Improved Deep Hashing With Soft Pairwise Similarity for Multi-Label Image Retrieval

Zheng Zhang, Qin Zou, *Senior Member, IEEE*, Yuewei Lin, Long Chen, and Song Wang, *Senior Member, IEEE*

***Abstract*—Hash coding has been widely used in the approximate nearest neighbor search for large-scale image retrieval. Recently, many deep hashing methods have been proposed and shown largely improved performance over traditional feature-learning methods. Most of these methods examine the pairwise similarity on the semantic-level labels, where the pairwise similarity is generally defined in a hard-assignment way. That is, the pairwise similarity is "1" if they share no less than one class label and "0" if they do not share any. However, such similarity definition cannot reflect the similarity ranking for pairwise images that hold multiple labels. In this paper, an improved deep hashing method is proposed to enhance the ability of multi-label image retrieval. We introduce a pairwise quantified similarity calculated on the normalized semantic labels. Based on this, we divide the pairwise similarity into two situations—"hard similarity" and "soft similarity," where cross-entropy loss and mean square error loss are adapted respectively for more robust feature learning and hash coding. Experiments on four popular datasets demonstrate that the proposed method outperforms the competing methods and achieves the state-of-the-art performance in multi-label image retrieval.**

***Index Terms*—Image retrieval, convolutional neural network, semantic label, pairwise similarity, deep hashing.**

I. Introduction

With the popular use of smartphone cameras, the amount of image data has been rapidly increasing, which calls for more efficient and accurate image retrieval. Generally, image retrieval is based on the approximate nearest neighbor search [1] and an image-retrieval system is often built on hashing [2]. In hashing methods, high dimensional data are transformed into compact binary codes and similar binary codes are expected to generate for similar data items. Due to the encouraging efficiency in both speed and storage, a number of hashing methods have been proposed in the past decade [3]–[14].

Generally, the existing hashing methods can be divided into two categories: unsupervised methods and supervised methods. The unsupervised methods use unlabeled data to generate hash functions. They focus on preserving the distance similarity in the Hamming space as in the feature space. The supervised methods incorporate human-interactive annotations, e.g., pairwise similarities of semantic labels, into the learning process to improve the quality of hashing, and often outperform the unsupervised methods. In the past five years, inspired by the success of deep neural networks that show superior feature-representation power in image classification [15]–[18], object detection [19], face recognition [20], and many other vision tasks [21]–[23], many supervised hashing methods based on deep neural networks were developed for image abstraction and hash-code learning [24]–[36]. These so called deep hashing methods have achieved the state-of-the-art performance on several popular benchmark datasets.

While these supervised deep hashing methods have produced impressive improvement in image retrieval, to the best of our knowledge, they only examine the similarity of pairwise images using the semantic-level labels, and define the similarity in a coarse way. That is, *the similarity of pairwise images is '1' if they share at least one object class and '0' (or '−1') if they do not share any object class*. However, such similarity definition cannot reflect the fine-grained similarity when the pairwise images both have multiple labels. An illustrative example is shown in Fig. 1 where the images in (a), (b) and (c) share the same class label 'sky' and each pair of them are taken as similar in the context of image retrieval. However, as the images in (a) and (b) share three class labels, i.e., 'sky', 'bridge', and 'water', the similarity between them should be ranked higher than that between (a) and (c) which have only one class label in common. It can be easily observed that, the traditional coarse similarity definition does not take the multi-label information into account and cannot rank the similarity for images with multiple class labels.

To solve this problem, we present a soft definition for the pairwise similarity with regarding to the semantic labels each image holds. Specifically, the pairwise similarity is quantified into a

Manuscript received June 25, 2018; revised April 16, 2019 and June 11, 2019; accepted July 9, 2019. Date of publication July 31, 2019; date of current version January 24, 2020. This work was supported in part by the National Natural Science Foundation of China under Grants 61872277, 41571437, 61672376, and U1803264, in part by the Hubei Provincial Natural Science Foundation under Grant 2018CFB482. The work of Y. Lin was supported by BNL LDRD 18-009. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Marco Bertini. *(Corresponding author: Qin Zou.)*

Z. Zhang and Q. Zou are with the School of Computer Science, Wuhan University, Wuhan 430072, China (e-mail: zhengzhang@whu.edu.cn; qzou@whu.edu.cn).

Y. Lin is with the Computational Science Initiative, Brookhaven National Laboratory, Upton, NY 11973 USA (e-mail: ywlin@bnl.gov).

L. Chen is with the School of Data and Computer Science, Sun Yat-Sen University, Guangzhou 518001, China (e-mail: chenl46@mail.sysu.edu.cn).

S. Wang is with the Department of Computer Science and Engineering, University of South Carolina, Columbia, SC 29201 USA, and also with the School of Computer Science and Technology, Tianjin University, Tianjin 300072, China (e-mail: songwang@cec.sc.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2019.2929957

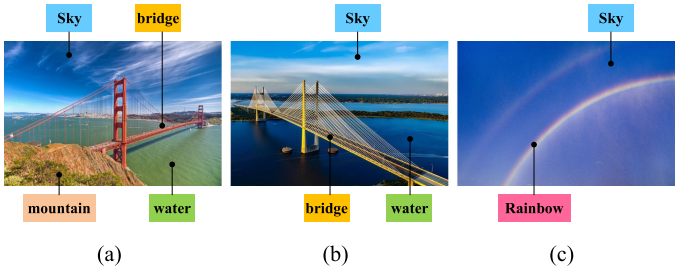


Fig. 1. Some examples of the multi-label images. Since the images in (a) and (b) share more class labels, the similarity between them is supposed to be higher than that between (a) and (c). However, in the traditional pairwise-similarity definition, the similarities between them are the same.

percentage using the normalized semantic labels. Based on the quantified similarity, we propose a deep method to improve the retrieval quality for multi-label image retrieval. For convenience, we abbreviate this improved deep hashing network as IDHN in the following description. Specifically, we divide the quantified similarity into two situations: one is 'hard similarity', which means a pair of images share either all object types or none; another is 'soft similarity', which means a pair of images share some object classes, but not all. For robustness and practicability, we construct cross entropy loss for 'hard similarity' situation and mean square error loss for 'soft similarity', for preserving the similarity of an image pair in hash space converging to their fine-grained semantic similarity in form of normalized labels. We evaluate the proposed deep hashing method on four popular multi-label image datasets and obtain significantly improved performance over the state-of-the-art hashing methods in image retrieval. The contributions of this work lie in three-fold:

- We propose a soft definition for the pairwise similarity by quantifying it into a percentage using the normalized semantic labels. To the best of our knowledge, IDHN is the first deep hashing method that directly uses pairwise quantified similarity which can reflect the fine-grained similarity between a pair of multi-label images for supervised learning.
- We divide the pairwise similarity into two situations – 'hard similarity' and 'soft similarity', and a joint loss-function of cross-entropy loss and mean square error loss are adapted for learning efficient, robust hash codes, and preserving the fine-grained semantic similarity based on the quantified similarity.
- Experiments have shown that the proposed method outperforms current state-of-the-art methods on four datasets in image retrieval and has good extensibility to deeper network architecture, which demonstrates the effectiveness of the proposed method.

The rest of this paper is organized as follows: Section II briefly reviews the related work. Section III describes the proposed quantified similarity deep hashing method which generates high-quality hash codes in a supervised learning manner. Section IV demonstrates the effectiveness of the proposed model by extensive experiments on four popular benchmark datasets, and Section V concludes our work.

II. Related Work

In the past two decades, many hashing methods have been proposed for approximate nearest neighbor search in the large-scale image retrieval. Hashing-based methods transform high dimensional data into compact binary codes with a fixed number of bits and generate similar binary codes for similar data items, which can greatly reduces the storage and calculation consumption. Generally, the existing hashing methods can be divided into two categories: unsupervised methods and supervised methods.

Unsupervised Methods: The unsupervised hashing methods learn hash functions to preserve the similarity distance in the Hamming space as in the feature space. Locality-Sensitive Hashing (LSH) [37] is one of the most well-known representative. LSH aims to maximize the probability that the similar items will be mapped to the same buckets. Spectral Hashing (SH) [3] and [38] consider hash encoding as a spectral graph partitioning problem, and learn a nonlinear mapping to preserve semantic similarity of the original data in the Hamming space. Iterative Quantization (ITQ) [8] searches for an orthogonal matrix by alternating optimization to learn the hash functions. Sparse Product Quantization (SPQ) [39] encodes the high-dimensional feature vectors into sparse representation by decomposing the feature space into a Cartesian product of low-dimensional subspaces and quantizing each subspace via K-means clustering, and the sparse representations are optimized by minimizing their quantization errors. [40] proposes to learn compact hash code by computing a sort of soft assignment within the k-means framework, which is called "multi-k-means", to void the expensive memory and computing requirements. Latent Semantic Minimal Hashing (LSMH) [41] refines latent semantic feature embedding in the image feature to refine original feature based on matrix decomposition, and a minimum encoding loss is combined with latent semantic feature learning process simultaneously to get discriminative obtained binary codes.

Supervised Methods: The supervised hashing methods use supervised information to learn compact hash codes, which usually achieve superior performance compared with the unsupervised methods. Binary Reconstruction Embedding (BRE) [4] constructs hash functions by minimizing the squared error loss between the original feature distances and the reconstructed Hamming distances. Semi-supervised hashing (SSH) [5] combines the characteristics of the labeled and unlabeled data to learning hash functions, where the supervised term tries to minimize the empirical error on the labeled data and the unsupervised term pursuits effective regularization by maximizing the variance and independence of hash bits over the whole data. Minimal Loss Hashing (MLH) [6] learns hash functions based on structural prediction with latent variables using a hinge-like loss function. Supervised Hashing with Kernels (KSH) [7] is a kernel based method which learns compact binary codes by maximizing the separability between similar and dissimilar pairs in the Hamming space. Online Hashing [42] is also a hot research area in image retrieval. [43] proposes an online multiple kernel learning method, which aims to find the optimal combination of multiple kernels for similarity learning, and [44] improves the online multi-kernel learning with semi-supervised way, which utilizes

supervision information to estimate the labels of the unlabeled images by introducing classification confidence that is also instructive to select the reliably labeled images for training.

In the last few years, approaches built on deep neural networks have achieved state-of-the-art performance on many vision tasks [15]–[17] as comparing to traditional methods [45]. Inspired by the powerful representation ability of deep neural networks, some deep hashing methods have been proposed, which show great progress compared with traditional hand-crafted feature based methods. A simple way to deep hashing learning is thresholding high level feature directly, the typical methods is DLBHC [46], which learns hash-like representations by inserting a latent hash layer before the last classification layer in AlexNet [15]. While the network is fine-tuned well on classification task, the feature of latent hash layer is considered to be discriminative, which indeed presents better performance than hand-crafted feature. CNNH [24] was proposed as a two-stage hashing method, which decomposes the hash learning process into a stage of learning approximate hash codes, and followed by a stage of deep network fine-tune to learn the image features and hash functions. DNNH [26] improves the two-stage CNNH in both the image representations and hash coding by using a joint learning process. DNNH and DSRCH [27] use image triplets as the input of deep network, which generate hash codes by minimizing the triplet ranking loss. Since the pairwise similarity is more straightforward than the triplet similarity, most of the latest deep hashing networks used pairwise labels for supervised hashing and further improved the performance of image retrieval, e.g., DHN [28], DQN [29] and DSH [30] etc. HashNet [31] proposes a deep hashing method to learn binary hash codes from imbalanced similarity data by continuation method with convergence guarantees.

Since the above-mentioned deep hashing methods are not designed for multi-label image retrieval, the fine-grained similarity of multi-label images are always neglected with coarse-grained definition between pair images. For multi-label retrieval, DSRH [25] tries to learn hash function by utilizing the ranking information of multi-level similarity, and proposes a surrogate losses to solve the optimization problem of ranking measures. IAH [47] focuses on learning instance-aware image representations and using the weighted triplet loss to preserve similarity ranking for multi-label images. However, the weighted triplet loss functions adapted by DSRH [25] and IAH [47] do not enforce direct restriction to learn fine-grained multilevel semantic similarity, since they are focusing on preserving the correct ranking of images according to their similarity degrees to the queries, which make it a room for improvement on the accuracies of top returned images. Based on this, DMSSPH [48] tries to construct hash functions to maximize the discriminability of the output space to preserve multilevel similarity between multi-label images. Althought DMSSPH [48] has utilized the fine-grained multilevel semantic similarity for pairwise similarity learning, there still are spaces for further exploration. A novel and effective method TALR was proposed in [36], which considered tied rankings on integer-valued Hamming distance and directly optimized the ranking-based evaluation metrics Mean Average Precision (MAP) [49] and Normalized Discounted Cumulative Gains (NDCG) [50]. It achieved high performance in several benchmark datasets. In [51], two new protocols were presented for the evaluation of supervised hashing methods, under the context of transfer learning.

In this work, we study to improve the hashing quality by exploring the diversities of pairwise semantic similarity on the multi-label dataset. Specifically, we propose to define the fine-grained pairwise similarity in the form of continuous value, and according to this definition, we divide the pairwise similarity into two situations and construct a joint pairwise loss function to perform simultaneous feature learning and hash-code generating.

III. Improved Deep Hashing Networks

A. Problem Definition

Given a training set of N images $I = \{I_1, I_2, \ldots, I_N\}$ and a pairwise similarity matrix $S = \{s_{ij} | i, j = 1, 2, \ldots, N\}$, the goal of hash learning for images is to learn a mapping $F : I \mapsto \{-1, 1\}^q$, so that an input image I_i can be encoded into a q-bit binary code $F(I_i)$, with the similarities of images being preserved. The similarity label s_{ij} is usually defined as $s_{ij} = 1$ if I_i and I_j have semantic label, i.e., object class label, in common and $s_{ij} = 0$ if I_i and I_j do not share any semantic label. As discussed in the introduction, this definitions does not take the multi-label information into account and cannot rank the similarity for images with multiple class labels. In our design, the pairwise similarity is quantified into percentages and the similarity value s_{ij} is defined as the cosine distance of pairwise label vectors:

$$s_{ij} = \frac{\langle l_i, l_j \rangle}{\|l_i\| \|l_j\|}, \tag{1}$$

where l_i and l_j denote the semantic label vector of image I_i and I_j, respectively, and $\langle l_i, l_j \rangle$ calculates the inner product. This cosine distance has been widely adopted in retrieval system, but it is always used to measure similarity of the feature vectors [29]. To the best of our knowledge, we are the first to use the cosine distance to quantify fine-grained semantic similarity of pair images.

According to Eq. (1), the similarity of pairwise images can be passed into three states: completely similar, partially similar, and dissimilar. For approximate nearest neighbor search, we demand that the binary codes $B = \{b_i\}_{i=1}^N$ should preserve the similarity in S. To be specific, given a pair of binary codes b_i and b_j, if $s_{ij} = 0$ which means pairwise images I_i and I_j do not share any object class, the Hamming distance between b_i and b_j should be large, i.e., be close to q in the q-bit hash coding case; if $s_{ij} = 1$, which means the pairwise images I_i and I_j have the same class labels, we expect the Hamming distance to be zero; otherwise, the binary codes b_i and b_j should have a suitable Hamming distance complying with the soft definition of similarity s_{ij}.

As a conclusion, we define the completely similar and dissimilar situation as 'hard similarity', which can be seen as equivalent to the similarity definition for single label images. Besides of these 'hard similarity' situations, the similarity between a pair

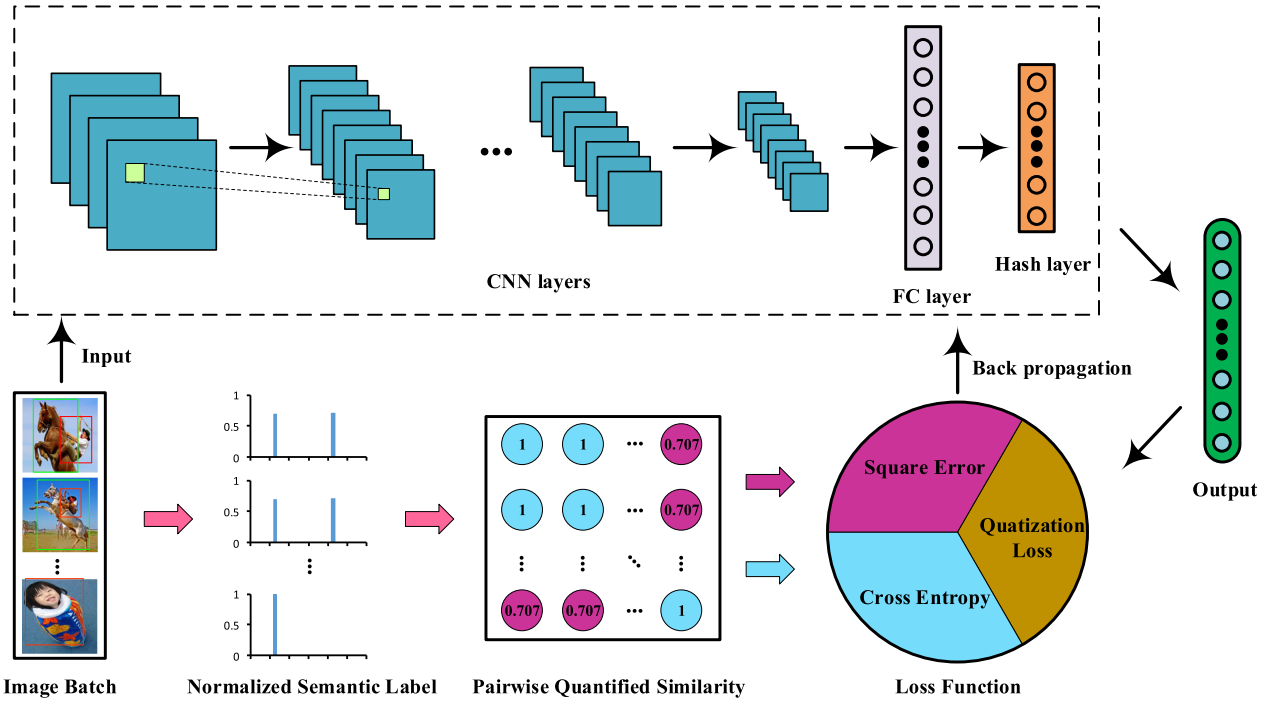


Fig. 2. An overview of the proposed deep hashing learning method. The top frame shows the deep architecture of neural network that produces the hash codes. The bottom frame shows the processing of pairwise quantified similarity and loss function construction. Cross entropy loss and mean square error loss are combined to preserving fine-grained pairwise similarity and a quantization loss is adapted to impose constraints for compact hash coding.

of images is more fine and complicated, which we define as 'soft similarity'.

Figure 2 shows the pipeline of the proposed deep hashing network for supervised hash-code learning. The proposed method accepts input images in a pairwise form (I_i, I_j, s_{ij}) and processes them through the deep representation learning and hash coding. It includes a sub-network with multiple convolution/pooling layers to perform image abstraction, fully-connected layer to approximate optimal dimension-reduced representation, and hash layer to generate q-bits hash codes. In this framework, a pairwise similarity loss is introduced for similarity-preserving learning, and a quantization loss is used to control the quality of hashing. The pairwise similarity loss consists of two parts–the cross entropy loss and the square error loss. Details will be introduced in the following of this section.

B. Deep Network Architecture

Since many deep hashing methods [28], [29], [31], [48] have adapted AlexNet [15] as base network, without loss of generality, we also adopt the AlexNet as our base network. AlexNet comprises of five convolutional layers $conv1$ - $conv5$ and three fully connected layers $fc6 - fc8$. After each hidden layer, a nonlinear mapping $z_i^l = a^l(W^l z_i^{l-1} + b^l)$ is learned by the activation function a^l, where z_i^l is the l-th layer feature representation for the original input, W^l and b^l are the weight and bias parameters of the l-th layer. We replace the $fc8$ layer of the softmax classifier in the original AlexNet with a new fully-connected hashing layer with q hidden nodes, which converts the learned deep features into a low-dimensional hash codes. In order to realize hash encoding, we introduce an activation function $a^l(x) = \frac{x}{|x|+1}$ to map the output of $fc8$ to be within $(-1, 1)$. Notice that, our method can be easily extended to other deep networks, such as GoogLeNet [17] and VGG19 [16], we will conduct experiments on these two classical networks to demonstrate the extensibility of our method.

C. Hash-Code Learning

For efficient nearest neighbor search, the semantic similarity of original images should be preserved in the Hamming space. In the following, we will discuss our proposed hashing methods with reference to 'hard similarity' and 'soft similarity' situation, respectively.

1) Hard Similarity: In these situations, according to the quantized pairwise similarity calculated by Eq. (1), the similarity of pairwise images s_{ij} can only get value 0 or 1, which is identical to the similarity definition in previous deep pair hashing methods [28], [30]. Given the hash codes B of all images and the pairwise similarity relation $S_h = \{s_{ij}\}$, the conditional probability $p(s_{ij}|B)$ of s_{ij} can be defined as follows:

$$p(s_{ij}|B) = \begin{cases} \sigma(\Omega_{ij}), & s_{ij} = 1, \\ 1 - \sigma(\Omega_{ij}), & s_{ij} = 0, \end{cases} \tag{2}$$

where $\sigma(x) = \frac{1}{1+e^{-x}}$ is the sigmoid function, which we use to transform the Hamming distance into a kind of measure of similarity. Previous works have shown that the inner product $\langle \cdot, \cdot \rangle$ is a good metric of the Hamming distance to quantify the pairwise

similarity [28], [29]. In this work, we construct an inner product $\Omega_{ij} = \langle b_i, b_j \rangle = b_i^T b_j$.

Here, we adapt negative log-likelihood as cost function to measure the pairwise similarity loss, as formulated by Eq. (3),

$$\begin{aligned}\mathcal{L}_1 &= -\sum_{s_{ij}\in S_h} log(p(s_{ij}|B)) \\ &= -\sum_{s_{ij}\in S_h} \big(s_{ij} log(\sigma(\Omega_{ij})) + (1-s_{ij}) log(1-\sigma(\Omega_{ij}))\big).\end{aligned} \tag{3}$$

Then, substituting the sigmoid function $\sigma(\Omega_{ij})$ with $\frac{1}{1+e^{-\Omega_{ij}}}$, we get

$$\mathcal{L}_1 = \sum_{s_{ij}\in S_h} \big(log(1+e^{\Omega_{ij}}) - s_{ij}\Omega_{ij}\big). \tag{4}$$

2) Soft Similarity: In this situation, the pairwise similarities defined by Eq. (1) are continuous value, we apply mean square error function to preserve the similarity of hash codes to fit the soft similarity. Thus, the pairwise similarity loss can be defined as:

$$\mathcal{L}_2 = \sum_{s_{ij}\in S_s} \left(\frac{\langle b_i, b_j\rangle + q}{2} - s_{ij}\cdot q\right)^2. \tag{5}$$

As the inner product $\langle b_i, b_j\rangle$ is within $[-q, q]$, the value of $\frac{\langle b_i,b_j\rangle+q}{2}$ will be non-negative and be within $[0, q]$, which has a same value range as $s_{ij}\cdot q$.

Although the cross entropy loss can also be used to measure the similarity error in the soft similarity, the mean square error loss shows better performance when multi-label images have more complicated semantic relation and more shared labels. We will discuss this in the experiments.

3) Joint Learning: For simultaneous learning of these two cases and make an unified form, we use M_{ij} to mark the two cases, where $M_{ij} = 1$ denotes the 'hard similarity' case, and $M_{ij} = 0$ denotes 'soft similarity' case. Hence, the pairwise similarity loss is rewritten as:

$$\begin{aligned}\mathcal{L} = \sum_{s_{ij}\in S} \big[&M_{ij}(log(1+e^{\Omega_{ij}}) - s_{ij}\Omega_{ij}) \\ &+\gamma\cdot(1-M_{ij})\left(\frac{\langle b_i, b_j\rangle + q}{2} - s_{ij}\cdot q\right)^2\big],\end{aligned} \tag{6}$$

where γ is a weight parameter to make a tradeoff between the cross entropy loss and mean square error loss.

It is challenging to directly optimize Eq. (6), because the binary constraint $b_i \in \{-1, 1\}^q$ requires thresholding the network outputs, which may result in the vanishing-gradient problem in back propagation during the training procedure. Following previous works [2], [28], [30], we apply the continuous relaxation to solve this problem. We use the output of deep hashing network u as a substitute for binary code b. Ω_{ij} is redefined as $\alpha u_i^T u_j$, where α is a positive hyper-parameter to control the constraint bandwidth. Since the network output is not the binary codes, we use a pairwise quantization loss to encourage the network output to be close to standard binary codes. The pairwise quantization loss is defined as

$$\mathcal{Q} = \sum_{i,j\in N} (\||u_i| - 1\|_1 + \||u_j| - 1\|_1), \tag{7}$$

where 1 is a vector of all ones, $\|\cdot\|_1$ is the L1-norm of the vector, $|\cdot|$ is the element-wise absolute value operation. By integrating the pairwise similarity loss and pairwise quantization loss, the final cost loss is defined as

$$\mathcal{C} = \mathcal{L} + \lambda\mathcal{Q}, \tag{8}$$

where λ is a weight coefficient for controlling the quantization loss.

D. Learning Algorithm

During the training process, the standard back-propagation algorithm with mini-batch gradient descent method is used to optimize the pairwise loss function. By combining Eq. (6) and Eq. (7), we rewrite the optimization objective function $\mathcal{C}$ as follows:

$$\begin{aligned}\mathcal{C} &= \mathcal{L} + \lambda\mathcal{Q} \\ &= \sum_{i,j\in N}\Big[M_{ij}\left(log\left(1+e^{\alpha u_i^T u_j}\right) - \alpha\cdot s_{ij}\cdot u_i^T u_j\right) \\ &\quad + \gamma\cdot(1-M_{ij})\left(\frac{u_i^T u_j + q}{2} - s_{ij}\cdot q\right)^2 \\ &\quad + \lambda\cdot(\||u_i| - 1\|_1 + \||u_j| - 1\|_1)\Big].\end{aligned} \tag{9}$$

In order to employ back propagation algorithm to optimize the network parameters, we need to compute the derivative of the objective function. The sub-gradients of Eq. (9) w.r.t. u_{ik} (k-th unit of the network output u_i) can be written as:

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial u_{ik}} &= \alpha\cdot M_{ij}\sum_{j\in N}(\sigma(\Omega_{ij}) - s_{ij})\cdot u_{jk} \\ &\quad + \gamma\cdot(1-M_{ij})\sum_{j\in N}(u_i^T u_j + q - 2\cdot s_{ij}\cdot q)\cdot u_{jk},\end{aligned} \tag{10}$$

and

$$\frac{\partial\mathcal{Q}}{\partial u_{ik}} = \begin{cases}1, & -1 < u_{ik} < 0, \\ -1, & \text{otherwise},\end{cases} \tag{11}$$

The gradient of u_{ik} w.r.t. $\hat{z}_{ik}^l$ (raw representations of hash layer before activation) can be calculated by

$$\frac{\partial u_{ik}}{\partial\hat{z}_{ik}^l} = sgn(\hat{z}_{ik}^l)\cdot\frac{1}{(|\hat{z}_{ik}^l| + 1)^2}, \tag{12}$$

where $sgn(\cdot)$ is an element-wise sign function and $\hat{z}_i^l = W^l z_i^{l-1} + b^l$ is the output of the l-th layer before activation. The gradient of the network parameter W^l is

$$\frac{\partial\mathcal{C}}{\partial W^l} = \sum\left(\frac{\partial\mathcal{L}}{\partial u_i} + \lambda\frac{\partial\mathcal{Q}}{\partial u_i}\right)\frac{\partial u_i}{\partial\hat{z}_i^l}\cdot z_i^{l-1}. \tag{13}$$

Since we have computed sub-gradients of the hash layer, the rest of the back-propagation procedure can be done in the standard manner. Note that, after the learning procedure, we have not obtained the corresponding binary codes of input images yet. The network only generates approximate hash codes that have values within $(-1, 1)$. To finally get the hash codes and evaluate the efficacy of the trained network, we need to treat the test query data as input and forward propagate the network to generate hash codes by using Eq. (14),

$$b_{ik} = sgn(u_{ik}). \tag{14}$$

In this way, we can train the deep neural network in an end-to-end fashion, and any new input images can be encoded into binary codes by the trained deep hashing model. Ranking the distance of these binary hash codes in the Hamming space, we can obtain an efficient image retrieval.

IV. Experiments and Results

A. Datasets

To verify the performance of the proposed method, we compare the proposed method with several baseline methods on four widely used benchmark datasets, i.e., NUS-WIDE, Flickr, VOC2012 and IAPRTC12.

NUS-WIDE [52] is a dataset containing 269,648 public web images. It is a multi-label dataset in which each image is annotated with one or more class labels from a total of 81 classes. We follow the settings in [26], [53] to use the subset of images associated with the 21 most frequent labels, where each label associates with at least 5,000 images, resulting in a total of 195,834 images. We resize the images of this subset to 227×227.

Flickr [54] is a dataset containing 25,000 images collected from Flickr. Each image contains at least one of the 38 semantic labels. We resize the images to 227×227.

VOC2012 [55] is a widely used dataset for object detection and segmentation, which contains 17,125 images, and each image belongs to at least one of the 20 semantic labels. We resize the images to 227×227.

IAPRTC12 [56] contains 20,000 images with segmentation masks. Each region of segmentation has been assigned with a label from a total of 276 pre-defined categories. We resize the images to 227×227.

B. Implementation Details

For NUS-WIDE, we randomly select 100 images per class to form a test query set of 2,100 images, and 500 images per class to form the training set. For Flickr, VOC2012 and IAPRTC12, we randomly select 1,000 images as the test query set, and 4,000 images as the train set. The remaining images in each of the four datasets are taken as query database.

We compare our method with several state-of-the-art hashing methods, including three unsupervised methods LSH [37], SH [3] and ITQ [8], and two traditional supervised methods MLH [6], KSH [7], and one classification-based deep hashing methods DLBHC [46], three deep hashing methods with coarse pairwise similarity definition–HashNet [31], DHN [28] and DQN [29], and one state-of-the-art deep hashing methods designed for multi-label image retrieval, DMSSPH [48]. Notice that, although the Jaccard coefficient based similarity has been used as one multiplier of the weight for each training pair in HashNet [31], the similarity of pairwise image is still in a coarse way. To make the comparison more extensive, four other typical deep hashing methods are also included, namely the DPSH [34], DSRH [25], DSDH [32] and DTSH [33]. Based on a coarse pairwise similarity, DPSH constructs the cross-entropy loss and absolute quantization loss based on a coarse pairwise similarity, and DSDH directly learns the discrete hash codes with the auxiliary of classification mask. DTSH and DSRH are two triplet-based methods which learn hash function from the local triplet ranking relation [57].

We implement the proposed IDHN[1] by the TensorFlow toolkit [58]. To make fair comparison, all deep hashing methods for comparison are reproduced by using the TensorFlow and based on the bone net of AlexNet. For other traditional methods, we use the open-source codes released by [59], which are implemented with MATLAB. We fine-tune the convolutional layers $conv1$–$conv5$ and fully-connected layers $fc6$–$fc7$ with network weight parameters copied from the pre-trained model, and train the hashing layer $fc8$, all via back-propagation. We use the Adam method for stochastic optimization with a mini-batch size of 128, and the learning rate decay after each 500 iterations with a decay rate of 0.5. For the deep learning based methods, we directly use the image pixels as the input. For the other traditional methods, i.e., LSH, SH, ITQ, MLH and KSH, feature maps of the fully-connected layer $fc7$ are extracted using the pre-trained model and taken as the input without other preprocessing.

C. Metrics

We evaluate the image retrieval quality using four widely-used metrics: Average Cumulative Gains (ACG) [60], Normalized Discounted Cumulative Gains (NDCG) [50], Mean Average Precision (MAP) [49] and Weighted Mean Average Precision (WAP) [25].

ACG represents the average number of shared labels between the query image and the top n retrieved images. Given a query image I_q, the ACG score of the top n retrieved images is calculated by

$$ACG@n = \frac{1}{n}\sum_{i}^{n} C(q, i), \tag{15}$$

where n denotes the number of top retrieval images and $C(q, i)$ is the number of shared class labels between I_q and I_i.

NDCG is a popular evaluation metric in information retrieval. Given a query image I_q, the DCG score of top n retrieved images is defined as

$$DCG@n = \sum_{i}^{n} \frac{2^{C(q,i)} - 1}{log(1 + i)}. \tag{16}$$

[1]Codes are available at https://sites.google.com/site/qinzoucn/

TABLE I
RESULTS OF MEAN AVERAGE PRECISION (MAP) FOR DIFFERENT PARAMETER VALUE OF α. NOTE THAT, q DENOTES THE LENGTH OF THE HASH CODES

α	NUS-WIDE		Flickr		VOC2012	
	24-bit	48-bit	24-bit	48-bit	24-bit	48-bit
1/q	0.7141	0.7194	0.7878	0.8049	0.6185	0.6374
5/q	**0.7560**	**0.7681**	**0.8462**	**0.8515**	0.6874	0.7032
10/q	0.7498	0.7661	0.8425	0.8472	**0.6886**	**0.7087**
20/q	0.7310	0.7528	0.8394	0.8492	0.6682	0.7014
1	0.7202	0.7079	0.8367	0.8428	0.6644	0.6674

Then, the normalized DCG (NDCG) score at the position n can be calculated by $NDCG@n = \frac{DCG@n}{Z_n}$, where Z_n is the maximum value of $DCG@n$, which constrains the value of NDCG in the range [0, 1].

MAP is the mean of average precision for each query, which can be calculated by

$$MAP = \frac{1}{Q}\sum_{q}^{Q} AP(q), \tag{17}$$

where

$$AP(q) = \frac{1}{N_{Tr}(q)@n}\sum_{i}^{n}\left(Tr(q,i)\frac{N_{Tr}(q)@i}{i}\right), \tag{18}$$

and $Tr(q,i) \in \{0,1\}$ is an indicator function that if I_q and I_i share some class labels, $Tr(q,i) = 1$; otherwise $Tr(q,i) = 0$. Q is the numbers of query sets and $N_{Tr}(q)@i$ indicates the number of relevant images w.r.t. the query image I_q within the top i images.

The definition of WAP is similar with MAP. The only difference is that WAP computes the average ACG scores at each top n retrieved image rather than average precision. WAP can be calculated by

$$WAP = \frac{1}{Q}\sum_{q}^{Q}\left(\frac{1}{N_{Tr}(q)@n}\sum_{i}^{n}(Tr(q,i)\times ACG@i)\right). \tag{19}$$

D. Results

1) Parameter Analysis: The parameter α is used to control the range of inner product value after normalization. We notice that the gradient of large absolute value is very small in the sigmoid function, which may cause gradient vanishing. In order to avoid this and accelerate the convergence, we employ the parameter α and set its value according to the length q of the hash codes. Table I shows the results of the proposed method by using different values of α. We set $\alpha = \frac{5}{q}$ to constrain the result of Ω_{ij} to be within $[-5, 5]$, which is relatively a suitable range. γ and λ are the coefficient of mean-square-error loss and quantization loss, respectively. We first study the influence of λ. It can be seen from Table II, it achieves the best results at $\lambda = 0.1$. It is because that a larger λ will result in more discrete but less similarity-preserved hash codes, and a smaller λ will make the quantization loss less effective. We also examine the influence of γ. It can be seen from Table III, $\gamma = \frac{0.1}{q}$ leads to the highest performance among all settings. By dividing q, the gradient of mean square error loss in Eq. (10) can be adaptively adjusted within a suitable range. Too larger or too smaller a γ value will destroy the balance between the cross-entropy loss and the mean-square-error loss.

TABLE II
RESULTS OF MEAN AVERAGE PRECISION (MAP) FOR DIFFERENT PARAMETER VALUES OF λ

λ	NUS-WIDE		Flickr		VOC2012	
	24-bit	48-bit	24-bit	48-bit	24-bit	48-bit
0	0.7259	0.7224	0.8020	0.7731	0.6393	0.6934
0.01	0.7263	0.7241	0.7983	0.7719	0.6419	0.6890
0.1	**0.7345**	**0.7298**	**0.8052**	**0.7756**	**0.6428**	**0.6954**
1.0	0.6662	0.6743	0.7651	0.7628	0.6334	0.6525
10.0	0.5382	0.5731	0.6878	0.6952	0.6228	0.4624

TABLE III
RESULTS OF MEAN AVERAGE PRECISION (MAP) FOR DIFFERENT PARAMETER VALUE OF γ

γ	NUS-WIDE		Flickr		VOC2012	
	24-bit	48-bit	24-bit	48-bit	24-bit	48-bit
0	0.7227	0.7309	0.8398	0.8454	0.6582	0.6721
0.01/q	0.7287	0.7542	0.8410	0.8501	0.6628	0.6779
0.1/q	**0.7600**	**0.7692**	**0.8462**	**0.8515**	**0.6874**	**0.7032**
1/q	0.7288	0.7275	0.8002	0.7757	0.6842	0.6933
10/q	0.6693	0.6685	0.7179	0.7194	0.6640	0.6677

TABLE IV
RESULTS OF MEAN AVERAGE PRECISION (MAP) FOR DIFFERENT NUMBERS OF BITS ON NUS-WIDE DATASET

Methods	12-bit	24-bit	36-bit	48-bit
IDHN	**0.7292**	**0.7585**	**0.7639**	**0.7692**
DQN [29]	0.7106	0.7327	0.7454	0.7493
DHN [28]	0.7187	0.7399	0.7595	0.7637
DMSSPH [48]	0.6713	0.6993	0.7173	0.7273
HashNet [4]	0.6429	0.6938	0.7371	0.7501
DLBHC [46]	0.5696	0.6160	0.6214	0.6351
KSH [7]	0.6556	0.6825	0.6934	0.7024
MLH [6]	0.5184	0.5319	0.5512	0.5458
SH [3]	0.4368	0.4412	0.4616	0.4596
ITQ [8]	0.5469	0.5666	0.5785	0.5876
LSH [37]	0.3854	0.4085	0.4452	0.4453

TABLE V
RESULTS OF MEAN AVERAGE PRECISION (MAP) FOR DIFFERENT NUMBERS OF BITS ON FLICKR DATASET

Methods	12-bit	24-bit	36-bit	48-bit
IDHN	**0.8327**	**0.8469**	**0.8490**	**0.8515**
DQN [29]	0.8092	0.8227	0.8298	0.8270
DHN [28]	0.8227	0.8393	0.8446	0.8471
DMSSPH [48]	0.7800	0.8080	0.8096	0.8159
HashNet [4]	0.7909	0.8262	0.8414	0.8483
DLBHC [46]	0.7236	0.7566	0.7573	0.7761
KSH [7]	0.7907	0.8070	0.8141	0.8181
MLH [6]	0.7033	0.7073	0.7163	0.7103
SH [3]	0.6451	0.6512	0.6505	0.6463
ITQ [8]	0.6845	0.6950	0.6973	0.6978
LSH [37]	0.5968	0.6086	0.6265	0.6369

2) Comparison With State-of-the-Art Methods: The results of MAP on NUS-WIDE, Flickr and VOC2012 datasets are shown from Table IV to VI. It can be observed that, the proposed IDHN method substantially outperforms all the comparison methods on these three datasets. Among the traditional hashing methods, KSH obtains the best results. Compared with

TABLE VI
RESULTS OF MEAN AVERAGE PRECISION (MAP) FOR DIFFERENT NUMBERS OF BITS ON VOC2012 DATASET

Methods	12-bit	24-bit	36-bit	48-bit
IDHN	**0.6561**	**0.6874**	**0.6991**	**0.7032**
DQN [29]	0.6303	0.6564	0.6675	0.6716
DHN [28]	0.6445	0.6704	0.6829	0.6928
DMSSPH [48]	0.6064	0.6298	0.6343	0.6420
HashNet [4]	0.6502	0.6809	0.6856	0.6871
DLBHC [46]	0.5284	0.5372	0.5895	0.6173
KSH [7]	0.5874	0.6088	0.6196	0.6209
MLH [6]	0.5074	0.5179	0.5305	0.5263
SH [3]	0.4465	0.4453	0.4493	0.4552
ITQ [8]	0.4966	0.5054	0.5110	0.5134
LSH [37]	0.3866	0.4039	0.4118	0.4222

TABLE VII
RESULTS OF MEAN AVERAGE PRECISION (MAP) FOR DIFFERENT NUMBERS OF BITS ON IAPRTC-12 DATASET

Methods	12-bit	24-bit	36-bit	48-bit
IDHN	**0.5495**	**0.5697**	**0.5779**	**0.5859**
DQN [29]	0.5409	0.5672	0.5728	0.5774
DHN [28]	0.5412	0.5672	0.5762	0.5781
DMSSPH [48]	0.4412	0.4745	0.4877	0.4928
HashNet [4]	0.4912	0.5242	0.5472	0.5612
DLBHC [46]	0.3218	0.3977	0.4199	0.4418
KSH [7]	0.5004	0.5211	0.5313	0.5363
MLH [6]	0.4618	0.4725	0.4763	0.4791
SH [3]	0.3883	0.3793	0.3930	0.3905
ITQ [8]	0.4507	0.4628	0.4659	0.4700
LSH [37]	0.3497	0.3553	0.3712	0.3891

KSH, the proposed method IDHN achieves an improvement of about 7.2%, 3.8% and 7.7% in average MAP for different bits on NUS-WIDE, Flickr and VOC2012, respectively. It can also be observed from Table IV to VI, the deep learning methods have obtained largely improved performance over the three traditional methods. Compared with DMSSPH, a deep hashing method designed for multi-label image retrieval, the proposed IDHN achieves an improvement of about 5.0%, 4.1% and 5.7% in average MAP on the three datasets, respectively. For other three deep hashing methods with coarse similarity definition, i.e., DQN, DHN and HashNet, high MAP values have been obtained. However, compared with these three methods, IDHN still holds a higher average MAP with significance, which is 2.0%, 2.2%, 2.9% higher than DQN, 0.9%, 0.5%, 1.3% higher than DHN, and 4.8%, 1.8%, 1.0% higher than HashNet on the three datasets, respectively. It indicates the effectiveness and the advantage of proposed fine pairwise similarity, which can preserve more fine-grained semantic similarity than the coarse similarity.

We also evaluate the performance of these methods on IAPRTC12. The results in Table VII show that, the proposed method achieves the best performance among all methods. The number of labels in IAPRTC12 is 276, which is much larger than NUS-WIDE's 21, Flickr's 38 and VOC2012's 20. The above results simply indicate that the proposed method can be effective in handling multi-label images with a small or large number of labels.

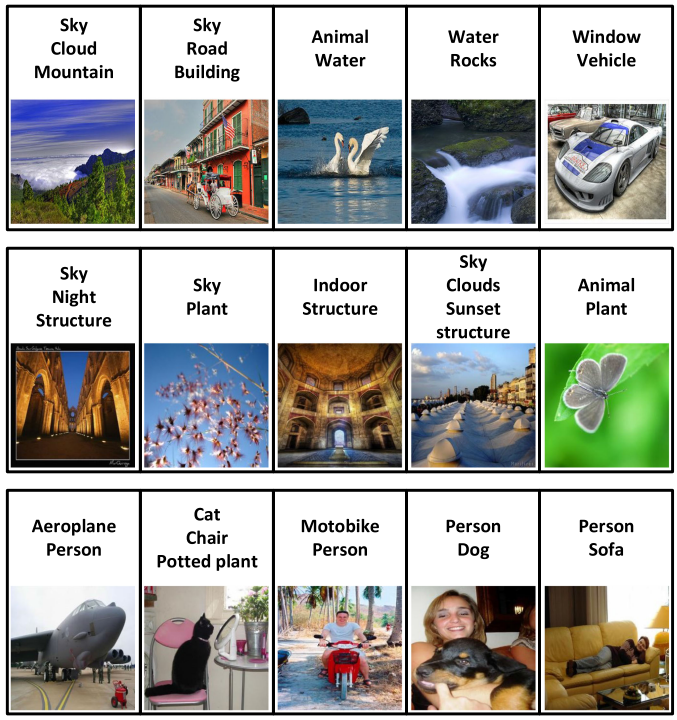


Fig. 3. Some sample images. From top row to the bottom row are the samples from NUS-WIDE, Flickr, and VOC2012, respectively. The labels have been given for each image as provided by the datasets.

Figure 4 shows the ACG, NDCG, and precision curves of compared hashing methods w.r.t. different numbers of top returned images with 12, 24, 36 and 48 bits on NUS-WIDE, respectively. On these metrics, the advantage of the proposed method is not great on 12 bits compared to deep hashing methods, DHN and DMSSPH. It may be because that a shorter code is less effective in representing the semantic similarity of multi-label images in a large-scale dataset. When the code length increases, the performance of the proposed IDHN improves rapidly and shows obvious advantage than other compared methods. DLBHC shows the worse results among these deep hashing methods, since it is essentially a classification task using the class label as supervised information rather than retrieval task to preserve the semantic similarity.

Similarly, Figure 5 and 6 show the ACG, NDCG and precision curves on the Flickr and VOC2012, respectively. It can be seen that, the proposed method achieves the highest performance on the two datasets. On Flickr, the proposed IDHN obtains significantly higher performance than the other methods. On VOC2012, among the comparison methods, HashNet achieves the most comparative results to IDHN, especially on 12 bits and 24 bits. However, when the hash code increases to 36 bits or 48 bits, the proposed IDHN holds an obvious higher performance than HashNet.

According to the definition of MAP, pairwise images that share at least one common object label will be considered as relevant images, and no more comparisons of fine-grained semantic relation between these images are included, which may not stay in step with user demand in multi-label image retrieval.

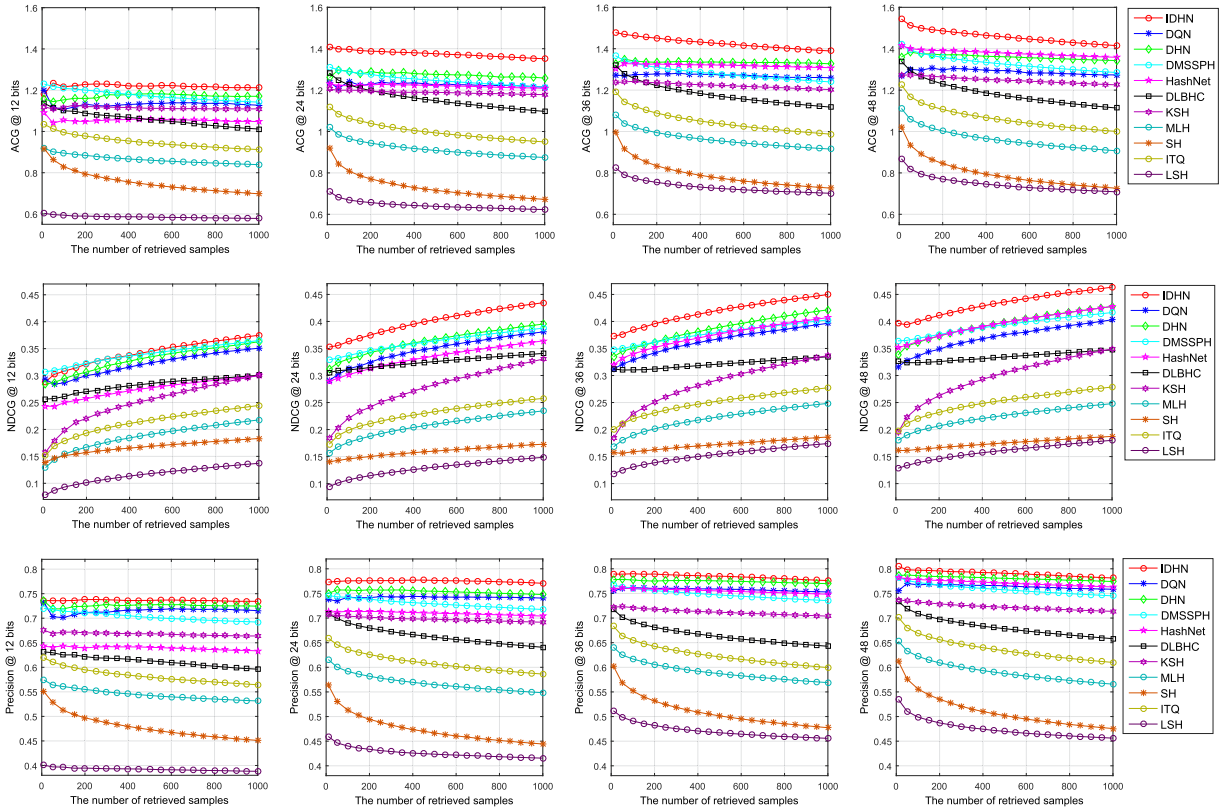


Fig. 4. Performance of different methods on the NUS-WIDE dataset. From top to bottom, there are ACG, NDCG and precision curves w.r.t. different top returned samples with hash codes of 12, 24, 36 and 48 bits, respectively.

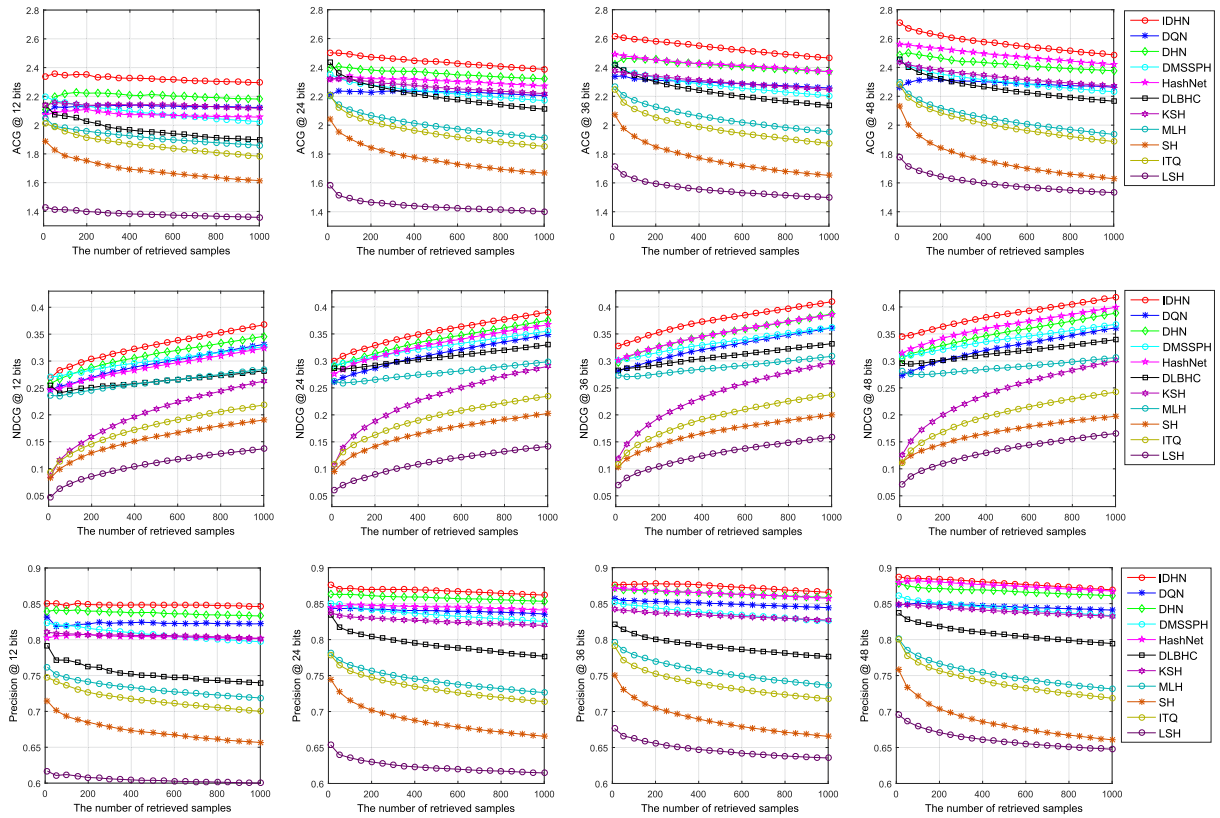


Fig. 5. Performance of different methods on the Flickr dataset. From top to bottom, there are ACG, NDCG and precision curves w.r.t. different top returned samples with hash codes of 12, 24, 36 and 48 bits, respectively.

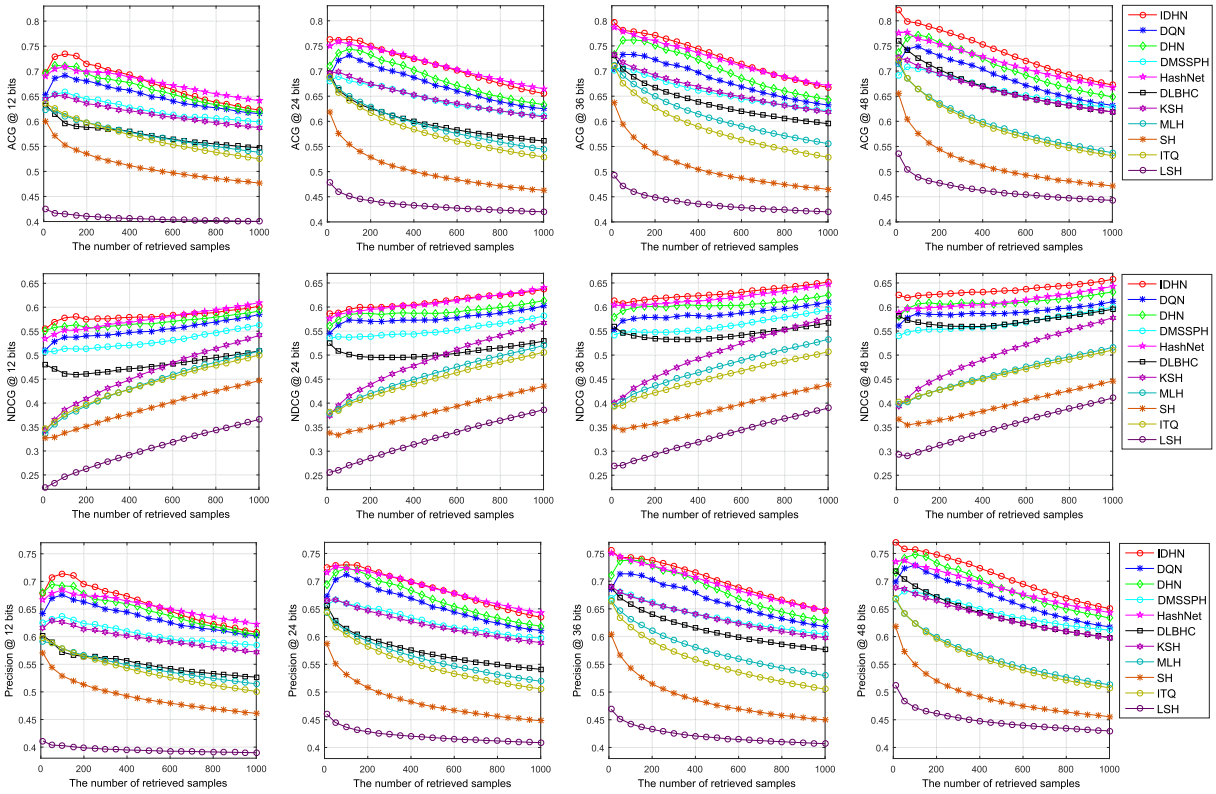


Fig. 6. Performance of difference methods on the VOC2012 dataset. From top to bottom, there are ACG, NDCG and precision curves w.r.t. different number of top returned samples with hash codes of 12, 24, 36 and 48 bits, respectively.

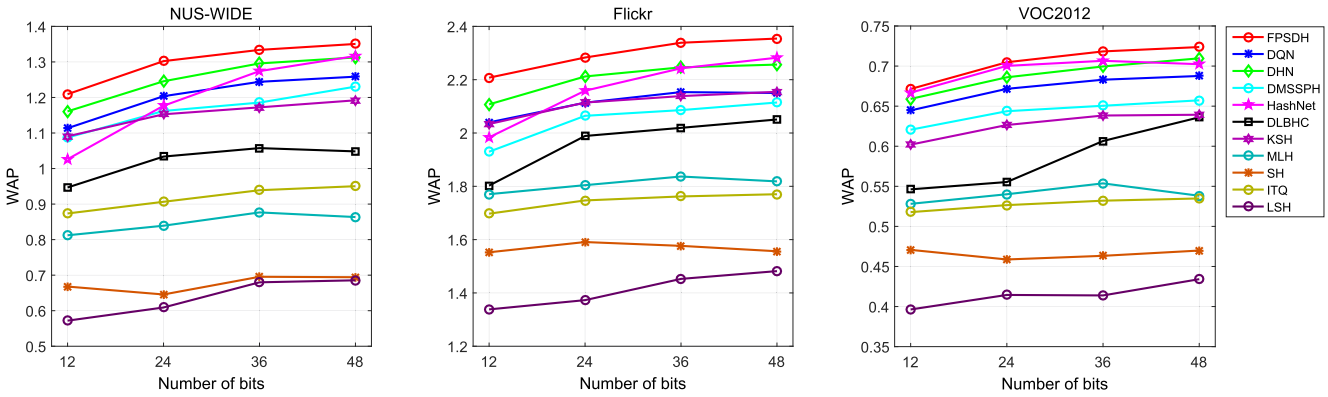


Fig. 7. Comparison of the IDHN method and other compared methods on $WAP@5000$ results.

We hope that high-quality retrieval results should have as more shared class labels as possible in the nearest retrieval image. Therefore, we also use WAP to measure the average number of shared class labels among these retrieved similar images. Figure 7 show the results of WAP for different numbers of bits. It can be seen that, although the range of WAP on the three datasets are very different, the WAPs of IDHN are stably higher than that of the comparison methods.

We also compare the proposed method with another four deep hashing methods, DPSH, DSDH, DSRH and DTSH. According to the results in Table VIII, the proposed method IDHN outperforms the two pairwise-similarity-based methods DPSH and DSDH with a significant margin. It indicates that, the soft similarity is helpful for learning high-quality hash codes. IDHN also achieves more robust results than the two triplet-based methods DSRH and DTSH on the four datasets at different code lengths.

3) Comparison With Different Settings: To justify the necessary of using the soft similarity definition and joint loss function, we conduct some comparison experiments. Table IX shows the results of MAP, WAP, ACG and NDCG metrics of the IDHN and its modifications, respectively. IDHN-fine-ce and IDHN-fine-mse are both trained under supervision of our proposed pairwise quantified similarity, where the difference between them is that IDHN-fine-ce only uses the cross entropy loss

TABLE VIII
RESULTS OF MAP AT DIFFERENT NUMBERS OF BITS ON FOUR DATASETS

Methods	NUS-WIDE			Flickr			VOC2012			IAPRTC12		
	12-bit	24-bit	48-bit	12-bit	24-bit	48-bit	12-bit	24-bit	48-bit	12-bit	24-bit	48-bit
IDHN	**0.7296**	**0.7586**	**0.7692**	**0.8327**	**0.8469**	**0.8515**	**0.6561**	**0.6845**	**0.7032**	**0.5495**	**0.5697**	**0.5859**
DPSH [34]	0.7088	0.7461	0.7606	0.8150	0.8369	0.8467	0.6123	0.6299	0.6461	0.5442	0.5548	0.5607
DSDH [32]	0.6333	0.6918	0.7317	0.7637	0.7824	0.8195	0.5913	0.6343	0.6494	0.5120	0.5458	0.5620
DSRH [25]	0.7029	0.7156	0.7321	0.8002	0.8140	0.8238	0.6518	0.6618	0.6675	0.5265	0.5397	0.5475
DTSH [33]	0.6972	0.7132	0.7462	0.8174	0.8283	0.8413	0.6552	0.6749	0.6795	0.5274	0.5452	0.5539

TABLE IX
RESULTS OF MAP, WAP, ACG AND NDCG AT DIFFERENT NUMBERS OF BITS ON NUS-WIDE, FLICKR AND VOC2012 DATASETS

Methods	NUS-WIDE				Flickr				VOC2012			
	12-bit	24-bit	36-bit	48-bit	12-bit	24-bit	36-bit	48-bit	12-bit	24-bit	36-bit	48-bit
MAP												
IDHN	0.7296	0.7586	0.7639	0.7692	0.8327	0.8469	0.8490	0.8515	0.6561	0.6874	0.6991	0.7032
IDHN-fine-ce	0.7237	0.7523	0.7601	0.7639	0.8252	0.8323	0.8406	0.8419	0.6603	0.6867	0.7022	0.7064
IDHN-fine-mse	0.7301	0.7553	0.7604	0.7641	0.8284	0.8373	0.8400	0.8439	0.6450	0.6712	0.6777	0.6803
IDHN-coarse-ce	0.7203	0.7413	0.7469	0.7598	0.8234	0.8342	0.8360	0.8401	0.6329	0.6630	0.6699	0.6709
IDHN-coarse-mse	0.6801	0.6921	0.7086	0.7116	0.7975	0.8045	0.8360	0.8401	0.6097	0.6359	0.6376	0.6398
IDHN-GoogLeNet	0.7512	0.7696	0.7738	0.7796	0.8552	0.8653	0.8668	0.8697	0.7499	0.7741	0.7841	0.7898
IDHN-VGG19	**0.7638**	**0.7769**	**0.7851**	**0.7884**	**0.8746**	**0.8830**	**0.8834**	**0.8843**	**0.7706**	**0.7842**	**0.7926**	**0.8020**
WAP												
IDHN	2.2076	2.2832	2.3386	2.3509	1.2092	1.3022	1.3339	1.3506	0.6714	0.7048	0.7182	0.7237
IDHN-fine-ce	2.2085	2.2662	2.3049	2.3204	1.2573	1.2906	1.3120	1.3261	0.6783	0.7061	0.7223	0.7267
IDHN-fine-mse	2.2141	2.2666	2.2845	2.3038	1.2467	1.3052	1.3248	1.3342	0.6612	0.6882	0.6915	0.6981
IDHN-coarse-ce	2.1426	2.1918	2.2134	2.2317	1.1822	1.2348	1.2628	1.2954	0.6473	0.6787	0.6859	0.6869
IDHN-coarse-mse	2.0046	2.0500	2.0592	2.0516	1.0523	1.0769	1.1198	1.1231	0.6235	0.6507	0.6521	0.6545
IDHN-GoogLeNet	2.2922	2.3302	2.3771	2.3879	1.2687	**1.3212**	1.3364	**1.3524**	0.7699	0.7956	0.8061	0.8119
IDHN-VGG19	**2.3422**	**2.4027**	**2.4357**	**2.4362**	**1.2882**	1.3048	**1.3405**	1.3430	**0.7907**	**0.8066**	**0.8160**	**0.8266**
ACG												
IDHN	2.0376	2.0775	2.0977	2.0919	1.1560	1.1942	1.2110	1.2135	0.5426	0.5555	0.5583	0.5586
IDHN-fine-ce	1.9838	2.0185	2.0362	2.0441	1.1498	1.1625	1.1750	1.1824	0.5483	0.5516	0.5563	0.5546
IDHN-fine-mse	2.0260	2.0503	2.0630	2.0680	1.1634	1.2003	1.2100	1.2118	0.5468	0.5507	0.5503	0.5517
IDHN-coarse-ce	1.9731	1.9859	2.0009	2.0033	1.1444	1.1770	1.1902	1.2035	0.5362	0.5432	0.5491	0.5501
IDHN-coarse-mse	1.8735	1.8821	1.8836	1.8861	1.0202	1.0372	1.0753	1.0798	0.5251	0.5312	0.5321	0.5334
IDHN-GoogLeNet	2.1277	2.1401	2.1827	2.1856	**1.2021**	**1.2077**	1.2163	1.2269	0.5702	0.5701	0.5733	0.5738
IDHN-VGG19	**2.1575**	**2.2227**	**2.2436**	**2.2458**	1.1988	1.2075	**1.2268**	**1.2282**	**0.5740**	**0.5785**	**0.5767**	**0.5816**
NDCG												
IDHN	0.5542	0.5705	0.5795	0.5786	0.5240	0.5602	0.5716	0.5780	0.7218	0.7502	0.7604	0.7641
IDHN-fine-ce	0.5440	0.5591	0.5672	0.5699	0.5407	0.5545	0.5623	0.5678	0.7445	0.7598	0.7674	0.7667
IDHN-fine-mse	0.5504	0.5621	0.5662	0.5705	0.5339	0.5593	0.5663	0.5687	0.7233	0.7355	0.7393	0.7415
IDHN-coarse-ce	0.5408	0.5484	0.5531	0.5559	0.5102	0.5330	0.5440	0.5581	0.7006	0.7207	0.7257	0.7286
IDHN-coarse-mse	0.5046	0.5095	0.5101	0.5108	0.4429	0.4527	0.4754	0.4795	0.6830	0.6855	0.6976	0.6989
IDHN-GoogLeNet	0.5742	0.5801	0.5925	0.5984	**0.5603**	**0.5738**	0.5807	**0.5876**	0.7890	0.7972	.8030	0.8050
IDHN-VGG19	**0.5892**	**0.6161**	**0.6211**	**0.6238**	0.5600	0.5692	**0.5834**	0.5845	**0.7949**	**0.8082**	**0.8139**	**0.8201**

and IDHN-fine-mse only uses mean square error loss. As a contrast, IDHN-coarse-ce and IDHN-coarse-mse are trained under supervision of coarse pairwise similarity that is adapted by most deep hashing methods at present. IDHN-coarse-ce only uses the cross entropy loss and IDHN-coarse-mse only uses mean square error loss.

Comparing IDHN-fine-ce with IDHN-coarse-ce, or IDHN-fine-mse with IDHN-coarse-mse, we can observe that using the pairwise quantified similarity as the supervised information can achieve results with higher semantic similarity than that obtained by the coarse similarity definition, whether adapting the cross entropy loss or mean square error loss. To some extent, the value of WAP and ACG can reflect the degree of shared labels between multi-label images. That is, the larger the value the larger the average number of shared labels over the whole dataset. It also means that, there are more complex semantic relation between images. Although IDHN-fine-ce has achieved outstanding performance on VOC2102 dataset, it does not show advantage on other two datasets – NUS-WIDE and Flickr, while there are complicated and abundant semantic relation on these two image datasets than VOC2012. It is interesting that with more fine-grained similarity information, deep hashing model using mean-square-error loss presents comparable performance compared to deep hashing model using cross-entropy loss, especially on dataset of complex semantic relations. The proposed IDHN method combines the advantages of cross entropy loss and mean square error loss, and shows sufficient powerful and robust performance on the three multi-label image datasets.

For fair comparison, all the above experiments are conducted based on AlexNet. We also extend our method to other deep network bases–GoogLeNet and VGG19, both of which achieve more accurate results than AlexNet on the ImageNet competition, to explore the extensibility of our hashing strategy. We denote these two modifications as 'IDHN-GoogLeNet' and 'IDHN-VGG19', respectively. From Table IX we can see that, with more powerful network bases, IDHN achieves improved performance on all these metrics, which indicates a good transfer capability of the proposed deep hashing strategy.

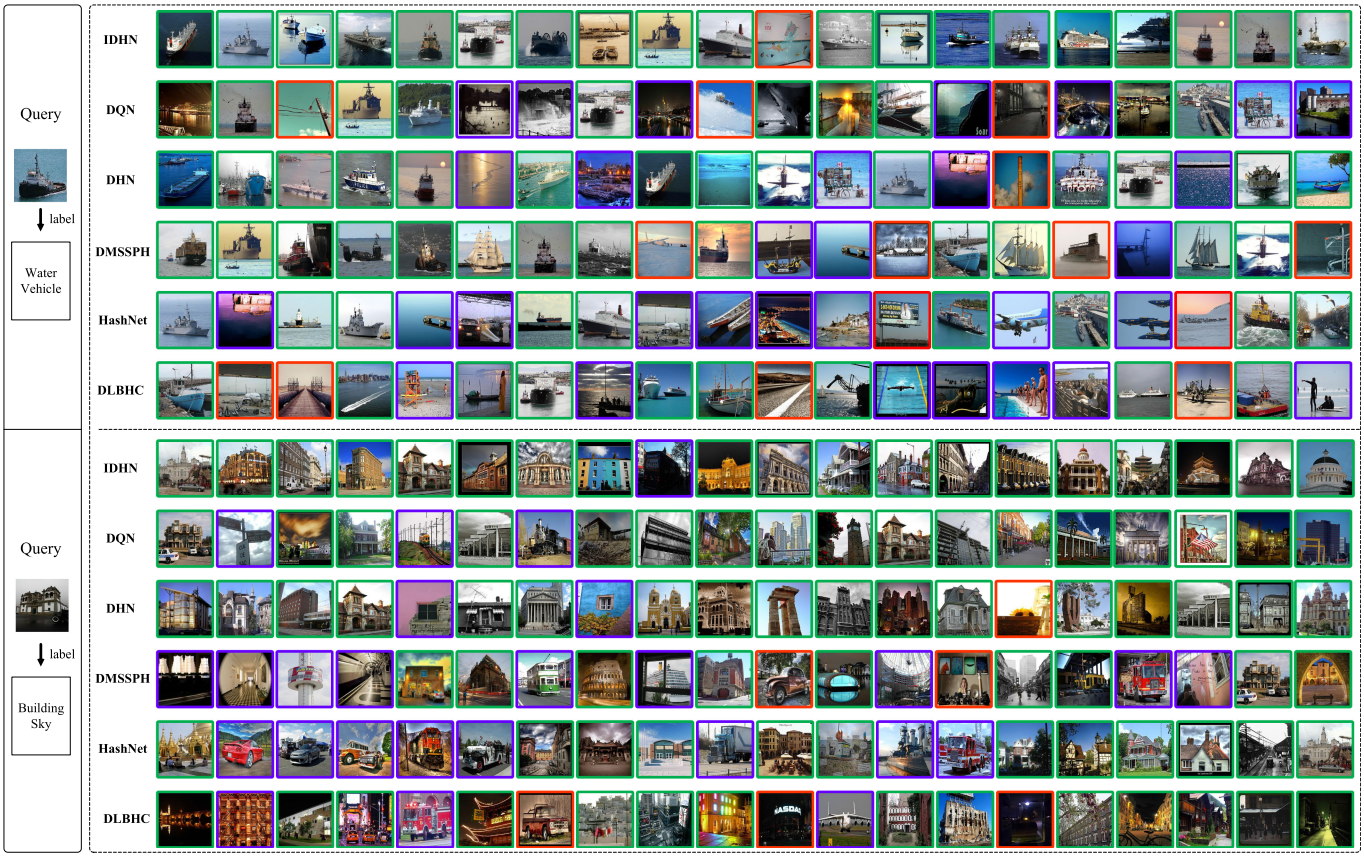


Fig. 8. Top 20 retrieved images of the proposed IDHN and five competing deep hashing methods using the Hamming ranking on 48-bit hash codes. The green box indicates that the retrieved image contains all the object classes in the query image, the blue box indicates the retrieved image partially contains the target classes, and the red box indicates the retrieved image does not contain any object classes in the query image.

4) Top Retrieval Results: Figure 8 shows some retrieval samples of six deep learning methods according to the ascending Hamming ranking. We mark the retrieval image with green box that includes all object classes in query image, blue box that includes part of the object classes, and red box which does not include any object classes in the query image. The first query image contains two semantic labels: water and vehicle. We can see that, among these six deep hashing methods, IDHN shows the best suitability between the retrieval images and query images, since the IDHN has the least incorrect retrieval (marked by red box) in top-20 retrieval results. The second query images contains two semantic labels: building and sky. On the top-20 retrieval images of each method, the results of IDHN are more similar to query images from the perspective of human vision. The top-20 results of DHN, DMSSPH and DLBHC have incorrect retrieval results, and the top-20 results of DQN and HashNet include some low similarity results, as denoted by the blue-box images. The results show the advantage of the proposed method for multi-label image retrieval.

V. Conclusion

In this paper, a novel deep hashing method was proposed for multi-label image retrieval, in which a quantified similarity definition was introduced to measure the fine-grained pairwise similarity. Compared with the traditional pairwise similarity, this fine-grained pairwise similarity can more effectively encode the information of fine-grained multi-label images. Based on the proposed pairwise similarity, a robust pairwise-similarity loss combining the cross-entropy loss and the mean-square-error loss was constructed for effective similarity-preserving learning. In addition, a quantization loss was introduced to control the quality of hashing. Extensive experiments on four multi-label datasets demonstrated that, the proposed IDHN outperformed the competing methods and achieved an effective feature learning and hash-code learning in the multi-label image retrieval.

References

[1] D. Tao, X. Tang, X. Li, and Y. Rui, "Direct kernel biased discriminant analysis: A new content-based image retrieval relevance feedback algorithm," *IEEE Trans. Multimedia*, vol. 8, no. 4, pp. 716–727, Aug. 2006.

[2] J. Wang, H. T. Shen, J. Song, and J. Ji, "Hashing for similarity search: A survey," 2014, *arXiv:1408.2927*.

[3] Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing," in *Proc. Advances Neural Inf. Process. Syst.*, 2009, pp. 1753–1760.

[4] B. Kulis and T. Darrell, "Learning to hash with binary reconstructive embeddings," in *Proc. Advances Neural Inf. Process. Syst.*, 2009, pp. 1042–1050.

[5] J. Wang, S. Kumar, and S.-F. Chang, "Semi-supervised hashing for scalable image retrieval," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2010, pp. 3424–3431.

[6] M. Norouzi and D. M. Blei, "Minimal loss hashing for compact binary codes," in *Proc. Int. Conf. Mach. Learn.*, 2011, pp. 353–360.

[7] W. Liu, J. Wang, R. Ji, Y.-G. Jiang, and S.-F. Chang, "Supervised hashing with kernels," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2012, pp. 2074–2081.

[8] Y. Gong, S. Lazebnik, A. Gordo, and F. Perronnin, "Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 12, pp. 2916–2929, Dec. 2013.

[9] C. Li, Q. Liu, J. Liu, and H. Lu, "Ordinal distance metric learning for image ranking," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 7, pp. 1551–1559, Jul. 2015.

[10] L. Liu and L. Shao, "Sequential compact code learning for unsupervised image hashing," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 27, no. 12, pp. 2526–2536, Dec. 2016.

[11] G. Jie, T. Liu, Z. Sun, D. Tao, and T. Tan, "Supervised discrete hashing with relaxation," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 3, pp. 608–617, Mar. 2018.

[12] Q. Liu *et al.*, "Reversed spectral hashing," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 6, pp. 2441–2449, Jun. 2018.

[13] F. Shen *et al.*, "Unsupervised deep hashing with similarity-adaptive and discrete optimization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 12, pp. 3034–3044, Dec. 2018.

[14] X. Zhou *et al.*, "Graph convolutional network hashing," *IEEE Trans. Cybern.*, pp. 1–13, 2018, doi: 10.1109/TCYB.2018.2883970.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Advances Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[16] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[17] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2015, pp. 1–9.

[18] Q. Zou, L. Ni, T. Zhang, and Q. Wang, "Deep learning based feature selection for remote sensing scene classification," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 11, pp. 2321–2325, Nov. 2015.

[19] C. Szegedy, A. Toshev, and D. Erhan, "Deep neural networks for object detection," in *Proc. Advances Neural Inf. Process. Syst.*, 2013, pp. 2553–2561.

[20] Y. Sun, Y. Chen, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," in *Proc. Advances Neural Inf. Process. Syst.*, 2014, pp. 1988–1996.

[21] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2015, pp. 3431–3440.

[22] J. Deng *et al.*, "Large-scale object classification using label relation graphs," in *Proc. Eur. Conf. Comput. Vision*, 2014, pp. 48–64.

[23] Q. Zou *et al.*, "Deepcrack: Learning hierarchical convolutional features for crack detection," *IEEE Trans. Image Process.*, vol. 28, no. 3, pp. 1498–1512, Mar. 2019.

[24] R. Xia, Y. Pan, H. Lai, C. Liu, and S. Yan, "Supervised hashing for image retrieval via image representation learning," in *Proc. AAAI Conf. Artif. Intell.*, vol. 1, 2014, pp. 2156–2162.

[25] F. Zhao, Y. Huang, L. Wang, and T. Tan, "Deep semantic ranking based hashing for multi-label image retrieval," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2015, pp. 1556–1564.

[26] H. Lai, Y. Pan, Y. Liu, and S. Yan, "Simultaneous feature learning and hash coding with deep neural networks," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2015, pp. 3270–3278.

[27] 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 Trans. Image Process.*, vol. 24, no. 12, pp. 4766–4779, Dec. 2015.

[28] H. Zhu, M. Long, J. Wang, and Y. Cao, "Deep hashing network for efficient similarity retrieval," in *Proc. AAAI Conf. Artif. Intell.*, 2016, pp. 2415–2421.

[29] Y. Cao, M. Long, J. Wang, H. Zhu, and Q. Wen, "Deep quantization network for efficient image retrieval," in *Proc. AAAI Conf. Artif. Intell.*, 2016, pp. 3457–3463.

[30] H. Liu, R. Wang, S. Shan, and X. Chen, "Deep supervised hashing for fast image retrieval," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2016, pp. 2064–2072.

[31] Z. Cao, M. Long, J. Wang, and P. S. Yu, "Hashnet: Deep learning to hash by continuation," 2017, *arXiv:1702.00758*.

[32] Q. Li, Z. Sun, R. He, and T. Tan, "Deep supervised discrete hashing," 2017, *arXiv:1705.10999*.

[33] X. Wang, Y. Shi, and K. M. Kitani, "Deep supervised hashing with triplet labels," in *Proc. Asian Conf. Comput. Vision*, 2016, pp. 70–84.

[34] W. Li, S. Wang, and W. Kang, "Feature learning based deep supervised hashing with pairwise labels," in *Proc. Int. Joint Conf. Artif. Intell.*, 2016, pp. 1711–1717.

[35] Q.-Y. Jiang and W.-J. Li, "Deep cross-modal hashing," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2017, pp. 3232–3240.

[36] K. He, F. Cakir, S. Adel Bargal, and S. Sclaroff, "Hashing as tie-aware learning to rank," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2018, pp. 4023–4032.

[37] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, "Locality-sensitive hashing scheme based on p-stable distributions," in *Proc. Annu. Symp. Comput. Geometry*, 2004, pp. 253–262.

[38] P. Li, M. Wang, J. Cheng, C. Xu, and H. Lu, "Spectral hashing with semantically consistent graph for image indexing," *IEEE Trans. Multimedia*, vol. 15, no. 1, pp. 141–152, Jan. 2013.

[39] Q. Ning, J. Zhu, Z. Zhong, S. C. H. Hoi, and C. Chen, "Scalable image retrieval by sparse product quantization," *IEEE Trans. Multimedia*, vol. 19, no. 3, pp. 586–597, Mar. 2017.

[40] S. Ercoli, M. Bertini, and A. D. Bimbo, "Compact hash codes for efficient visual descriptors retrieval in large scale databases," *IEEE Trans. Multimedia*, vol. 19, no. 11, pp. 2521–2532, Nov. 2017.

[41] X. Lu, X. Zheng, and X. Li, "Latent semantic minimal hashing for image retrieval," *IEEE Trans. Image Process.*, vol. 26, no. 1, pp. 355–368, Jan. 2017.

[42] L. K. Huang, Q. Yang, and W. S. Zheng, "Online hashing," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29 no. 6, pp. 2309–2322, Jun. 2018.

[43] H. Xia, S. C. H. Hoi, R. Jin, and P. Zhao, "Online multiple kernel similarity learning for visual search," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 3, pp. 536–549, Mar. 2014.

[44] J. Liang, Q. Hu, W. Wang, and Y. Han, "Semisupervised online multikernel similarity learning for image retrieval," *IEEE Trans. Multimedia*, vol. 19, no. 5, pp. 1077–1089, May 2017.

[45] Q. Zou, L. Ni, Q. Wang, Q. Li, and S. Wang, "Robust gait recognition by integrating inertial and RGBD sensors," *IEEE Trans. Cybern.*, vol. 48, no. 4, pp. 1136–1150, Apr. 2018.

[46] K. Lin, H. F. Yang, J. H. Hsiao, and C. S. Chen, "Deep learning of binary hash codes for fast image retrieval," in *Proc. Comput. Vision Pattern Recognit. Workshops*, 2015, pp. 27–35.

[47] H. Lai, P. Yan, X. Shu, Y. Wei, and S. Yan, "Instance-aware hashing for multi-label image retrieval," *IEEE Trans. Image Process.*, vol. 25, no. 6, pp. 2469–2479, Jun. 2016.

[48] D. Wu, Z. Lin, B. Li, M. Ye, and W. Wang, "Deep supervised hashing for multi-label and large-scale image retrieval," in *Proc. Int. Conf. Multimedia Retrieval*, 2017, pp. 150–158.

[49] R. Baeza-Yates *et al.*, *Modern Information Retrieval*. vol. 463, New York, NY, USA: ACM Press, 1999.

[50] K. Jarvelin and J. Kekalainen, "Cumulated gain-based evaluation of IR techniques," *ACM Trans. Inf. Syst.*, vol. 20, no. 4, pp. 422–446, 2002.

[51] A. Sablayrolles, M. Douze, N. Usunier, and H. Jégou, "How should we evaluate supervised hashing?" in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2017, pp. 1732–1736.

[52] T.-S. Chua *et al.*, "NUS-WIDE: A real-world web image database from National University of Singapore," in *Proc. ACM Int. Conf. Image Video Retrieval*, 2009, pp. 1–9.

[53] W. Liu, J. Wang, S. Kumar, and S.-F. Chang, "Hashing with graphs," in *Proc. Int. Conf. Mach. Learn.*, 2011, pp. 1–8.

[54] M. J. Huiskes and M. S. Lew, "The MIR Flickr retrieval evaluation," in *Proc. ACM Int. Conf. Multimedia Inf. Retrieval*, 2008, pp. 39–43.

[55] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *Int. J. Comput. Vision*, vol. 88, no. 2, pp. 303–338, 2010.

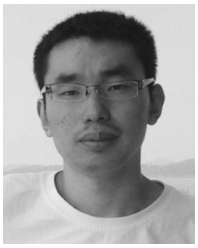
[56] H. J. Escalante *et al.*, "The segmented and annotated IAPR TC-12 benchmark," *Comput. Vision Image Understanding*, vol. 114, no. 4, pp. 419–428, 2010.

[57] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, 2015, pp. 815–823.

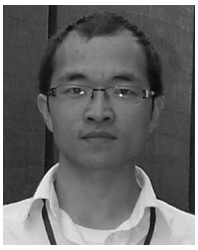
[58] M. Abadi *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," 2016, *arXiv:1603.04467*.

[59] Y. Yuan, Habir: Hashing baseline for image retrieval, 2019. [Online]. Available: https://github.com/willard-yuan/hashing-baseline-for-image-retrieval

[60] K. Jarvelin and J. Kekalainen, "IR evaluation methods for retrieving highly relevant documents," in *Proc. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2000, pp. 41–48.

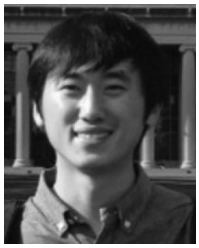


Zheng Zhang received the B.S. degree in computer science from Wuhan University, Wuhan, China, in 2015, and is now working toward the master's degree at the School of Computer Science, Wuhan University. His research interests include deep learning and its applications in image classification and retrieval. Mr. Zhang was the recipient of the first prize in "China Undergraduate Contest in Internet of Things" in 2015, and the second prize in "China Graduate Contest on Application, Design and Innovation of Mobile-Terminal" in 2016.

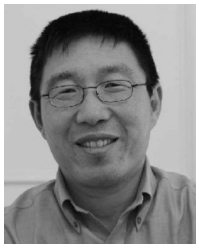


Qin Zou (M'13–SM'19) received the B.E. degree in information engineering and the Ph.D. degree in computer vision from Wuhan University, Wuhan, China, in 2004 and 2012, respectively. From 2010 to 2011, he was a visiting Ph.D. student at the Computer Vision Lab, University of South Carolina, Columbia, SC, USA. Currently, he is an Associate Professor with the School of Computer Science, Wuhan University. His research activities include computer vision, pattern recognition, and machine learning. Prof. Zou is a member of the ACM. He is a co-recipient of the National Technology Invention Award of China 2015.

Yuewei Lin received the B.S. degree in optical information science from Sichuan University, Chengdu, China; the M.E. degree in optical engineering from Chongqing University, Chongqing, China; and the Ph.D. degree in computer science and engineering from the University of South Carolina, Columbia, SC, USA. He is currently an Associate Scientist with Brookhaven National Laboratory, Upton, NY, USA. His research interests include computer vision, machine learning, image/video analysis, and the scientific data applications.



Long Chen received the B.Sc. degree in communication engineering and the Ph.D. degree in signal and information processing from Wuhan University, Wuhan, China, in 2007 and in 2013, respectively. From 2010 to 2012, he was visiting Ph.D. student at the National University of Singapore, Singapore. He is currently an Associate Professor with the School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China. His areas of interest include visual perception and its applications.



Song Wang (M'02–SM'13) received the Ph.D. degree in electrical and computer engineering from the University of Illinois at Urbana-Champaign (UIUC), Champaign, IL, USA, in 2002. From 1998 to 2002, he also worked as a Research Assistant with the Image Formation and Processing Group, Beckman Institute of UIUC. In 2002, he was with the Department of Computer Science and Engineering, University of South Carolina, Columbia, SC, USA, where he is currently a Professor. His research interests include computer vision, medical image processing, and machine learning. Prof. Wang is currently serving as an Associate Editor of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE and an Associate Editor of Pattern Recognition Letters. He is a senior member of the IEEE Computer Society.