Joint Kernel-Based Supervised Hashing for Scalable Histopathological Image Analysis

Menglin Jiang[1], Shaoting Zhang[2,⋆], Junzhou Huang[3],
Lin Yang[4], and Dimitris N. Metaxas[1]

[1] Department of Computer Science, Rutgers University, Piscataway, NJ, USA
[2] Department of Computer Science, UNC Charlotte, Charlotte, NC, USA
[3] Department of Computer Science and Engineering, UT Arlington, Arlington, TX, USA
[4] Department of Biomedical Engineering, University of Florida, Gainesville, FL, USA

Abstract. Histopathology is crucial to diagnosis of cancer, yet its interpretation is tedious and challenging. To facilitate this procedure, content-based image retrieval methods have been developed as case-based reasoning tools. Recently, with the rapid growth of histopathological images, hashing-based retrieval approaches are gaining popularity due to their exceptional scalability. In this paper, we exploit a joint kernel-based supervised hashing (JKSH) framework for fusion of complementary features. Specifically, hashing functions are designed based on linearly combined kernel functions associated with individual features, and supervised information is incorporated to bridge the semantic gap between low-level features and high-level diagnosis. An alternating optimization method is utilized to learn the kernel combination and hashing functions. The obtained hashing functions compress high-dimensional features into tens of binary bits, enabling fast retrieval from a large database. Our approach is extensively validated on thousands of breast-tissue histopathological images by distinguishing between actionable and benign cases. It achieves 88.1% retrieval precision and 91.2% classification accuracy within 14.0 ms query time, comparing favorably with traditional methods.

1 Introduction

For years, histopathology has played a key role in the early diagnosis of breast cancer, which is the second leading cause of cancer-related death among women. Unfortunately, examination of histopathological images is very tedious and error-prone due to their large size, inter- and intra-observer variability among pathologists, and several other factors [11]. To facilitate this procedure, many content-based image retrieval (CBIR) methods have been proposed as computer-aided diagnosis (CAD) tools [1, 3, 13, 14]. These approaches compare a query histopathological image with previously diagnosed cases stored in a database, and return the most similar cases along with the likelihood of abnormality of the query. Compared with classifier-based CAD methods [2, 5], CBIR approaches could provide more clinical evidence to assist the diagnosis. In addition, they can also contribute to digital slide archiving, pathologist training, and various

⋆ Corresponding author.

N. Navab et al. (Eds.): MICCAI 2015, Part III, LNCS 9351, pp. 366–373, 2015.
DOI: 10.1007/978-3-319-24574-4_44

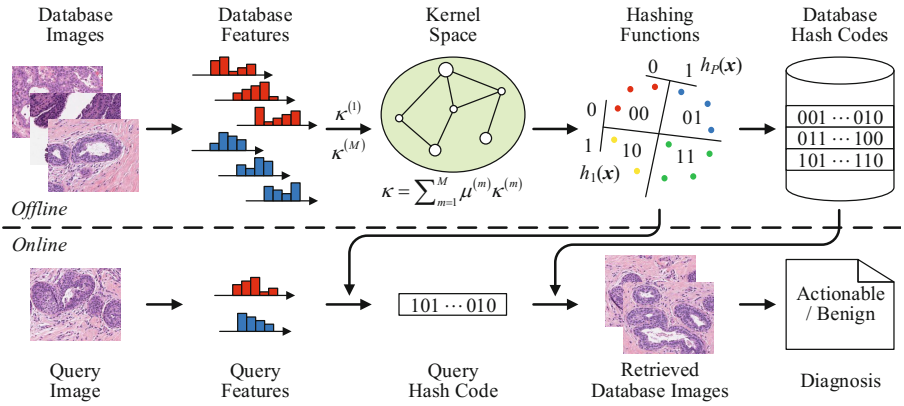

Fig. 1. Overview of the proposed approach.

other applications. Especially, along with the dramatic increase in digital histopathology, hashing-based retrieval methods are drawing more and more attention because of their remarkable computational efficiency and excellent scalability [13, 14].

In the image retrieval community, it is a common practice to employ multiple features to improve performance. Nevertheless, few hashing-based methods have put this principle into practice. A pioneer work [14] adopts the strategy of affinity aggregation. In particular, several affinity matrices calculated using individual features are averaged, and traditional hashing methods are applied to the combined matrix. However, this approach is not suitable for those features that need different kernel functions during the hashing process. Besides, it introduces extra parameters (i.e. the weights of all the matrices) which need to be elaborately tuned. Other widely used feature fusion methods in medical image retrieval include feature concatenation [1] and result-level fusion [12]. The former approach simply concatenates several features to form a new one. Similar to affinity aggregation, it is not appropriate for intrinsically different features even with feature normalization, because various features may need different similarity measures during subsequent feature matching. The latter method first conducts similarity search using individual features, and then integrates their results. Obviously, this approach compromises the computational efficiency, since its processing time will be at least the sum of time required by each feature.

To overcome the above drawbacks, we employ joint kernel-based supervised hashing (JKSH) [6,8,9] to incorporate feature fusion into the supervised hashing framework, and apply it to scalable analysis of histopathological images [13, 14]. The overview of our approach is shown in Fig. 1. Specifically, a joint kernel function is defined as a linear combination of the kernels for individual features, and a series of hashing functions are constructed based on this kernel. Diagnostic information of histopathological images in the database is utilized to learn the weights of individual kernels and the hash functions, which bridges the semantic gap between low-level features and high-level diagnosis. With the learned hashing functions, high-dimensional features are compressed into tens of binary hash bits, enabling efficient search from a large-scale database. At last, a query

image is classified (as actionable or benign) according to a weighted majority vote of its retrieved database images [7].

Our approach has many advantages over current methods. First, it could adopt multiple kernel functions as similarity measures for various features, rather than employ the same kernel. Second, the parameter tuning issue is solved, as all the important parameters are automatically learned. Finally, utilizing the kernel representation, compressing multiple features into hash code has little computational overhead than using a single feature.

2 Methodology

In this section, we first formulate the multiple-feature hashing problem as a linear combination of individual kernels, and then explain how to simultaneously learn the kernel weights and hashing functions. After obtaining the hashing functions, features extracted from the database histopathological images can be mapped to compact hash codes and stored in a hash table. Given a query histopathological image, its hash code is computed using the same hashing functions and searched from the hash table to find the most similar database images, which then vote to determine its diagnosis [7].

Joint Kernel-Based Hashing: Suppose we extract M features from N histopathological images. Denote $\boldsymbol{x}_n^{(m)} \in \mathbb{R}^{d^{(m)}}$ as the m-th feature of the n-th image, which is a $d^{(m)}$-dimensional column vector. Then $\boldsymbol{x}_n = \left[\left(\boldsymbol{x}_n^{(1)}\right)^T, \cdots, \left(\boldsymbol{x}_n^{(M)}\right)^T\right]^T \in \mathbb{R}^d$ is the concatenation of all features extracted from the n-th image, where $d = \sum_{m=1}^{M} d^{(m)}$. A hashing method aims at finding P hashing functions $\{h_1, \cdots, h_P\}$, where P is the desired number of hash bits. Each hashing function, $h_p : \mathbb{R}^d \mapsto \{-1, 1\}$, maps a concatenated feature vector into a binary bit. The n-th image is represented as $\boldsymbol{y}_n = [h_1(\boldsymbol{x}_n), \cdots, h_P(\boldsymbol{x}_n)]^T$.

When designing hashing functions, a classic idea is to preserve "local sensitivity", i.e., similar feature vectors are compressed into similar hash codes. Unfortunately, sometimes it is difficult to distinguish between the original features. To solve this problem, kernel functions are introduced to operate the data in an *implicit* higher-dimensional feature space [6]. Given M features, we can choose M kernel functions $\left\{\kappa^{(1)}, \cdots, \kappa^{(M)}\right\}$, where each kernel $\kappa^{(m)}$ is associated with an implicit feature mapping function $\varphi^{(m)}$, i.e. $\kappa^{(m)}\left(\boldsymbol{x}_i^{(m)}, \boldsymbol{x}_j^{(m)}\right) = \varphi^{(m)}\left(\boldsymbol{x}_i^{(m)}\right)^T \varphi^{(m)}\left(\boldsymbol{x}_j^{(m)}\right)$. Without ever computing the mapped features $\varphi^{(m)}\left(\boldsymbol{x}_i^{(m)}\right)$ and $\varphi^{(m)}\left(\boldsymbol{x}_j^{(m)}\right)$, $\kappa^{(m)}$ directly calculates their inner product. Such "kernel trick" improves computational efficiency dramatically. Following [3], the joint mapping function φ and corresponding kernel κ are defined as:

$$\varphi(\boldsymbol{x}_n) = \left[\sqrt{\mu^{(1)}}\varphi^{(1)}\left(\boldsymbol{x}_n^{(1)}\right)^T, \cdots, \sqrt{\mu^{(M)}}\varphi^{(M)}\left(\boldsymbol{x}_n^{(M)}\right)^T\right]^T, \tag{1}$$

$$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \varphi(\boldsymbol{x}_i)^T \varphi(\boldsymbol{x}_j) = \sum\nolimits_{m=1}^{M} \mu^{(m)} \kappa^{(m)}\left(\boldsymbol{x}_i^{(m)}, \boldsymbol{x}_j^{(m)}\right), \tag{2}$$

where $\mu^{(m)}$ is the weight for the m-th feature. Later we will show how to automatically learn the weight vector $\boldsymbol{\mu} = \left[\mu^{(1)}, \cdots, \mu^{(M)}\right]^T$. Eq. (2) demonstrates that $\kappa = \sum_{m=1}^{M} \mu^{(m)} \kappa^{(m)}$ is actually a linear combination of individual kernels for each feature.

To reduce computational complexity, R $(R \ll N)$ landmark points, denoted as $\{\boldsymbol{z}_1, \cdots, \boldsymbol{z}_R\}$, are randomly selected from all the database feature vectors $\{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N\}$. Then, for the p-th hashing function h_p, its hyperplane vector $\boldsymbol{v}_p$ in the kernel space is represented as a linear combination of projections of landmarks in that space:

$$\boldsymbol{v}_p = \sum\nolimits_{r=1}^{R} W(r,p)\, \varphi(\boldsymbol{z}_r),\ p = 1, \cdots, P, \tag{3}$$

where W is a $R \times P$-dimensional matrix, its element $W(r,p)$ denotes the weight of $\boldsymbol{z}_r$ for $\boldsymbol{v}_p$. $h_p(\boldsymbol{x}_n)$ is defined based on the projection of $\varphi(\boldsymbol{x}_n)$ on $\boldsymbol{v}_p$:

$$h_p(\boldsymbol{x}_n) = \operatorname{sgn}\left(\boldsymbol{v}_p^T \varphi(\boldsymbol{x}_n) + b_p\right),\ p = 1, \cdots, P, \tag{4}$$

where b_p is the threshold parameter. Denote $\boldsymbol{b} = [b_1, \cdots, b_P]^T$ as the threshold vector, $K_{R\times N} = [\kappa(\boldsymbol{z}_r, \boldsymbol{x}_n)]_{R\times N}$ as the kernel matrix between R landmarks and N database features, and $K_{R\times N}(:,n)$ as the n-th column of K. Utilizing the fact $\kappa = \sum_{m=1}^{M} \mu^{(m)} \kappa^{(m)}$, we can represent the hash code of the n-th image in a kernel form:

$$\boldsymbol{y}_n = \operatorname{sgn}\left(W^T K_{R\times N}(:,n) + \boldsymbol{b}\right),\ n = 1, \cdots, N. \tag{5}$$

W and $\boldsymbol{b}$ determine the hashing functions, and they are learned using supervised information along with $\boldsymbol{\mu}$.

Supervised Optimization: In the image retrieval field, "semantic gap", which refers to the difference between low-level features and high-level concepts, is a long-standing problem. Supervised methods, such as kernel-based supervised hashing (KSH) [8], offer a promise to address this issue. Developed from the idea of "local sensitivity", supervised hashing approaches map semantically similar images to similar hash codes. To this end, we incorporate diagnostic information into affinity matrix S. $S(i,j)$, representing the similarity score between the i-th and the j-th images, is defined as:

$$S(i,j) = \begin{cases} \exp\left(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{\sigma^2}\right), & \text{if } \boldsymbol{x}_j \text{ is among the } k \text{ nearest neighbors} \\ & \text{of } \boldsymbol{x}_i \text{ with the same label} \\ 0, & \text{otherwise} \end{cases}, \tag{6}$$

where σ is a scaling parameter estimated from the data. Note that S is a sparse matrix, i.e., most of its elements are 0.

The objective function of the proposed JKSH is formulated as:

$$\begin{aligned} &\min_{W,\boldsymbol{b},\boldsymbol{\mu}} \ \tfrac{1}{2}\sum\nolimits_{i,j=1}^{N} S(i,j)\left\|\boldsymbol{y}_i - \boldsymbol{y}_j\right\|^2 + \lambda\|V\|_F^2 = \operatorname{Tr}\left(YLY^T\right) + \lambda\|V\|_F^2 \\ &\text{s.t. } \sum\nolimits_{n=1}^{N} \boldsymbol{y}_n = \boldsymbol{0},\ \tfrac{1}{N}\sum\nolimits_{n=1}^{N} \boldsymbol{y}_n \boldsymbol{y}_n^T = I,\ \mathbf{1}^T\boldsymbol{\mu} = 1,\ \boldsymbol{\mu} \succcurlyeq \boldsymbol{0}. \end{aligned} \tag{7}$$

Here $\sum_{i,j=1}^{N} S(i,j)\left\|\boldsymbol{y}_i - \boldsymbol{y}_j\right\|^2$ guarantees that histopathological images with the same label and similar features are compressed into similar hash codes, $V = [\boldsymbol{v}_1, \cdots, \boldsymbol{v}_P]$

is the hyperplane matrix, $\|V\|_F^2$ is a regularized term used to control the smoothness of hashing functions, $Y = [\boldsymbol{y}_1, \cdots, \boldsymbol{y}_N]$ includes all the database hash codes, $L = \mathrm{diag}(S\mathbf{1}) - S$ is the graph Laplacian matrix, the constraints ensure that the generated hash codes are balanced and uncorrelated. To solve this NP-hard problem, we employ spectral relaxation, which ignores the discrete constraint for $\boldsymbol{y}_n$ in Eq. (5) and allows $\boldsymbol{y}_n = W^T K_{R\times N}(:, n) + \boldsymbol{b} \in \mathbb{R}^P$.

The above problem, when either $(W, \boldsymbol{b})$ or $\boldsymbol{\mu}$ is fixed, is convex with respect to the other. Therefore we perform an alternating optimization algorithm, which mainly consists of the following two steps.

Step 1. Optimize $(W, \boldsymbol{b})$ ***for given*** $\boldsymbol{\mu}$**.** Similar to [6], we can find the optimal W by solving the following problem using eigen-decomposition:

$$\min_{W} \ \mathrm{Tr}\left(W^T C W\right) \ \text{s.t.}\ W^T G W = I\,, \tag{8}$$

where $C = K_{R\times N} L K_{R\times N}^T + \lambda K_{R\times R}$, $K_{R\times R} = \left[\kappa(\boldsymbol{z}_i, \boldsymbol{z}_j)\right]_{R\times R}$ is the kernel matrix between R landmarks, and $G = (1/N)\, K_{R\times N}\left(I - (1/N)\,\mathbf{1}\mathbf{1}^T\right) K_{R\times N}^T$. $\boldsymbol{b}$ has a close-form solution depending on W:

$$\boldsymbol{b} = -\frac{1}{N} W^T K_{R\times N}\mathbf{1}\,. \tag{9}$$

Step 2. Optimize $\boldsymbol{\mu}$ ***for given*** $(W, \boldsymbol{b})$**.** The optimal $\boldsymbol{\mu}$ can be obtained by solving the following quadratic programming problem:

$$\min_{\boldsymbol{\mu}} \ \frac{1}{2}\boldsymbol{\mu}^T E \boldsymbol{\mu} + \boldsymbol{f}^T \boldsymbol{\mu} \ \text{s.t.}\ \mathbf{1}^T\boldsymbol{\mu} = 1,\ \boldsymbol{\mu} \succcurlyeq \mathbf{0}\,, \tag{10}$$

where E is defined as $E(i,j) = 2\,\mathrm{Tr}\left(W^T K_{R\times N}^{(i)} L \left(K_{R\times N}^{(j)}\right)^T W\right)$ $(i, j = 1, \cdots, M)$, $\boldsymbol{f} = \left[f^{(1)}, \cdots, f^{(M)}\right]^T$, $f^{(m)} = \lambda\,\mathrm{Tr}\left(W^T K_{R\times R}^{(m)} W\right)$, $K_{R\times N}^{(m)} = \left[\kappa^{(m)}\left(\boldsymbol{z}_r^{(m)}, \boldsymbol{x}_n^{(m)}\right)\right]_{R\times N}$ is the kernel matrix for the m-th feature between R landmarks and N database images, and $K_{R\times R}^{(m)} = \left[\kappa^{(m)}\left(\boldsymbol{z}_i^{(m)}, \boldsymbol{z}_j^{(m)}\right)\right]_{R\times R}$ is the kernel matrix for the m-th feature between R landmarks $(m = 1, \cdots, M)$.

In summary, the optimization approach works as follows. First, it initializes $\boldsymbol{\mu} = [1/M, \cdots, 1/M]^T$. Then, it iteratively updates $(W, \boldsymbol{b})$ according to step 1 and updates $\boldsymbol{\mu}$ according to step 2 until they converge. In practice, our method usually finds the optimal $(W, \boldsymbol{b})$ and $\boldsymbol{\mu}$ within a few iterations.

3 Experiments

Experimental Settings: Our experiments are carried out on the breast-tissue microscopic image dataset built in [5, 13]. Briefly speaking, this dataset comprises 20 actionable (atypical ductal hyperplasia, ADH, and ductal carcinoma in situ, DCIS) cases and 20 benign (usual ductal hyperplasia, UDH) cases. 654 and 1723 images, each of which

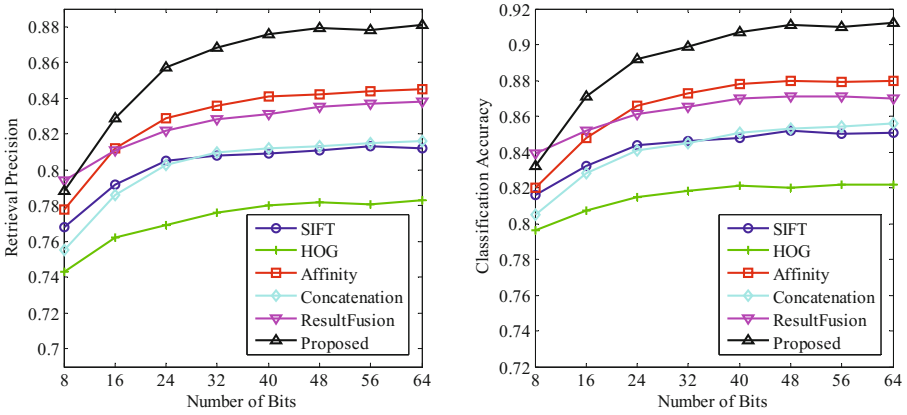

Fig. 2. Retrieval precision (left) and classification accuracy (right) at different hash code lengths.

has about 2.25M pixels, are sampled from the slides of these two categories. Four-fold cross-validation is performed for reliable results. That is, both actionable and benign cases are divided into four parts, and the proposed approach is evaluated four times. During each time, images in three parts form a database, and images in the other part are used as queries. Note that the query and database images are selected from different cases to avoid positive bias. The average performance from four runs is reported.

The proposed approach employs two texture-related features, namely scale-invariant feature transform (SIFT) [10] and histogram of oriented gradient (HOG) [4], which have demonstrated good performance in medical image retrieval and analysis [2, 3, 7, 13, 14]. In particular, scale-invariant keypoints are first detected by finding local extrema in difference-of-Gaussian (DoG) space, then SIFT and HOG features are calculated around these keypoints. Both features are quantized using bag-of-words (BoW) method [2, 3] and represented as 2000-dimensional histograms. As for kernel function, the proposed approach adopts Gaussian radial basis function (RBF) for both SIFT and HOG. Gaussian RBF is very popular in kernelized learning methods, and has been successfully applied to medical image analysis [2].

Five baseline methods are implemented for comparison. The first two methods, following [13], apply traditional KSH [8] on SIFT and HOG BoW respectively. The other three methods exploit KSH on both features, which are unified through affinity aggregation [14], feature catenation [1], and result-level fusion [12], respectively. These feature fusion methods have been widely used in medical image retrieval and demonstrated good performance.

Results and Analysis: We first evaluate the *retrieval precision* of all the methods, which is defined as the percentage of retrieved database images that are relevant to query image. The top 20 retrieved images are considered for this purpose. To demonstrate parameter sensitivity, each method uses a series of hash code lengths, ranging from 8 to 64 bits. The results are summarized in Fig. 2. For all the methods, as the hash code length increases to 64 bits, the precision scores first improve and then converge. This observation indicates that hashing-based methods can transform high-dimensional

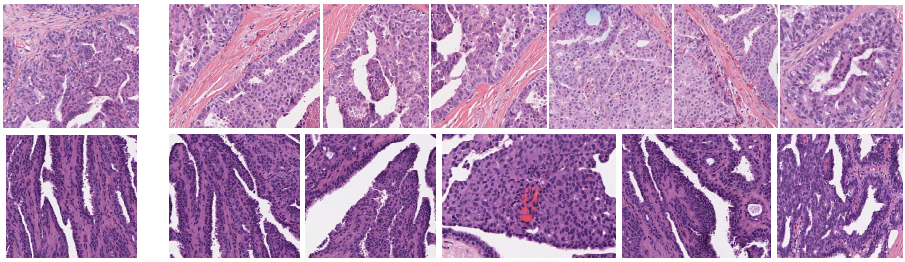

Fig. 3. Two query images (left) and their retrieved database images obtained by our approach (right). Images in the top row are all actionable, and images in the bottom row are all benign.

Table 1. Query time with 64-bit hash code

Method	SIFT	HOG	Affinity	Concatenation	ResultFusion	Proposed
Time (millisecond)	13.2	13.1	14.1	13.8	26.4	14.0

image features to compact yet descriptive image "signatures". Among all the six approaches, the two methods utilizing single feature perform worst. As for the fusion methods, feature concatenation gains marginal improvement over "SIFT + KSH", while affinity aggregation and result-level fusion obtain considerable improvement. The proposed approach substantially outperforms all compared methods, and achieves a precision score of 88.1% when using 64-bit hash code. Two retrieval examples are provided in Fig. 3 for visual evaluation, which demonstrate that our approach could find visually and semantically similar database images for queries.

Then, *classification accuracy* is measured, which refers to the percentage of query images that are correctly classified. Remember that a query image is classified as actionable or benign tissue according to a weighted majority vote of its retrieved database images [7]. The accuracy scores are reported in Fig. 2. Apparently, these scores exhibit similar intra- and inter-method trends to those of the precision scores. Furthermore, the accuracy scores are systematically higher than the precision scores, since irrelevant retrieved images would not cause a misclassification as long as they remain a minority of the retrieval set. Once again, our approach considerably surpasses all the baseline methods. Especially, it achieves a satisfactory classification accuracy of 91.2% at 64-bit hash code length.

Finally, *query time*, i.e. the time needed to retrieve and classify a query image, is investigated when using 64-bit hash code. Here, the time cost of SIFT and HOG BoW calculation is not taken into account, since it remains fixed as the database expands and therefore is not the bottleneck for large-scale image analysis. As shown in Table 1, these hashing-based methods exhibit outstanding computational efficiency. This is attributed to the compactness of hash codes, as well as adoption of hash table and "kernel trick". It is noteworthy that our approach, along with affinity aggregation and feature concatenation, has only a small computational overhead compared with methods using a single feature. As expected, the time cost for result-level fusion is the sum of those for "SIFT + KSH" and "HOG + KSH".

4 Conclusion

In this paper, we adopt joint kernel-based supervised hashing (JKSH) for fusion of complementary features. Multiple-feature hashing is transformed to a similarity preserving problem with linearly combined kernel functions, which are corresponding to the similarity measures for individual features. An alternating optimization algorithm is performed to learn both the kernel combination and hashing functions efficiently. Superior to traditional fusion methods, the proposed approach is suitable for heterogeneous features and doesn't introduce new parameters. Extensive experiments on breast cancer histopathological images demonstrate the efficacy of our approach. Future endeavors will be devoted to improve the performance by choosing better image features and kernel functions. In addition, we plan to extend our approach with online learning so that it could efficiently update hashing functions as new images are added into the database.

References

1. Akakin, H.C., Gurcan, M.N.: Content-based microscopic image retrieval system for multi-image queries. IEEE TITB 16(4), 758–769 (2012)
2. Caicedo, J.C., Cruz, A., González, F.A.: Histopathology image classification using bag of features and kernel functions. In: Combi, C., Shahar, Y., Abu-Hanna, A. (eds.) AIME 2009. LNCS, vol. 5651, pp. 126–135. Springer, Heidelberg (2009)
3. Caicedo, J.C., González, F.A., Romero, E.: Content-based histopathology image retrieval using a kernel-based semantic annotation framework. JBI 44(4), 519–528 (2011)
4. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Proc. IEEE CVPR, pp. 886–893 (2005)
5. Dundar, M.M., Badve, S., Bilgin, G., Raykar, V.C., Jain, R.K., Sertel, O., Gurcan, M.N.: Computerized classification of intraductal breast lesions using histopathological images. IEEE TBME 58(7), 1977–1984 (2011)
6. He, J., Liu, W., Chang, S.: Scalable similarity search with optimized kernel hashing. In: Proc. ACM SIGKDD, pp. 1129–1138 (2010)
7. Jiang, M., Zhang, S., Li, H., Metaxas, D.N.: Computer-aided diagnosis of mammographic masses using scalable image retrieval. IEEE TBME 62(2), 783–792 (2015)
8. Liu, W., Wang, J., Ji, R., Jiang, Y., Chang, S.: Supervised hashing with kernels. In: Proc. IEEE CVPR, pp. 2074–2081 (2012)
9. Liu, X., He, J., Lang, B.: Multiple feature kernel hashing for large-scale visual search. PR 47(2), 748–757 (2014)
10. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV 60(2), 91–110 (2004)
11. Veta, M., Pluim, J.P.W., van Diest, P.J., Viergever, M.A.: Breast cancer histopathology image analysis: A review. IEEE TBME 61(5), 1400–1411 (2014)
12. Wei, C.H., Li, Y., Huang, P.J.: Mammogram retrieval through machine learning within BI-RADS standards. JBI 44(4), 607–614 (2011)
13. Zhang, X., Liu, W., Dundar, M., Badve, S., Zhang, S.: Towards large-scale histopathological image analysis: Hashing-based image retrieval. IEEE TMI 34(2), 496–506 (2015)
14. Zhang, X., Yang, L., Liu, W., Su, H., Zhang, S.: Mining histopathological images via composite hashing and online learning. In: Golland, P., Hata, N., Barillot, C., Hornegger, J., Howe, R. (eds.) MICCAI 2014, Part II. LNCS, vol. 8674, pp. 479–486. Springer, Heidelberg (2014)