Pairwise based Deep Ranking Hashing For Histopathology Image Classification and Retrieval

Xiaoshuang Shi, Manish Sapkota, Fuyong Xing, Fujun Liu, Lei Cui, Lin Yang

PII: S0031-3203(18)30105-5
DOI: 10.1016/j.patcog.2018.03.015
Reference: PR 6493

To appear in: *Pattern Recognition*

Received date: 15 August 2017
Revised date: 31 January 2018
Accepted date: 20 March 2018

Please cite this article as: Xiaoshuang Shi, Manish Sapkota, Fuyong Xing, Fujun Liu, Lei Cui, Lin Yang, Pairwise based Deep Ranking Hashing For Histopathology Image Classification and Retrieval, *Pattern Recognition* (2018), doi: 10.1016/j.patcog.2018.03.015

Highlights

- We design a pairwise matrix to preserve both ranking and label information.
- We propose a novel objective function to learn binary codes.
- We propose a novel pairwise based supervised deep learning framework for fast image classification and retrieval.

Pairwise based Deep Ranking Hashing For Histopathology Image Classification and Retrieval

Xiaoshuang Shi[a], Manish Sapkota[b], Fuyong Xing[c], Fujun Liu[b], Lei Cui[d], Lin Yang[a,b*]

[a] *J. Crayton Pruitt Family Department of Biomedical Engineering, University of Florida*
[b] *Electrical and Computer Engineering, University of Florida*
[c] *Department of Biostatistics and Informatics, University of Colorado Denver*
[d] *School of Information and Technology, Northwest University, China*

Abstract

Hashing has become a popular tool on histopathology image analysis due to the significant gain in both computation and storage. However, most of current hashing techniques learn features and binary codes individually from whole images, or emphasize the inter-class difference but neglect the relevance order within the same classes. To alleviate these issues, in this paper, we propose a novel pairwise based deep ranking hashing framework. We first define a pairwise matrix to preserve intra-class relevance and inter-class difference. Then we propose an objective function that utilizes two identical continuous matrices generated by the hyperbolic tangent (tanh) function to approximate the pairwise matrix. Finally, we incorporate the objective function into a deep learning architecture to learn features and binary codes simultaneously. The proposed framework is validated on 5,356 skeletal muscle and 2,176 lung cancer images with four types of diseases, and

*Corresponding author
Email address: lin.yang@bme.ufl.edu (Lin Yang[a,b])

it can achieve 97.49% classification accuracy, 97.49% mean average precision (MAP) with 100 returned images, and 0.51 NDCG score with 50 retrieved neighbors on 2,032 query images.

Keywords: Histopathology images, classification, retrieval, ranking hashing, deep learning

1. Introduction

Histopathology images play a significant role in early disease detection and grading, such as lung, breast and brain cancers [1], [2], [3], [4]. However, manual assessment is labor expensive, time consuming, subjective and error-prone. To reduce the workload of pathologists and improve the objectivity of image analysis, computer aided diagnosis (CAD) systems including image processing and modern machine learning techniques have been widely applied to histopathology image computing. Generally, CAD systems can be roughly classified into two categories: classifier-based CAD and content-based image retrieval (CBIR) [5] [6], [7]. Compared to classifier-based CAD that directly provides diagnosis results or grading scores, CBIR can not only be utilized to classify query images but also retrieve and visualize images with the most similar morphological profiles [8], [9]. Therefore, CBIR techniques have attracted considerable attention for histopathology image analysis [10], [11], [12], [13], [14], [15], [16].

Although traditional CBIR systems have exhibited their advantages on providing pathologists with diagnosis support in visualizing relevant images and diagnosis information, most of them are suitable for disease diagnosis with only tens or hundreds images and fail to tackle large-scale datasets

due to the computational efficiency and storage costs. However, the large number of annotated medical images might reduce the semantic gap, which characterizes the difference between image representation and label (disease) description, between images and diagnosis information with modern data-driven methods [14], [17], [18], [19], [20]. To handle large-scale image data, hashing-based retrieval methods have become attractive [21], [22], [23], because hashing can encode the high-dimensional data into compact binary codes with maintaining the similarity among neighbors [24], [25], leading to significant gains in both computation and storage [26], [27], [28].

Based on whether employing semantic information, hashing methods can be classified into two groups: (i) unsupervised hashing that aims to explore the intrinsic structure of data to maintain the similarity among neighbors without any semantic information; (ii) supervised hashing that utilizes semantic information to produce binary codes. Due to the semantic gap, supervised hashing is more preferred in histopathology image analysis. Additionally, since deep learning methods [29] [30] have powerful ability to learn features from large-scale raw image data, some hashing methods [31], [32] utilize deep learning architectures to learn binary representations of images. Although these deep hashing methods have show great strength to encode raw images, they usually have one or two following shortcomings: (i) learning image features and binary codes individually, which usually exhibit inferior performance to simultaneous learning [33]; (ii) emphasizing the distinction between images among different classes but neglecting the relevance order within the same class. This would cause image retrieval to become a classification problem [34].

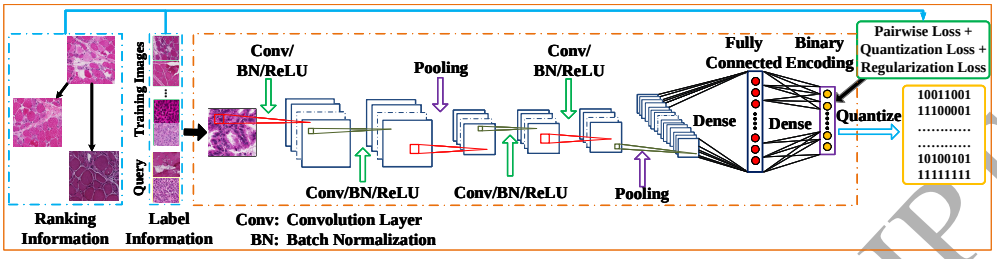


Fig 1: The flowchart of the proposed deep hashing framework. Binary encoding layer is to encode the extracted features obtained from the fully connected layer into binary codes.

Motivated by aforementioned observations, in this paper, we propose a novel pairwise based deep ranking hashing (PDRH) algorithm that can extract features from images and learn their binary representations simultaneously. Meanwhile, it can preserve the inter-class difference for image classification and maintain the intra-class relevance order for image retrieval. To achieve these goals, we first design a pairwise matrix based on image labels and their relevance order within the same class. Then we propose an objective function to learn binary representations of images. Next, we design a supervised deep learning framework by incorporating the proposed objective function to simultaneously learn features and corresponding binary codes. For clarity, we illustrate the flowchart of our framework in Fig 1. Extensive experiments on thousands of histopathology images demonstrate the effectiveness and efficiency of our proposed framework.

2. Related Work

Based on the usage of semantic information, supervised hashing methods can be roughly grouped into three categories: point-wise, multi-wise and pairwise.

Point-wise and multi-wise based hashing: Point-wise based hashing, like supervised discrete hashing (SDH) [28] and supervised quantization (SQ) [34], formulates the searching as a classification problem to learn binary codes. Their objective functions are usually developed on the basis of a regression model. Multi-wise (ranking) based hashing aims to learn hashing functions to project original high-dimensional data into a binary space and meanwhile maximize the agreement of similarity orders over more than two items. Several popular algorithms are: triplet ranking hashing (TRH) [35] that proposes a triplet ranking loss function based on the pairwise hinge loss; ranking supervision hashing (RSH) [36] that incorporates the ranking triplet information into a listwise matrix to learn binary codes; ranking preserving hashing (RPH) [37] that directly optimizes Normalized Discounted Cumulative Gain (NDCG) [38] to learn binary codes with high ranking accuracy. These algorithms cannot learn features and binary codes simultaneously, and later deep semantic ranking based hashing (DSRH) [39], adding a adaptive weight into a triplet hinge loss function, has been proposed to handle the multi-label retrieval problem.

Pairwise based hashing is to utilize the element-wise product of two binary vectors to preserve the Hamming affinity of data pairs. Several popular pairwise based hashing methods used in pathology image analysis are: kernel based supervised hashing (KSH) [26] that leverages the Hamming distance between pairs and utilizes kernels to explore the nonlinear structure hidden in data; joint kernel graph hashing (JKGH) [12] that builds a graph to preserve the relationship of pairs, and constructs and weights sub-kernels based on each feature of data points; unlike KSH and JKGH using a sym-

metric relaxation strategy and a subsequent threshold to learn binary codes, kernel based supervised discrete hashing (KSDH) [40] utilizes asymmetric relaxation to directly learn binary codes in order to reduce the accumulated quantization errors between discrete and continuous matrices. These pairwise and pointwise based hashing algorithms often produce binary codes after obtaining features extracted by using GIST [41], HOG [42], SIFT [43] or convolutional neural network (CNN) [29], which might decrease their retrieval accuracy due to learning features and their binary representation individually. To further improve the searching performance, a few deep hashing models [31], [32], [33] have been proposed to conduct simultaneous learning.

Since point-wise based hashing learns binary codes by formulating the searching into a classification problem, it usually neglects the similarity order among neighbors. Additionally, multi-wise based hashing is expensive to construct the triplet loss for large-scale training data, e.g., the time complexity is $\mathcal{O}(n^3)$ for n training images. Moreover, most of current pairwise based deep learning hashing algorithms [31], [32] often ignore the relevance order of images within the same classes. To alleviate these issues, in this paper, we propose a pairwise-based deep ranking hashing framework to simultaneously learn feature representation and binary codes by employing a deep learning framework and a pairwise matrix to describe the difference and relevance among images, with the time complexity $\mathcal{O}(n^2)$ building the pairwise matrix.

3. Methods

3.1. Definitions and Notations

Given data $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n] \in \mathbb{R}^{n \times d}$, where n and d are the number of data points and dimensions, respectively. Suppose these data have c classes, with each class containing n_k ($\sum_{k=1}^{c} n_k = n$) data points. Let $(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{M}$ if $\mathbf{x}_i$ and $\mathbf{x}_j$ are in the same class; otherwise, $(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{C}$, where $\mathcal{M}$ and $\mathcal{C}$ represent the neighbor-pair and nonneighbor-pair sets, respectively. Assume that a data point $\mathbf{x} \in \mathbb{R}^d$ belongs to the k-th class and it has different relevance to the data points in the same class, the relevance list can be written as:

$$r(\mathbf{x}, \mathbf{X}_k) = \left\{ r_1^k, r_2^k, \cdots, r_{n_k}^k \right\}, \tag{1}$$

where $\mathbf{X}_k$ is a set containing all data points belonging to the k-th class, $r_j^k > 0$ represents the relevance of data point $\mathbf{x}_j^k$ to $\mathbf{x}$, and $r_j^k > r_l^k$ means that $\mathbf{x}$ is more similar to $\mathbf{x}_j^k$ than that to $\mathbf{x}_l^k$.

Hashing is to encode the high-dimensional data into a set of compact binary codes. Specifically, for a data point $\mathbf{x}$, its k-th hashing function is defined as:

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

where $f(\mathbf{x}) \in \mathbb{R}^p$ is row vector representing p features extracted from $\mathbf{x}$, $\mathbf{a}_k \in \mathbb{R}^p$ is a column vector to project the high-dimensional features into a low-dimensional space and b_k is a basis. In this paper, we define $h_k(\mathbf{x}) = 1$ if $f(\mathbf{x})\mathbf{a}_k + b_k \geq 0$; otherwise, $h_k(\mathbf{x}) = -1$. Let m-bit hash codes of $\mathbf{x}$ be $code_m(\mathbf{x}) = [h_1, h_2, \cdots, h_m]$, and then we can obtain $-m \leq code_m(\mathbf{x}_i) \circ code_m(\mathbf{x}_j) \leq m$. For three data points $\mathbf{x}_i$, $\mathbf{x}_j$ and $\mathbf{x}_k$, if $\mathbf{x}_i$ is more similar to

$\mathbf{x}_j$ than $\mathbf{x}_k$, we have $code_m(\mathbf{x}_i) \circ code_m(\mathbf{x}_j) > code_m(\mathbf{x}_i) \circ code_m(\mathbf{x}_k)$. In oder to distinguish data points belonging to different classes, we set $code_m(\mathbf{x}_i) \circ code_m(\mathbf{x}_j) > 0$ when $(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{M}$ and $code_m(\mathbf{x}_i) \circ code_m(\mathbf{x}_j) < 0$ when $(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{C}$. Considering the intra-class relevance of data points, we define a pairwise matrix $\mathbf{S} \in \mathbb{R}^{n \times n}$ to describe the relationship of data pairs as follows:

$$s_{ij} = \begin{cases} r(\mathbf{x}_i, \mathbf{x}_j) & (x_i, x_j) \in \mathcal{M}, \\ -\gamma & (x_i, x_j) \in \mathcal{C}, \end{cases} \tag{3}$$

where $0 < r(\mathbf{x}_i, \mathbf{x}_j) \leq m$ is the relevance between $\mathbf{x}_i$ and $\mathbf{x}_j$, and $0 < \gamma \leq m$ is a constant. Although the largest γ can be m, empirically, we choose a relatively small γ in order to loose the constraint of hashing codes. In this paper we set $\gamma = 1$.

3.2. Problem Formulation

In this subsection, we propose a novel objective function to learn binary codes. The proposed objective function contains three main parts: a pairwise loss function, a quantization loss term and a regularization loss term.

Hashing aims to learn compact binary codes to preserve the relations among original high-dimensional data. Since $code_m(\mathbf{x}) = [h_1, h_2, \cdots, h_m]$, $-m \leq code_m(\mathbf{x}_i) \circ code_m(\mathbf{x}_j) \leq m$, and $-\gamma \leq s_{ij} \leq r_{max}$, similar to [25], the objective function can be intuitively written as:

$$\min_{\mathbf{H}} \frac{1}{4} \left\| \frac{r_{max}}{m} \mathbf{H}\mathbf{H}^T - \mathbf{S} \right\|_F^2, \tag{4}$$

where r_{max} is the maximum element in $\mathbf{S}$, $\mathbf{H} = sgn(f(\mathbf{X})\mathbf{A} + \mathbf{1}_n\mathbf{b})$, $\mathbf{H} \in \{-1, 1\}^{n \times m}$, $\mathbf{A} \in \mathbb{R}^{p \times m}$, $\mathbf{b} \in \mathbb{R}^m$ and $\mathbf{1}_n \in \mathbb{R}^n$ is a column vector with all elements being one.

Unfortunately, Eq. (4) is non-differential and thus it is difficult to be directly solved. To learn the projection matrix $\mathbf{A}$, we relax $\mathbf{H} = sgn(f(\mathbf{X})\mathbf{A} + \mathbf{1}_n\mathbf{b}) \in \{-1, 1\}^{n\times m}$ into $\mathbf{Y} = tanh(f(\mathbf{X})\mathbf{A} + \mathbf{1}_n\mathbf{b}) \in [-1, 1]^{n\times m}$ based on the following observations: (1) $[-1, 1]^{n\times m}$ is the closest convex region to the non-convex region $\{-1, 1\}^{n\times m}$; (2) $\mathbf{Y} = tanh(f(\mathbf{X})\mathbf{A} + \mathbf{1}_n\mathbf{b})$ is differentiable with respect to $\mathbf{A}$ and $\mathbf{b}$, while $\mathbf{H} = sgn(f(\mathbf{X})\mathbf{A} + \mathbf{1}_n\mathbf{b})$ is non-differentiable due to the non-smooth function $sgn(\cdot)$. Then Eq. (4) can be reformulated as:

$$\begin{aligned} J_1 &= \min_{\mathbf{A},\mathbf{b}} \tfrac{1}{4} \left\| \tfrac{r_{max}}{m}\mathbf{Y}\mathbf{Y}^T - \mathbf{S} \right\|_F^2, \\ s.t.\ \ &\mathbf{Y} = tanh(f(\mathbf{X})\mathbf{A} + \mathbf{1}_n\mathbf{b}), \end{aligned} \tag{5}$$

which is a pairwise loss function that preserves the semantic information into $\mathbf{Y}$. Because a large accumulated quantization error between $\mathbf{H}$ and the relaxed $\mathbf{Y}$ will decrease the retrieval accuracy, we add a quantization loss term $J_2 = \frac{\lambda_1}{2} \|\mathbf{H} - \mathbf{Y}\|_F^2$ into Eq. (5) to make the projection matrix $\mathbf{A}$ and the vector $\mathbf{b}$ reduce the accumulated error, where λ_1 is a weight coefficient. Furthermore, since the variance of the projection matrix $\mathbf{A}$ is important to obtain a robust and stable solution, we also incorporate a regularization loss term $J_3 = \frac{\lambda_2}{4} \left\| \mathbf{A}^T\mathbf{A} - \mathbf{I}_m \right\|_F^2$ into Eq. (5), where $\mathbf{I}_m \in \mathbb{R}^{m\times m}$ is an identity matrix and λ_2 is a regularization coefficient. Therefore, Eq. (5) becomes:

$$\begin{aligned} J &= \min_{\mathbf{A},\mathbf{b}} \tfrac{1}{4} \left\| \tfrac{r_{max}}{m}\mathbf{Y}\mathbf{Y}^T - \mathbf{S} \right\|_F^2 + \tfrac{\lambda_1}{2} \|\mathbf{H} - \mathbf{Y}\|_F^2 + \tfrac{\lambda_2}{4} \left\| \mathbf{A}\mathbf{A}^T - \mathbf{I}_m \right\|_F^2, \\ &s.t.\ \ \mathbf{Y} = tanh(f(\mathbf{X})\mathbf{A} + \mathbf{1}_n\mathbf{b}), \mathbf{H} = sgn(\mathbf{Y}), \end{aligned} \tag{6}$$

which is our proposed objective function. In Eq. (6), λ_1 is mainly used to balance the preserved semantic information in $\mathbf{Y}$ and the accumulated errors between $\mathbf{H}$ and $\mathbf{Y}$, i.e. the larger λ_1, the smaller accumulated errors yet the less preserved semantic information; λ_2 is to control the variance of $\mathbf{A}$,

and a too large λ_2 might increase the accumulated errors and decrease the preserved semantic information.

3.3. Network Optimization

In this subsection, we will show the optimization procedure of the proposed objective function Eq. (6) embedded in a CNN architecture.

Suppose that the proposed network contains L layers with parameters $\left(\mathbf{A}^l, \mathbf{b}^l\right)_{l=1}^{L}$, where $\mathbf{A}^l$ denotes the weight connection between the $(l-1)$-th and l-th layers, and $\mathbf{b}^l$ represents the bias in the l-th layer. The previous $L-1$ layers are provided by a CNN architecture (please refer to Fig 1). Eq. (6) is the objective function of the L-th (binary encoding) layer after the $L-1$-th (fully connected) layer, and thus $\mathbf{A}^L = \mathbf{A}$ and $\mathbf{b}^L = \mathbf{b}$. The output of the l-th layer is:

$$\mathbf{Z}^l = \sigma(\mathbf{Z}^{l-1}\mathbf{A}^l + \mathbf{1}_n\mathbf{b}^l), \tag{7}$$

where $\sigma(\cdot)$ represents the activation function. Hence, the parameters $f(\mathbf{X})$ and $\mathbf{Y}$ in Eq. (6) are equivalent to $\mathbf{Z}^{L-1}$ and $\mathbf{Z}^L$ in the network, respectively.

To calculate the gradients of the parameters $\left(\mathbf{A}^l, \mathbf{b}^l\right)_{l=1}^{L}$, we need to first calculate the partial derivatives $\frac{\partial J}{\partial \mathbf{A}^L}$ and $\frac{\partial J}{\partial \mathbf{b}^L}$ in the L-th layer as follows:

$$\frac{\partial J}{\partial \mathbf{A}^L} = \frac{\partial J_1}{\partial \mathbf{Z}^L}\frac{\partial \mathbf{Z}^L}{\partial \mathbf{A}^L} + \frac{\partial J_2}{\partial \mathbf{Z}^L}\frac{\partial \mathbf{Z}^L}{\partial \mathbf{A}^L} + \frac{\partial J_3}{\partial \mathbf{A}^L} = \mathbf{Z}^{L-1T}\Delta^L + \lambda_2\mathbf{A}^L(\mathbf{A}^{LT}\mathbf{A}^L - \mathbf{I}_m) \tag{8a}$$

$$\frac{\partial J}{\partial \mathbf{b}^L} = \frac{\partial J_1}{\partial \mathbf{Z}^L}\frac{\partial \mathbf{Z}^L}{\partial \mathbf{b}^L} + \frac{\partial J_2}{\partial \mathbf{Z}^L}\frac{\partial \mathbf{Z}^L}{\partial \mathbf{b}^L} = \frac{1}{n}\mathbf{1}_n\Delta^L \tag{8b}$$

where $\Delta^L = (\frac{r_{max}}{m}(\frac{r_{max}}{m}\mathbf{Z}^L\mathbf{Z}^{LT} - \mathbf{S})\mathbf{Z}^L + \lambda_1(\mathbf{H} - \mathbf{Z}^L)) \bigodot (\mathbf{1}_n\mathbf{1}_m^T - \mathbf{Z}^L \bigodot \mathbf{Z}^L)$, and $\bigodot$ denotes element-wise multiplication.

The partial derivatives $\frac{\partial J}{\partial \mathbf{A}^l}$ and $\frac{\partial J}{\partial \mathbf{b}^l}$ in the l-th ($l < L$) layer are calculated as follows:

$$\frac{\partial J}{\partial \mathbf{A}^l} = \frac{\partial J}{\partial \mathbf{Z}^l}\frac{\partial \mathbf{Z}^l}{\partial \mathbf{A}^l} = \mathbf{Z}^{l-1T}\Delta^l, \tag{9a}$$

$$\frac{\partial J}{\partial \mathbf{b}^l} = \frac{\partial J}{\partial \mathbf{Z}^l}\frac{\partial \mathbf{Z}^l}{\partial \mathbf{b}^l} = \frac{1}{n}\mathbf{1}_n\Delta^l, \tag{9b}$$

where $\Delta^l = \Delta^{l+1}\mathbf{A}^{l+1T} \bigodot \sigma'(\mathbf{Z}^{l-1}\mathbf{A}^l + \mathbf{1}_n\mathbf{b}^l)$, and $\sigma'(\mathbf{Z}^{l-1}\mathbf{A}^l + \mathbf{1}_n\mathbf{b}^l)$ denotes the derivative of $\mathbf{Z}^l$.

The parameters $\left(\mathbf{A}^l, \mathbf{b}^l\right)_{l=1}^{L}$ are updated by using the gradient descent algorithm as follows:

$$\mathbf{A}^l = \mathbf{A}^l - \eta\frac{\partial J}{\partial \mathbf{A}^l}, \tag{10a}$$

$$\mathbf{b}^l = \mathbf{b}^l - \eta\frac{\partial J}{\partial \mathbf{b}^l}. \tag{10b}$$

For better illustration, we present the whole procedure of PDRH in Algorithm 1.

3.4. Implementation Details

We implement the proposed PDRH with Keras [44] and TensorFlow [45]. Our network architecture is displayed in Table 1. It contains 4 convolutional, 4 batch normalization [46], 4 ReLU [47] and 2 max-pooling layers, 1 fully connected (FC) layer and following by 1 binary layer. The convolutional layers consist of 32, 32, 64 and 64 kernels with size 3×3. In pooling layers, each window has the size 2×2 with stride 2 and 0 padding. The fully connected layer and the binary layer contain 512 and m nodes, respectively, where m is the length of binary codes. Here, we respectively set m to be 8, 16 and 32. Note that we adopt a relatively simple neural network due to the limitation of available computation resources. PDRH is easy to be employed

Algorithm 1: PDRH

Input: Training data $\mathbf{X}$, pairwise matrix $\mathbf{S}$, parameters λ_1 and λ_2, learning rate η, number of layers L, number of iterations N.

Output: $\{\mathbf{A}^l, \mathbf{b}^l\}_{l=1}^{L}$.

Optimization with back propagation

```
for i = 1 to N do
    for l = 1 to L do
        Perform feedforward computation for Z^l using Eq. (7) for other layers
        and Eq. (6) for the L-th layer;
    end
    for l = L to 1 do
        Calculate the gradients based on Eq. (8) or Eq. (9);
    end
    for l = L to 1 do
        Update the parameters based on Eq. (10);
    end
end
```

in more complicated neural works and obtain better binary representations for images.

Similar to [48], we initialize the weights in our network with uniform distribution scaled by the square root of the number of inputs and neurons, and set the mini-batch size to be 32. We train the network 50 epochs by using the ADADELTA optimization algorithm [49], since it does not require a specific hyperparameter tuning.

Table 1: Configuration of the proposed network for input images with the size $128 \times 128 \times 3$. (Note that the batch normalization followed by the ReLU layer are in the middle of two convolution layers or the convolution and max-pooling layers.)

Type	Filter size/stride	Output size
Convolution	$3 \times 3 \times 3 \times 32/1$	$126 \times 126 \times 32$
Convolution	$3 \times 3 \times 32 \times 32/1$	$124 \times 124 \times 32$
Pool	$2 \times 2/2$	$62 \times 62 \times 32$
Convolution	$3 \times 3 \times 32 \times 64/1$	$60 \times 60 \times 64$
Convolution	$3 \times 3 \times 64 \times 64/1$	$58 \times 58 \times 64$
Pool	$2 \times 2/2$	$29 \times 29 \times 64$
FC	-	512
Binary	-	m

4. Experiments and Analysis

To evaluate the proposed algorithm PDRH, we conduct experiments on an image dataset including histopathological skeletal muscle and lung cancer images, which are stained with Haematoxylin and eosin (H&E). The skeletal muscle images contain two major classes of Idiopathic Inflammatory Myopathy (IIM), Polymyositis (PM) and Dermatomyositis (DM). The lung cancer images are with two types of diseases, adenocarcinoma (AC) and squamous cell carcinoma (SC). We collect and crop 5,256 (2,572 PM and 2,678 DM) skeletal muscle images corresponding to 41 individual subjects from the Medical College of Wisconsin Neuromuscular Laboratory (MCWNL), and select and crop 2,904 (1,456 AC and 1,448 SC) lung cancer images of 126 patients from The Cancer Genome Altas (TCGA). The skeletal muscle images are captured at a 40× objective with pixel resolution of 0.25 micron. All ground-

truth are provided by MCWNL. All of the muscle images are analyzed by 3 independent pathologists and a final label is assigned to each image based on the common consensus among them. The lung cancer images are downloaded from the The Cancer Genome Atlas (TCGA) Data Portal. TCGA consists of a collection of cancer specimens with clinical information about participants including metadata about the samples, histopathology slide images from sample portions and molecular information derived from the samples. The lung cancer images are supervised by National Cancer Institute (NCI) and National Human Genome Research Institute (NHGRI) and are freely available to researchers. Here, we randomly partition all images into training and testing sets with a (approximate) ratio 3:1. Specifically, we collect 6,128 images for training, including 3,952 (2,010 PM and 1,942 DM) skeletal muscle images and 2,176 (1,092 AC and 1,084 SC) lung cancer images, and the remaining 2,032 (562 PM, 736 DM, 364 AC and 364 SC) images are used for testing. In our experiments, the RGB raw image data are directly used as input for all deep hashing methods, and they are wrapped to patches with a size of 128×128 before inputting to the learning pipeline. Moreover, for images in the same class, we calculate their Euclidean distance between any two images, and then divide these images into eight subsets based on the distance. The relevance of the images in the subset with the smallest distance is 8 and that with the largest distance is 1. Note that in practice the relevance can be defined based on the applications.

4.1. Experimental Setting

We compare PDRH against eight state-of-the-art non-deep hashing methods, including six non-ranking hashing algorithms, spectral hashing (SH) [24],

KSH [26], column sampling based discrete supervised hashing (COSIDISH) [50], SDH [28], KSDH [40] and asymmetric discrete graph hashing (ADGH) [51], as well as two ranking hashing algorithms RSH [36] and RPH [37], and two popular deep hashing algorithms, convolutional neural network hashing (CNNH) [32] and deep learning of binary hash codes (DLBHC) [31]. For non-deep hashing algorithms, we extract the holistic high-dimensional features from the whole image as the input. Specifically, we first detect scale-invariant feature transform (SIFT) key points from the whole image and then employ SIFT to extract features around these key points. Then, we encode these features into 2,000-dimensional histograms using the bag-of-words (BoW) method [6]. For comparison, we also utilize the output of the fully connected layer of PDRH as the input for the non-deep hashing methods, and these deep learning features are named DLF. In PDRH, we empirically set the three essential parameters $\gamma = 1$, $\lambda_1 = r_{max}$ and $\lambda_2 = 0.1$ in this experiment. To evaluate the performance of hashing algorithms, we employ five main criterion: classification accuracy, MAP, precision-recall (PR) curve, REL and NDCG. For one query image, its REL and NDCG scores are calculated as:

$$REL = \frac{1}{q}\sum_{i=1}^{q} r_i, \tag{11}$$

where q is the number of retrieved samples, and r_i is the relevance of the i-th nearest neighbors to the query image.

$$NDCG = \frac{1}{Z}\sum_{i=1}^{q} \frac{2^{r_i}-1}{log(i+1)}, \tag{12}$$

where Z is a constant to make the maximum of NDCG to be one.

4.2. Experimental Results

To calculate the retrieval accuracy of various hashing algorithms, we first adopt each algorithm to encode training and query images into binary codes, and then utilize the binary codes to calculate the Hamming distance between query and training images. Next, we can select the nearest training images for each query image based on their Hamming distance. Finally, we can compute the retrieval accuracy of each query image based on the corrected selected training images. We calculate the average value of all query images based on the five criterion and report them in Tables 2-3, and Figs 2-5.

Table 2 shows the average retrieval accuracy of all query images, including classification accuracy and MAP of various methods, e.g., the non-deep algorithms SH, KSH, SDH, COSDISH, KSDH, ADGH, RSH, RPH, and deep hashing algorithms CNNH, DLBHC and PDRH, on 8-, 16- and 32-bit hashing codes. As we can see, the non-deep hashing algorithms with DLF achieve significantly better performance than that with BoW. Additionally, COSIDISH, KSDH and ADGH with DLF outperform the deep hashing algorithms CNNH and DLBHC. PDRH obtains higher accuracy (97.49%) and MAP (97.49 % and 97.33 %) than COSIDISH, KSDH and ASDH with DLF at 8-bit, and they have similar performance (the difference is within 0.5%) at 16- and 32-bit.

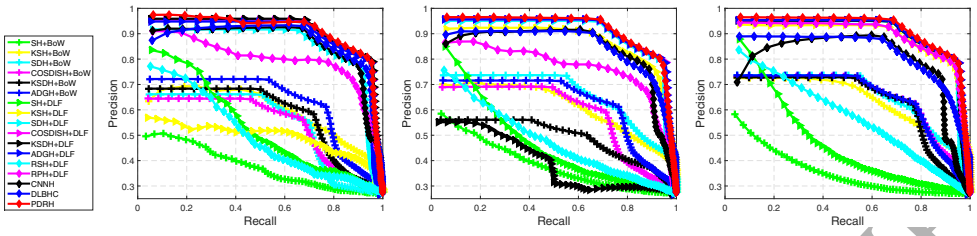
Fig 2 displays the PR curve of various algorithms at 8-, 16- and 32-bit. It suggests that with DLF features, the non-deep algorithms, SH, KSH, SDH, COSDISH, KSDH and ADGH outperform those with BoW features. At 8-bit, PDRH achieves better performance than the other algorithms when the recall is smaller than 0.3; At 16- and 32-bit, PDRH can obtain the best retrieval

Table 2: Retrieval performance (%) measured as classification accuracy and MAP with the top 100 and 500 returned neighbors, respectively.

Method	Accuracy			MAP (Top 100)			MAP (Top 500)		
	8	16	32	8	16	32	8	16	32
SH[24]+BoW	49.31	63.48	69.73	54.86	65.16	65.38	55.22	57.95	56.16
KSH[26]+BoW	64.22	75.30	77.66	64.19	72.33	73.55	71.09	71.56	73.32
SDH[28]+BoW	66.14	73.67	73.67	66.14	73.67	73.67	66.14	73.67	73.67
COSDISH[50]+BoW	64.42	69.29	73.08	68.22	70.63	76.26	68.68	71.30	76.44
KSDH[40]+BoW	68.36	66.10	72.83	68.36	66.10	72.83	68.36	66.10	72.83
ADGH[51]+BoW	72.15	71.56	73.57	72.15	71.56	73.57	72.15	71.56	73.57
SH[24]+DLF	85.73	90.26	92.57	87.25	88.86	91.42	83.98	80.08	82.56
KSH[26]+DLF	57.48	92.86	93.65	57.02	91.90	94.40	56.96	93.22	94.16
SDH[28]+DLF	91.68	95.08	94.98	91.68	95.08	94.98	91.68	95.08	94.98
COSDISH[50]+DLF	96.56	**96.70**	**96.41**	96.69	**96.70**	**96.49**	96.70	**96.70**	**96.50**
KSDH[40]+DLF	96.90	96.15	96.11	96.90	96.15	96.11	96.90	96.15	96.11
ADGH[51]+DLF	96.56	**96.70**	**96.41**	96.69	**96.70**	**96.49**	96.70	**96.70**	**96.70**
RSH[36]+DLF	80.27	80.36	88.63	83.16	81.38	87.65	78.88	74.02	81.33
RPH [37]+DLF	92.22	85.29	94.24	91.56	88.01	95.06	92.39	88.55	94.61
CNNH[32]	92.86	92.32	66.19	88.48	82.57	67.14	92.81	90.01	85.03
DLBHC[31]	94.49	92.13	90.80	82.92	88.30	89.44	90.86	91.50	89.27
PDRH	**97.49**	**96.75**	**96.65**	**97.49**	**96.80**	**96.66**	**97.33**	**96.65**	**96.52**

performance among all hashing algorithms. Fig 3 presents the precision of various algorithms at all 8, 16 and 32 bits using Hamming radius $r = 1, 2$ and 3. When $r = 1$ or 2, PDRH attains the best precision at 8- and 16-bit; When $r = 3$, PDRH outperforms other algorithms at 16- and 32-bit.

Table 3 presents the ranking performance (REL and NDCG) of the proposed PDRH and the comparative algorithms with DLF features at 16-bit on 5, 10 and 50 returned neighbors. PDRH obtains the highest REL score



@8-bit @16-bit @32-bit

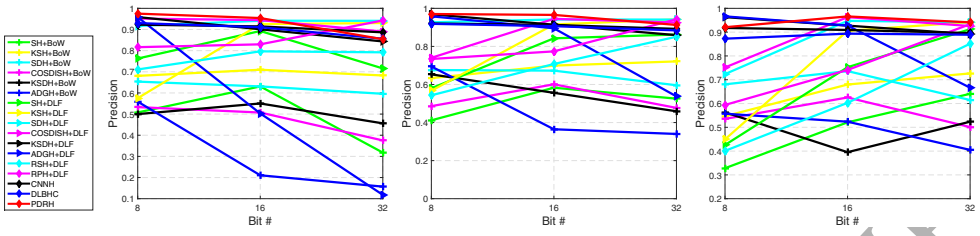
Fig 2: PR curve of various algorithms at different number of bits.

5.74, 5.74 and 5.76 on 5, 10 and 50 returned neighbors, respectively. Fig 4 shows the relevance of above algorithms on 5, 10, 20, 50 and 100 retrieved images at all the three bits. It further illustrates that PDRH achieves higher REL scores than the other algorithms on different returned neighbors or bits. Table 3 also demonstrates that with 16-bit hashing codes, PDRH, SH+DLF and RSH+DLF obtain the best NDCG score than the others when 5 images are retrieved; With 10 and 50 images returned, PDRH achieves the highest NDCG score 0.49 and 0.51, respectively. Fig 5 shows the NDCG score of various algorithms on 5, 10, 20, 50 and 100 retrieved images. At 8-bit, PDRH achieves slightly worse score than RSH+DLF, while it significantly outperforms the others; At 16-bit, PDRH and RSH+DLF achieve similar NDCG scores on 5, 10 and 20 retrieved images, and PDRH obtains the best score on 50 and 100 returned neighbors; At 32-bit, PDRH outperforms the others on 10, 20, 50 and 100 returned samples.

4.3. Discussion and Analysis

Based on experimental results in Tables 2-3 and Figs 2-5, we have the following observations:

- Non-deep hashing algorithms SH, KSH, SDH, COSDISH, KSDH and



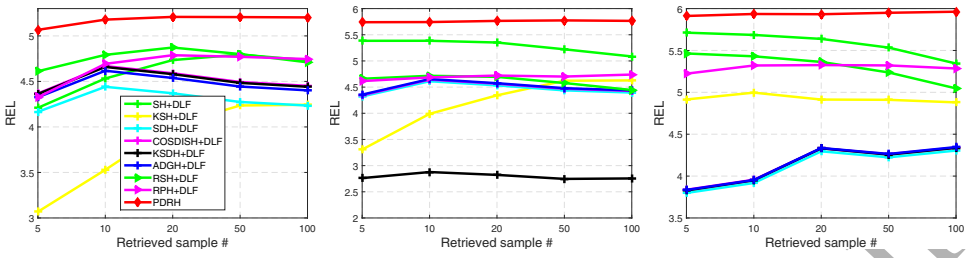
@r=1 @r=2 @r=3

Fig 3: Precision vs bit using various algorithms on different Hamming radiuses (@r).

Table 3: Ranking performance using REL and NDCG with 16-bit hashing codes on 5, 10 and 50 retrieved images.

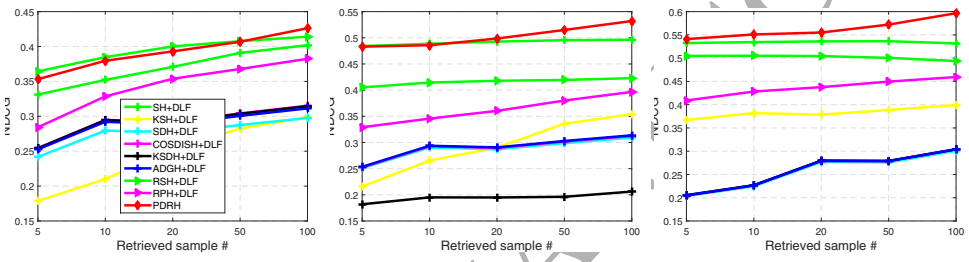
Method	REL			NDCG		
	5	10	50	5	10	50
SH[24]+DLF	5.38	5.38	5.22	**0.48**	0.48	0.49
KSH[26]+DLF	3.31	3.99	4.62	0.22	0.27	0.34
SDH[28]+DLF	4.32	4.61	4.44	0.25	0.29	0.30
COSDISH[50]+DLF	4.34	4.64	4.47	0.25	0.29	0.30
KSDH[40]+DLF	3.76	3.87	3.82	0.21	0.22	0.28
ADGH[51]+DLF	4.36	4.65	4.57	0.25	0.29	0.29
RSH[36]+DLF	5.38	5.39	5.22	**0.48**	0.48	0.49
RPH[37]+DLF	4.62	4.68	4.70	0.33	0.35	0.38
PDRH	**5.74**	**5.74**	**5.76**	**0.48**	**0.49**	**0.51**

ADGH with DLF obtain better retrieval performance including classification accuracy, MAP and PR curve than those with BoW features. PDRH containing a convolutional neural network has powerful ability to extract features from the original histopathology images with preserving the significant semantic information. By contrast, BoW ex-



@8-bit @16-bit @32-bit

Fig 4: REL with different number of retrieved images.



@8-bit @16-bit @32-bit

Fig 5: NDCG with different number of retrieved images.

plores the intrinsic structure to extract features with maintaining the significant information without any supervision, and thus it might neglect some significant semantic information.

- Although the non-deep hashing algorithms including the non-rank and rank hashing can perform well with DLF features, they usually achieve similar or inferior performance to PDRH. The main possible reason is that PDRH extracts features and learns binary representations simultaneously, leading to better and more stable solutions.

- The non-ranking hashing including KSH, SDH, COSDISH, KSDH and ADGH can deliver fair retrieval performance, while their ranking per-

formance, including REL and NDCG, is relatively poor. This is because their affinity matrices maintain the difference of images among different classes, while they do not consider the similarity order of images within the same classes.

- Although the ranking hashing algorithms, RSH and RPH, can obtain better ranking performance than most of non-ranking hashing, they usually achieve worse retrieval performance. This can be attributed to the fact that RSH and RPH focus on the intra-class difference among images, but they neglect the inter-class difference among images.

5. Conclusion

In this paper, we present a novel pairwise based deep ranking hashing algorithm to simultaneously extract features from histopathology images and learn their binary representations, with preserving the inter-class difference for image classification and maintaining the intra-class relevance order in the same classes. We first define a pairwise matrix to preserve the class and ranking information of images, and then propose a novel objective function to learn binary codes. Then, we incorporate the objective function into a convolutional neural network architecture to simultaneously extract features and learn binary codes. Experiments on muscle and lung cancer images demonstrate the effectiveness and efficiency of the proposed algorithm. Since the objective function approximates the pairwise matrix using a relaxed tanh function, which might generate accumulated quantization errors between the discrete and continuous matrices, in the future, we will work on approximating the pairwise matrix directly using discrete matrices. Moreover, we will

learn a more general and robust deep hashing model by collecting more types of histopathology images.

References

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

[2] L. Yang, W. Chen, P. Meer, G. Salaru, M. D. Feldman, D. J. Foran, High throughput analysis of breast cancer specimens on the grid, in: Proc. Int. Conf. Medical Imag. Comput. Computer-Assist. Interv., pp. 617-625, 2007.

[3] F. Xing and L. Yang, Robust Nucleus/Cell Detection and Segmentation in Digital Pathology and Microscopy Images: A Comprehensive Review, IEEE Reviews in Biomedical Engineering, 9(2016), pp. 234-263.

[4] F. Xing and Y. Xie and L. Yang, An Automatic Learning-Based Framework for Robust Nucleus Segmentation, IEEE Trans. Medical Imag., 35(2)(2016), pp. 550-566.

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

[6] J. C. Caicedo, A. Cruz, F. A. Gonzalez, Histopathology image classification using bag of features and kernel functions, in: Proc. Conf. Artificial Intell. Med. Euro., pp. 126-135, 2009.

[7] A. N. Basavanhally, S. Ganesan, S. Agner, J. P. Monaco, M. D. Feldman, J. E. Tomaszewski, G. Bhanot, A. Madabhushi, Computerized image-based detection and grading of lymphocytic infiltration in HER2+ breast cancer histopathology, IEEE Trans. Biomed. Eng., 57(3)(2010), pp. 642-653.

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

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

[10] X. Zhang, H. Su, L. Yang, S. Zhang, Weighted Hashing with Multiple Cues for Cell-Level Analysis of Histopathological Images, Med. Imag. Anal., (2015), pp. 303-314.

[11] X. Zhang, H. Su, L. Yang, S. Zhang, Fine-grained histopathological image analysis via robust segmentation and large-scale retrieval, in: Proc. Int. Conf. Comput. Vision Pattern Recog., pp. 5361-5368, 2015.

[12] M. Jiang, S. Zhang, J. Huang, L. Yang, D. N. Metaxas, Joint Kernel-Based Supervised Hashing for Scalable Histopathological Image Analysis, in: Proc. Med. Image Comput. Comput. Assist. Interv., pp. 366-373, 2015.

[13] X. Shi, F. Xing, K. Xu, Y. Xie, H. Su, L. Yang, Supervised graph hashing

for histopathology image retrieval and classification, Med. Imag. Anal., 2017.

[14] S. Zhang, D. Metaxas, Large-Scale medical image analytics: Recent methodologies, applications and Future directions, Med. Imag. Anal., 33(2016), pp. 98-101.

[15] X. Zhang, W. Liu, M. Dundar, S. Badve, S. Zhang, Towards large-scale histopathological image analysis: Hashing-based image retrieval, IEEE Trans. Med. Imag., 34(2)(2015), pp. 496-506.

[16] X. Shi, F. Xing, Y. Xie, H. Su, L. Yang, Cell Encoding for Histopathology Image Classification, in Proc. Med. Image Comput. Comput. Assist. Interv., 2017.

[17] A. Wiliem, C. Sanderson, Y. Wong, P. Hobson, R. F. Minchin, B. C. Lovell, Automatic classification of human epithelial type 2 cell indirect immunofluorescence images using cell pyramid matching, Pattern Recog., 47(7) (2014), 2315-2324.

[18] Z. Zhang, Y. Xie, F. Xing, M. McGough, L. Yang, Mdnet: A semantically and visually interpretable medical image diagnosis network, in: Proc. IEEE Conf. Comput. Vision Pattern Recog., pp. 6428-6436, 2017.

[19] Z. Zhang, P. Chen, M. Sapkota, L. Yang, TandemNet: Distilling Knowledge from Medical Images Using Diagnostic Reports as Optional Semantic References, in: Proc. Int. Conf. Medical Image Comput. Computer-Assisted Interv., pp. 320-328, 2017.

[20] Y. Xie, F. Xing, X. Shi, X. Kong, H. Su, L. Yang, Efficient and robust cell detection: A structured regression approach, Med. Image Anal., 2017.

[21] Y. Zhang, H. Lu, L. Zhang, X. Ruan, S. Sakai, Video anomaly detection based on locality sensitive hashing filters. Pattern Recog., 59(2016), 302-311.

[22] Y. L. Lai, Z. Jin, A. B. J. Teoh, B. M. Goi, W. S. Yap, T. Y. Chai, C. Rathgeb, Cancellable iris template generation based on Indexing-First-One hashing, Pattern Recog., 64 (2017), 105-117.

[23] Y. Luo, Y. Yang, F. Shen, Z. Huang, P. Zhou, H. T. Shen, Robust discrete code modeling for supervised hashing, Pattern Recog., (2017)

[24] Y. Weiss, A. Torralba, R. Fergus, Spectral hashing, in: Proc. Neur. Inf. Process. Syst., pp. 1753-1760, 2009.

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

[26] W. Liu, J. Wang, R. Ji, Y. Jiang, S. Chang, Supervised hashing with kernels, in: Proc. Int. Conf. Comput. Vision Pattern Recog., pp. 2074-2081, 2012.

[27] W. Liu, C. Mu, S. Kumar, S. Chang, Discrete graph hashing, in: Proc. Neur. Inf. Process. Syst., pp. 3419-3427, 2014.

[28] F. Shen, C. Shen, W. Liu, H. Shen, Supervised Discrete Hashing, in: Proc. Int. Conf. Comput. Vision Pattern Recog., pp. 37-45, 2015.

[29] L. Yann, Y. Bengio, G. Hinton, Deep learning, Nature, 521(7553)(2015), pp. 436-444.

[30] S. Ding, L. Lin, G. Wang, H. Chao, Deep feature learning with relative distance comparison for person re-identification, Pattern Recog., 48(10) (2015), 2993-3003.

[31] K. Lin, H. Yang, J. Hsiao, C. Chen, Deep learning of binary hash codes for fast image retrieval, in: Proc. Int. Conf. Comput. Vision Pattern Recog. Workshops, pp. 27-35, 2015.

[32] R. Xia, Y. Pan, H. Lai, C. Liu, S. Yan, Supervised Hashing for Image Retrieval via Image Representation Learning, in: Proc. Association. Adv. Artificial Intell., pp. 2, 2014.

[33] H. Lai, Y. Pan, Y. Liu, S. Yan, Simultaneous feature learning and hash coding with deep neural networks, in: Proc. Int. Conf. Comput. Vision Pattern Recog., pp. 3270-3278, 2015.

[34] X. Wang, T. Zhang, G. Qi, J. Tang, J. Wang, Supervised Quantization for Similarity Search, in: Proc. Int. Conf. Comput. Vision Pattern Recog., 2016.

[35] M. Norouzi, D. J. Fleet, R. R. Salakhutdinov, Hamming distance metric learning, in: Proc. Neur. Inf. Process. Syst., pp. 1061-1069, 2012.

[36] J. Wang, W. Liu, A. X. Sun, Y. G. Jiang, Learning hash codes with listwise supervision, in: Proc. Int. Conf. Comput. Vision, pp. 3032-3039, 2013.

[37] Q. Wang, Z. Zhang, L. Si, Ranking Preserving Hashing for Fast Similarity Search, in: Proc. Int. Joint Conf. Artificial Intell., pp. 3911-3917, 2015.

[38] K. Järvelin, J. Kekäläinen, IR evaluation methods for retrieving highly relevant documents, ACM SIGIR, pp. 41-48, 2000.

[39] F. Zhao, Y. Huang, L. Wang, T. Tan, Deep semantic ranking based hashing for multi-label image retrieval, in: Proc. Int. Conf. Comput. Vision Pattern Recog., pp. 1556-1564, 2015.

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

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

[42] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, pp. 886-893, 2005.

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

[44] C. Franois, keras, GitHub repository, `https://github.com/fchollet/keras`, 2015.

[45] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, and others, Tensorflow: Large-scale machine learning on heterogeneous distributed systems, arXiv preprint arXiv:1603.04467, 2016.

[46] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, arXiv preprint arXiv:1502.03167, 2015.

[47] V. Nair, G. E. Hinton, Rectified linear units improve restricted boltzmann machines, in: Proc.Int. Conf. Mach. Learn., pp. 807-814, 2010.

[48] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, Aistats, pp. 249-256, 2010.

[49] M. D. Zeiler, ADADELTA: an adaptive learning rate method, arXiv preprint arXiv:1212.5701, 2012.

[50] W. Kang, W. Li, Z. Zhou, Column sampling based discrete supervised hashing, Association. Adv. Artificial Intell., 2016.

[51] X. Shi, F. Xing, K. Xu, M. Sapkota, L. Yang, Asymmetric Discrete Graph Hashing, in: Proc. Association. Adv. Artificial Intell., 2017.

Xiaoshuang Shi received the B.S. degree in automation from Northwestern Polytechnical University, China, and M.S. degree in automation from Tsinghua University, China, in 2009 and 2013, respectively. From Sep. 2013 to Apr. 2015, he was a Research Assistant in Shenzhen Key Laboratory of Broadband Network & Multimedia, Graduate School at Shenzhen, Tsinghua University, China. Now, he is pursuing a PhD degree in the J. Crayton Pruitt Family Department of Biomedical Engineering at University of Florida, Gainesville, USA. His current research interests include large-scale image retrieval, pattern recognition and medical image analysis.

Manish Sapkota received the Computer Engineering degree from Purbanchal University - Nepal in 2007. He is currently a PhD candidate in Department of Electrical and Computer Engineering at University of Florida, USA. His research interests include computer vision, machine learning, content-based image retrieval and high performance computing

Fuyong Xing received the bachelors degree from Xian Jiaotong University, Xian, China, the M.S. degree from Rutgers University, New Brunswick, NJ, USA, and the Ph.D. degree in electrical and computer engineering from the University of Florida, Gainesville, FL, USA, in 2017. He is currently an Assistant Professor with the Department of Biostatistics and Informatics, Colorado School of Public Health, University of Colorado Denver, Denver, CO, USA. His current research interests include biomedical image computing, imaging informatics, computer vision, machine learning, and deep learning.

Fujun Liu received the Bachelor of Science (B.S.) degree in Communication Engineering from Shandong University, China, and Master of Science (M.S.) degree in Information and Communication Engineering from Chinese

Academy of Sciences, China, in 2007 and 2010, respectively. Dr. Liu obtained his Doctor of Philosophy (Ph.D) degree in Electrical and Computer Engineering at University of Florida in 2017. During his Ph.D study, Dr. Liu's research interests included medical image processing, computer vision, machine learning, deep learning, etc.

Lei Cui is a Ph.D. candidate at Northwest University. His research interests include medical image analysis, machine learning and computer-aided diagnosis.

Lin Yang is an associate professor in the J. Crayton Pruitt Family Department of Biomedical Engineering, the Department of Electrical and Computer Engineering, and the Department of Computer and Information Science and Engineering at University of Florida. He was an assistant professor in the Department of Radiology and Pathology, and graduate faculty in the Department of Biomedical Engineering at Rutgers University from 2009-2011. He was an assistant professor in the Department of Biomedical Informatics and the Department of Computer Science at University of Kentucky from 2011-2014. His major research interest is focused on biomedical image analysis, imaging informatics, computer vision, biomedical informatics, and machine learning. He is also working on high performance computing and computed aided health care and information technologies. He leads the Biomedical Image Computing and Imaging Informatics (BICI2) Lab: http://www.bme.ufl.edu/labs/yang/.