Supervised Graph Hashing for Histopathology Image Retrieval and Classification

Xiaoshuang Shi, Fuyong Xing, Kadi Xu, Yuanpu Xie, Hai Su, Lin Yang

PII: S1361-8415(17)30123-8
DOI: 10.1016/j.media.2017.07.009
Reference: MEDIMA 1285

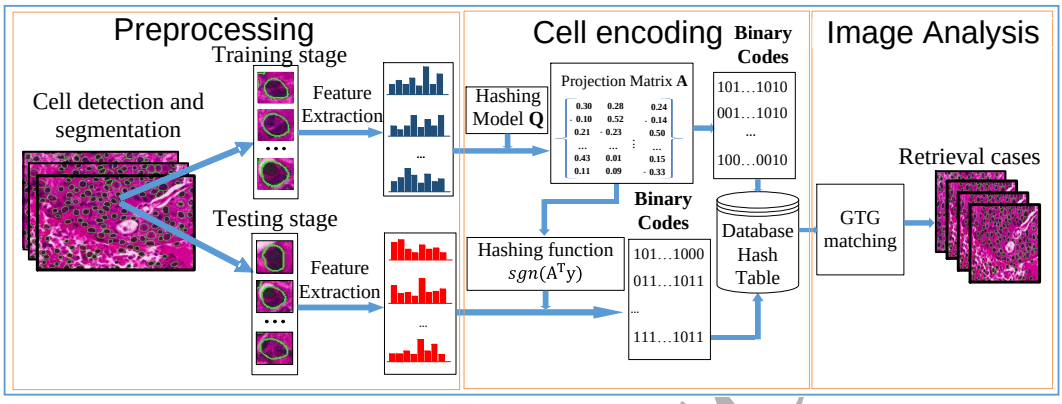
To appear in: *Medical Image Analysis*

Received date: 4 November 2016
Revised date: 25 July 2017
Accepted date: 31 July 2017

Please cite this article as: Xiaoshuang Shi, Fuyong Xing, Kadi Xu, Yuanpu Xie, Hai Su, Lin Yang, Supervised Graph Hashing for Histopathology Image Retrieval and Classification, *Medical Image Analysis* (2017), doi: 10.1016/j.media.2017.07.009

Research highlights

(1) An framework based on cell encoding for large-scale histopathological image analysis is proposed.
(2) A supervised graph-based model via asymmetric relaxation and its scalable version are proposed.
(3) A group-to-group matching method to retrieve images based on binary codes of cells is proposed.



Supervised Graph Hashing for Histopathology Image Retrieval and Classification

Xiaoshuang Shi[a], Fuyong Xing[b], Kadi Xu[c], Yuanpu Xie[a], Hai Su[a], Lin Yang[a,*]

[a] *J. Crayton Pruitt Family Department of Biomedical Engineering University of Florida, Gainesville, FL, 32611-6130, U.S.A*
[b] *Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL, 32611-6130, U.S.A*
[c] *Department of Computer Science and Engineering, University of Florida, Gainesville, FL, 32611-6130, U.S.A*

Abstract

In pathology image analysis, morphological characteristics of cells are critical to grade many diseases. With the development of cell detection and segmentation techniques, it is possible to extract cell-level information for further analysis in pathology images. However, it is challenging to conduct efficient analysis of cell-level information on a large-scale image dataset because each image usually contains hundreds or thousands of cells. In this paper, we propose a novel image retrieval based framework for large-scale pathology image analysis. For each image, we encode each cell into binary codes to generate image representation using a novel graph based hashing model and then conduct image retrieval by applying a group-to-group matching method to similarity measurement. In order to improve both computational efficiency and memory requirement, we further introduce matrix factorization into the hashing model for scalable image retrieval. The proposed framework is extensively validated with thousands of lung cancer images, and it achieves 97.98% classification accuracy and 97.50% retrieval precision with all cells of each query image used.

Keywords: Image retrieval, large-scale images, hashing, histopathology image analysis.

*Corresponding author
Email address: lin.yang@bme.ufl.edu (Lin Yang)

1. Introduction

Histopathology plays an important role in the early diagnosis of different cancers, such as lung and breast cancers. However, manual examination of histopathological images is labor intensive, time consuming and error-prone due to high-resolution and subjective assessment of doctors. To reduce the workload of pathologists and provide more reliable and consistent analysis of histopathological images, image process techniques and modern machine learning algorithms have been widely used for medical diagnosis, disease detection and decision support (Petushi et al., 2006), (Yang et al., 2007), (Caicedo et al., 2009), (Basavanhally et al., 2010), (Dundar et al., 2011), (Tabesh et al., 2007), (Xing and Yang, 2016). Compared to classifier-based computer-aided diagnosis (CAD) systems that directly provide diagnosis results or grading scores, content-based image retrieval (CBIR) (Comaniciu et al., 1999), (Schnorrenberg et al., 2000), (Zheng et al., 2003), (El-Naqa et al., 2004), (Liu et al., 2016) is able to provide more clinical evidence to support the diagnosis, because CBIR methods can be used to not only classify the query image but also retrieve and visualize the images with morphological profiles most relevant (Greenspan and Pinhas, 2007), (Akakin and Gurcan, 2012), (Zhang et al., 2014) (Zhang et al., 2015a), (Zhang et al., 2015b), (Jiang et al., 2015), (Zhang and Metaxas, 2016), (Jiang et al., 2016).

Cell-level information, including shape, area, and nuclear and cytoplasm appearances, plays a significant role in disease grading. In clinical practice, they do have extensive applications: 1) Two major types of non-small cell lung cancer, adenocarcinoma and squamous carcinoma, often contain cells that exhibit a mixture of representative morphologies, and the disease classification can be achieved using a majority voting of different cells. 2) Gliomas highly depends on the cellular-level information. For example, 1p/19q co-deletion greatly relies on the shape of the cell (roundness), and searching for cells that exhibit the similar morphology has important prognosis values because this would allow the clinician researchers to check whether there exists important known clinical information in the existing image database. 3) Bladder cancer is another example. Finding the nucleus with prominent nucleolus inside and their content-wise similar nucleus in the database is very important to differentiate low- and high-grade bladder cancer. Therefore, rigorously measuring and analyzing each individual cell can significantly assist

pathologists for diagnosis and disease detection (Zhang et al., 2015c). However, it is a challenging task because there often exist hundreds of thousands of cells in one single digitized image. When using high-dimensional features for large-scale cell images, traditional CBIR methods usually exhibit low computational efficiency. As a result, most previous methods (Doyle et al., 2008), (Zhang et al., 2015a), (Jiang et al., 2015) encode the whole image as a holistic high-dimensional features by representing the statistics of cell-level information, and then compress the high-dimensional features for computational efficiency. Nevertheless, some significant information might lose in such holistic representation.

To enable large-scale image analysis, recently hashing-based retrieval methods are considerably attractive since they can significantly improve the requirement of computer memory and query time cost (Zhang et al., 2015c), (Wang et al., 2016), thereby facilitating fast image retrieval in a large-scale database. Hashing encodes the data using a set of discrete binary codes, which can be *easily stored and quickly searched*. Thus it is often used to retrieve nearest neighbors based on a certain similarity measurement in large-scale databases (Wang et al., 2012). Many hashing methods (Weiss et al., 2009), (Liu et al., 2012), (Shen et al., 2015), (Jiang et al., 2015), (Shi et al., 2017) have been proposed, and they have achieved great success in computer vision and data mining. In general, these methods can be roughly classified into two categories: (1) unsupervised hashing (Weiss et al., 2009), which aims to explore the intrinsic structure of data to preserve the similarity of neighbors without any supervision; (2) supervised hashing (Liu et al., 2012), (Shen et al., 2015), which utilizes the semantic information to assist searching and retrieval. Due to the semantic gap (Wang et al., 2012), unsupervised hashing models usually exhibit inferior performance to supervised hashing approaches. Therefore, supervised hashing methods are more preferred in histopathological image analysis.

Unfortunately, when hashing encodes each cell image into binary codes, there are two major concerns: (1) Directly learning discrete binary codes is an NP-hard problem (Liu et al., 2014), (Shen et al., 2015). To address this problem, most of hashing methods utilize the strategy of symmetric relaxation followed by a threshold to obtain binary codes (Weiss et al., 2009). However, this strategy would generate accumulated quantization errors between the discrete binary code matrix and its relaxed continuous matrix, which would decrease the retrieval and classification accuracy (Shen et al., 2015). (2) It is a non-trivial task to effectively utilize binary codes of cells in a query image

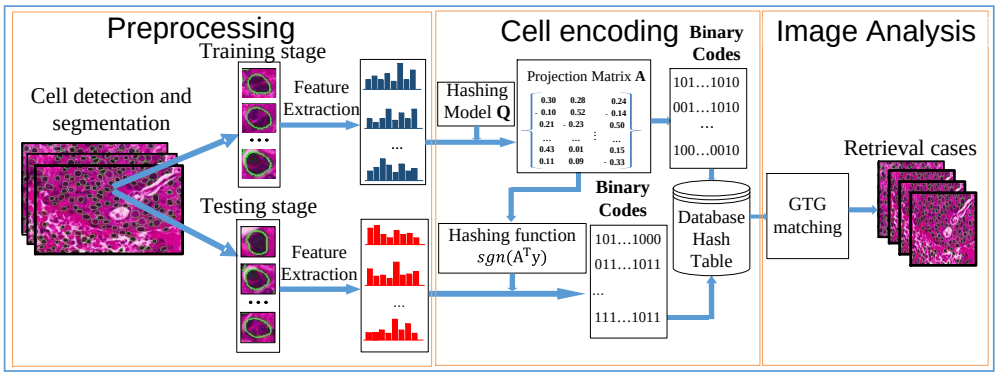
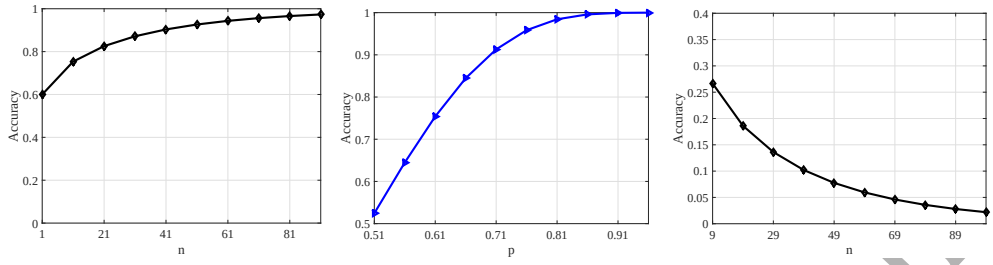


Fig 1: The flowchart of the proposed framework for pathology image analysis.

to retrieve relevant images. Usually, hashing methods retrieve images by a point-to-point matching. In other words, they encode the entire images into a set of binary codes and then calculate the Hamming distance between two images. This strategy is not suitable for cell based medical image retrieval since one pathology image contains more than one cell.

In this paper, we propose a novel framework for histopathological image analysis via encoding large-scale cells (see Fig 1). Specifically, instead of using symmetric relaxation followed by a threshold, we propose a supervised graph-based hashing (GH) algorithm via asymmetric relaxation, which preserves the hashing function in the objective function and thus effectively reduces the accumulated quantization errors between the discrete and continuous matrices (Shi et al., 2016b). Additionally, the GH model jointly learns binary codes and a projection matrix, usually exhibiting better and more robust performance than that learning them individually (Shi et al., 2015). In order to reduce the cost of memory storage and computational time, we further improve the GH model to enable scalable graph-based hashing (SGH), which selects a subset of cells from training cells to build an asymmetric graph to preserve the similarity of neighbors. The complexity of building a graph in SGH is significantly lower than that in GH. Next, we propose a novel method, namely group-to-group matching, to conduct image retrieval by using the cellular information in each image. In summary, our contributions are listed as follows:

- We propose a novel image retrieval framework for large-scale histopathological image analysis based on cell encoding.



(a) @ p=0.6 (b) @ n=9 (c) @ p=0.4

Fig 2: The change of image Classification accuracy with different n and p. (a) Accuracy vs n when $p > 0.5$, (b) Accuracy vs p, (a) Accuracy vs n when $p < 0.5$

- We propose a supervised graph-based model via asymmetric relaxation to learn binary codes and meanwhile reduce the accumulated quantization errors between the discrete and continuous matrices. Furthermore, we improve the model to allow scalable image retrieval by reducing both space and time complexity.

- We propose a group-to-group matching method to retrieve images based on the binary codes of cells.

The rest of this paper is structured as follows: Section 2 explains the reason why cell examination is helpful for histopathological image classification. Section 3 introduces the proposed graph-based model and its scalable version. Section 4 shows the novel strategy, group-to-group matching. Section 5 presents and interprets experimental results on lung cancer images. Finally, section 6 concludes this paper and points out the future work.

2. Why cell examination is helpful for histopathological image classification

In this section, we explain the reason why cell examination is helpful for histopathological image classification from a statistical viewpoint. Given two types of cells: adenocarcinoma (class 1) and squamous carcinoma (class 2) in lung cancer images, suppose that an adenocarcinoma image I contains n (without loss of generality, n is assumed to be an odd constant) cells, with p representing the probability of each cell belonging to adenocarcinoma type, where $p \in (0.5, 1)$. Based on the popular major voting strategy (Penrose,

1946), (Gans and Smart, 1996), the probability of the image I belongs to the adenocarcinoma type is:

$$P_I = E(l_I = 1|p, n) = \sum_{i=\frac{n+1}{2}}^{n} C_n^i p^i (1-p)^{n-i}, \tag{1}$$

where l_I represents the predicted label of the image I.

With n fixed, the probability P_I increases when p grows; with p fixed, a larger n means a higher P_I. For clarity, we present these two cases in Fig 2a-b, which suggest that in a query image, when $p \in (0.5, 1)$, a larger number of cells means a better image classification accuracy (Fig 2a), and a better cell classification accuracy will also lead to a better image classification accuracy (Fig 2b). These two observations show the relations between cell examination and pathological image classification, which is one major motivation of our framework. Note that when $p \in (0, 0.5)$, a larger number of cells will lead to a worse image classification accuracy (Fig 2c).

3. Supervised Graph Hashing

3.1. Graph-based Hashing

3.1.1. Problem formulation

Each cell is cropped as an image patch (see Fig 1), from which feature representation is calculated. Given a set of N cells $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N] \in \mathbb{R}^{d \times N}$, the goal of hashing is to find c hash functions $H = [h_1, h_2, \cdots, h_c]$ leading to c-bit Hamming embedding of $\mathbf{X}$. The hash function can be written as:

$$h_k(\mathbf{x}_i) = sgn(\mathbf{a}_k^T \mathbf{x}_i + b_k), \tag{2}$$

where h_k is the k-th hash function with learnable parameters $\mathbf{a}_k \in \mathbb{R}^d$ and b_k. Usually b_k is equivalent to $-\frac{1}{N}\sum_{i=1}^{n} \mathbf{a}_k^T \mathbf{x}_i$ and will be zero if $\mathbf{X}$ is normalized to have zero-mean.

Intuitively, adjacent cells in the original feature space should have similar binary codes after embedding, meaning that they should still be close to each other in the embedding space. Graph embedding is an attractive technique by considering the intrinsic dimensionality. Traditional graph embedding (Yan et al., 2007), (Nie et al., 2011) aims to find an optimal low-dimensional projection matrix $\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_c] \in \mathbb{R}^{d \times c}$ to preserve the similarity among

data points, and the optimization problem is formulated as:

$$\begin{aligned} &\min_{\mathbf{A}} \ Tr\left\{\mathbf{A}^T\mathbf{X}\mathbf{L}\mathbf{X}^T\mathbf{A}\right\}, \\ &s.t. \ \mathbf{A}^T\mathbf{X}\mathbf{X}^T\mathbf{A} = \mathbf{I}_c, \end{aligned} \tag{3}$$

where $\mathbf{L} = \mathbf{D} - \mathbf{W}$, $\mathbf{D}$ is a diagonal matrix with the i-th diagonal element $d_{ii} = \sum_{j=1}^{n} w_{ij}$, and $\mathbf{W}$ is a symmetric matrix with the weight w_{ij} between data points $\mathbf{x}_i$ and $\mathbf{x}_j$. In order to produce binary codes for hashing encoding, the linear projection function $\mathbf{A}^T\mathbf{X}$ is usually replaced by the hashing function $sgn(\mathbf{A}^T\mathbf{X})$, and thus Eq. (3) becomes:

$$\begin{aligned} &\min_{\mathbf{A}} \ Tr\left\{sgn(\mathbf{A}^T\mathbf{X})\mathbf{L}sgn(\mathbf{X}^T\mathbf{A})\right\}, \\ &s.t. \ sgn(\mathbf{A}^T\mathbf{X})sgn(\mathbf{X}^T\mathbf{A}) = N\mathbf{I}_c. \end{aligned} \tag{4}$$

By using the fact that $Tr\left\{sgn(\mathbf{A}^T\mathbf{X})\mathbf{D}sgn(\mathbf{X}^T\mathbf{A})\right\}$ is a constant and $\mathbf{L} = \mathbf{D} - \mathbf{W}$ as well as symmetric relaxation of the hashing function $sgn(\mathbf{A}^T\mathbf{X})$ (Weiss et al., 2009), (Jiang et al., 2015), Eq. (4) can be written as:

$$\begin{aligned} &\max_{\mathbf{A}} \ Tr\left\{\mathbf{A}^T\mathbf{X}\mathbf{W}\mathbf{X}^T\mathbf{A}\right\}, \\ &s.t. \ \mathbf{A}^T\mathbf{X}\mathbf{X}^T\mathbf{A} = N\mathbf{I}_c, \end{aligned} \tag{5}$$

where the constraint can reduce the redundancy among data points (Shi et al., 2016a).

One major problem of Eq. (5) is that the accumulated quantization error between the hashing function (binary code matrix) and the linear projection function might largely affect the retrieval accuracy, especially for a large number of training data (Shen et al., 2015). In order to reduce accumulated quantization errors, we utilize an asymmetric relaxation strategy (Shi et al., 2016b) and propose the following optimization model:

$$\begin{aligned} &\max_{\mathbf{A},\mathbf{B}\in\{-1,1\}^{c\times N}} \ Tr\left\{\mathbf{B}\mathbf{W}\mathbf{X}^T\mathbf{A}\right\} - \tfrac{\alpha}{2}\left\|\mathbf{B} - \mathbf{A}^T\mathbf{X}\right\|_F^2, \\ &s.t. \ \left\|\mathbf{A}^T\mathbf{X}\mathbf{X}^T\mathbf{A} - N\mathbf{I}_c\right\|_F^2 \leq \epsilon, \end{aligned} \tag{6}$$

where $\mathbf{B}$ is a discrete matrix representing the binary codes of training data, and the Frobenius norm regularization term aims to reduce the accumulated quantization error between the binary code matrix $\mathbf{B}$ and the linear projection function $\mathbf{A}^T\mathbf{X}$. Note that we utilize the constraint $\left\|\mathbf{A}^T\mathbf{X}\mathbf{X}^T\mathbf{A} - N\mathbf{I}_c\right\|_F^2 \leq \epsilon$ rather than $\mathbf{A}^T\mathbf{X}\mathbf{X}^T\mathbf{A} = N\mathbf{I}_c$ to improve the robustness of the projection

matrix $\mathbf{A}$. In addition, jointly learning the binary codes $\mathbf{B}$ and projection matrix $\mathbf{A}$ in Eq. (6) can further improve the robustness of the projection matrix $\mathbf{A}$ (Shi et al., 2015). Eq. (6) seems to be similar to the objective function in (Liu et al., 2014), however, they are intrinsically different in the following two aspects: (1) Eq. (6) is to learn the projection matrix $\mathbf{A}$ for supervised hashing, while (Liu et al., 2014) focuses on unsupervised hashing; (2) The discrete constraint in the objective function of (Liu et al., 2014) is symmetric, but an asymmetric discrete constraint is used in Eq. (6), which can be solved with lower time costs.

3.1.2. Optimization procedure

It is difficult to simultaneously calculate $\mathbf{B}$ and $\mathbf{A}$ in Eq. (6). By introducing an auxiliary $\mathbf{C}$, the optimization problem in Eq. (6) can be optimized by solving the following three subproblems:

B-subproblem:

$$\max_{\mathbf{B}^{c\times N}\in\{-1,1\}} Tr\left\{\mathbf{B}(\mathbf{W}+\alpha\mathbf{I}_N)\mathbf{X}^T\mathbf{A}\right\}. \tag{7}$$

Eq. (7) can be obtained from Eq. (6) because $Tr\left\{\mathbf{A}^T\mathbf{X}\mathbf{X}^T\mathbf{A}\right\}$ and $Tr\left\{\mathbf{B}\mathbf{B}^T\right\}$ are constants. The solution to Eq. (7) is $\mathbf{B}=sign(\mathbf{A}^T\mathbf{X}(\mathbf{W}+\alpha\mathbf{I}_N))$.

C-subproblem:

$$\begin{aligned}&\max_{\mathbf{C}\in\{-1,1\}^{N\times c}} Tr\left\{\mathbf{B}(\mathbf{W}+\alpha\mathbf{I}_N)\mathbf{C}\right\},\\ &s.t.\ \mathbf{C}^T\mathbf{C}=N\mathbf{I}_c, \mathbf{1}_N\mathbf{C}=0,\end{aligned} \tag{8}$$

where $\mathbf{I}_c\in\mathbb{R}^{c\times c}$ is an unit matrix, $\mathbf{1}_N\in\mathbb{R}^N$ is a row vector with all elements being one, and $\mathbf{1}_N\mathbf{C}=0$ because of $\mathbf{X}\mathbf{1}_N^T=0$. We can obtain $\mathbf{C}$ based on the singular value decomposition (SVD) of $\mathbf{B}(\mathbf{W}+\alpha\mathbf{I}_N)\mathbf{J}=\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, where $\mathbf{J}=\mathbf{I}_N-\frac{1}{N}\mathbf{1}_N^T\mathbf{1}_N$, and $\mathbf{C}=\sqrt{N}\mathbf{V}(:,1:c)\mathbf{U}^T$.

A-subproblem:

$$\min_{\mathbf{A}} \left\|\mathbf{C}-\mathbf{X}^T\mathbf{A}\right\|_F^2+\gamma\left\|\mathbf{A}\right\|_F^2, \tag{9}$$

whose solution is $\mathbf{A}=(\mathbf{X}\mathbf{X}^T+\gamma\mathbf{I}_d)^{-1}\mathbf{X}\mathbf{C}$. When $\gamma=0$, $\mathbf{C}=\mathbf{X}^T\mathbf{A}$ and then $\mathbf{A}^T\mathbf{X}\mathbf{X}^T\mathbf{A}=N\mathbf{I}_c$; when $\gamma>0$, $\left\|\mathbf{A}^T\mathbf{X}\mathbf{X}^T\mathbf{A}-N\mathbf{I}_c\right\|_F^2\leq\epsilon$, and a larger γ means a larger ϵ.

The detailed procedure to solve Eq. (6), namely graph-based hashing (GH), is summarized in Algorithm 1. Since each step in Algorithm 1 has a

Algorithm 1: Graph-based Hashing (GH)

Input: Data matrix $\mathbf{X} \in \mathbb{R}^{d\times N}$, weight matrix $\mathbf{W} \in \mathbb{R}^{N\times N}$, regularization parameter α, γ, the number of bits c, the maximum iterations k.

Output: Binary code matrix $\mathbf{B} \in \mathbb{R}^{c\times N}$, projection matrix $\mathbf{A} \in \mathbb{R}^{d\times c}$.

1. Calculate $\mathbf{A}$ corresponding to the c largest eigenvalues of $\mathbf{X}(\mathbf{W} + \alpha\mathbf{I}_N)\mathbf{X}^T$.
2. Calculate the matrix $\mathbf{B} = sgn(\mathbf{A}^T\mathbf{X})$.
3. Loop until convergence or reach maximum iterations
   3.1 Update $\mathbf{B} = sign(\mathbf{A}^T\mathbf{X}(\mathbf{W} + \alpha\mathbf{I}_N))$;
   3.2 Update $\mathbf{C} = \sqrt{N}\mathbf{V}(:, 1:c)\mathbf{U}^T$, where $\mathbf{U}\Sigma\mathbf{V}^T = \mathbf{B}(\mathbf{W} + \alpha\mathbf{I}_N)\mathbf{J}$;
   3.3 Update $\mathbf{A} = (\mathbf{X}\mathbf{X}^T + \gamma\mathbf{I}_d)^{-1}\mathbf{X}\mathbf{C}$.

closed form solution, empirically we can obtain approximately optimal solutions within a few iterations (e.g. f=5), where f is the number of iterations.

3.1.3. Time complexity analysis of GH

In Algorithm 1, the calculation of the matrix $\mathbf{A}$ in step 1 requires $\mathcal{O}(N^2d)$ operations because of $N > d$, and step 2 needs $\mathcal{O}(Ndc)$ operations to calculate the matrix $\mathbf{B}$. In step 3, the calculation of the matrices $\mathbf{B}$, $\mathbf{C}$ and $\mathbf{A}$ requires $\mathcal{O}(fN^2d)$, $\mathcal{O}(fN^2c)$ and $\mathcal{O}(fNd^2)$ operations, respectively. Usually $c < d < N$, and thus the total computational complexity of Algorithm 1 is $\mathcal{O}(fN^2d)$.

3.2. Scalable Graph-based Hashing

In GH, the complexity of both storage and computational time to construct a graph with N data points is $\mathcal{O}(N^2)$, and thus GH might not adapt to large-scale datasets (a large N). To improve memory requirement and computational efficiency, we present a fast hashing algorithm, namely scalable graph-based hashing (SGH), to approximate the model in Eq. (6).

3.2.1. Problem formulation

Dimensionality reduction methods, like principal component analysis (PCA) (Zou et al., 2006) and nonnegative matrix factorization (NMF) (Recht et al., 2012), suggest that a matrix with high-dimensionality yet low rank can be represented or approximated by a linear combination of basis vectors. In our

problem, the weight matrix $\mathbf{W}$ can be approximated by $\mathbf{W} \approx \mathbf{TP}$, where $\mathbf{T} \in \mathbb{R}^{N\times M}$, $\mathbf{T} \subset \mathbf{W}$ and $\mathbf{P} \in \mathbb{R}^{M\times N}$ is a weight matrix. Therefore, Eq. (6) can be reformulated as:

$$\begin{aligned} \max_{\mathbf{A},\mathbf{B}\in\{-1,1\}^{c\times N}} & Tr\left\{\mathbf{BTPX}^T\mathbf{A}\right\} - \tfrac{\alpha}{2}\left\|\mathbf{B}-\mathbf{A}^T\mathbf{X}\right\|_F^2, \\ s.t. & \left\|\mathbf{A}^T\mathbf{X}\mathbf{X}^T\mathbf{A} - N\mathbf{I}_c\right\|_F^2 \le \epsilon. \end{aligned} \tag{10}$$

Although $\mathbf{T}$ can be a subset of $\mathbf{W}$, computing $\mathbf{P}$ is expensive. To avoid the calculation of $\mathbf{P}$, we suppose $\mathbf{Z} = \mathbf{XP}^T$ be generated new data (anchors) that are used to construct graph, and $\mathbf{T}$ can be seen as a weight matrix characterizing the relationship between the training data $\mathbf{X}$ and anchors $\mathbf{Z}$. Since the projection matrix mainly depends on the data matrix $\mathbf{X}$ due to $M << N$, we ignore the effect of the new anchors on the projection matrix and thus neglect the calculation of $\mathbf{P}$. Let $\mathbf{D} = \mathbf{Z}^T\mathbf{A}$, Eq. (10) can be rewritten as:

$$\begin{aligned} \max_{\mathbf{A},\mathbf{D},\mathbf{B}\in\{-1,1\}^{c\times N}} & Tr\left\{\mathbf{BTD}\right\} - \tfrac{\alpha}{2}\left\|\mathbf{B}-\mathbf{A}^T\mathbf{X}\right\|_F^2, \\ s.t.\ \mathbf{D}^T\mathbf{D} = M\mathbf{I}_c, & \left\|\mathbf{A}^T\mathbf{X}\mathbf{X}^T\mathbf{A} - N\mathbf{I}_c\right\|_F^2 \le \epsilon, \end{aligned} \tag{11}$$

where the constraint $\mathbf{D}^T\mathbf{D} = M\mathbf{I}_c$ is derived from the constraint $\mathbf{A}^T\mathbf{X}\mathbf{X}^T\mathbf{A} = N\mathbf{I}_c$.

3.2.2. Optimization procedure

Similar to solving Eq. (6), we adopt an iterative and alternative strategy to compute the discrete binary codes $\mathbf{B}$ and the projection matrix $\mathbf{A}$ in Eq. (11). We divide the optimization problem in Eq. (11) into four subproblems as follows and present the details of scalable graph-based hashing (SGH) in Algorithm 2.

B-subproblem:

$$\max_{\mathbf{B}^{c\times N}\in\{-1,1\}} Tr\left\{\mathbf{BTD} + \alpha\mathbf{BX}^T\mathbf{A}\right\}, \tag{12}$$

whose solution is $\mathbf{B} = sign(\mathbf{D}^T\mathbf{T}^T + \alpha\mathbf{A}^T\mathbf{X})$.

D-subproblem:

$$\max_{\mathbf{D}} Tr\left\{\mathbf{BTD}\right\}, s.t.\ \mathbf{D}^T\mathbf{D} = M\mathbf{I}_c. \tag{13}$$

The solution to Eq. (13) is $\mathbf{D} = \sqrt{M}\mathbf{V}_D(:, 1:c)\mathbf{U}_D^T$, where $\mathbf{U}_D$ and $\mathbf{V}_D$ can be obtained by the SVD of $\mathbf{BT} = \mathbf{U}_D\Sigma_D\mathbf{V}_D^T$.

Algorithm 2: Scalable Graph-based Hashing (SGH)

Input: Data matrix $\mathbf{X} \in \mathbb{R}^{d\times N}$, graph matrix $\mathbf{T} \in \mathbb{R}^{N\times M}$, parameters α, γ, the number of bits c, the number of iterations k.

Output: Binary code matrix $\mathbf{B} \in \mathbb{R}^{c\times N}$, projection matrix $\mathbf{A} \in \mathbb{R}^{d\times c}$.

1. Calculate $\mathbf{A}$ corresponding to the c largest eigenvalues of $\mathbf{X}\mathbf{T}\mathbf{T}^T\mathbf{X}^T$;
2. Calculate the matrix $\mathbf{B} = sgn(\mathbf{A}^T\mathbf{X})$;
3. Loop until converge or reach maximum iterations
   3.1 Update $\mathbf{B} = sgn(\mathbf{D}^T\mathbf{T}^T + \alpha\mathbf{A}^T\mathbf{X})$;
   3.2 Update $\mathbf{D} = \sqrt{M}\mathbf{V}_D(:, 1:c)\mathbf{U}_D^T$, where $\mathbf{B}\mathbf{T} = \mathbf{U}_D\Sigma_D\mathbf{V}_D^T$;
   3.3 Update $\mathbf{C} = \sqrt{N}\mathbf{V}_C(:, 1:c)\mathbf{U}_C^T$, where $\mathbf{B}\mathbf{J} = \mathbf{U}_C\Sigma_C\mathbf{V}_C^T$;
   3.4 Update $\mathbf{A} = (\mathbf{X}\mathbf{X}^T + \gamma\mathbf{I}_d)^{-1}\mathbf{X}\mathbf{C}$;

C-subproblem:

$$\max_{\mathbf{C}} Tr\left\{\mathbf{B}\mathbf{C}\right\}, s.t.\ \mathbf{C}^T\mathbf{C} = N\mathbf{I}_c, \mathbf{1}_N\mathbf{C} = 0. \tag{14}$$

The solution to Eq. (14) is $\mathbf{C} = \sqrt{N}\mathbf{V}_C(:, 1:c)\mathbf{U}_C^T$, where $\mathbf{U}_C$ and $\mathbf{V}_C$ can be obtained by the SVD of $\mathbf{B}\mathbf{J} = \mathbf{U}_C\Sigma_C\mathbf{V}_C^T$.

A-subproblem: This subproblem is the same to Eq. (9).

Since supervised hashing methods usually have better performance than unsupervised hashing, in this paper, we focus on supervised hashing method to retrieve cells, and the weight matrix $\mathbf{W} \in \mathbb{R}^{N\times N}$ that utilizes semantic information is defined as follows:

$$w_{ij} = \begin{cases} \frac{1}{N_k} & if\ x_i, x_j \in k\text{-th } class, \\ 0 & otherwise, \end{cases} \tag{15}$$

where N_k is the number of samples in the k-th class, $1 \le k \le K$ and K is the number of classes. Since $\mathbf{T} \in \mathbb{R}^{N\times M}$ is the subset of $\mathbf{W}$, we can attain $\mathbf{T}$ by randomly selecting M_k data points as anchors to construct $\mathbf{T}$, where M_k is the selected number of anchors from the k-th class.

3.2.3. Time complexity analysis

In Algorithm 2, the computational complexity of calculating $\mathbf{A}$ in step 1 is $max(\mathcal{O}(NMd), \mathcal{O}(d^3))$. The calculation of the matrix $\mathbf{B}$ in step 2 requires $\mathcal{O}(Ndc)$ operations. In step 3, calculating the matrices $\mathbf{B}$, $\mathbf{D}$, $\mathbf{C}$ and $\mathbf{A}$ needs

$max(\mathcal{O}(fNMc), \mathcal{O}(fNdc))$, $\mathcal{O}(fNMc)$, $\mathcal{O}(fNc^2)$ and $\mathcal{O}(fNd^2)$ operations, respectively. Since usually $fc < d$, the total time complexity of Algorithm 2 is $max(\mathcal{O}(fNd^2), \mathcal{O}(NMd))$.

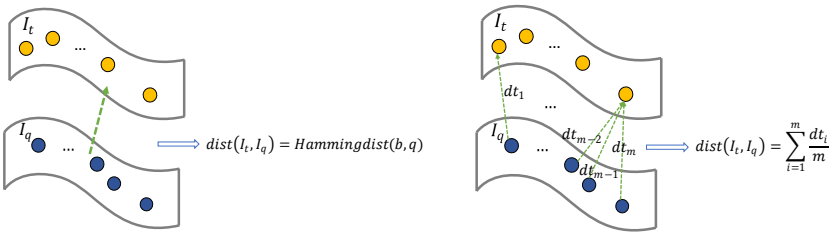
3.3. Relations to other related hashing algorithms

There are many hashing algorithms proposed in the literature, and to better explain the proposed methods, in this section we discuss the relations between ours and several popular and related hashing algorithms.

Spectral hashing (SH) (Weiss et al., 2009) is a popular graph based hashing algorithm, but it has two major limitations: (1) The complexity of both storage and computational time to construct a graph with N data points is $\mathcal{O}(N^2)$; (2) Symmetric relaxation is used to solve the NP-hard optimization problem, which would generate accumulated errors between the discrete binary code matrix and its relaxed continuous matrix. To reduce the complexity of building a graph, anchor graph hashing (AGH) (Liu et al., 2011) constructs an approximative symmetric graph with a small number of anchors to preserve the similarity between neighbors. However, AGH utilizes the symmetric relaxation strategy to solve the NP-hard optimization problem. To reduce the accumulated quantitative errors, discrete graph hashing (DGH) (Liu et al., 2014) preserves the symmetric discrete constraint. Compared to GH that preserves the asymmetric discrete constraint, the optimization procedure of DGH is more complex and takes more training time. In addition, both AGH and DGH focus on learning binary codes in an unsupervised manner such that it is difficult to directly utilize semantic (label) information to learn binary codes.

KSH (Liu et al., 2012) is a kernel based hashing algorithm applied to pathological image analysis. It utilizes symmetric relaxation followed by greedy optimization to solve its non-differentiable objective function. This strategy is not able to greatly reduce the accumulated errors and usually costs a large amount of training time. Joint kernel graph hashing (JKGH) (Jiang et al., 2015) also uses symmetric relaxation to solve its NP-hard optimization problem. It constructs the kernel of each data with weighting each sub-kernel constructed by each feature and thus achieves better performance than KSH. However, JKGH usually requires considerable training and test time to construct the kernel. Hence it is not suitable for tackling large-scale data with high-dimensional features.

Although kernel-based supervised discrete hashing (KSDH) (Shi et al., 2016b) also utilizes the asymmetric relaxation strategy to directly learn bi-



(a) (b)

Fig 3: The ideas of point-to-point and group-to-group. (a) point-to-point, (b) group-to-group. (Each point in a plane represents a cell in one image, the discrete vectors **b** and **q** represent binary codes of the entire target and query images, respectively.)

nary codes, it does not learn binary codes and a projection matrix simultaneously, which might decrease its performance (Shi et al., 2015).

Compared to KSH, JKSH and kernel-based supervised discrete hashing (KSDH) (Shi et al., 2016b), the proposed algorithms have three major advantages: (1) GH and SGH are linear methods and thus they do not require kernel selection and construction; (2) GH and SGH utilize asymmetric relaxation to solve the NP-hard optimization problem, and thus they can reduce the accumulated quantitative errors; (3) KSH, JKGH and KSDH require $\mathcal{O}(N^2)$ operations to construct their weight matrix, while SGH needs only $\mathcal{O}(MN)$ $(M << N)$ operations.

4. Group-to-group matching for pathological image retrieval

One single histopathology image usually contains multiple cells, and this would fail the traditional point-to-point matching (Fig 3a) in image retrieval, which calculates the Hamming distance between two entire images that are encoded into a set of binary codes. To address this problem, we propose a novel strategy: group-to-group (GTG) matching, for image retrieval using cell-level information. Given one target image I_t containing n cells and one query image I_q including m cells, denote their binary vectors by $[\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_n]$ and $[\mathbf{q}_1, \mathbf{q}_2, \cdots, \mathbf{q}_m]$, respectively, where $\mathbf{b}_i \in \{0,1\}^c$, $\mathbf{q}_j \in \{0,1\}^c$, $1 \leq i \leq n$ and $1 \leq j \leq m$. We can regard that the cells in one target image $\mathbf{I}_t$ form a plane $\mathcal{P}_t$, and each query cell can be regarded as a point outside of the plane $\mathcal{P}_t$. Then the distance between the point $\mathbf{q}_i$ and

Algorithm 3: Group-to-group (GTG)

Input: Training binary matrix $[\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_n] \in \mathbb{R}^{n \times c}$, query binary matrix $[\mathbf{q}_1, \mathbf{q}_2, \cdots, \mathbf{q}_m] \in \mathbb{R}^{m \times c}$.

Output: $dist(I_t, I_q)$.

for $j = 1$ to m do
  for $i = 1$ to n do
    $dt(i) = Hammingdist(\mathbf{b}_i, \mathbf{q}_j)$,
  end;
$dist(j, I_t) \leftarrow min(dt)$,
end;
$dist(I_t, I_q) \leftarrow \frac{1}{m} \sum_{j=1}^{m} dist(j, I_t)$.

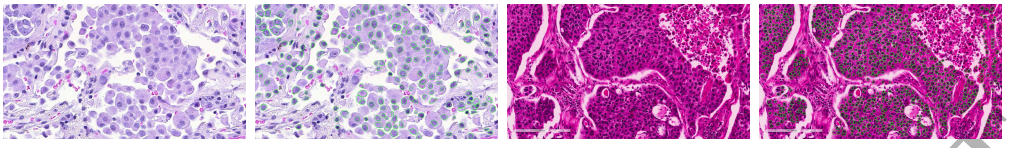
the plane $\mathcal{P}_t$ is equivalent to $dist(\mathbf{q}_j, \mathcal{P}_t) = \min\limits_{1 \le i \le n} dist(\mathbf{q}_j, \mathbf{b}_i)$, and the distance between these two images is defined as follows:

$$dist(I_t, I_q) = \frac{1}{m} \sum_{j=1}^{m} \min\limits_{1 \le i \le n} dist(\mathbf{q}_j, \mathbf{b}_i). \tag{16}$$

For better illustration, we explain this idea in Fig 3b. The details of the GTG matching to calculate the distance between two images are shown in Algorithm 3. Note that compared to the major voting strategy, GTG can be viewed as a weighted voting strategy, because the cells in one query image has different distances to the target image.

5. Experiments

To evaluate the proposed hashing algorithms (GH and SGH), we conduct extensive experiments on the lung cancer image dataset with two types of diseases: squamous cell carcinoma and adenocarcinoma. The lung cancer dataset is collected from The Cancer Genome Altas (TCGA) (Institute:, 2013), which consists of 1240 images (630 squamous and 610 adenocarcinoma images). These images contains around 1248K squamous and 589K adenocarcinoma cells. All cells are detected and segmented using the method in (Xing et al., 2014). Then all cells are cropped as image patches with corresponding labels. For each cell, we extract a 1024-dimensional feature vector by GIST (Oliva and Torralba, 2001) and HOG (Dalal and Triggs, 2005), respectively. Additionally, we also utilize LeNet (LeCun et al., 2015) to extract a 300-dimensional feature vector from each cell. The sample images and their segmented cells are shown in Fig 4.



(a) (b) (c) (d)

Fig 4: Sample images and segmented cells. (a) and (b) Adenocarcinoma image and cell segmentation results, (c) and (d) squamous cell carcinoma image and cell segmentation results.

5.1. Experimental setting:

We compare GH and SGH against two state-of-the-art supervised hashing algorithms: KSH (Liu et al., 2012) and SDH (Shen et al., 2015). After using hashing algorithms to encode each cell into binary codes, the GTG matching is used for image retrieval. To better show the characteristics of GTG, we also present the classification accuracy of the hashing algorithms with major voting (MV). In addition, we compare with two popular classifiers: support vector machine (SVM) (Fan et al., 2008) and nearest neighbors (NN). We utilize SVM and NN to classify cells and then applying major voting to image classification, and also show the baseline results obtained by directly applying SVM and NN on the holistic high-dimensional features extracted from the whole image. Specifically, similar to (Jiang et al., 2015), (Zhang et al., 2015a), we first detect scale-invariant keypoints from the whole pathology image, and then employ SIFT (Lowe, 2004) and HOG to extract features around these keypoints. Next, both descriptors are encoded as 2000-dimensional histograms using bag-of-words (BoW) methods (Caicedo et al., 2009), (Caicedo et al., 2011). Finally, SVM and NN are applied to these histograms for image classification.

We adopt five criteria including classification accuracy, precision, mean average precision (MAP), training and test time, to evaluate the hashing algorithms. Classification accuracy evaluates the classification performance; precision measures the retrieval accuracy (returned corrected cells over total returned samples), MAP evaluates the ranking performance of total returned cells, training and test time are used to measure the time cost on training and query, respectively. For GH and SGH, we set $\alpha = 10^{-4}$ and $\alpha = 10^{-3}$, respectively, and $\gamma = 10^{-3}$ in the following experiments, and we set $M = 0.1N$ in SGH. For KSH, SDH, KSDH_H and KSDH_B, we randomly select

500 training cells to construct the kernels.

We first randomly select 100 images from each category for training and the remaining images are used for testing. We randomly select 100 cells from each training image to constitute a training cell dataset. In total, there are 20K cells in the training dataset. To form a test cell dataset we randomly select m cells from each test image. After obtaining the binary codes of these test cells, we use them to classify test images and retrieve relevant images. Since m is an important hyperparameter, we present the performance of all hashing algorithms with respect to different m's. We repeat this process 10 times and calculate the mean accuracy of different methods. All experiments are conducted using MATLAB on a 3.50GHz Intel Xeon CPU with 128GB memory.

5.2. Experimental results

Table 1 shows image classification results of two popular classifiers, SVM and NN, using the holistic high-dimensional HOG and SIFT features extracted from the whole image, and cell classification results of eight algorithms using HOG and GIST features extracted from cell images. We use GIST rather than SIFT to extract cell features because SIFT is much slower to extract features from a larger number of cell images. Additionally, we also present cell classification results of eight algorithms using CNN features. As shown in Table 1, GH achieves the best cell classification accuracy with HOG, GIST and CNN features. Although SGH has slightly worse cell classification accuracy than SVM, SDH, KSDH_H and KSDH_B with HOG features, it significantly outperforms NN and KSH. When GIST and CNN features are used, SGH attains better performance than NN, KSH, SDH, KSDH_H and KSDH_B. The results in Table 1 are used as the baseline for comparison.

Table 2 presents the classification results of two classifiers SVM, NN and six retrieval algorithms KSH, SDH, KSDH_H, KSDH_B, GH and SGH with MV, and the classification and retrieval results of the six retrieval algorithms with GTG. Table 2 suggests that the retrieval algorithms with GTG achieve better classification accuracy than those with MV in most of cases. Meanwhile, when 9 cells are selected from each query image, KSH+MV and SGH+MV obtain worse image classification performance, since the classification accuracy of KSH (using HOG, GIST or CNN features) and SGH (using HOG features) on squamous cells is very low ($< 50\%$). On the contrary, KSH+GTG and SGH+GTG achieve much better image classification

Table 1: Baseline classification accuracy (%) on images and cells. (Hashing algorithms encode each cell into 10-bit binary codes.)

(a)

	Method	Adeno	Squam	Mean	Adeno	Squam	Mean
		HOG			**SIFT**		
Image	SVM	40.39	81.51	61.35	74.90	81.51	78.27
	NN	61.57	59.25	60.38	74.71	83.02	78.94

(b)

		HOG			**GIST**			**CNN**		
Cell ($m = 9$)	SVM	64.60	60.88	62.70	64.99	64.63	64.81	78.54	71.20	74.60
	NN	42.57	63.88	53.43	56.21	56.67	56.44	70.07	75.05	72.61
	KSH	83.66	24.19	53.35	91.42	12.54	51.22	99.11	0.01	49.08
	SDH	59.98	68.76	62.46	69.04	59.16	64.01	75.75	72.50	74.10
	KSDH_H	71.74	53.98	62.69	73.90	55.24	64.39	80.15	67.95	73.94
	KSDH_B	69.63	56.39	62.88	72.57	56.48	64.37	78.96	69.63	74.20
	GH	64.77	61.28	**62.99**	68.39	61.74	**65.00**	76.75	73.50	**75.10**
	SGH	75.99	49.27	62.37	74.14	55.22	64.50	79.98	68.33	74.27

performance. This indicates the superior performance of GTG to MV. In addition, with 9 cells selected from each query image, SVM, NN and six hashing algorithms with CNN features can attain better performance than that with HOG or GIST features; among these algorithms, SGH+GTG achieves the best classification accuracy (88.83%), precision (88.50%) and MAP (88.73%), GH+GTG with CNN features achieves similar performance to the other hashing algorithms except SGH+GTG.

Table 2: Classification accuracy, precision and MAP (%) of different algorithms on lung cancer images. ('Accuracy' represents the classification accuracy in table and 'All' means that all cells in the query image are used for retrieval. Hashing algorithms encode each cell into 10-bit binary codes.)

	Method	**HOG**				
		Accuracy			**Precision**	**MAP**
		Adeno	Squam	Mean	(Top 100)	(Top 100)
$m = 9$	SVM+MV	76.73	81.96	71.70	-	-
	NN+MV	30.59	75.09	53.27	-	-
	KSH+MV	98.43	5.28	50.96	-	-
	SDH+MV	71.75	82.96	77.46	-	-
	KSDH_H+MV	90.91	60.00	75.75	-	-
	KSDH_B+MV	89.27	62.36	76.07	-	-
	GH+MV	82.35	71.32	76.73	-	-
	SGH+MV	99.22	10.19	53.85	-	-

$m = 9$	KSH+GTG	79.98	80.70	**80.35**	66.34	72.14
	SDH+GTG	74.71	79.81	77.31	77.31	77.31
	KSDH_H+GTG	80.76	75.42	78.04	77.89	83.78
	KSDH_B+GTG	89.20	76.74	77.94	78.09	78.06
	GH+GTG	82.35	73.21	77.69	77.89	77.99
	SGH+GTG	85.69	73.21	79.33	**79.02**	**85.54**
All	SVM+MV	98.04	91.89	94.90	-	-
	KSH+GTG	93.12	99.00	**96.12**	79.47	87.83
	SDH+GTG	90.59	98.68	94.71	94.71	94.71
	KSDH_H+GTG	96.08	94.15	95.10	94.64	95.03
	KSDH_B+GTG	96.08	94.15	95.10	95.10	95.10
	GH+GTG	97.65	94.34	95.96	**96.02**	**95.91**
	SGH+GTG	98.24	93.02	95.58	95.23	94.93
		GIST				
$m = 9$	SVM+MV	81.25	80.00	80.26	-	-
$m = 9$	NN+MV	71.18	61.70	66.35	-	-
	KSH+MV	100	0.19	49.13	-	-
	SDH+MV	87.84	73.21	80.38	-	-
	KSDH_H+MV	92.80	48.00	70.40	-	-
	KSDH_B+MV	91.20	56.80	74.00	-	-
	GH+MV	87.45	71.17	82.22	-	-
	SGH+MV	95.10	57.74	76.06	-	-
$m = 9$	KSH+GTG	90.39	72.45	81.25	65.84	72.04
	SDH+GTG	88.04	73.58	80.67	80.67	80.67
	KSDH_H+GTG	90.00	73.58	81.63	81.34	84.07
	KSDH_B+GTG	90.20	73.77	81.83	81.83	81.83
	GH+GTG	88.24	74.91	81.44	81.37	82.12
	SGH+GTG	91.76	76.98	**84.23**	**84.10**	**86.81**
All	SVM+MV	98.61	96.82	97.75	-	-
	KSH+GTG	96.65	99.00	97.85	81.04	89.82
	SDH+GTG	100	92.83	96.35	96.35	96.35
	KSDH_H+GTG	100	94.91	97.40	96.83	97.20
	KSDH_B+GTG	99.80	94.89	97.30	97.30	96.05
	GH+GTG	99.61	95.47	97.56	97.56	**97.56**
	SGH+GTG	99.80	96.23	**97.98**	**97.50**	96.52
		CNN				
$m = 9$	SVM+MV	88.43	83.77	86.06	-	-
	NN+MV	88.24	86.23	87.21	-	-
	KSH+MV	100	0	49.04	-	-
	SDH+MV	91.57	82.08	86.73	-	-
	KSDH_H+MV	91.50	76.23	85.48	-	-
	KSDH_B+MV	92.94	81.32	87.02	-	-
	GH+MV	89.02	84.34	86.63	-	-
	SGH+MV	88.82	86.79	**87.79**	-	-

$m = 9$	KSH+GTG	92.55	63.58	77.79	68.31	73.14
	SDH+GTG	91.26	83.46	87.29	87.28	87.19
	KSDH_H+GTG	92.55	82.64	87.39	87.39	87.39
	KSDH_B+GTG	90.76	85.04	87.85	87.85	87.85
	GH+GTG	90.22	84.47	88.31	88.31	88.31
	SGH+GTG	91.41	86.34	**88.83**	**88.50**	**88.73**
All	SVM+MV	94.71	93.40	94.04	-	-
	KSH+GTG	97.25	93.40	95.29	73.54	82.15
	SDH+GTG	96.86	92.83	94.81	94.81	94.81
	KSDH_H+GTG	97.65	91.32	94.42	94.12	94.25
	KSDH_B+GTG	96.86	92.82	94.80	94.80	94.80
	GH+GTG	97.06	93.21	95.10	95.10	95.10
	SGH+GTG	97.45	93.40	**95.38**	**95.38**	**95.38**

When all cells are selected, both GH+GTG and SGH+GTG outperform SDH on classification accuracy, precision and MAP; GH+GTG has similar performance to KSDH_H and KSDH_B, while SGH+GTG has slightly better performance; GH+GTH, SGH+GTG and KSH+GTG achieve almost the same classification accuracy, however, GH+GTH and SGH+GTG have significantly better precision and MAP than KSH. Compared to the case with only 9 cells selected, SVM+MV, KSH+GTG, SDH+GTG, KSDH_H+GTG, KSDH_B+GTG, GH+GTG and SGH+GTG using all cells of each query image can achieve better performance, especially for SGH+GTG, which using GIST features can achieve the best classification accuracy (97.98%) and precision (97.50%), and the sub-best MAP (96.52%). Note that we do not show the results of NN+MV when all cells are selected, because it is very computationally expensive. Compared to Table 1, Table 2 shows that image classification using features extracted from cells can obtain better accuracy than that using the holistic high-dimensional features extracted from the whole image. To better understand our framework and hashing algorithms, we also present a retrieval example in Fig 5, which further exhibits better performance of GH and SGH than the other hashing methods. Note that the similarity between the query and training images is calculated by using GTG, which determines the weight of each query cell. Usually, the smaller Hamming distance between a query cell and a training image, the larger weight of the query cell. If the weight of query cells changes, different returned images might be obtained. Moreover, the number of query cells in one image also affects the retrieved images.

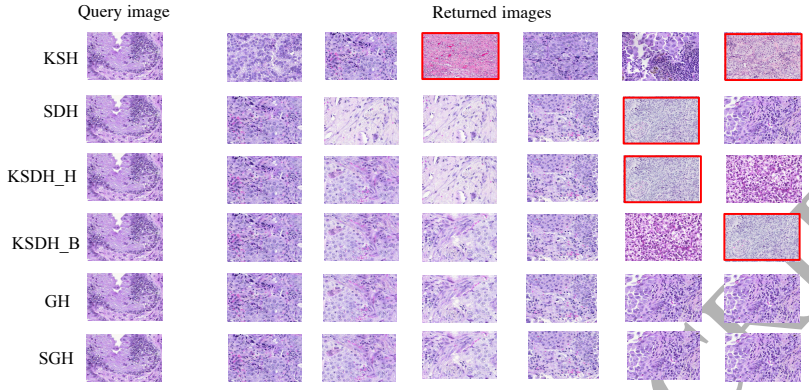


Fig 5: A query image and its top six returned images obtained by using the six hashing algorithms and GIST features. (The error images are marked by a red rectangle block. Additionally, since all hashing algorithms use the strategy GTG, we do not show its name in this figure for simplicity.)

Table 3: Training and query time (second) comparison of different retrieval algorithms (Since all hashing algorithms use the strategy GTG, we do not show its name in this table for simplicity. Training time is the time with all training data learning binary codes, and query time is the querying time of one test image used. 'All' means that all cells in the query image are used for retrieval. In addition, hashing algorithms encode each cell into 10-bit binary codes.)

Time	**KSH**	**SDH**	**KSDH_H**	**KSDH_B**	**GH**	**SGH**
Training	1.7×10^3	1.6	3.0×10^1	2.0×10^1	1.9×10^1	7.6
Query ($m = 9$)	8.8×10^{-3}	8.7×10^{-3}	8.8×10^{-3}	8.8×10^{-3}	8.5×10^{-3}	8.5×10^{-3}
Query (All)	3.1	3.1	3.1	3.1	3.0	3.0

Table 3 lists the training time of KSH, SDH, KSDH_H, KSDH_B, GH and SGH, and their query time as well. KSH has the highest training time cost, while SDH takes the lowest. SGH costs lower training time than KSDH_H, KSDH_B and GH, and in contrast with SDH, its training time cost is acceptably lower. When 9 cells are selected, all algorithms spend less query time than all cells being selected. Note that in Table 3, we utilize only one hash table to retrieve relevant images, the query time can be reduced if multiple hashing tables are used.

5.3. Retrieval performance vs. query cells

Fig 6 presents the classification accuracy, precision and MAP of different hashing algorithms with respect to different number m of cells selected from each query image. Fig 6a-c, Fig 6d-f and Fig 6g-i correspond to the cases

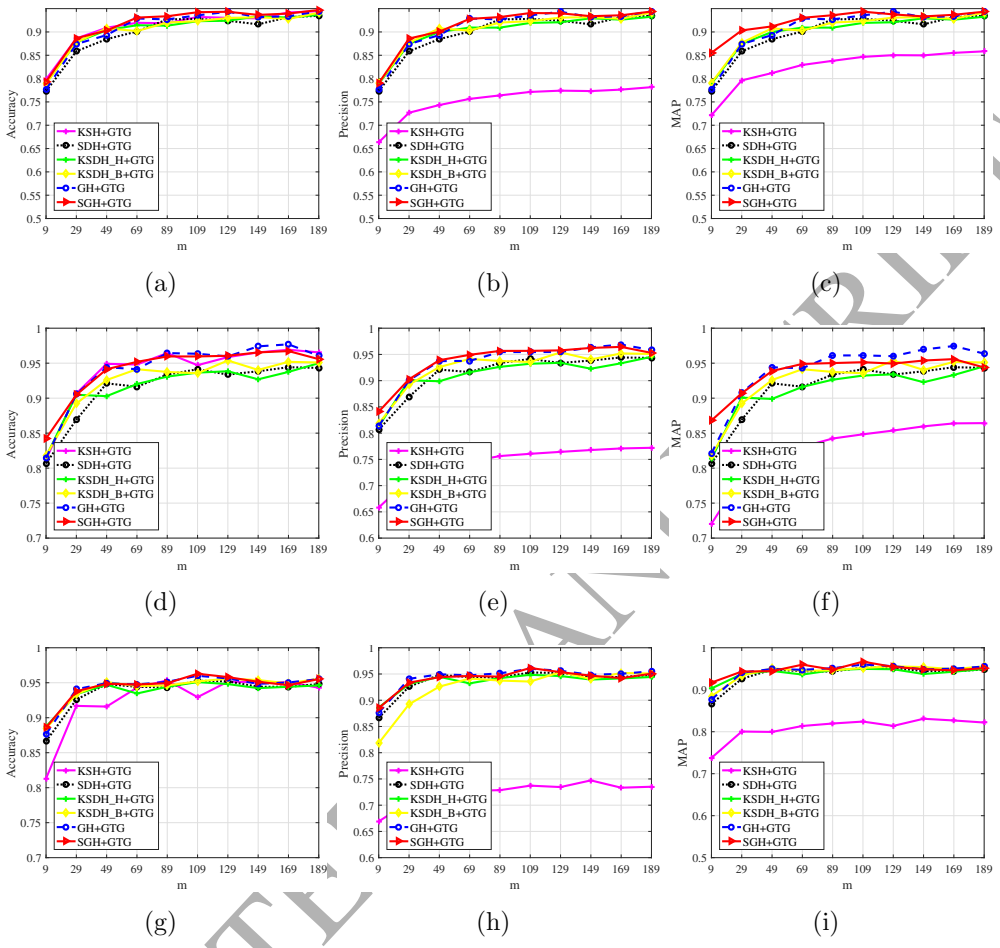


Fig 6: Retrieval accuracy vs the number of selected cells m. (a)-(c) Accuracy, precision and MAP vs m, respectively, with HOG features; (d)-(f) accuracy, precision and MAP vs m, respectively, with GIST features; (g)-(i) accuracy, precision and MAP vs m, respectively, with CNN features.

using HOG, GIST and CNN features, respectively. In Fig 6a-c, the classification accuracy, precision and MAP improve with the increasing of m, and GH+GTG and SGH+GTG perform better than KSH+GTG, KSDH_H, KSDH_B and SDH+GTG in almost all cases. When m=189, GH+GTG achieves 94.54% classification accuracy, 94.34% precision and 94.42% MAP, respectively. SGH+GTG can also obtain the similar results. In Fig 6d-f hashing algorithms with GIST features have the similar trends to those with

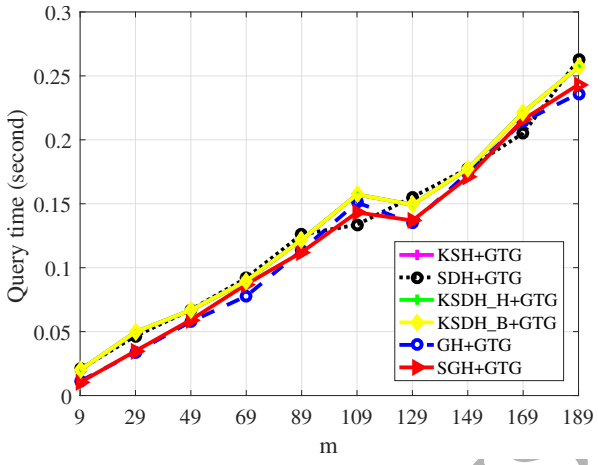


Fig 7: Query time vs the number m of selected anchors.

HOG features, and GH+GTG and SGH+GTG outperform KSH+GTG and SDH+GTG. When $m = 189$, GH+GTG attains the best classification accuracy (96.06%), precision (95.84%) and MAP (96.36%) among all hashing algorithms. In Fig 6g-i, the hashing algorithms with CNN features have similar trends to that with HOG or GIST features. When $m = 189$, GH+GTG attains the best classification accuracy (95.02%), precision (95.02%) and MAP (95.02%) among all hashing algorithms. It is worthy noting that hashing algorithms with CNN features can achieve their best or sub-best performance using less number of selected cells than that with HOG or GIST features. Although the hashing algorithms can achieve better results with a larger number of cells selected, Fig 7 suggests that a larger m means a higher query time cost. Therefore, in practice, we make a trade-off between the retrieval accuracy and query time.

5.4. Retrieval performance vs. dimension

In this section, we show the retrieval accuracy of various hashing algorithms with different number c of dimensions. Fig 8 presents the performance of various hashing algorithms with c during $[2, 20]$, respectively. Fig 8 shows that SDH+GTG and KSDH_B can achieve robust performance are more robust on different number of dimensions with HOG, GIST or CNN features; Fig 8 also displays that SGH+GTG and KSDH_H+GTG obtain robust performance with HOG or GIST features, while SGH+GTG and KSDH_H+GTG with CNN features achieve bad performance at 2- and 4-bit, respectively. GH+GTG achieves robust performance on CNN features, while

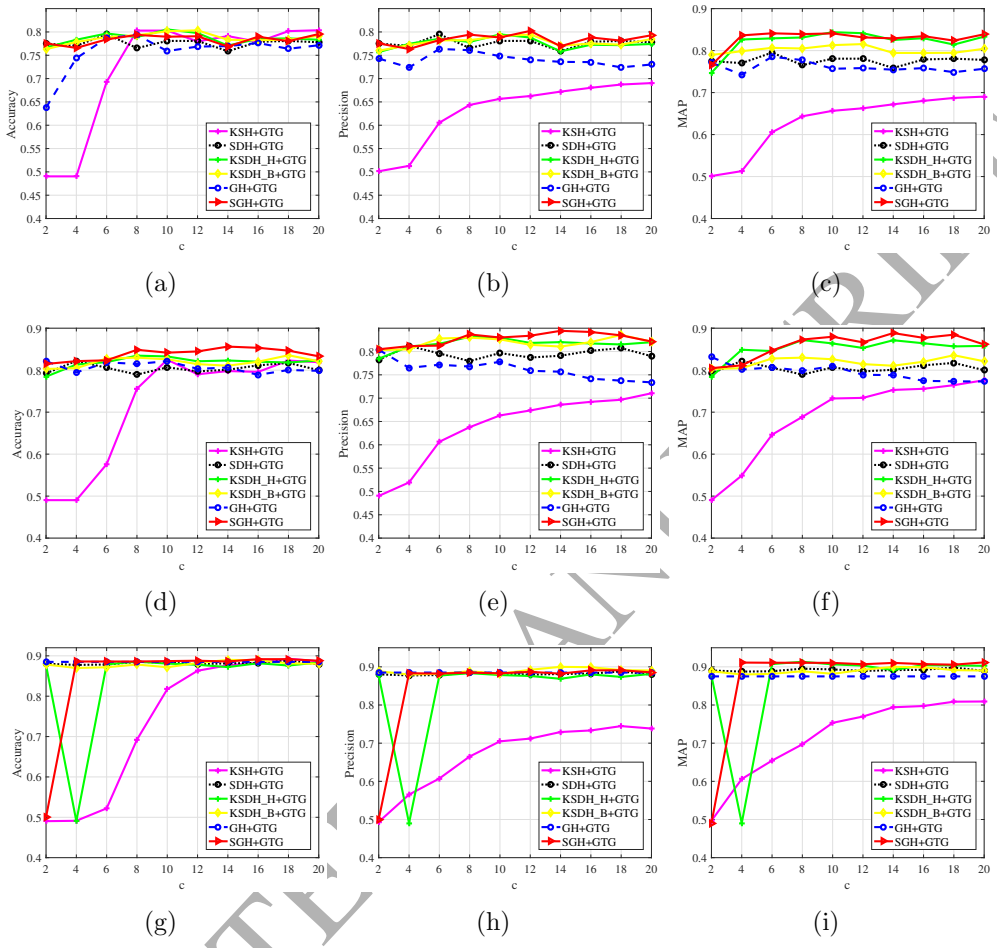


Fig 8: Retrieval accuracy vs the number of dimensions c. (a)-(c) Accuracy, precision and MAP vs c, respectively, with HOG features; (d)-(f) accuracy, precision and MAP vs c, respectively, with GIST features; (g)-(i) accuracy, precision and MAP vs c, respectively, with CNN features.

its performance is slightly changing with different number of dimensions on HOG or GIST features. KSH has worse retrieval accuracy on the three types of features when c is small, probably because it utilizes the greedy strategy to attain the approximated optimal solutions. When encoding cells into 10-bit binary codes, all hashing algorithms except KSH can achieve their best or sub-best performance on HOG, GIST and CNN features.

5.5. Parameters analysis

In the proposed algorithm GH, two essential parameters α and γ affect its classification and retrieval performance. In SGH, besides α and γ, M also largely affects its performance. We show their classification accuracy on different parameters with GIST features used in Fig 9. Since M is determined by the number of N, we use $Rate$ to describe the relationship between M and N in Fig 9, where $Rate = \frac{M}{N}$. As shown in Fig 9, the range of both α and γ is $[10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10]$ and the range of $Rate$ is from 2% to 40% with the step 2%. Fig 9a shows the accuracy of GH with different α and γ, which can achieve the best or sub-best accuracy during the range $[10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}]$ and $[10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}]$, respectively; Fixing $Rate = 10\%$, Fig 9b displays the accuracy of SGH with different α and γ, which can achieve the best or sub-best accuracy during the range $[10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}]$ and $[10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}]$, respectively; Fixing $\gamma = 10^{-3}$, Fig 9c presents the accuracy of SGH with different $Rate$ and α, which can achieve the best or sub-best accuracy during the range from 4% to 28%, and $[10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}]$, respectively; Fixing $\alpha = 10^{-3}$, Fig 9d shows the accuracy of SGH with different $Rate$ and γ, which can achieve the best or sub-best accuracy during the range from 4% to 40%, and $[10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}]$, respectively. Similar findings can be observed on HOG and CNN features. In our experiments, we choose $\gamma = 10^{-3}$ for both GH and SGH, set $\alpha = 10^{-4}$ for GH, and select $\alpha = 10^{-3}$ and $Rate = 10\%$ for SGH. Using these parameters, the proposed methods GH and SGH can obtain the approximately best and robust performance.

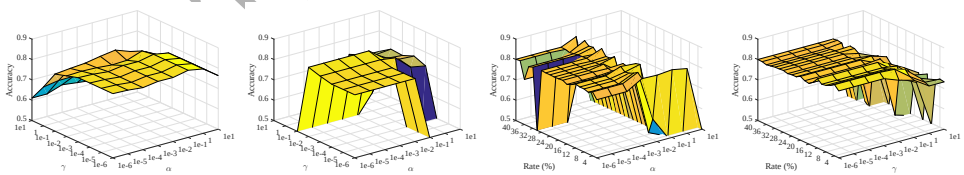


Fig 9: Classification accuracy on different parameters with GIST features. (a) Classification accuracy of GH on parameters α and γ; (b) Accuracy of SGH on parameters α and γ with $M = 0.1N$; (c) Accuracy of SGH on parameters $Rate$ and α with $\gamma = 10^{-3}$; (d) Accuracy of SGH on parameters $Rate$ and γ with $\alpha = 10^{-3}$.

5.6. Discussion and future work

Based on experiments on the lung cancer database, we can observe that: (1) SVM+MV, SDH+MV and GH+MV obtain reasonable image classification accuracy, but NN+MV and KSH+MV sometimes perform worse than NN and KSH. (2) The performance of all eight algorithms is affected by extracted features. GIST obtains better performance than HOG and CNN features when all cells are selected, although CNN features can achieve better performance than GIST and HOG features when 9 cells are selected. (3) The GTG matching can be effectively combined with the hashing algorithm for image retrieval and exhibits superior performance to MV. (4) Among six hashing algorithms, SGH and GH achieve superior retrieval performance to KSH, SDH, KSDH_H and KSDH_B in many cases; (5) Although all hashing algorithms can quickly retrieve the most relevant images when 9 cells are selected, the query time is relatively high when all cells are selected. The main reasons are summarized as follows:

- SVM+MV, SDH+MV and GH+MV provide promising performance, because their classification accuracy on each type of cells is larger than 50%. Based on Eq. 1 and Fig 2, for $p \in (0.5, 1)$, the classification accuracy will increase when the number of cells grows. By contrast, the classification accuracy will decrease if $p \in (0, 0.5)$, and this is the main reason why NN+MV and KSH+MV has inferior performance to NN and KSH.

- GIST and HOG are much faster than some other popular feature extraction methods like SIFT, and thus they are more suitable for tackling the large-scale cell problem. Usually, GIST extracting features via describing the entire cell image achieves better performance than HOG extracting cell texture and edge information. It suggests that global information of cell images is more important than their texture and edge information on cell and image classification, probably because global information contains the shape and area of the cell, which is beneficial to the cell and image classification. CNN features show superior performance to GIST and HOG features when a small number of cells are selected, while with all cells selected, they exhibit slightly worse performance than GIST and even HOG features, probably because: (1) CNN features contain high-level information, while HOG and GIST extract low-level information. This leads to better performance of CNN features when few cells are selected; (2) When all cells are selected, a large

number of cells in one query image will decrease the gap of accuracy obtained by GIST, HOG and CNN features, and CNN features might lead to overfitting to some extent.

- For GTG, when most of cells of one query image have smaller distances to one certain training image than that to the other training images, the distance of these two images will be the smallest in most of cases. Thus, GTG is effective for image classification and retrieval. Compared to MV, GTG is a weighted voting strategy which can better measure the similarity between the query image and those in the database, because the cells in one query image might have different distance to a specific target image. This explains why KSH+GTG, SDH+GTG, KSDH_H+GTG, KSDH_B+GTG, GH+GTG and SGH+GTG outperform KSH+MV, SDH+MV, KSDH_H+MV, KSDH_B+MV, GH+MV and SGH+MV, especially for KSH, whose classification accuracy on squamous cells is smaller than 50%.

- Compared with KSH that uses the symmetric relaxation + greedy strategy to learn binary codes, GH and SGH directly preserve the discrete binary code matrices to reduce the accumulated quantization error between the discrete and its relaxed matrices. This might be the main reason for the superior performance of GH and SGH to KSH. Additionally, GH and SGH can be easily solved and the optimization procedure quickly converges, and thus they require lower training time costs than KSH. Compared with SDH, GH and SGH produce better performance in almost all cases, probably because the performance of SDH is largely affected by the selection of kernels, and the graph structure in our methods can better preserve the similarity (label) information than the regression model used by SDH. Compared with KSDH_H and KSDH_B, SGH and GH can have superior performance in many cases, probably because (1) KSDH_H and KSDH_B do not jointly learn binary codes and projection matrices, and (2) their performance highly relies on the kernels; however, SGH and GH formulate binary code and projection matrix learning into a joint optimization problem, and the parameters α and γ can achieve their best performance during a wide range.

- When all query cells are selected, the time cost of image searching is high. To alleviate this issue, currently we can utilize multiple hash-

ing tables to reduce time costs. In our future work, we will design a novel weighted voting algorithm with lower querying time costs and meanwhile obtaining similar or even better performance. Moreover, our framework utilizes the segmentation method (Xing et al., 2014) to crop cell patches, which takes a high relatively time cost for large-scale image data. One potential solution is to design an end-to-end deep learning model for efficient cell segmentation, which will be our future work.

- In addition to lung images, we also conduct some experiments on other pathological applications including breast and brain cancers diagnosis, and our proposed framework can produce good performance. Moreover, they might be also suitable for 3D/4D pathological images with effective features (e.g., CNN features), since the proposed framework including the hashing algorithms are general methods.

6. Conclusion

In this paper, we present a hashing-based image retrieval framework for pathology image analysis. Firstly, we explain the reason why image classification can benefit from cell-level information. Then we present a graph-based hashing algorithm via asymmetric relaxation to encode each cell into a set of binary codes. To improve the scalability of the proposed graph hashing algorithm, we further propose a novel algorithm, namely scalable graph-based hashing. Next, we propose a group-to-group matching method to retrieve images based on binary codes of cells. Experimental results on lung cancer images demonstrate the effectiveness and efficiency of our framework. Since the method group-to-group matching takes relatively high searching time for the large number of cells in one query image. In the future, we will design a novel weighted voting method with lower time costs and meanwhile obtaining similar or even better performance. Additionally, since training and query images usually contain many noise cells, which usually affect the performance of the framework, in the future we will focus on selecting robust cells to further improve the robustness of the proposed framework. Moreover, based on the proposed framework we will design an end-to-end deep learning model for efficient and fast cell segmentation so that the total time cost is decreased.

References

Akakin, H. C., Gurcan, M. N., 2012. Content-based microscopic image retrieval system for multi-image queries. IEEE Trans. Inf. Tech. Biomed. 16 (4), 758–769.

Basavanhally, A. N., Ganesan, S., Agner, S., Monaco, J. P., Feldman, M. D., Tomaszewski, J. E., Bhanot, G., Madabhushi, A., 2010. Computerized image-based detection and grading of lymphocytic infiltration in her2+ breast cancer histopathology. IEEE Trans. Biomed. Eng. 57 (3), 642–653.

Caicedo, J. C., Cruz, A., Gonzalez, F. A., 2009. Histopathology image classification using bag of features and kernel functions. In: Proc. Conf. Artificial Intell. Med. Euro. Springer, pp. 126–135.

Caicedo, J. C., González, F. A., Romero, E., 2011. Content-based histopathology image retrieval using a kernel-based semantic annotation framework. J. Biomed. Inf. 44 (4), 519–528.

Comaniciu, D., Meer, P., Foran, D. J., 1999. Image-guided decision support system for pathology. Mach. Vision App. 11 (4), 213–224.

Dalal, N., Triggs, B., 2005. Histograms of oriented gradients for human detection. In: Proc. Int. Conf. Comput. Vision Pattern Recog. Vol. 1. IEEE, pp. 886–893.

Doyle, S., Agner, S., Madabhushi, A., Feldman, M., Tomaszewski, J., 2008. Automated grading of breast cancer histopathology using spectral clustering with textural and architectural image features. In: IEEE Int. Sym. on Biomed. Imag. IEEE, pp. 496–499.

Dundar, M. M., Badve, S., Bilgin, G., Raykar, V., Jain, R., Sertel, O., Gurcan, M. N., 2011. Computerized classification of intraductal breast lesions using histopathological images. IEEE Trans. Biomed. Eng. 58 (7), 1977–1984.

El-Naqa, I., Yang, Y., Galatsanos, N. P., Nishikawa, R. M., Wernick, M. N., 2004. A similarity learning approach to content-based image retrieval: application to digital mammography. IEEE Trans. Med. Imag. 23 (10), 1233–1244.

649 Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., Lin, C.-J., 2008. Liblin-
650 ear: A library for large linear classification. J. Mach. Learn, Res. 9 (Aug),
651 1871–1874.

652 Gans, J. S., Smart, M., 1996. Majority voting with single-crossing prefer-
653 ences. J. Public Economics 59 (2), 219–237.

654 Greenspan, H., Pinhas, A. T., 2007. Medical image categorization and re-
655 trieval for pacs using the gmm-kl framework. IEEE Trans. Inf. Tech.
656 Biomed. 11 (2), 190–202.

657 Institute:, N. C., 2013. The cancer genome atals retrieved from. https://tcga-
658 data.nci.nih.gov.

659 Jiang, M., Zhang, S., Huang, J., Yang, L., Metaxas, D. N., 2015. Joint kernel-
660 based supervised hashing for scalable histopathological image analysis. In:
661 Med. Image Comput. Comput. Assist. Interv. pp. 366–373.

662 Jiang, M., Zhang, S., Huang, J., Yang, L., Metaxas, D. N., 2016. Scalable
663 histopathological image analysis via supervised hashing with multiple fea-
664 tures. Medical Image Analysis 34, 3–12.

665 LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521 (7553),
666 436–444.

667 Liu, J., Zhang, S., Liu, W., Deng, C., Zheng, Y., Metaxas, D. N., 2016. Scal-
668 able mammogram retrieval using composite anchor graph hashing with it-
669 erative quantization. IEEE Transactions on Circuits and Systems for Video
670 Technology.

671 Liu, W., Mu, C., Kumar, S., Chang, S., 2014. Discrete graph hashing. In:
672 Proc. Neur. Inf. Process. Syst. pp. 3419–3427.

673 Liu, W., Wang, J., Ji, R., Jiang, Y., Chang, S., 2012. Supervised hashing
674 with kernels. In: Proc. Int. Conf. Comput. Vision Pattern Recog. pp. 2074–
675 2081.

676 Liu, W., Wang, J., Kumar, S., Chang, S.-F., 2011. Hashing with graphs. In:
677 Proc. Int. Conf. Mach. Learn. pp. 1–8.

678 Lowe, D. G., 2004. Distinctive image features from scale-invariant keypoints.
679 Int. J. Comput. Vision 60 (2), 91–110.

Nie, F., Wang, H., Huang, H., Ding, C., 2011. Unsupervised and semi-supervised learning via 1-norm graph. In: Proc. Int. Conf. Comput. Vision. IEEE, pp. 2268–2273.

Oliva, A., Torralba, A., 2001. Modeling the shape of the scene: A holistic representation of the spatial envelope. Int. J. Comput. Vision 42 (3), 145–175.

Penrose, L. S., 1946. The elementary statistics of majority voting. J. Royal Statist. Society 109 (1), 53–57.

Petushi, S., Garcia, F. U., Haber, M. M., Katsinis, C., Tozeren, A., 2006. Large-scale computations on histology images reveal grade-differentiating parameters for breast cancer. Bio. Med. Comput. Med. Imag. 6 (1), 1.

Recht, B., Re, C., Tropp, J., Bittorf, V., 2012. Factoring nonnegative matrices with linear programs. In: Proc. Neur. Inf. Process. Syst. pp. 1214–1222.

Schnorrenberg, F., Pattichis, C., Schizas, C., Kyriacou, K., 2000. Content-based retrieval of breast cancer biopsy slides. Tech. Health Care 8 (5), 291–297.

Shen, F., Shen, C., Liu, W., Shen, H., 2015. Supervised discrete hashing. In: Proc. Int. Conf. Comput. Vision Pattern Recog. pp. 37–45.

Shi, X., Guo, Z., Lai, Z., Yang, Y., Bao, Z., Zhang, D., 2015. A framework of joint graph embedding and sparse regression for dimensionality reduction. IEEE Trans. Imag. Process. 24 (4), 1341–1355.

Shi, X., Guo, Z., Nie, F., Yang, L., You, J., Tao, D., 2016a. Two-dimensional whitening reconstruction for enhancing robustness of principal component analysis. IEEE Trans. Pattern Anal. Mach. Intell. 38 (10), 2130–2136.

Shi, X., Xing, F., Cai, J., Zhang, Z., Xie, Y., Yang, L., 2016b. Kernel-based supervised discrete hashing for image retrieval. In: Proc. Euro. Conf. Comput. Vision. pp. 419–433.

Shi, X., Xing, F., Xu, K., Sapkota, M., Yang, L., 2017. Asymmetric discrete graph hashing. In: Thirty-First AAAI Conference on Artificial Intelligence.

709 Tabesh, A., Teverovskiy, M., Pang, H.-Y., Kumar, V. P., Verbel, D., Kot-
710 sianti, A., Saidi, O., 2007. Multifeature prostate cancer diagnosis and glea-
711 son grading of histological images. IEEE Trans. Med. Imag. 26 (10), 1366–
712 1378.

713 Wang, J., Kumar, S., Chang, S., 2012. Semi-supervised hashing for large-
714 scale search. IEEE Trans. Pattern Anal. Mach. Intell. 34 (12), 2393–2406.

715 Wang, J., Liu, W., Kumar, S., Chang, S.-F., 2016. Learning to hash for
716 indexing big dataa survey. Proceedings of the IEEE 104 (1), 34–57.

717 Weiss, Y., Torralba, A., Fergus, R., 2009. Spectral hashing. In: Proc. Neur.
718 Inf. Process. Syst. pp. 1753–1760.

719 Xing, F., Su, H., Neltner, J., Yang, L., 2014. Automatic ki-67 counting using
720 robust cell detection and online dictionary learning. IEEE Trans. Bio. Med.
721 Eng. 61 (3), 859–870.

722 Xing, F., Yang, L., 2016. Robust nucleus/cell detection and segmentation in
723 digital pathology and microscopy images: A comprehensive review. IEEE
724 Reviews in Biomedical Engineering PP (99), 1–1.

725 Yan, S., Xu, D., Zhang, B., Zhang, H., Yang, Q., Lin, S., 2007. Graph em-
726 bedding and extensions: a general framework for dimensionality reduction.
727 IEEE Trans. Pattern Anal. Mach. Intell. 29 (1), 40–51.

728 Yang, L., Chen, W., Meer, P., Salaru, G., Feldman, M. D., Foran, D. J., 2007.
729 High throughput analysis of breast cancer specimens on the grid. In: Proc.
730 Int. Conf. Medical Imag. Comput. Computer-Assist. Interv. Springer, pp.
731 617–625.

732 Zhang, S., Metaxas, D., 2016. Large-scale medical image analytics: Recent
733 methodologies, applications and future directions. Medical Image Analysis
734 33, 98–101.

735 Zhang, X., Liu, W., Dundar, M., Badve, S., Zhang, S., 2015a. Towards large-
736 scale histopathological image analysis: Hashing-based image retrieval.
737 IEEE Trans. Med. Imag. 34 (2), 496–506.

738 Zhang, X., Su, H., Yang, L., Zhang, S., 2015b. Fine-grained histopathological
739 image analysis via robust segmentation and large-scale retrieval. In: Proc.
740 Comput. Vision Pattern Recog. pp. 5361–5368.

Zhang, X., Su, H., Yang, L., Zhang, S., 2015c. Weighted hashing with multiple cues for cell-level analysis of histopathological images. In: Med. Imag. Analysis. pp. 303–314.

Zhang, X., Yang, L., Liu, W., Su, H., Zhang, S., 2014. Mining histopathological images via composite hashing and online learning. In: MICCAI (2). pp. 479–486.

Zheng, L., Wetzel, A. W., Gilbertson, J., Becich, M. J., 2003. Design and analysis of a content-based pathology image retrieval system. IEEE Trans. Inf. Tech. Biomed. 7 (4), 249–255.

Zou, H., Hastie, T., Tibshirani, R., 2006. Sparse principal component analysis. J. Comput. Graph. Statist. 15 (2), 265–286.