhance the robustness of sparse codes, we apply spatial pyramid matching (SPM) [14] and max-pooling [28] to the codes for final image signature generation.

2.2 Nearest Neighbor Searching with Kd-tree

Kd-tree is one type of binary trees designed to support range and nearest-neighbor queries [3,11]. In our implementation of the Kd-tree with dimension W, each node represents one W-dimensional feature point (SPM-based sparse code). The non-leaf nodes are divided into two half-spaces by a splitting hyperplane which is perpendicular to this dimension axis of the node [3,11]. The left child has smaller coordinate of this dimension than the corresponding value of the tree root, whist the right child has larger coordinate of the dimension. In next level, the nodes are split based on another dimension. The first dimension will be revisited when all dimensions are exhausted, and this procedure is repeated until the whole Kd-tree with depth $h = \log_2 N$ (N is the number of images in the database) is built. In this paper, we implement independent Kd-trees (IKdt) for parallel computing. IKdt is a baseline brute force approach [1]. We divide the image dataset into multiple subsets, each corresponding to one machine which builds its own independent Kd-tree. A single root machine accepts a query image and send it to all the machines which will traverse their own IKdt. Finally, the root machine collects and ranks the returned images for output [1]. The framework of Kd-tree construction is shown in Figure 1 (a), and the detailed description is given in Algorithm 1.

Given a query point $x \in R^{W \times 1}$, the nearest neighbors are sought using the branch-and-bound search method. It traverses from the left or right side of the

Algorithm 1: Kd Tree Construction
Input: A set of points $X = \{x_1, x_2, ..., x_N\} \in R^{W \times N}$ and depth h.
Output: Balanced Kd Tree.

```
Choose split $s_r$ axis defined by $h$ mod $k$(To make sure axis cycles through all valid values).
Compute median $m$ by $s_r$ from $\{x_i\}$
$node \leftarrow$ create internal node storing $d_r$ and median $m$.
$left[node] \leftarrow$ recursively build tree for $x[s_r] \leq m$ and $depth + 1$.
$right[node] \leftarrow$ recursively build tree for $x[s_r] > m$ and $depth + 1$.
return $node$.
```

tree depending on whether the query point is less or greater than the current node in the split dimension. For backtracking, a test is made to determine if it is necessary to consider points on the other side of the splitting plane that are closer to the search point than the current best candidate. This is done by checking if the splitting hyperplanes with radius equal to the current nearest distance, and the search point as center intersect, similar to bounds-overlap-ball defined in [11]. The searching algorithm requires only h scalar comparison and will terminate once all unexplored branches in the tree are traversed. The details of Kd-tree nearest neighbor searching is listed in Algorithm 2.

Algorithm 2: Kd-Tree Search
Input: Kd-Tree T, set of points $X \in R^{W \times N}$ used to build the tree, the query vector $Q = \{q_1, ..., q_N\}$.
Output: A set of K nearest neighbors $\{n_k\}$ and their distances $\{s_k\}$ to query vector.

```
$node \leftarrow$ root node. Initialize the priority queue $P$ to store a $list$ of the $K$ closest nodes that have been visited and their distances to the query.
Recursively traverse from root to node of the Kd-Tree $T$.
If $node$ is leaf node
    Store $node$ as current best.
    Update $list$.
end.
While $node \neq root$
    $node \leftarrow$ Unwind recursion of the $T$.
    if $node$ is closer than current best
     Update $list$.
    end.
end.
Find the $K$ nearest neighbors to $Q$ in $list$ and return the sorted list $\{n_k\}$ and their distances $\{s_k\}$.
```

2.3 Multiple Query MapReduce Kd-tree Retrieval Algorithm

In order to accommodate the high throughput analysis query requirements, we have implemented a distributed image retrieval method based on Kd-Tree using MapReduce. The flowchart is shown in Figure 1 (b). MapReduce is a programming framework to support parallel computing for large-scale datasets on computer clusters [6], which has been successfully applied to many real-world applications [15,20,21,24,32] including text processing, bioinformatics/medical informatics, image processing, data mining, machine learning, etc. Using two major functions Map and Reduce, this programming model takes a set of input key/value pairs, and produces a set of output key/value pairs. Map function processes on the input key/value pairs to generate a set of intermediate key/value pairs, and Reduce function merges all the intermediate values associated with the same intermediate key. Programs written in this functional style are automatically distributed and executed on a large cluster of machines. MapReduce is a simple, reliable and scalable software framework, and can automatically handle data partition, inter-machine communications, program execution across multiple machines, machine failure, etc [6].

Based on MapReduce architecture, we have implemented the independent Kd-tree (IKdt). At the training phase, the MapReduce directs the training image signatures into different machines to build their Kd-trees, see Figure 1 (a). At query phase, MapReduce distributes the query image signatures to all the machines, and the performs searching, counting, and sorting, see Figure 1 (b). The Map and Reduce functions during the query phase is outlined in Algorithm 3. Map function distributes the query images according to the image name. Reduce function loads the Kd-Tree from HDFS to perform the search against the query, and outputs the ranked list of search results.

Algorithm 3: The parallel content based image retrieval using MapReduce.
Input: A query image Q=$\{q_1, ..., q_n\}$.
Output: A set of indices of top K retrieved images $I = \{i_1, ..., i_K\}$.

```
Function Map (key, val)
    $k \leftarrow val.imagename$.
    $v \leftarrow val.imagename$.
    Emit($k$, $v$)
Function Reduce (key, val)
    $T \leftarrow$ Load Kd-Tree.
    $kindices \leftarrow$ The indices of the K nearest neighbors NN($Q$,$K$) using $T$.
    Emit($key$, $kindices$).
```

3 Experiments

The parallel implementation of the proposed content based image retrieval framework is tested using a lung cancer image database that contains 1000 images, which are cropped from over 50 whole-slide scan lung cancer specimens with 40× magnification. The dataset consists of 500 adenocarcinoma and 500 squamous cell carcinoma images. We densely extract the 128-dimensional SIFT feature using a 16 × 16 window on a grid with a step of 4 pixels. K-means is applied to generate 1024 codewords. We perform spatial pyramid matching with three levels $[1 \times 1, 2 \times 2, 4 \times 4]$, and max-pooling with local size 5. The dimensionality of the final image signatures is 21054. In order to test the MapReduce functionality, a local cluster which is composed of four machines is configured. Hadoop 0.20.2 has been installed in all these machines for the Master/Slave and Slave machines. Each machine has an Inter (R) Core (TM) i3-2410M CPU @2.30GHz and 16.00GB RAM.

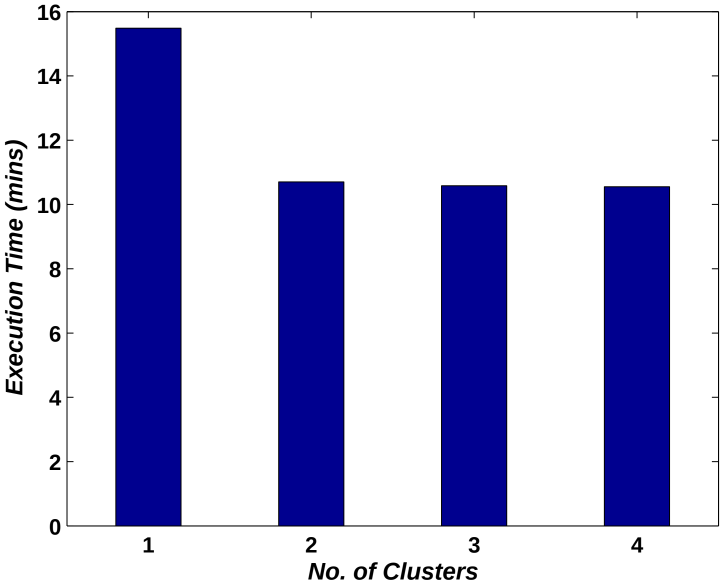


Fig. 2. The execution times in minutes of the distributed image retrieval system in a database containing 1000 images and 200 query images. Figure shows the execution time plotted against the number of clusters.

Figure 2 shows the execution time of the multiple queries for content retrieval image retrieval with various number of nodes. In total 200 queries are conducted and the execution time using a cluster with 1 to 4 computational nodes is recorded. As we illustrate in Figure 2, the execution time decreases as we increase the number of computational nodes. There is a significant decrease in execution time as we increase the cluster size from 1 to 2. However, when we increase the number of computational nodes from 2 to 4 there is no significant speed-up. This can be explained mainly because of the communication overhead between the master node and the slaves. These results demonstrate that we can employ MapReduce framework to efficiently accommodate the high throughput content based image queries in a large image dataset.

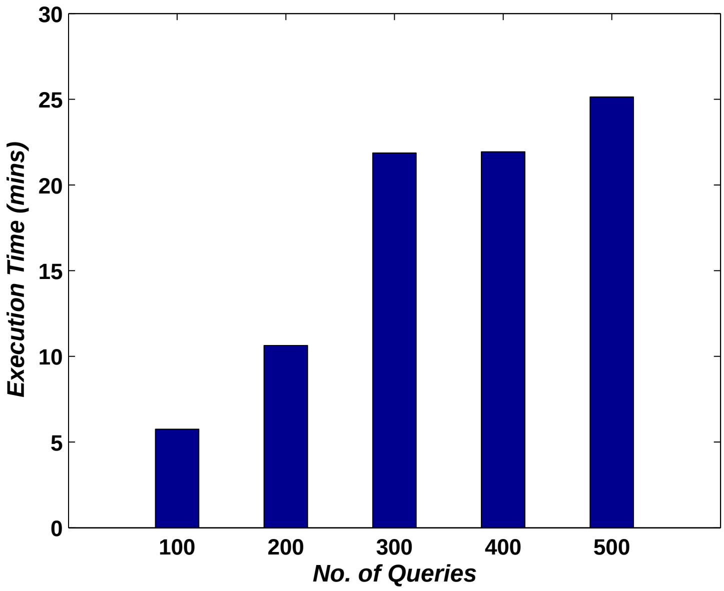


Fig. 3. The execution times in minutes of the distributed image retrieval system in a database containing 1000 images for different numbers of queries. The experiment concludes the scalability of the proposed framework.

Figure 3 shows the execution time of the range of multiple queries from 100 to 500. For this experiment we used all of the 4 computational nodes in the cluster. As we illustrate in Figure 3, the execution time increases significantly for query size from 100 to 300. For query size of 300 to 500, there is no significant increase in the execution time. These results demonstrate that the distributed retrieval setup is well suited for large scale retrieval task. Scalability of proposed method is only limited by the hardware configuration. We display the corresponding retrieved images given one adenocarcinoma query and one squamous cell carcinoma query in Figure 4, which demonstrate that the system performance is fairly good with respect to visual similarity.

4 Conclusion

In this paper, we have implemented an independent Kd-Tree distributed in different machines using the MapReduce framework for content based lung cancer image retrieval. We present an efficient and robust feature coding approach to generate image signatures for content-based image retrieval. In the future, we plan to apply the proposed framework to a cluster with more computational nodes in a larger database.

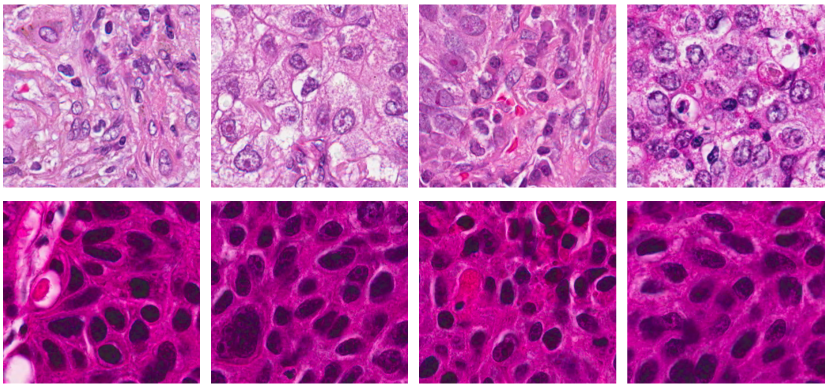


Fig. 4. Sample retrieved images for adenocarcinoma (row 1) and squamous cell carcinoma (row 2). Column 1 denote the query images, and columns 2,3,4 represent the corresponding returned ranked images.

References

1. Aly, M., Munich, M., Perona, P.: Distributed kd-trees for ultra large scale object recognition. In: Proc. BMVC. pp. 40.1–40.11 (2011)
2. Anagnostou, V.K., Dimou, A.T., Botsis, T., Killiam, E.J., Gustavson, M.D., Homer, R.J., Boffa, D., Zolota, V., Dougenis, D., Tanoue, L., Gettinger, S.N., Detterbeck, F.C., Syrigos, K.N., Bepler, G., Rimm, D.L.: Molecular classification of nonsmall cell lung cancer using a 4-protein quantitative assay. Cancer 118(6), 1607–1618 (2012)
3. Bentley, J.: Multidimensional binary search trees used for associative searching. the Communications of the ACM 18(9), 509517 (1975)
4. Charikar, M.: Similarity estimation techniques from rounding algorithms. In: ACM STOC. pp. 380–388 (2002)
5. Csurka, G., Bray, C., Dance, C., Fan, L.: Visual categorization with bags of keypoints. In: Proc. ECCV Intl Workshop Statistical Learning in Computer Vision (2004)
6. Dean, J., Ghemawat, S.: Mapreduce: simplified data processing on large clusters. the Communications of the ACM 51(1), 107113 (2008)
7. Efron, B., Hastie, T., Johnstone, I., Tibshirani, R.: Least angle regression. The Annals of Statistics 32(2), 407499 (2004)
8. Foncubierta-Rodriguez, A., Depeursinge, A., Muller, H.: Using multiscale visual words for lung texture classification and retrieval. In: Medical Content-Based Retrieval for Clinical Decision Support. vol. 7075, pp. 69–79 (2012)
9. Foncubierta-Rodriguez, A., Vargas, A., Platon, A., Poletti, P., Muller, H., Depeursinge, A.: Retrieval of 4D dual energy CT for pulmonary embolism diagnosis. In: Medical Content-Based Retrieval for Clinical Decision Support. vol. 7723, pp. 45–55 (2013)
10. Foran, D.J., Yang, L., Chen, W., Hu, J., Goodell, L.A., Reiss, M., Wang, F., Kurc, T., Pan, T., Sharma, A., Saltz, J.H.: Imageminer: a software system for comparative analysis of tissue microarrays using content-based image retrieval, high-performance computing, and grid technology. JAMIA 18(4), 403–415 (2011)

11. Friedman, J., Bentley, J., Finkel, R.: An algorithm for nding best matches in logarithmic expected time. ACM TMS 3(9), 209226 (1977)
12. Huang, Y., Wu, Z., Wang, L., Tan, T.: Feature coding in image classification: A comprehensive study. TPAMI 36(3), 493–506 (2014)
13. Jemal, A., Bray, F., Center, M.M., Ferlay, J., Ward, E., Forman, D.: Global cancer statistics. CA: Cancer J. Clin. 61(2), 69–90 (2011)
14. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: CVPR. pp. 2169–2178 (2006)
15. Lin, J., Dyer, C.: Data-intensive text processing with mapreduce. Synthesis Lectures on Human Language Technologies 3(1), 1–177 (2010)
16. Liu, J., Zhang, S., Liu, W., Zhang, X., Metaxas, D.: Scalable mammogram retrieval using anchor graph hashing. In: ISBI (2014)
17. Lowe, D.: Distinctive image features from scale-invariant key-points. IJCV 60(2), 91–110 (2004)
18. Muller, H., Michoux, N., Bandon, D., Geissbuhler, A.: A review of content-based image retrieval systems in medical applications-clinical benefits and future directions. IJMI 73(1), 123 (2004)
19. Rahman, M.M., Antani, S.K., Thoma, G.R.: A learning-based similarity fusion and filtering approach for biomedical image retrieval using SVM classification and relevance feedback. TITB 15(4), 640–646 (2011)
20. Sadasivam, G.S., Baktavatchalam, G.: A novel approach to multiple sequence alignment using hadoop data grids. In: Proceedings of the 2010 Workshop on Massive Data Analytics on the Cloud. p. 2 (2010)
21. Sandholm, T., Lai, K.: Mapreduce optimization using regulated dynamic prioritization. In: Proceedings of the Eleventh International Joint Conference on Measurement and Modeling of Computer Systems. pp. 299–310 (2009)
22. Simonyan, K., Modat, M., Ourselin, S., Cash, D., Criminisi, A., Zisserman, A.: Immediate roi search for 3-D medical images. In: Medical Content-Based Retrieval for Clinical Decision Support. vol. 7723, pp. 56–67 (2013)
23. Simonyan, K., Zisserman, A., Criminisi, A.: Immediate structured visual search for medical images. In: MICCAI. pp. 288–296 (2011)
24. Sweeney, C., Liu, L., Arietta, S., Lawrence, J.: Hipi: A hadoop image processing interface for image-based mapreduce tasks. Chris. University of Virginia (2011)
25. Tropp, J.: Greed is good: algorithmic results for sparse approximation. TIT 50(10), 2231–2242 (2004)
26. Wang, J., Yang, J., Yu, K., Lv, F., Huang, T., Gong, Y.: Locality-constrained linear coding for image classification. In: CVPR. pp. 3360–3367 (2010)
27. Weiss, W., Torralba, A., Fergus, R.: Spectral hashing. In: NIPS. pp. 1753–1760 (2008)
28. Yang, J., Yu, K., Gong, Y., Huang, T.: Linear spatial pyramid matching using sparse coding for image classification. In: CVPR. pp. 1794–1801 (2009)
29. Yang, L., Qi, X., Xing, F., Kurc, T., Saltz, J., Foran, D.J.: Parallel content-based sub-image retrieval using hierarchical searching. Bioinformatics 30(7), 996–1002 (2013)
30. Yu, K., Zhang, T., Gong, Y.: Nonlinear learning using local coordinate coding. In: NIPS. pp. 2223–2231 (2009)
31. Zhang, X., Liu, W., Zhang, S.: Mining histopathological images via hashing-based scalable image retrieval. In: ISBI (2014)
32. Zhao, W., Ma, H., He, Q.: Parallel k-means clustering based on mapreduce. In: Cloud Computing, pp. 674–679 (2009)

33. Zheng, L., Wetzel, A.W., Gilbertson, J., Becich, M.J.: Design and analysis of a content-based pathology image retrieval system. TITB 7(4), 245–255 (2003)