Deep Convolutional Hashing for Low Dimensional Binary Embedding of Histopathological Images

Manish Sapkota, *Student Member, IEEE,* Xiaoshuang Shi, *Student Member, IEEE,* Fuyong Xing, *Member, IEEE,* and Lin Yang, *Member, IEEE*

***Abstract*—Compact binary representations of histopathology images using hashing methods provide efficient approximate nearest neighbor search for direct visual query in large-scale databases. They can be utilized to measure the probability of the abnormality of the query image based on the retrieved similar cases, thereby providing support for medical diagnosis. They also allow for efficient managing of large-scale image databases because of a low storage requirement. However, the effectiveness of binary representations heavily relies on the visual descriptors that represent the semantic information in the histopathological images. Traditional approaches with hand-crafted visual descriptors might fail due to significant variations in image appearance. Recently, deep learning architectures provide promising solutions to address this problem using effective semantic representations. In this paper, we propose a Deep Convolutional Hashing (DCH) method that can be trained "point-wise" to simultaneously learn both semantic and binary representations of histopathological images. Specifically, we propose a convolutional neural network (CNN) that introduces a latent binary encoding (LBE) layer for low dimensional feature embedding to learn binary codes. We design a joint optimization objective function that encourages the network to learn discriminative representations from the label information, and reduce the gap between the real-valued low dimensional embedded features and desired binary values. The binary encoding for new images can be obtained by forward propagating through the network and quantizing the output of the LBE layer. Experimental results on a large-scale histopathological image dataset demonstrate the effectiveness of the proposed method.**

***Index Terms*—Binary hashing, convolutional neural network, deep learning, feature embedding, microscopic images, search and retrieval.**

I. Introduction

Recently, digitized histopathological images are frequently analyzed and interpreted to diagnose and grade different diseases. Unfortunately, the process of manual interpretation is time consuming, challenging and error-prone due to their large size and inter/intra-observer variation [1]–[3]. To address these problems and improve the efficiency of the procedure, different content-based image retrieval (CBIR) methods for automated computer aided diagnosis (CAD) have been proposed [4]–[10]. CBIR aims to directly utilize a visual query to find the most similar cases (nearest neighbors) among the images previously diagnosed and stored in databases. Pathologists could directly utilize these retrieved cases to measure the likelihood of the abnormality of the query image and render better diagnosis. In addition, CBIR systems could also be used for various other applications including effective archiving and management of digitized histopathology images and pathologists' training [11].

In the biomedical image retrieval community, a typical approach to find nearest neighbors is using predefined distance measurement and compact image representations (features/signatures), which are usually real-valued [12], [13]. The nearest neighbor search could be performed with an exhaustive comparison of the query image via each of the samples in the database [7]; nevertheless, this method becomes infeasible as the number of samples in the database increases rapidly. This restricts the scalability of the exhaustive search and makes it unsuitable for practical settings where a large number of images are available. Alternatively, it has been shown that *Approximate nearest neighbor* (ANN) methods could be a sufficient solution in many applications that aim at large-scale information search and retrieval [14]. Subsequently, ANN that utilizes the real-valued image features has been used for CBIR in large-scale biomedical image databases [4], [5]. However, the real-valued image representations become a critical bottleneck, because the storage requirement for these representations increases with the number of samples.

Fortunately, the problem could be addressed with similarity-preserved *hashing methods* for ANN search, which has low storage requirement and high retrieval efficiency [15], [16]. In recent years, many hashing methods, particularly learning-based, have been proposed for both natural and biomedical images [17]–[23]. These hashing techniques map image representations in the original feature space to more compact binary representations in a Hamming space, yet preserving the semantic similarity of the original feature space. Specifically, these techniques aim to generate neighboring binary representations for the data points that are close in the original space, and vice versa. One of the advantages of binary representations is that the fast image search could be carried out by simple Hamming distance (binary pattern matching) measurement, which has a low computation cost.

Supervised hashing methods [6], [18], [19], [24]–[29], which take advantage of supervised information of the images such as class labels, have shown superior performance and great potentials to learn binary code representations. These methods rely on image representations and similarity information. Traditionally, most of them first encode the images as a set of hand-crafted visual descriptors, e.g., scale-invariant

M. Sapkota is with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL, 32611 USA, F. Xing is with the Department of Biostatistics and Informatics, Colorado School of Public Health, University of Colorado Denver, Denver, CO, 80045, USA and X. Shi and L. Yang are with J. Crayton Pruitt Family Department of Biomedical engineering, University of Florida, Gainesville, FL, 32611 USA. E-mail:lin.yang@bme.ufl.edu

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JBHI.2018.2827703, IEEE Journal of Biomedical and Health Informatics

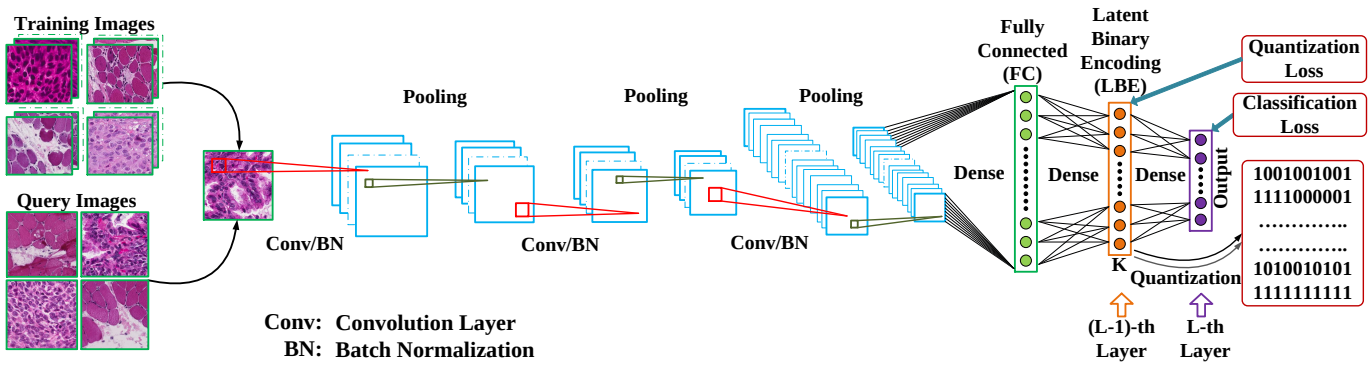


Fig. 1: The network architecture of the proposed DCH method.

feature transform (SIFT) [30], to capture image information before learning binary representations. However, these visual descriptors, which are usually extracted in an unsupervised manner, do not guarantee the accurate representation of semantic information. This limits the performance of many hashing methods. Recently, deep learning based methods have shown that the performance of various vision tasks including hashing could be significantly improved by learning successive non-linear image representations using large datasets [31]–[37]. Most of these deep learning based hashing methods optimize the network based on the semantic "pair-wise" relation, which requires creating a computationally inefficient pair-wise similarity matrix from the label information. Although these methods achieve good retrieval performance, they are limited in scalability due to the high storage and computational requirement of pair-wise optimization problem. Therefore, direct "point-wise" optimization of deep learning methods, similar to other vision tasks, to learn binary codes is preferable since it does not restrict the scalability of these methods.

In this work, we are interested in taking advantage of convolutional neural network (**CNN**) models, which can be trained "point-wise", to generate efficient compact binary codes for histopathological image representation. To this end, we propose a DCH that exploits the CNN structure for binary encoding, as shown in Figure 1. It introduces a LBE layer (between fully connected layers and the classification layer) for low dimensional feature embedding. Unlike many other existing hashing methods that utilize pair-wise inputs, we devise a joint *point-wise* optimization approach with a carefully designed loss function, which could leverage the supervised label information to generate optimized binary codes for linear classification. To guarantee optimal learning of binary codes, we impose two simultaneous constraints at the LBE layer and the top classification layer of the DCH network: (1) the quantization loss between real values and quantized binary values of the LBE is minimized, and (2) the classification loss between original and learned labels is minimized. The joint constraints encourage the real values of the low dimensional feature embedding layer to be close to quantized values while simultaneously improving the discriminative power of the learned binary codes, thereby preserving the semantic structure in the Hamming space. We argue that the learned binary codes are the non-linear low dimensional embedded feature vectors of the original data, and the label information is exploited to easily differentiate these learned binary feature vectors. Finally, binary image representations are obtained by first feedforward propagating through the network and then quantizing the output of the LBE layer. The overall contributions of the paper can be summarized as follows:

- We propose an effective CNN-based method to simultaneously learn binary codes and feature representations. The key ideas are to introduce a binary encoding layer in the CNN architecture and a loss function that generates the binary codes with minimized quantization errors for classification. The overall architecture could be easily implemented such that it could be trained seamlessly with a joint optimization of modified loss functions in a *point-wise* manner. The point-wise learning of binary codes makes it easier to scale the method to large dataset in comparison with traditional pair-wised approaches.
- The proposed DCH method is evaluated on a large-scale histopathological image dataset comprised of skeletal muscle disease and lung cancer specimens, and the superior experimental results demonstrate the effectiveness of the proposed method over other state-of-the-art hashing approaches.

The novelty of the proposed method is to introduce a binary encoding layer into the conventional CNN for binary representation learning and a modified loss function to minimize quantization errors for classification. The proposed framework can be trained "point-wise" and end-to-end with the stochastic gradient descent algorithm.

II. Related Works

A. Hashing methods

Various hashing methods to learn binary representations have been proposed in the literature over the years [15], [16], [20], [38]–[44]. Traditional *data-independent* methods, such as the family of Locality Sensitive Hashing (LSH) [15], [45], [46] that use random projections to generate compact binary codes, in general, require longer binary codes to achieve desired performance. This in-turn introduces huge storage overhead that limits practical applications on large datasets. Therefore, tremendous effort to improve LSH is observed. In particular,

learning-based data-dependent hashing methods, which use available training data to learn the similarity preserved hashing functions, have gained more popularity. Some representative approaches include **unsupervised hashing** (Spectral Hashing (SH) [38], Discrete Graph Hashing (DGH) [44], Iterative Quantization (ITQ) [40]), **supervised/semi-supervised hashing** (Graph Cuts Coding (GCC) [47], CCA-ITQ [40]; Minimum Loss Hashing (MLH) [24], Semi-Supervised Hashing (SSH) [42]), **kernel-based hashing** (Binary Reconstruction Embedding (BRE) [16]; Kernel-Based Supervised Hashing (KSH) [41], Supervised Discrete Hashing (SDH) [20]) and **deep learning based hashing** ([25]–[27], [33], [48], [49]).

The main goal of most learning-based data-dependent hashing methods is to preserve data similarity in the Hamming space. Unsupervised hashing methods aim to learn image representations in the Hamming space and preserve the *Euclidean similarity* defined in the original feature space. For instance, ITQ [40] learns binary codes by imposing the binary constraints to minimize the quantization error to reduce the gap between image representations in the original feature space and binary representations in the Hamming space. The major drawback of these methods is that they cannot effectively eliminate the "semantic gap", which is the difference between high-level semantic meanings of images that are represented by similar low-level features. Therefore, to address these problems, supervised hashing methods leverage semantic labels of the training dataset. MLH [24] uses weakly labelled training data to minimize a hinge-like loss to learn the hashing functions, and SSH [42] minimizes the empirical loss over the labeled training samples and maximizes the variance and independence of individual bits for both labeled and unlabeled training samples. Furthermore, kernel-based hashing methods advocate the usage of kernels to learn the hashing functions in the non-linear space to capture data patterns. One of the most prominent existing methods in this category is SDH [20]. Specifically, SDH jointly optimizes the classification loss and the fitting error of the binary codes using a continuous embedding function to achieve highly discriminative binary representations. However, the kernel-based hashing methods use explicitly defined non-linear functions, which lead to the problem of scalability.

All of the aforementioned methods use hand-crafted image representations, which do not necessarily generalize well in the real-world image data that show redundancy and high-range variance, and thus degrade the retrieval performance. Recently, deep learning based methods, which are non-linear models that can be learned from raw image data, have shown tremendous success in various computer vision tasks. Following along the success, different deep learning based hashing methods have also been proposed [25]–[27], [33], [48]–[50] to simultaneously learn non-linear image representations and compact binary codes. All of these methods train a CNN to learn the binary representations using pairwise similarity matrices. However, the large memory requirement to compute and store these similarity matrices could restrict the amount of data for CNN training. This could degrade the performance of the CNN-based methods and could be a serious drawback. To deal with the issue, the work in [33] provides an alternative online approach to create a similarity matrix during training, but it creates additional computational overhead. The method in [27] suggests direct utilization of label information and point-wise train CNN with binary-like hidden representations as features for image classification tasks. This approach does not require computation of the pairwise similarity, and it enforces the network to learn binary-like outputs to preserve the semantic similarity based on classification performance. However, this approach simply binarizes the real-valued outputs of hidden layers such that it does not guarantee optimal binary codes. Our method builds on top of this approach to address this issue.

B. CAD with Hashing

The importance of CBIR systems for CAD has been recognized [12], [51], [52] for a long time in the biomedical community. CBIR is valuable in medical image analysis, because it provides doctors with diagnostic support for case based and evidence based reasoning, which usually requires studies of archived cases. However, the vast majority of the CBIR methods fall short of scalability and cannot handle the increasing amount of data for practical applications. With the increasing popularity of binary hashing methods, the problem of scalability is greatly addressed, and there is a recent trend to utilize them for the CAD [6], [8], [17], [22], [28], [34], [39], [53]. For instance, authors in [22] have investigated a scalable mammogram retrieval system for breast cancer diagnosis, which uses anchor graph hashing (AGH) method for binary hashing and aggregated features to improve the search accuracy. Similarly, authors in [8] have proposed a CAD system to diagnose masses in mammogram. They have reported a scalable retrieval and diagnosis system to query mammographic regions of interest, which are represented as SIFT features and quantized in a vocabulary tree.

In the context of histopathological images, authors in [6] have proposed a joint kernel-based supervised hashing method to map the multiple high-dimensional features of breast cancer images into much smaller binary features and search in a Hamming space to distinguish between actionable and benign cases. In [28], authors have proposed a modification to KSH to handle online training and generate efficient binary codes. The binary representations are then utilized to diagnose the breast cancer images based on a majority voting. Therefore, the compact binary representations allow for scalable CBIR for medical image analysis. Additionally, compact representations have a low storage requirement that allows to easily store and manage the growing histopathology image collection.

III. Methods

In this section, we first introduce the general binary hashing framework and then formulate our proposed DCH method for binary hashing as a classification problem with a hard binary constraint at the LBE layer. Next, we define a continuous relaxation of the binary constraint and introduce the quantization error in the loss function to improve binary representation, as shown in Figure 1. Finally, we show how to minimize the loss function using the back-propagation algorithm to train the network.

A. Binary Hashing

The goal of hashing methods is to learn compact binary code representations for images such that the codes are discriminative enough for dissimilar data. Formally, given N training samples $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N] \in \mathbb{R}^{d \times N}$, where $\mathbf{x}_n \in \mathbb{R}^d$ is the n-th training sample in $\mathbf{X}$, the hashing methods aim to learn multiple hashing functions that map and quantize the samples into a set of compact binary codes $\mathbf{B} = [\mathbf{b}_1, \ldots, \mathbf{b}_N] \in \{-1, 1\}^{K \times N}$, where $\mathbf{b}_n \in \{-1, 1\}^{K \times 1}$ is a K-bit binary code for $\mathbf{x}_n$. The mapping of $\mathbf{x}_n$ to the k-th binary bit b_{kn} is computed as:

$$b_{kn} = g_k(\mathbf{x}_n) = sgn(f_k(\mathbf{x}_n)) = sgn(\mathbf{w}_k \mathbf{x}_n), \tag{1}$$

where $\mathbf{w}_k \in \mathbb{R}^{1 \times d}$ is a projection vector in k-th hashing function g_k, and $sgn(u) = 1$ if $u > 0$ and $sgn(u) = -1$ otherwise. Therefore, given a projection matrix $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_K] \in \mathbb{R}^{K \times d}$ the mapping of the $\mathbf{x}_n$ is computed as: $f(\mathbf{x}_n) = \mathbf{W}\mathbf{x}_n$ that is binarized to get the binary codes as:

$$g(\mathbf{x}_n) = \mathbf{b}_n = sgn(f(\mathbf{x}_n)). \tag{2}$$

Different hashing methods to learn the projection matrix $\mathbf{W}$ have been proposed. Most of earlier methods suffer from the limitation of linear projections, unscalable non-linear representations or inefficient hand-crafted features. Recently, deep learning based hashing methods have been proposed to learn the non-linear image representations and compact binary codes simultaneously. In this work, we propose an end-to-end CNN based hashing approach, which inserts a binary representation layer between the fully connected and classification layer. We train the network "point-wise" using a set of images and their corresponding labels such that the learned binary image representations can preserve the semantic similarity. The loss function has been designed to encourage the activation of the binary layer to approach discrete values suited for good classification. In this scenario, the optimized network would learn similar binary representations for the data with the same label.

B. DCH

The proposed DCH network consists of consecutive layers of convolution, max-pooling, non-linear transformations, fully connected layers, followed by an LBE and finally a classification layer, as shown in Figure 1. The binary representations for $\mathbf{X}$ in the RGB space is obtained with first forward propagation through the network and then quantization of the output of the LBE layer. Assume that there are L layers in the DCH network and each layer $(l|l = 1, ..., L)$ has m^l units. Then the output of the first hidden layer, for a given sample $\mathbf{x}_n \in \mathbb{R}^d$, is : $\mathbf{z}^1 = h(\mathbf{W}^1 \mathbf{x}_n + \mathbf{c}^1)$, where $\mathbf{W}^1 \in \mathbb{R}^{m^1 \times d}$ is the learned projection matrix, $\mathbf{c}^1 \in \mathbb{R}^{m^1}$ is the bias, and $h(\cdot)$ is a non-linear activation function. The result of the previous hidden layer is passed to the next hidden layer. Therefore, we compute the output of the l-th layer of the network as follows:

$$\mathbf{z}^l = h(\mathbf{W}^l \mathbf{z}^{l-1} + \mathbf{c}^l), \tag{3}$$

where $\mathbf{z}^0 = \mathbf{x}_n$, $\mathbf{W}^l \in \mathbb{R}^{m^l \times m^{l-1}}$ and $\mathbf{c}^l \in \mathbb{R}^{m^l}$ are the learned projection matrix and bias for the l-th layer, respectively. Similarly, the output for the LBE layer in the proposed network is:

$$\mathbf{z}^{L-1} = h(\mathbf{W}^{L-1} \mathbf{z}^{L-2} + \mathbf{c}^{L-1}). \tag{4}$$

We binarize the output of the LBE layer of the network to get the binary codes as follows:

$$\mathbf{b}_n = sgn(\mathbf{z}^{L-1}). \tag{5}$$

With Equations (1), (2), (3) and (5), $g(\mathbf{x}_n) = sgn(\mathbf{z}^{L-1})$ maps the sample in the RGB space to K binary bits: $g(\cdot) : \mathbb{R}^d \rightarrow \mathbb{R}^{m^{L-1}}$, such that $m^{L-1} = K$ and $g(\cdot)$ is parameterized by $\{\mathbf{W}^l, \mathbf{c}^l\}_{l=1}^L$, which are learned.

Our goal is to optimize the parameters of the DCH network such that the mapping from the sample space to the K-bit binary code space preserves the semantic similarity among images. To this end, we design a loss function to optimize the binary codes for classification that preserves semantic structure in the Hamming space. Specifically, given M_b mini-batches of training images with individual training example $\mathbf{x}_n$ and its corresponding label, $\mathbf{y} = [y_1, ..., y_\varsigma]$, where $y_j = 1$ if and only if the instance $\mathbf{x}_n$ is associated with the j-th label, we formulate the mapping, similar to [20], into the following optimization problem:

$$\begin{aligned} \min_{\mathbf{W},\mathbf{c}} E &= \sum_{n=1}^{M_b} E_\varsigma(\hat{\mathbf{y}}, \mathbf{y}), \\ &\text{s.t. } \mathbf{b}_n \in \{-1, 1\}^{K \times 1}, \end{aligned} \tag{6}$$

where $\hat{\mathbf{y}} = h(\mathbf{a}^L)$ is the output of the classification layer, $\mathbf{a}^L = \mathbf{W}^L \mathbf{b} + \mathbf{c}^L$ is the linear activation at the output layer, $E_\varsigma(\hat{\mathbf{y}}, \mathbf{y})$ is the classification loss function. Here, we introduce an additional notation for linear activations at the l-th layer, $\mathbf{a}^l = \mathbf{W}^l \mathbf{z}^{l-1} + \mathbf{c}^l$, to simplify the subsequent computational representation.

C. Relaxation

Unfortunately, the discrete variable $\mathbf{b}$ in Equation (6) will make the problem intractable. Direct optimization of binary codes using the entire training set [20], [44] might not be suitable, since the CNN is trained on batches of data which are much less and the optimality of binary codes produced from the CNN with the same formulation will be questionable. To address this problem, we relax the hard-quantization constraint $\mathbf{b}$ into continuous embedding $\hat{\mathbf{b}} = \mathbf{tanh}(\mathbf{a}^{\mathbf{L-1}}) \in [-1, 1]^{K \times 1}$. Therefore, we rewrite the objective function as follows:

$$\begin{aligned} \min_{\mathbf{W},\mathbf{c}} E &= \sum_{n=1}^{M_b} E_\varsigma(\hat{\mathbf{y}}, \mathbf{y}) \\ &\text{s.t. } \hat{\mathbf{b}}_n = \tanh(\mathbf{a}^{L-1}), \end{aligned} \tag{7}$$

where $\hat{\mathbf{y}} = h(\mathbf{a}^L)$. However, the relaxation will introduce the accumulated quantization errors between $\mathbf{b}$ and relaxed $\hat{\mathbf{b}}$, and subsequently decrease the retrieval accuracy. Therefore, to reduce the accumulated errors, we add a penalty term $E_\mathcal{Q}(\mathbf{b}_n, \hat{\mathbf{b}}_n^{L-1})$ with a regularization coefficient γ into Equation (6). Then Equation (6) can be rewritten as:

$$\min_{\mathbf{W},\mathbf{c}} E = \sum_{n=1}^{M_b} E_\varsigma(\hat{\mathbf{y}}, \mathbf{y}) + \frac{\gamma}{2} \sum_{n=1}^{M_b} E_\mathcal{Q}(\mathbf{b}_n, \hat{\mathbf{b}}_n) \quad \text{s.t. } \hat{\mathbf{b}}_n = \tanh(\mathbf{a}^{L-1}), \mathbf{b}_n = sgn(\hat{\mathbf{b}}_n). \tag{8}$$

We adopt a joint optimization of softmax and quantization losses to train the DCH for binary feature learning. We choose softmax at the output layer for DCH because we formulate the binary hashing as a multiclass classification problem. The formulation is given as follows:

$$\min_{\mathbf{W},\mathbf{c}} E = -\sum_{n=1}^{M_b} log \frac{e^{a_{y_n}^L}}{\sum_j e^{a_j^L}} + \frac{\gamma}{2} \sum_{n=1}^{M_b} \|\mathbf{b}_n - \hat{\mathbf{b}}_n\|^2 \quad \text{s.t. } \hat{\mathbf{b}}_n = \tanh(\mathbf{a}^{L-1}), \mathbf{b}_n = sgn(\hat{\mathbf{b}}_n). \tag{9}$$

In Equation (9), y_n represents the class membership of $\mathbf{x}_n$, and a_j^L represents the linear activation of the j-th neuron in the output layer. With this modified objective function, the network is trainable and can be optimized using standard stochastic gradient descent (SGD) with the back-propagation algorithm.

Our proposed method is inspired by the work in [27]; however, we modify its loss function and impose a quantization constraint to learn better binary representations. Different from [20], our approach uses a deep CNN to simultaneously learn the feature representation and binary codes, and we solve the entire optimization problem with the back-propagation algorithm.

D. Backpropagation

To train the network, we need to compute the gradients of the objective function in Equation (7) with respect to the parameters $\{\mathbf{W}^l, \mathbf{c}^l\}_{l=1}^L$ of the network. In the following, we define several notations. w_{ji}^l is a component of projection matrix $\mathbf{W}^l$ and denotes the weighted connection of the i-th neuron in the $(l-1)$-th layer to the j-th neuron in the l-th layer; c_j^l and z_j^l are components of the vectors $\mathbf{c}$ and $\mathbf{z}$, and denote the bias and the activation, respectively, of the j-th neuron in the l-th layer; $a_j^l = \sum_i w_{ji}^l z_i^{l-1} + c_j^l$ is a component of the vector $\mathbf{a}^l$ and denotes the input to the activation function of the j-th neuron in the l-th layer, i.e., the weighted sum of outputs of all neurons in the $(l-1)$-th layer.

To compute the gradients, we compute the partial derivatives $\frac{\partial E_{\mathbf{x}_n}}{\partial w_{ji}^l}$ and $\frac{\partial E_{\mathbf{x}_n}}{\partial c_j^l}$ for a single training example $\mathbf{x}_n$, and then recover the overall gradients by averaging over all the training examples. Therefore, the gradient of the loss function with respect to the parameters at the output layer is computed as follows:

$$\begin{aligned} \frac{\partial E_{\mathbf{x}_n}}{\partial w_{ji}^L} &= \frac{\partial E_\varsigma}{\partial a_j^L} \frac{\partial a_j^L}{\partial w_{ji}^L}, \\ &= b_i \delta_j^L, \\ \frac{\partial E_{\mathbf{x}_n}}{\partial c_j^L} &= \frac{\partial E_\varsigma}{\partial a_j^L} \frac{\partial a_j^L}{\partial c_j^L}, \\ &= \delta_j^L. \end{aligned} \tag{10}$$

Algorithm 1: DCH binary feature learning algorithm

Input : Training examples $\{\mathbf{x}_n\}$ with their corresponding labels $\{\mathbf{y}_n\}$, mini batch size (M_b) of training examples, learning rate α, number of iterations T, parameter γ.

Output: $\{\mathbf{W}^l, \mathbf{c}^l\}_{l=1}^L$.

```
for t = 1, 2, ..., T do
    foreach training examples x_n in M_b do
        Set z^0 = x_n
        for l = 1, 2, 3, ..., L do
            Perform feedforward computation for z^1 using
              (3) for other layers, and (4) and
              b̂_n = tanh(a^{L-1}) for LBE layer.
        end
        for l = L, L-1, ..., 1 do
            Compute the gradients according to (10)-(16).
        end
        for l = L, L-1, ..., 1 do
            Update the weights and biases according to
              (17)-(18).
        end
    end
end
```

Return: $\{\mathbf{W}^l, \mathbf{c}^l\}_{l=1}^L$

Here, we introduce an error term for j-th neuron of L-th layer, $\delta_j^L = \frac{\partial E_\varsigma}{\partial a_j^L} h'(a_j^L)$, where $h'(.)$ is the derivative of the non-linear function. For the classification loss with a softmax cross entropy as defined in Equation (9), we can compute the error using the chain rule of derivatives [54], [55] as:

$$\delta_j^L = \hat{y}_j - y_j. \tag{11}$$

In a vector form, we have

$$\delta^L = \hat{\mathbf{y}} - \mathbf{y}. \tag{12}$$

Next, we compute the gradient of the objective function with respect to the parameters of the LBE layer as follows:

$$\begin{aligned} \frac{\partial E_{\mathbf{x}_n}}{\partial w_{ji}^{L-1}} &= \frac{\partial E_\varsigma}{\partial a_j^{L-1}} \frac{\partial a_j^{L-1}}{\partial w_{ji}^{L-1}} + \frac{\gamma}{2} \frac{\partial E_\mathcal{Q}}{\partial a_j^{L-1}} \frac{\partial a_j^{L-1}}{\partial w_{ji}^{L-1}}, \\ &= z_i^{L-2} (\vartheta_j^{L-1} + \frac{\gamma}{2} \varrho_j), \\ &= z_i^{L-2} \delta_j^{L-1}, \\ \frac{\partial E_{\mathbf{x}_n}}{\partial c_j^{L-1}} &= \frac{\partial E_\varsigma}{\partial a_j^{L-1}} \frac{\partial a_j^{L-1}}{\partial c_j^{L-1}} + \frac{\gamma}{2} \frac{\partial E_Q}{\partial a_j^{L-1}} \frac{\partial a_j^{L-1}}{\partial c_j^{L-1}}, \\ &= \vartheta_j^{L-1} + \frac{\gamma}{2} \varrho_j, \\ &= \delta_j^{L-1}, \end{aligned} \tag{13}$$

where δ_j^{L-1} is the error at the j-th neuron of the LBE layer, and it consists of two components: ϑ_j error due to classification loss and ϱ_j error due to quantization loss. For the loss function defined in Equation (9), the components are computed using the chain rule of derivative and represented in a vector form as follows:

$$\begin{aligned}\vartheta^{L-1} &= ((\mathbf{W}^L)^T\delta^L) \odot (1-\tanh^2(\mathbf{a}^{L-1})),\\ \varrho &= -2(\mathbf{b}-\mathbf{z}^{L-1}) \odot (1-\tanh^2(\mathbf{a}^{L-1})),\end{aligned} \tag{14}$$

where $\mathbf{z}^{L-1} = \hat{\mathbf{b}}$ and $\odot$ denotes the Hadamard product. Similarly, we can compute the gradient for the rest of the layers using standard back-propagation as follows:

$$\begin{aligned}\frac{\partial E_{\mathbf{x}_n}}{\partial w^l_{ji}} &= z^{l-1}_i \delta^l_j,\\ \frac{\partial E_{\mathbf{x}_n}}{\partial c^l_{ji}} &= \delta^l_j,\end{aligned} \tag{15}$$

where δ^l_j is the j-th component of the error vector at the l-th layer computed as:

$$\delta^l = ((\mathbf{W}^l)^T\delta^{l+1}) \odot h'(\mathbf{a}^l), \tag{16}$$

The parameters are then updated for all the training samples in mini-batches (M_b) using the gradient descent algorithm as follows:

$$w^l_{ji} = w^l_{ji} - \alpha \sum_{n=1}^{M_b} \frac{\partial E_{\mathbf{x}_n}}{\partial w^l_{ji}}, \tag{17}$$

$$c^l_{ji} = c^l_{ji} - \alpha \sum_{n=1}^{M_b} \frac{\partial E_{\mathbf{x}_n}}{\partial c^l_{ji}}, \tag{18}$$

where α is the learning rate. The overall learning algorithm for the proposed DCH is given in Algorithm 1.

IV. Implementation Details

Architecture: The proposed DCH method is implemented with Keras [57]. The network architecture used in the paper is illustrated in Table I. It consists of four convolution-pooling layers followed by one fully connected layer, one LBE layer and finally the output layer. The convolution layers are comprised of 32, 32, 64, and 64 kernels of size 3×3. The architecture uses 2 max-pooling, each with a window of size 2×2 with stride 2. Additionally, batch normalization [58] layers are used before the nonlinear activation for each of the convolution and fully connected layers. The fully connected layer consists of 2048 nodes, and the LBE layer consists of K nodes, where K is the length of binary code. For all of the convolution and fully connected layers, ReLU [56] is used as the nonlinear activation function. We choose ReLU in hidden layers of DCH because networks with ReLU nonlinear activations, which are non-saturating, can be trained faster than those with a saturating nonlinearity [59]. Note that we design a relatively simple architecture because of the limitation of the available computation resources and training dataset. The proposed DCH method is generic, and a more complicated architecture could be easily used to achieve better binary representation of the images.

Parameters & Training: The network is initialized with a uniform distribution scaled by the square root of the number of inputs [60] during training. We use a fixed mini-batch of 32 images to optimize the DCH network. We train the network using the adaptive learning rate optimization method called ADADELTA [61] because of its desirable property that does not require a specific hyperparameter tunning. The network is trained with 200 epochs.

To obtain different binary codes using DCH, it is required to train distinct models with different numbers of neurons (K) at the LBE layer. Training all the models from scratch is a severe waste of the computation time, and results in slow convergence. Additionally, the number of parameters that need to be learned increases with the code length, and the network becomes more prone to overfitting. One can observe that for all of these models, layers preceding the LBE layers are common, and these parameters could be pre-trained and easily shared by all the models. Pre-training has also been reported to improve the performance of the network [33]. Therefore, we choose to pre-train the network without the LBE layer to learn a pure classification model and then fine-tune the network with a desired code length in the LBE layer.

V. Experiments

A. Datasets and Evaluation Metrics

To verify the effectiveness of our proposed DCH method and other state-of-the-art methods, we create a large dataset comprised of digitized histopathological skeletal muscle and lung cancer images. The skeletal muscle images represent two major categories of Idiopathic Inflammatory Myopathy (IIM): Polymyositis (PM) and Dermatomyositis (DM); The lung cancer images represent two major categories of the disease: adenocarcinoma (AC) and squamous cell carcinoma (SC). We select non-overlapping regions and crop 5256 (2572 PM and 2678 DM) skeletal muscle images corresponding to 41 individual subjects and 2904 (1456 AC and 1448 SC) lung cancer images corresponding to 126 individual subjects, respectively. Details on the dataset are provided as follows.

Skeletal Muscle Images: The whole slide scanned skeletal muscle images are prepared by the Medical College of Wisconsin Neuromuscular Laboratory (MCWNL) using a Hamamatsu NanoZoomer Microscope. The images are captured at a $40\times$ objective with pixel resolution of 0.25 micron. All ground-truth are provided by MCWNL. All of the images are analyzed by 3 independent pathologists and a final label is assigned to each image based on the common consensus among them.

Lung Cancer Images: . The 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 images are supervised by National Cancer Institute (NCI) and National Human Genome Research Institute (NHGRI) and are freely available to researchers.

In our experiments, the raw RGB image data are directly used as input for all of the deep learning based methods, and they are wrapped to patches with a size of 128×128 before inputting to the learning pipeline. The semantic similarity between the images are defined by their corresponding labels, i.e., images from the same category are considered similar, and

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JBHI.2018.2827703, IEEE Journal of Biomedical and Health Informatics

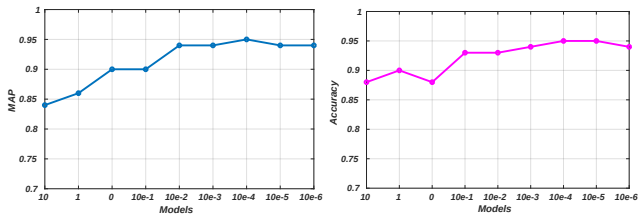
TABLE I: The network architecture used for the proposed DCH method. Two of the convolution layers are followed by max-pooling. All of the layers except pooling and the final dense layer are followed by batch normalization. In the Table, $(3, 32)_{/1,0} \times 2$ represents two successive convolution layers with 32 kernels of size 3×3, and the stride of 1 and 0 padding. Similarly, $2_{/2,0}$ represents a max-pooling layer with kernels of size 2×2 with stride of 2 and 0 padding. K in the LBE layer represents the number of bits. For all the convolution and fully connected layers, ReLU is used as the nonlinear activation [56].

	Stage 1	Stage 2		Stage 3		Stage 5	Stage 6	Stage 7
Layer	**Input**	**Conv**	**Pool**	**Conv**	**Pool**	**Dense**	**Dense (LBE)**	**Dense**
Size	128	$(3, 32)_{/1,0} \times 2$	$2_{/2,0}$	$(3, 64)_{/1,0} \times 2$	$2_{/2,0}$	2048	K	4

dissimilar otherwise. We sample and use 25% of the images per class for testing and rest of the images for training. The image patches obtained from the same case are never used simultaneously in training and testing stage. For conventional hashing methods, images are represented by the 2048-D deep learning features and 2000-D histogram using bag-of-words BoW [62], [63] that encodes the SIFT [64] features.

Following previous work, we use following evaluation metrics to compare the performance of different methods: (1) accuracy based on the top-ranked retrieved images; (2) mean average precision (MAP) computed as area under the precision-recall curve for different bit values; (3) precision-recall curve for different bit values; and (4) mean precision for Hamming look-up within the Hamming radius of r.

B. Effects of Parameter γ



(a) (b)

Fig. 2: MAP (a) and classification accuracy (b) of the models with respect to γ, which regulates the strength of quantization error in overall loss computation.

The hyperparameter γ controls the strength of the quantization loss and balances the two loss functions. The parameter is essential to train our DCH model. Therefore, we conduct an experiment to investigate the sensitiveness of the learned binary values with respect to this parameter. We vary the value of γ from 10 to $10e^{-6}$ to learn different models. Without loss of generality, we only test for the number of bits $K = 8$ for the proposed DCH method. We report the MAP and the classification accuracy using the learned binary codes.

Figure 2 summarizes the experimental results. We can observe that the retrieval performance improves significantly by setting the value of γ in a reasonable range (i.e. [$10e^{-1}$, $10e^{-6}$]). We also see that the accuracy and MAP increase from $10e^{-1}$ to $10e^{-4}$, and the performance drops slightly as we further decrease the value of gamma. On the other hand, for the value of $\gamma = 0$ (equivalent to having no influence of quantization error) inferior performsance is observed. The range to improve accuracy suggests that the imposed quantization constraint can reduce the gap between real-valued representations and binary codes, therefore improving the overall retrieval accuracy. Increasing the influence of the quantization error (setting value of $\gamma = \{1, 10\}$) has adverse effects on the retrieval performance, observed both in MAP and the accuracy. This is expected since weakening the influence of classification error decreases the discriminative power of the binary codes, and the neighboring points in the output space are mapped to different binary representations. We find that the proposed method attends highest MAP and accuracy at value of $\gamma = 10e^{-4}$. Therefore, we set the value of $gamma$ to $10e^{-4}$ for all the experiments in the paper.

C. Comparison with State of the Arts

Setup: We compare our DCH method with the following state-of-the-art approaches: KSDH [65], KSH [41], SDH [20], CCA [19], COSDISH [66], DLBHC [27], CNNH [67], SFL [26], and DSRH [49]. All of these methods are implemented using the source codes provided by the authors except for SFL[1] and DSRH[1]. For fairness, all of the deep learning based methods (i.e., DLBHC, CNNH, SFL, and DSRH) are trained using the same network architecture.

For all of the comparative methods, we use the parameters suggested by the authors in their respective publications. For our proposed DCH, we set the value of hyperparameter $\gamma = 10e^{-4}$. Additional experiments on the sensitivity of network with respect to the parameter γ is reported in Section V-B. Furthermore, following observations are made during training:

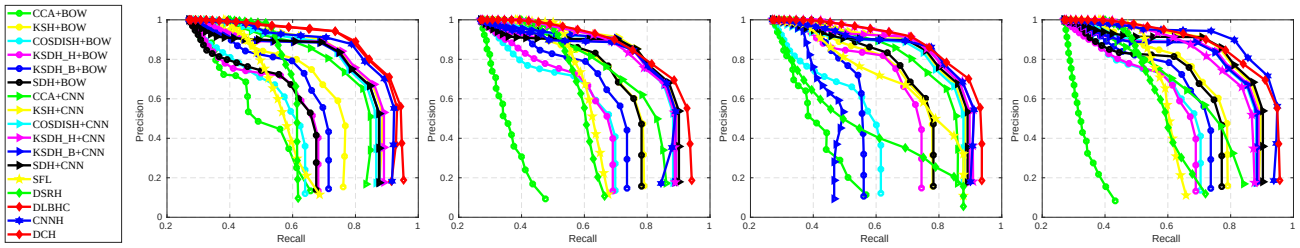
1) To achieve equivalent performance, the DCH network trained with ADADELTA converges faster than SGD with momentum.
2) With the addition of batch normalization before ReLU activation, we have a performance (measured as classification accuracy on validation dataset) boost of 6% for our dataset. Even without the dropout layer, no overfitting is observed while using batch normalization; Adding dropout with batch normalization actually degraded the performance of the network.
3) Comparable performance is observed for networks trained on various mini-batch with size ranging from 8 to 64. In our implementation, the mini-batch size of 32 is chosen as a tradeoff between the training speed and memory requirement.

[1]We used our implementation of the method for experiment because source codes for the method is not publicly available.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JBHI.2018.2827703, IEEE Journal of Biomedical and Health Informatics

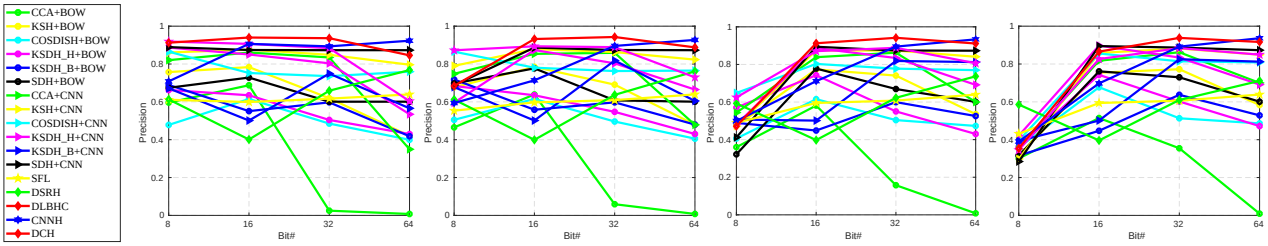
TABLE II: Retrieval performance measured as Accuracy based on the top 10 ranked retrieved images, and MAP with respect to different numbers of bits on the dataset. We calculate the MAP values within the top 1000 returned neighbors for all the compared methods. [∗]+CNNF in the table are the traditional methods trained with deep learned features.

Method	Accuracy				MAP (Top 1000)			
	8	16	32	64	8	16	32	64
KSDH_H [65]	0.68	0.75	0.69	0.69	0.74	0.81	0.75	0.75
KSDH_B [65]	0.71	0.56	0.74	0.74	0.79	0.69	0.81	0.80
KSH [41]	0.64	0.78	0.79	0.79	0.81	0.82	0.83	0.83
SDH [20]	0.68	0.78	0.78	0.77	0.74	0.85	0.85	0.82
CCA [19]	0.65	0.72	0.66	0.64	0.57	0.46	0.39	0.36
COSDISH [66]	0.64	0.62	0.69	0.70	0.70	0.69	0.75	0.76
KSDH_H [65]+CNNF	0.89	0.90	0.89	0.88	0.91	0.92	0.91	0.91
KSDH_B [65]+CNNF	0.88	0.47	0.89	0.88	0.90	0.62	0.91	0.90
KSH [41]+CNNF	0.66	0.88	0.87	0.88	0.90	0.90	0.91	0.91
SDH [20]+CNNF	0.88	0.89	0.90	0.90	0.90	0.92	0.91	0.92
CCA [19]+CNNF	0.57	0.78	0.88	0.91	0.85	0.85	0.81	0.74
COSDISH [66]+CNNF	0.87	0.88	0.88	0.88	0.91	0.90	0.91	0.91
SFL [26]	0.58	0.61	0.61	0.59	0.61	0.65	0.66	0.64
DSRH [49]	0.52	0.37	0.60	0.64	0.62	0.41	0.63	0.65
DLBHC [27]	0.73	0.63	0.73	0.73	0.88	0.91	0.80	0.88
CNNH [67]	0.50	0.68	0.50	0.51	0.92	0.91	0.89	0.95
DCH	**0.94**	**0.95**	**0.95**	**0.96**	**0.95**	**0.94**	**0.94**	**0.96**



(a) @8-bit (b) @16-bit (c) @32-bit (d) @64-bit

Fig. 3: Precision-recall curves for different algorithms using different numbers of bits. For each curve, we compute the precision and recall in the range of 300 to 6000 retrieved images with with an interval of 300.



(a) @r=2 (b) @r=3 (c) @r=4 (d) @r=5

Fig. 4: Precision vs bit using different algorithms on Top 1000 results for different Hamming radiuses (@r).

Results: Table II shows the accuracy based on top 10 ranked retrieved images and the MAP using different contrastive hashing methods. As we can see, our proposed DCH achieves the best accuracy in all of the cases among all the hashing methods. This suggests better ranking of the retrieved images. Additionally, superior ranking performance is observed for DCH with MAP ranging from 0.94 to 0.96 using different numbers of bits, outperforming all the other methods by 2%-4%. CNNH shows similar MAP compared to DCH for 64-bit representation but underperforms for all the other bits. CNNH and DLBHC provide comparable ranking performance in terms of MAP but exhibit inferior accuracy compared to DCH. SFL and DSRH, which use triplet ranking based losses to train the networks, show inferior performance both in terms of accuracy and MAP. Meanwhile, all of the traditional methods trained on handcrafted features underperform significantly. We can observe the performance boost ranging from 5% to 20% for these methods when trained with deep learning features, which makes the results comparable to other end-to-end trained deep learning based hashing methods.

Figure 3 and Figure 4 show the precision-recall curve using different numbers of bits and the precision using different numbers of bits with different hamming radiuses (@r), respectively. The results are consistent with previous observations, which demonstrate that our proposed DCH method provides superior performance. Examples with qualitative representa-

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/JBHI.2018.2827703, IEEE Journal of Biomedical and Health Informatics

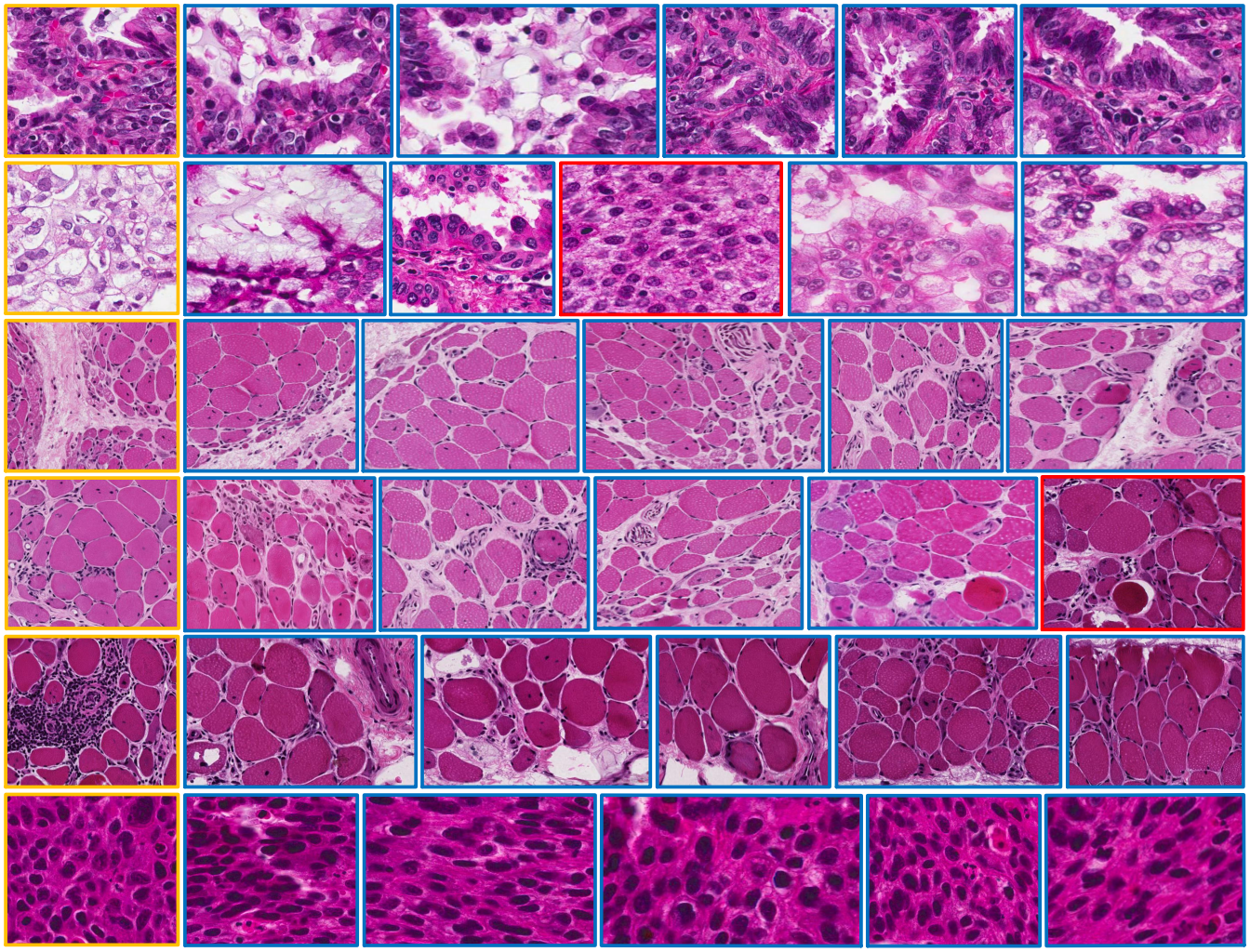


Fig. 5: Six examples of top 5 ranked retrieval results (query marked in yellow, correct retrieved images marked in blue, and incorrect retrieved images are marked in red) for the proposed DCH using 8-bit image representation. Rows 1 and 2 represent images AC; 3 and 4 represent images DM; 5 and 6 represent images PM, and SC, respectively, where DM and PM are types of myopathies and AC and SC are lung carcinomas.

tions of the retrieval results using the proposed DCH with 8-bit representation are shown in Figure 5. It also shows incorrectly retrieved results matched to different class for a case of DM and a case of AC. The incorrectly retrieved cases are observed to share common visual and morphological attributes, which is common in histopathological images [68].

D. Discussion

In the traditional methods such as KSDH_H and KSDH_B, binary encoding with learned features outperforms that using the handcrafted features. This validates the effectiveness of the learned features over the handcrafted features, and also demonstrates the importance of good feature representations for effective hashing. Although these results are comparable to other deep learning based hashing methods, the two-stage strategy, which consists of first training the deep learning network for feature extraction and then utilizing these features to learn the hash function for binary encoding, makes the training process complicated. Therefore, the end-to-end learning capability of deep learning based hashing methods make them preferable to the two stage methods.

Among the deep learning based methods, it can be observed that our DCH provides the highest ranking performance. Higher accuracy and MAP of DCH suggest that the images that match the most to the query image are among the top-ranked results. The CNNH method first computes the discriminative binary codes and then trains the model to fit the pre-computed codes. The relatively poor performance of the CNNH suggests that separating the binary code and network learning generates a lot of noise in the training labels and cannot optimize the final binary codes. DSRH and SFL methods use label information to create a triplet ranking matrix and minimize variations of the triplet loss to learn binary codes. Triplet loss based methods show relatively poor performance among the deep learning based methods probably because the loss is based on the distance between the features

of positive and negative image pairs rather than directly utilize the supervised information to optimize the network. DLBHC, on the other hand, takes an approach similar to ours where the model is trained with binary-like hidden layers optimized for classification. The relatively poor performance of the encoded binary values suggests that there is discrepancy between the learned real values and discrete binary values. However, it should be noted that the proposed approach to "point-wise" train the network still captures the semantic relation between similar images (as well as dissimilar image). Our DCH method further reduces the discrepancy between real-valued feature representations and binary codes by penalizing the network to reduce the gap between them, therefore improving the performance.

E. Training Time and Space Complexity Analysis

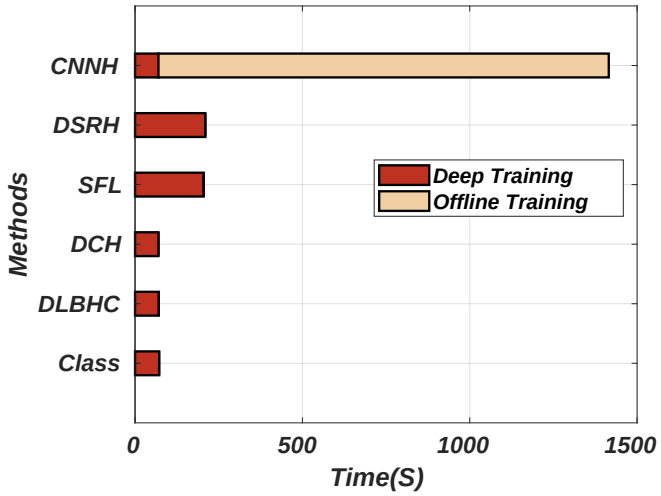


Fig. 6: Training time for different deep learning based methods for 64 bit hashing. **Deep Training** represents the time to run 5 epochs of training iteration averaged over 5 independent training runs, and **Offline Training** represents additional time spent to learn the binary codes.

This section analyzes the computation complexity of the loss functions of our DCH method and other deep learning based hashing methods: DLBHC, CNNH, SFL, DSRH and base classification model (Class). For fair comparison, all of the methods are implemented with the same version of Keras. The experiments are carried out on a PC with Intel i7-5930 processor and 128GB memory, and 12GB NVIDIA Titan GPU with CUDA-8.0.

First, the CNN architecture used in this work has approximately 130M parameters. For all of the deep learning based methods, computational complexity of the loss function is very limited compared to the training time for the entire CNN. We should also note that all of the deep learning based method add no additional test time since binary representation for images is obtained by forward pass through the network. Next, we compare the training time of the deep learning based methods as shown in Figure 6. It shows the time plot to run 5 epochs of training iteration averaged over 5 independent runs for all the deep learning based methods. Without loss of generality, we only report the time consumption for 64 bit codes. As illustrated in figure, DCH and DLBHC methods, which are trained *point-wise*, have similar training time compared to the base Class model. This suggests that they do not have additional computational overhead. SFL and DSRH have greater training time because they require to gather and employ a triplet ranking matrix, in which each sample is repeatedly used. Training the network for the CNNH has comparable training time to DCH but it requires additional offline processing time to learn the binary codes. This suggests that our proposed method has less training overhead and better training time than other pairwise trained deep learning methods.

Additionally, during the optimization of the network, the maximum storage complexity for the DCH and DLBHC is $\mathcal{O}(M_b)$, which is comparable to the base Class model. For CNNH, network optimization has similar space requirement as DCH but the offline training to learn binary codes requires $\mathcal{O}(N^2)$ space for the pairwise similarity matrix. SFL and DSRH require additional memory to store the triplet ranking matrix. Therefore, the storage complexity for these methods are at least $\mathcal{O}(M_b^3)$. For most of the pairwise similarity based approaches such as [33], they require at least $\mathcal{O}(M_b^2)$ space to store the pairwise similarity matrix. This suggests our proposed method is memory efficient than other pairwise similarity based methods.

VI. Conclusion And Future Work

In this paper, we propose a deep learning based hashing method called DCH to encode images into binary codes. The proposed network consists of an embedded LBE layer that can be trained in the "point-wise" manner. The effectiveness of the proposed method is illustrated by the superior performance on the histopathology image dataset. The improved performance can be directly attributed to modified loss functions that jointly optimize the network for classification and binary representation learning. The classification loss in the optimization encourages the network to preserve semantic similarity of learned binary codes and the quantization loss helps the network to reduce the gap between the real-valued low dimensional embedded features and desired binary values. The proposed approach is scalable and general; more complicated networks could be easily exploited to further improve the effectiveness of the learned binary codes. In practice, the improved binary representation can be utilized for efficient management of the histopathological images. This framework provides a fast image query and retrieval of similar cases, thereby assists experts in the evidence-based study of the diseases for diagnosis. They can also study the retrieved similar cases to understand the morphological and biological characteristics of a disease. This could help experts in early diagnosis and provide effective personalized treatment.

Currently, the model optimization is relaxed using a saturating non-linear $tanh$ function, which might restrict the performance of the learned binary codes. In the future, we will work on improving the quantization loss with better optimization formulation. Additionally, this work assumes image labels do

not have noise, i.e., data annotation is consistent. Label noise would have a negative impact on the learning of the proposed model and then in the ultimate disease diagnosis. Therefore, in the future, we will work on developing noise-insensitive learning algorithms to train the network and also statistically evaluate the robustness of the proposed method on disease diagnosis. We will also focus on diagnosis of the whole slide images or images at the organ level based on the analysis of the multiple regions of interest from the same organ. This might require reformulating the problem with new constraint to learn the anatomical spatial organization of the images. We will also compare DCH with human pathologists to measure interrater or annotator reliability using standard statistical tests.

ACKNOWLEDGMENT

This research is funded, in part, by NIH R01 AR065479-02.

REFERENCES

[1] 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*, vol. 9, pp. 234–263, 2016.

[2] M. Veta, J. P. Pluim, P. J. Van Diest, and M. A. Viergever, "Breast cancer histopathology image analysis: A review," *IEEE Transactions on Biomedical Engineering*, vol. 61, no. 5, pp. 1400–1411, 2014.

[3] F. Xing, Y. Xie, H. Su, F. Liu, and L. Yang, "Deep learning in microscopy image analysis: A survey," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–19, 2017.

[4] W. Yang, Z. Lu, M. Yu, M. Huang, Q. Feng, and W. Chen, "Content-based retrieval of focal liver lesions using bag-of-visual-words representations of single-and multiphase contrast-enhanced ct images," *Journal of digital imaging*, vol. 25, no. 6, pp. 708–719, 2012.

[5] Y. L. Zou, C. Li, Z. Boukhers, K. Shirahama, T. Jiang, and M. Grzegorzek, "Environmental microbiological content-based image retrieval system using internal structure histogram," in *Proceedings of the 9th International Conference on Computer Recognition Systems CORES 2015*. Springer, 2016, pp. 543–552.

[6] M. Jiang, S. Zhang, J. Huang, L. Yang, and D. N. Metaxas, "Scalable histopathological image analysis via supervised hashing with multiple features," *Medical Image Analysis*, vol. 34, pp. 3–12, 2016.

[7] A. Kamen, S. Sun, S. Wan, S. Kluckner, T. Chen, A. M. Gigler, E. Simon, M. Fleischer, M. Javed, S. Daali *et al.*, "Automatic tissue differentiation based on confocal endomicroscopic images for intraoperative guidance in neurosurgery," *BioMed research international*, vol. 2016, 2016.

[8] M. Jiang, S. Zhang, H. Li, and D. N. Metaxas, "Computer-aided diagnosis of mammographic masses using scalable image retrieval," *IEEE Transactions on Biomedical Engineering*, vol. 62, no. 2, pp. 783–792, 2015.

[9] L. Zheng, A. W. Wetzel, J. Gilbertson, and M. J. Becich, "Design and analysis of a content-based pathology image retrieval system," *IEEE Transactions on Information Technology in Biomedicine*, vol. 7, no. 4, pp. 249–255, 2003.

[10] M. Sapkota, F. Liu, Y. Xie, H. Su, F. Xing, and L. Yang, "Aiimds: An integrated framework of automatic idiopathic inflammatory myopathy diagnosis for muscle," *IEEE Journal of Biomedical and Health Informatics*, vol. PP, no. 99, pp. 1–1, 2017.

[11] D. J. Foran, L. Yang, W. Chen, J. Hu, L. A. Goodell, M. Reiss, F. Wang, T. Kurc, T. Pan, A. Sharma *et al.*, "Imageminer: a software system for comparative analysis of tissue microarrays using content-based image retrieval, high-performance computing, and grid technology," *Journal of the American Medical Informatics Association*, vol. 18, no. 4, pp. 403–415, 2011.

[12] A. Kumar, J. Kim, W. Cai, M. Fulham, and D. Feng, "Content-based medical image retrieval: a survey of applications to multidimensional and multimodality data," *Journal of digital imaging*, vol. 26, no. 6, pp. 1025–1039, 2013.

[13] X. Zhang, H. Dou, T. Ju, J. Xu, and S. Zhang, "Fusing heterogeneous features from stacked sparse autoencoder for histopathological image analysis," *IEEE journal of biomedical and health informatics*, vol. 20, no. 5, pp. 1377–1383, 2016.

[14] P. Indyk and R. Motwani, "Approximate nearest neighbors: towards removing the curse of dimensionality," in *Proceedings of the thirtieth annual ACM symposium on Theory of computing*. ACM, 1998, pp. 604–613.

[15] A. Gionis, P. Indyk, R. Motwani *et al.*, "Similarity search in high dimensions via hashing," in *VLDB*, vol. 99, no. 6, 1999, pp. 518–529.

[16] B. Kulis and T. Darrell, "Learning to hash with binary reconstructive embeddings," in *Advances in neural information processing systems*, 2009, pp. 1042–1050.

[17] D. Zhang, J. Wang, D. Cai, and J. Lu, "Self-taught hashing for fast similarity search," in *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2010, pp. 18–25.

[18] G. Lin, C. Shen, D. Suter, and A. van den Hengel, "A general two-step approach to learning-based hashing," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 2552–2559.

[19] G. Lin, C. Shen, Q. Shi, A. van den Hengel, and D. Suter, "Fast supervised hashing with decision trees for high-dimensional data," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1963–1970.

[20] F. Shen, C. Shen, W. Liu, and H. Tao Shen, "Supervised discrete hashing," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 37–45.

[21] R. Zhang, L. Lin, R. Zhang, W. Zuo, and L. Zhang, "Bit-scalable deep hashing with regularized similarity learning for image retrieval and person re-identification," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 4766–4779, 2015.

[22] J. Liu, S. Zhang, W. Liu, X. Zhang, and D. N. Metaxas, "Scalable mammogram retrieval using anchor graph hashing," in *Biomedical Imaging (ISBI), 2014 IEEE 11th International Symposium on*. IEEE, 2014, pp. 898–901.

[23] L. Liu, A. Rahimpour, A. Taalimi, and H. Qi, "End-to-end Binary Representation Learning via Direct Binary Embedding," *ArXiv e-prints*, Mar. 2017.

[24] M. Norouzi and D. M. Blei, "Minimal loss hashing for compact binary codes," in *Proceedings of the 28th international conference on machine learning (ICML-11)*, 2011, pp. 353–360.

[25] W.-J. Li, S. Wang, and W.-C. Kang, "Feature learning based deep supervised hashing with pairwise labels," *arXiv preprint arXiv:1511.03855*, 2015.

[26] H. Lai, Y. Pan, Y. Liu, and S. Yan, "Simultaneous feature learning and hash coding with deep neural networks," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[27] K. Lin, H.-F. Yang, J.-H. Hsiao, and C.-S. Chen, "Deep learning of binary hash codes for fast image retrieval," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2015, pp. 27–35.

[28] X. Zhang, W. Liu, M. Dundar, S. Badve, and S. Zhang, "Towards large-scale histopathological image analysis: Hashing-based image retrieval," *IEEE Transactions on Medical Imaging*, vol. 34, no. 2, pp. 496–506, 2015.

[29] Y. Xu, F. Shen, X. Xu, L. Gao, Y. Wang, and X. Tan, "Large-scale image retrieval with supervised sparse hashing," *Neurocomputing*, vol. 229, pp. 45–53, 2017.

[30] D. G. Lowe, "Object recognition from local scale-invariant features," in *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, vol. 2. Ieee, 1999, pp. 1150–1157.

[31] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3431–3440.

[32] M. Sapkota, F. Xing, H. Su, and L. Yang, "Automatic muscle perimysium annotation using deep convolutional neural network," in *2015 IEEE 12th International Symposium on Biomedical Imaging (ISBI)*, April 2015, pp. 205–208.

[33] H. Liu, R. Wang, S. Shan, and X. Chen, "Deep supervised hashing for fast image retrieval," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2064–2072.

[34] J. Shi, J. Wu, Y. Li, Q. Zhang, and S. Ying, "Histopathological image classification with color pattern random binary hashing based pcanet and matrix-form classifier," *IEEE Journal of Biomedical and Health Informatics*, 2016.

[35] S. Conjeti, A. G. Roy, A. Katouzian, and N. Navab, "Deep residual hashing," *arXiv preprint arXiv:1612.05400*, 2016.

[36] S. Conjeti, M. Paschali, A. Katouzian, and N. Navab, "Learning robust hash codes for multiple instance image retrieval," *arXiv preprint arXiv:1703.05724*, 2017.

[37] X. Shi, M. Sapkota, F. Xing, F. Liu, L. Cui, and L. Yang, "Pairwise based deep ranking hashing for histopathology image classification and retrieval," *Pattern Recognition*, 2018.

[38] Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing," in *Advances in neural information processing systems*, 2009, pp. 1753–1760.

[39] W. Liu, J. Wang, S. Kumar, and S.-F. Chang, "Hashing with graphs," in *Proceedings of the 28th international conference on machine learning (ICML-11)*, 2011, pp. 1–8.

[40] Y. Gong and S. Lazebnik, "Iterative quantization: A procrustean approach to learning binary codes," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 817–824.

[41] W. Liu, J. Wang, R. Ji, Y.-G. Jiang, and S.-F. Chang, "Supervised hashing with kernels," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 2074–2081.

[42] J. Wang, S. Kumar, and S.-F. Chang, "Semi-supervised hashing for large-scale search," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 12, pp. 2393–2406, 2012.

[43] K. He, F. Wen, and J. Sun, "K-means hashing: An affinity-preserving quantization method for learning binary compact codes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 2938–2945.

[44] W. Liu, C. Mu, S. Kumar, and S.-F. Chang, "Discrete graph hashing," in *Advances in Neural Information Processing Systems*, 2014, pp. 3419–3427.

[45] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, "Locality-sensitive hashing scheme based on p-stable distributions," in *Proceedings of the twentieth annual symposium on Computational geometry*. ACM, 2004, pp. 253–262.

[46] M. Raginsky and S. Lazebnik, "Locality-sensitive binary codes from shift-invariant kernels," in *Advances in neural information processing systems*, 2009, pp. 1509–1517.

[47] T. Ge, K. He, and J. Sun, "Graph cuts for supervised binary coding," in *European Conference on Computer Vision*. Springer, 2014, pp. 250–264.

[48] V. Erin Liong, J. Lu, G. Wang, P. Moulin, and J. Zhou, "Deep hashing for compact binary codes learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2475–2483.

[49] F. Zhao, Y. Huang, L. Wang, and T. Tan, "Deep semantic ranking based hashing for multi-label image retrieval," in *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*. IEEE, 2015, pp. 1556–1564.

[50] Z. Xia, X. Feng, J. Lin, and A. Hadid, "Deep convolutional hashing using pairwise multi-label supervision for large-scale visual search," *Signal Processing: Image Communication*, vol. 59, pp. 109–116, 2017.

[51] H. Müller, N. Michoux, D. Bandon, and A. Geissbuhler, "A review of content-based image retrieval systems in medical applicationsclinical benefits and future directions," *International journal of medical informatics*, vol. 73, no. 1, pp. 1–23, 2004.

[52] Z. Li, X. Zhang, H. Müller, and S. Zhang, "Large-scale retrieval for medical image analytics: A comprehensive review," *Medical image analysis*, vol. 43, pp. 66–84, 2018.

[53] X. Zhang, L. Yang, W. Liu, H. Su, and S. Zhang, "Mining histopathological images via composite hashing and online learning." in *MICCAI (2)*, 2014, pp. 479–486.

[54] J. S. Bridle, "Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters," in *Advances in neural information processing systems*, 1990, pp. 211–217.

[55] P. Sadowski, "Notes on backpropagation," 2016. [Online]. Available: https://www.ics.uci.edu/%7Epjsadows/notes.pdf

[56] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

[57] F. Chollet, "keras," https://github.com/fchollet/keras, 2015.

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

[59] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[60] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks." in *Aistats*, vol. 9, 2010, pp. 249–256.

[61] M. D. Zeiler, "Adadelta: an adaptive learning rate method," *arXiv preprint arXiv:1212.5701*, 2012.

[62] J. C. Caicedo, A. Cruz, and F. A. Gonzalez, "Histopathology image classification using bag of features and kernel functions," in *Conference on Artificial Intelligence in Medicine in Europe*. Springer, 2009, pp. 126–135.

[63] J. C. Caicedo, F. A. González, and E. Romero, "Content-based histopathology image retrieval using a kernel-based semantic annotation framework," *Journal of biomedical informatics*, vol. 44, no. 4, pp. 519–528, 2011.

[64] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[65] X. Shi, F. Xing, J. Cai, Z. Zhang, Y. Xie, and L. Yang, "Kernel-based supervised discrete hashing for image retrieval," in *European Conference on Computer Vision*. Springer, 2016, pp. 419–433.

[66] W.-C. Kang, W.-J. Li, and Z.-H. Zhou, "Column sampling based discrete supervised hashing." in *AAAI*, 2016, pp. 1230–1236.

[67] R. Xia, Y. Pan, H. Lai, C. Liu, and S. Yan, "Supervised hashing for image retrieval via image representation learning." in *AAAI*, vol. 1, 2014, p. 2.

[68] M. C. Dalakas, "Muscle biopsy findings in inflammatory myopathies," *Rheumatic Disease Clinics*, vol. 28, no. 4, pp. 779–798, 2002.

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, computer aided diagnosis system.

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.

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.

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/.