Deep Semantic Reconstruction Hashing for Similarity Retrieval

Yunbo Wang, Xianfeng Ou, Jian Liang, and Zhenan Sun, *Senior Member, IEEE*

***Abstract*— Hashing has shown enormous potentials in preserving semantic similarity for large-scale data retrieval. Existing methods widely retain the similarity within two binary codes towards their discrete semantic affinity, i.e., 1 or −1. However, such a discrete reconstruction approach has obvious drawbacks. First, two unrelated dissimilar samples would have similar binary codes when both of them are the most dissimilar with an anchor sample. Second, the fine-grained semantic similarity cannot be shown in the generated binary codes among data with multiple semantic concepts. Furthermore, existing approaches generally adopt a point-wise error-minimizing strategy to enforce the real-valued codes close to its associated discrete codes, resulting in the well-learned paired semantic similarity being unintentionally damaged when performing quantization. To address these issues, we propose a novel deep hashing method with pairwise similarity-preserving quantization constraint, termed Deep Semantic Reconstruction Hashing (DSRH), which defines a high-level semantic affinity within each data pair to learn compact binary codes. Specifically, DSRH is expected to learn the specific binary codes whose similarity can reconstruct their high-level semantic similarity. Besides, we adopt a pairwise similarity-preserving quantization constraint instead of the traditional point-wise quantization technique, which is conducive to maintain the well-learned paired semantic similarity when performing quantization. Extensive experiments are conducted on four representative image retrieval benchmarks, and the proposed DSRH outperforms the state-of-the-art deep-learning methods with respect to different evaluation metrics.**

Manuscript received July 10, 2019; revised December 23, 2019; accepted February 1, 2020. Date of publication February 18, 2020; date of current version January 7, 2021. This work was supported in part by the National Key Research and Development Program of China under Grant 2016YFB1001000, Grant 2016YFB1001001, and Grant 2017YFC0821602, and in part by the National Natural Science Foundation of China under Grant U1836217, Grant 61427811, Grant 61573360, and Grant 61721004. This article was recommended by Associate Editor Z.-J. Zha. *(Corresponding author: Zhenan Sun.)*

Yunbo Wang is with the Center for Research on Intelligent Perception and Computing, National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: wangybyz@yeah.net).

Xianfeng Ou is with the School of Science Information and Engineering, Hunan Institute of Science and Technology, Yueyang 414006, China (e-mail: ouxf@hnist.edu.cn).

Jian Liang is with the Department of Electrical and Computer Engineering, National University of Singapore, Singapore 117583 (e-mail: liangjian92@gmail.com).

Zhenan Sun is with the Center for Research on Intelligent Perception and Computing, National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with the Center for Excellence in Brain Science and Intelligence Technology, Chinese Academy of Sciences, Beijing 100190, China, and also with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: znsun@nlpr.ia.ac.cn).

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

Digital Object Identifier 10.1109/TCSVT.2020.2974768

***Index Terms*— Deep hashing, high-level semantic similarity, similarity-preserving quantization, similarity retrieval.**

I. Introduction

With the explosive growth of multimedia data in search engines and social networks, it is highly desirable that the data should be organized and indexed efficiently and accurately. As an approximate nearest neighbor (ANN) search technique, hashing [7], [46], [48], [57] has shown superior potential for dealing with the large-scale data. Generally, hashing employs a set of hashing functions to encode each data into binary codes, meanwhile preserving the similarity from original space. Based on the binary representation, the storage cost can be dramatically decreased and we can achieve constant or sub-linear search speed [27], [44], [45]. Due to the encouraging efficiency in both storage cost and search speed, more and more hashing methods are proposed for real-world similarity retrieval tasks recently [1], [5], [13], [24].

Existing hashing could be roughly classified into two categories according to the type of hashing function: data-independent hashing [7], [17], [40] and data-dependent hashing (also known as learning-based hashing) [15], [53], [55]. Local Sensitive Hashing is a typical data-independent method, which randomly generates a set of hashing functions to encode each data into binary codes. However, the learning-based hashing resorts to training data to learn more effective hashing functions. In this paper, we focus on learning-based hashing with the application to similarity retrieval.

A fruitful of learning-based hashing methods [15], [59] have been designed for efficient ANN search, where the efficiency comes from the compact binary codes that are orders of magnitude smaller than high-dimensional feature descriptors. The learning-based hashing generally includes unsupervised and supervised approaches in real-world applications. For the unsupervised hashing [8], [23], [27], [50], they need to learn the hashing functions under no ground-truth label, and the retrieval accuracy is not desirable. The supervised hashing usually construct the pairwise label or directly employ the point-wise label to learn hashing functions, showing a better retrieval result [18]. Some representative works include Minimal Loss Hashing [37], Supervised Discrete Hashing [42] and Fast Supervised Discrete Hashing [34]. Among these methods, the input data is usually represented by hand-crafted feature descriptors such as SIFT [31] and GIST [38], followed by separate projection and quantization. Research [18] demonstrates that the hand-crafted feature based hashing is suboptimal.

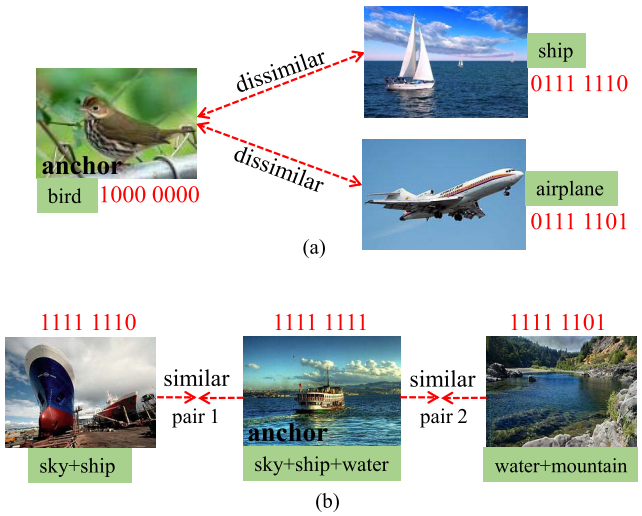


Fig. 1. (a) gives a paradox example about two dissimilar data pairs with a shared anchor. The two unrelated samples 'ship' and 'airplane' would have similar codes when they are most similar with the anchor in existing hashing methods. (b) shows another example about two similar data pairs *'pair 1'* and *'pair 2'* with a shared anchor. *'pair 1'* and *'pair 2'* have a consistent Hamming distance, but it is contradicted with sharing different number of semantic concepts. In addition, two unrelated non-anchor samples also undesignedly have similar codes.

Recently, deep learning-based hashing methods have been proposed to simultaneously learn effective feature representation and hash functions, which have shown superior performance over the hand-crafted feature based hashing methods. Specifically, deep hashing methods with pairwise labels [3], [60] generate similarity-preserving binary codes in terms of their semantic similarity. The triplet-wise affinity based deep hashing methods [18], [54], [58] obtain the relative similar codes within a triplet tuple. What's more, it proves crucial to jointly learn the similarity-preserving representation and minimize the quantization error of converting continuous representation to binary codes.

Although many supervised hashing methods have been proposed with promising results, they confront some common flaws in learning similarity-preserving representations and controlling the quantization error. In learning similarity, the supervision information (i.e., pairwise similarity) is simply constructed based on label information. Specifically, the similarity affinity is widely defined as 1 if two samples have at least a common semantic label, otherwise -1 [34]. However, the brutal discrete definition is unreasonable for effective hash learning, and it would bring in some issues: (1) two unrelated dissimilar samples have similar codes when they are the most dissimilar with the same anchor, as shown in Figure 1(a). When utilizing the affinity -1 to maximize the Hamming distance of '$bird$' and '$ship$' as well as '$bird$' and '$airplane$' to obtain dissimilar codes, it leads to an unexpected truth that the unrelated samples '$ship$' and '$airplane$' have similar codes; (2) the fine-grained semantic similarity cannot be shown in data with multiple semantic concepts, resulting in the Hamming distance of two similar pairs to be same or similar, as shown in Figure 1(b). It is observed that *'pair 1'* shares two semantic concepts and *'pair 2'* only shares one semantic concept. When utilizing the affinity 1 to minimize their Hamming distance to get similar codes, it may lead to the consistent Hamming distance in *'pair 1'* and *'pair 2'*. Furthermore, it undesignedly makes two unrelated non-anchor samples have similar codes (i.e., the first and the third image). Obviously, the discrete similarity $1/-1$ based supervised hashing is insufficient to describe the high-level semantic affinity of data pair, failing to generate compact binary codes.

Besides, quantization is a very intractable issue for learning-based hashing. Current work mainly focuses on minimizing the point-wise quantization error [5], [26] that enforces the real-valued codes close to their discrete codes. The classical point-wise error-minimizing quantization constraints include L_1-norm regularizer [19], [33], L_2-norm regularizer [21], [42] and $Tanh()$ smooth function [3], [52]. However, whatever the point-wise quantization constraint, it inevitably introduces the quantization error, and results in information loss. What's worse, it produces a larger approximation error by adopting the similarity of real-valued codes as the surrogate of that of binary codes in similarity measure. After quantization, the well-learned paired similarity is unintentionally seriously damaged.

To address these above issues, we propose a novel deep hashing method with pairwise similarity-preserving quantization constraint, termed Deep Semantic Reconstruction Hashing (DSRH), which adopts an end-to-end trainable way to learn compact binary codes. Figure 2 shows the framework of the proposed DSRH. Generally, our main contributions are summarized as follows:

- The similarity affinity of data pairs is elaborately designed and redefined to characterize their relationship in DSRH. The semantic similarity of a similar pair is continuous, and the similarity of a dissimilar pair is also not limited to a single value -1 but a local variable within a batch. Based on the redefined similarity affinity, the high-level semantics of data pairs can be fully exploited, generating more compact binary codes.
- We develop a novel pairwise similarity-preserving quantization constraint to maintain the semantic similarity of data pairs when performing quantization. Compared with conventional point-wise error-minimizing quantization schema, the proposed method enables improving the quality of binary codes and maintaining the well-learned paired semantic similarity for similarity retrieval.
- Extensive experiments are conducted on four public benchmark datasets. The retrieval results demonstrate the superiority of the proposed DSRH over the state-of-the-art supervised hashing methods.

The rest of this paper is organized as follows: Section II gives a brief review of related work about deep hashing and point-wise quantization constraint. Section III presents the procedure of the proposed DSRH. Section IV shows the details, results, and analyses of the experiment. Section V concludes the paper.

II. Related Work

Learning-based hashing has become an important research topic in multimedia retrieval, which trades off efficacy from

efficiency. In this section, we review the related works which motivate our method. They can be roughly cast into two categories below.

A. Deep Hashing

Considering the promising performance of deep learning [10], [16], [30], [36] on many computer vision tasks, e.g., image classification, object detection, and semantic segmentation, more and mare hashing works [6], [29], [39], [56] try to exploit the Convolutional Neural Networks (CNN) [11] to project images into compact binary codes. Specifically, these methods jointly perform feature representation learning and hash coding in an end-to-end manner, showing superior performance over traditional hashing methods with hand-crafted feature [46]. The first proposed deep hashing work is Convolutional Neural Network Hashing (CNNH) [51], which adopts the well-known architecture in [16] to learn discriminative and compact binary codes with pairwise constraint. CNNH consists of two stages to learn the feature representation and binary codes. Nevertheless, the feature representation cannot make feedback to hash coding and it cannot fully show the efficiency of CNNs in hash learning. Based on CNNH, Network In Network Hashing (DNNH) [18] integrates image representation and hash coding in a unified framework. Besides, DNNH employs a triplet-based ranking constraint to maximize the margin between similar data pair and dissimilar data pair, and it designs a divide-and-encode module to reduce the redundancy among binary codes. Furthermore, Deep Hashing Network (DHN) [60] is a representative pairwise deep hashing work in a unified framework. It employs a cross-entropy loss to enforce similar(dissimilar) pairs to have small(large) Hamming distance and formally controls the point-wise quantization error by a designed smooth surrogate of the L_1-norm. To better control quantization error, HashNet [3] proposes a continuous scale strategy to approximately approach the discrete binary codes, and takes into consideration class imbalance to obtain similar codes in similar data pairs. DPH [4] also takes into consideration class imbalance for supervised hashing, and integrates the prior information into obtaining compact binary codes in data pairs. Other typical deep hashing methods can be found in [19], [23], [32], [47], [49].

Among these methods above, they simply construct data pairs' similarity as the ground-truth label for supervised hash learning. Specifically, the similarity is defined as 1 if two samples share at least one semantic concept, otherwise -1. Then they expect to obtain minimal or maximal Hamming distance within a pair of codes according to the hard-assigned discrete similarity. However, the simply defined similarity cannot show the fine-grained semantic similarity among data pairs. Meanwhile, minimizing or maximizing Hamming distance within a pair of codes would unexpectedly result in two unrelated data having similar codes. Therefore, redefining the similarity affinity is necessary for effective supervised learning-based learning in the pairwise scenario.

B. Point-Wise Error-Minimizing Quantization

Quantization [5], [6], [9] is an important factor in hash learning. To make the real-valued codes close to the discrete codes, existing hashing works generally adopt the point-wise error-minimizing strategy [21], [25], [33] to narrow their gap (quantization error), and the specific manner includes L_1-norm regularizer, L_2-norm regularizer and $tanh(\cdot)$ smooth function. Such as Deep Supervised Hashing [25] adopts the L_1-norm regularizer to reduce the gap. Deep Pairwise-Supervised Hashing [21] utilizes the L_2-norm regularizer to narrow their Euclidean distance. HashNet [3] proposes a $tanh(\cdot)$ function based smooth scale technique to make the real-valued codes close to discrete codes. Deep Priority Hashing [4] also employs bi-modal Laplacian prior probability based on L_1-norm to model this gap, forcing the learned real-valued code to be assigned to $\{-1, 1\}$ with the largest probability. Besides, Deep Hashing via Discrepancy Minimization [6] attempts to transform the discrete objective over binary codes to a continuous objective over hashing functions through a Taylor expansion, reducing the quantization error.

In hash coding, the uncontrollable point-wise quantization error is inevitable. Besides, it will further produce a larger approximate error by adopting the similarity of real-valued codes as the surrogate of that of binary codes for similarity retrieval. After quantization, the well-learned paired similarity must be damaged.

III. The Proposed DSRH

Given a training set of N points $\{\boldsymbol{x}_i\}_{i=1}^n$, each data point is represented by a d-dimensional feature vector $\boldsymbol{x}_i \in R^d$. The goal of learning-based hashing is to learn a set of hashing functions $F = \{f_1, f_2, \ldots, f_k\}$, which encode each data point $\boldsymbol{x}_i$ into a compact k-bit binary codes $\boldsymbol{b}_i = F(x_i) \in \{-1, 1\}^k$. The corresponding label matrix is denoted as $T = \{\boldsymbol{t}_i\}_i^n \in R^{n \times c}$ and c denotes the number of classes. The term $\boldsymbol{t}_{ik}$ is the k-th element of $\boldsymbol{t}_i$ and $\boldsymbol{t}_{ik} = 1$ if $\boldsymbol{x}_i$ is from class k, otherwise $\boldsymbol{t}_{ik} = 0$. Then, existing hashing generally denotes the similarity affinity $s_{ij} = 1$ if two samples share at least one class label, otherwise $s_{ij} = -1$ [43].

As in [18], [33], we use linear projections followed by an element-wise transformation function as our hashing functions. Firstly, we can obtain the output of the hashing layer by linear projection, and the specific output is listed as follows:

$$\boldsymbol{h}_i = \boldsymbol{W}_H^T \boldsymbol{x}_i + \nu_H, \tag{1}$$

where $\boldsymbol{W}_H \in \mathbb{R}^{d \times k}$ denotes the weight in the hashing layer, and $\nu_H \in \mathbb{R}^{k \times 1}$ denotes the bias parameter. Obviously, the output of the hashing layer $\boldsymbol{h}_i \in \mathbb{R}^k$ is continuous value. In order to obtain discrete binary codes $\boldsymbol{b}_i \in \mathbb{R}^k$, the element-wise transformation is defined as:

$$\boldsymbol{b}_i = sign(\boldsymbol{h}_i), \tag{2}$$

where $sign(\cdot)$ denotes a sign function, i.e., $sign(x) = 1$ if $x > 0$, otherwise $sign(x) = -1$. To learn discriminative and compact binary codes, we introduce details of the proposed DSRH in the next part.

For a pair of codes $(\boldsymbol{b}_i, \boldsymbol{b}_j, s_{ij})$, their semantic similarity should be effectively preserved in Hamming space. Besides, there exists a close linear relationship between their Hamming

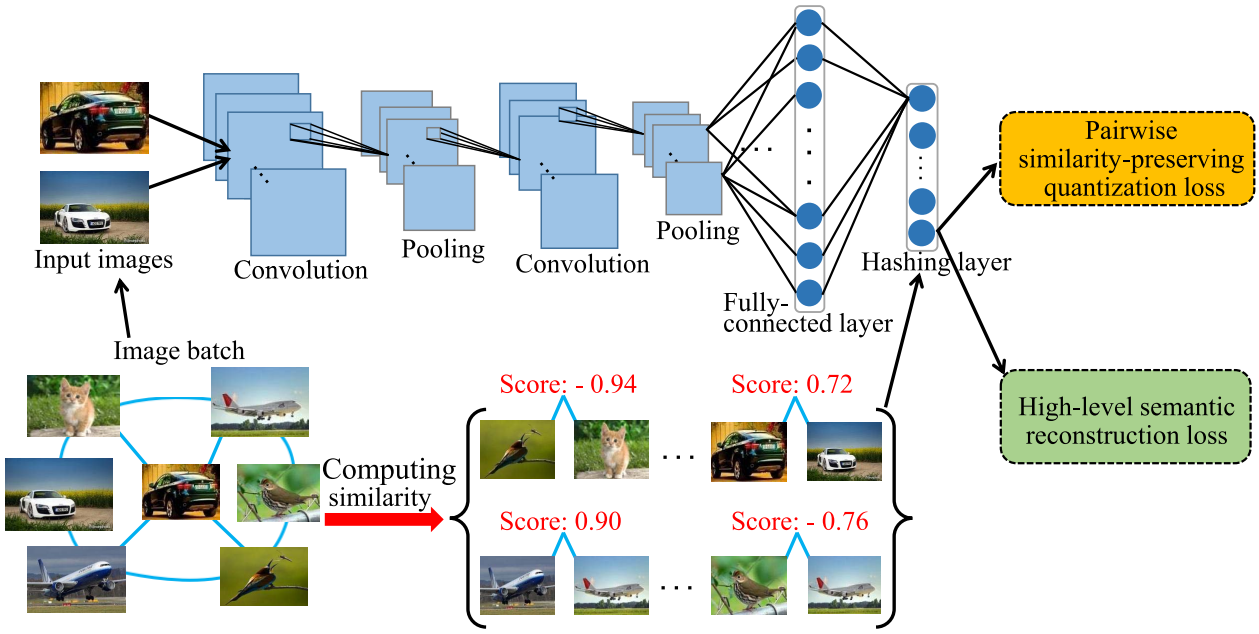


Fig. 2. An overview of the proposed deep hashing method DSRH, which accepts paired images as its input. In this framework, a deep convolutional neural network is exploited for extracting image representation, followed by a hashing layer fch with $\boldsymbol{k}$ neural units, which transforms the representation into $\boldsymbol{k}$-bit binary codes. For each pair of images, we redefine their similarity affinity to reconstruct the high-level semantic similarity in Hamming space. Meanwhile, we employ a pairwise similarity-preserving quantization constraint to maintain the well-learned paired similarity when performing quantization.

distance $D_H(\boldsymbol{b}_i, \boldsymbol{b}_i)$ and their inner-product $\boldsymbol{b}_i^T \cdot \boldsymbol{b}_j$:

$$D_H(\boldsymbol{b}_i, \boldsymbol{b}_j) = \frac{1}{2}(k - \boldsymbol{b}_i^T \cdot \boldsymbol{b}_j), \tag{3}$$

if the inner-product of two binary codes is small, their Hamming distance will be large, and vice versa. Hence in the sequel, we will the inner-product as a good surrogate of the Hamming distance to quantify the pairwise similarity, as in [4], [20]. Based on the similarity affinity s_{ij}, the general objective function of learning-based hashing can be formulated as:

$$\min \sum_{i,j} ||\boldsymbol{h}_i^T \cdot \boldsymbol{h}_j - k * s_{ij}||_1 + \eta \sum_i ||\boldsymbol{h}_i - \boldsymbol{b}_i||_1, \tag{4}$$

where $\boldsymbol{h}_i^T$ is the transposition of $\boldsymbol{h}_i$. The first term is used to learn similarity-preserving representations, and the second term is employed to narrow the point-wise quantization error. The η is a hyper-parameter for balancing the importance of the quantization error term.

Figure 2 shows the framework of the proposed DSRH, which accepts paired images as the input and processes them through the deep representations learning and hash coding. It includes a sub-network with multiple convolution/pooling layers to perform image abstraction, two fully-connected layers to obtain optimal representations, a hashing layer fch to generate k-bits binary codes. In the DSRH, we redefine the similarity of data pairs, and we perform a high-level semantic reconstruction to obtain specific binary codes centered on this similarity. Furthermore, we develop a pairwise similarity-preserving quantization term to maintain the well-learned paired similarity when performing quantization.

A. High-Level Semantic Reconstruction

As discussed in the introduction, the discrete similarity, i.e., $s_{ij} = 1$ or -1, leads to several issues: (1) it fails to capture the fine-grained semantic affinity among images with multiple semantic concepts, resulting in the generated codes becoming unified among similar data pairs and so as to have similar codes between two non-anchor samples; (2) in dissimilar pairs, two unrelated data may also have similar codes when they are the most dissimilar to the same anchor. Subsequently, it inevitably yields suboptimal results on hash coding. In order to effectively perform hash learning, we redefine the similarity affinity in the light of label information or the ratio of similar data pairs as follows:

$$\hat{s}_{ij} = \frac{m - 1 + c_{ij}}{m} s_{ij}, \tag{5}$$

where the term c_{ij} is designed by the following formula:

$$c_{ij} = \begin{cases} \dfrac{\boldsymbol{t}_i^T \cdot \boldsymbol{t}_j}{\|\boldsymbol{t}_i\| \|\boldsymbol{t}_j\|}, & s_{ij} = 1 \\ \dfrac{\|S_i^1\| + \|S_j^1\|}{\|S_i\| + \|S_j\|}, & s_{ij} = -1. \end{cases} \tag{6}$$

In Equation. (6), $\boldsymbol{t}_i$ and $\boldsymbol{t}_j$ denote the semantic label vectors of data x_i and x_j, respectively. $S_i = \{s_{ik} : \forall k\}$ is the set of data pairs which contain specific sample x_i. $S_i^1 = \{s_{ik} : s_{ik} = 1\}$ is the subset of similar data pairs with x_i. The c_{ij} will be adaptively determined in the range of $[0, 1]$. The m is a parameter, and need to meet this requirement: $m - 1 > 0$.

For similar data pairs, the similarity affinity $\hat{s}_{ij}$ is a continuous variable, which is proportional to the $cosin$ similarity of two label vectors. Meanwhile, it is expected that the value of $\hat{s}_{ij}$ would be greater than $(m - 1)/m$. It is observed if a pair of data have the same class labels, the $\hat{s}_{ij}$ just equals to 1. If a pair of data shares partial labels, the similarity value is less than 1. Based on this similarity, the generated binary codes can characterize the potential fine-grained semantic

similarity in terms of sharing semantic concepts, avoiding that all related similar pairs would have similar codes. Specifically, the number of different hash bits between two similar data can be up to $\lfloor \frac{k}{2m} \rfloor$,[1] which can be inferred by jointing Eq. (3) and (4). Besides, the minimum of $\hat{s}_{ij}$ is designed to be greater than $(m-1)/m$, which is the lower bound of similarity level, aiming to keep a certain similarity on similar data pairs at least. Therefore, the proposed similarity affinity $\frac{m-1+c_{ij}}{m}$ not only shows the fine-grained similarity affinity, but also keeps a certain similarity on similar data pair.

In addition, two partial similar sample don't have the same code, and the number of different bits is $\frac{k}{2m}(1-c_{ij})$ for similar data pair (x_i, x_j). For another similar data pair (x_i, x_p) based on x_i, the number of different bits is $\frac{k}{2m}(1-c_{ip})$. After reasoning, The different bits of x_j and x_p can be up to $\frac{k}{2m}(2-c_{ij}-c_{ip})$ at most. This makes sure a certain compatibility between x_j and x_p if they are dissimilar.

For dissimilar data pairs, the similarity affinity $\hat{s}_{ij}$ dynamically changes with the ratio of similar pairs related to data point x_i and x_j in a batch of data. The purpose is to obtain Hamming-compatible binary codes, avoiding two unrelated data having similar codes when they are the most dissimilar with an anchor. From Equations. (5) and (6), we can observe that the higher the ratio of total similar pairs $(x_i, x_p : s_{ip}=1)$ and $(x_j, x_q : s_{jq}=1)$ is, the closer the $\hat{s}_{ij}$ is to -1. When the ratio of similar pairs is higher, it is acceptable that the base dissimilar pair (x_i, x_j) can tolerate a larger Hamming distance in the light of similar data pair having similar codes. When the ratio of similar pairs is fewer, it needs to consider keeping a certain distance about other dissimilar pairs related to x_i or x_j. Thus, in order to obtain the Hamming-compatible codes, the basic data pair (x_i, x_j) only holds an appropriate Hamming distance instead of the maximized distance. Furthermore, we empirically set the value of $\hat{s}_{ij}$ being less than $-(m-1)/m$, and expect to differentiate dissimilar pairs with a supporting of minimal Hamming distance $k(1-\frac{1}{2m})$. Hence, the novel similarity of dissimilar pairs is helpful to generate Hamming-compatible binary codes.

Substituting the redefined $\hat{s}_{ij}$ into Equation. (4), we can get a novel deep reconstructive hashing to retain their high-level semantic similarity. As the term of $||\boldsymbol{h}_i^T \cdot \boldsymbol{h}_j - k * s_{ij}||_1$ might be sensitive to outliers, we adopt the normalized inner-product to increase the robustness of the hashing procedure in this study. In addition, since the exponential manner $\exp(|x|)$ can obtain a stronger gradient information for effective back propagation and parameter update, we use the exponential manner to compute a loss. The specific formulation can be obtained as follows:

$$L = \sum_{i,j} \exp\ (||\frac{\boldsymbol{h}_i^T \cdot \boldsymbol{h}_j}{k} - \hat{s}_{ij}||_1). \tag{7}$$

The above defined similarity affinity$\hat{s}_{ij}$ is a natural extension of the original hard-assigned similarity, i.e., -1 or 1. Technically, we consider the fine-grained semantic affinity between similar pairs. Meanwhile, we integrate the ratio of similar data pairs into computing a continuous similarity affinity among dissimilar pairs for semantic reconstruction.

[1] $\lfloor \cdot \rfloor$ denotes the operation of rounding down.

B. Pairwise Similarity-Preserving Quantization

Since discrete optimization of the Equation. (7) with binary codes is very challenging, the binary constraint is usually replaced with real-valued codes as in [3], [21], [42], [60]. Considering the subsequent error (quantization error) introduced by real-valued codes, existing deep hashing [20], [33] generally imposes a L_1/L_2-norm regularizer on this error, making the real-valued codes close the discrete codes [25].

Although the point-wise quantization constraint can narrow the quantization error to some extent, this uncontrollable error is inevitable. What's worse, it will produce a larger approximate error by adopting the similarity of real-valued codes as the surrogate of that of binary codes. After quantization, the well-learned paired similarity must be damaged, whereas there is no defense.

Given two real-valued vectors $\boldsymbol{h}_i$ and $\boldsymbol{h}_j$, and take their nth-dim value $\boldsymbol{h}_{in}=0.8$ and $\boldsymbol{h}_{jn}=0.9$ for example. According to the element-wise transformation function used in our work, we can obtain the corresponding binary codes $\boldsymbol{b}_{in}=1$ and $\boldsymbol{b}_{jn}=1$. Based on the given value, the point-wise quantization error in the nth-dim is $\|\boldsymbol{h}_{in}-\boldsymbol{b}_{in}\| = \|0.8-1\| = 0.2$ and $\|\boldsymbol{h}_{jn}-\boldsymbol{b}_{jn}\| = \|0.9-1\| = 0.1$, respectively. However, the approximate error in similarity measure is $\|\boldsymbol{h}_{in}\cdot\boldsymbol{h}_{jn}-\boldsymbol{b}_{in}\cdot\boldsymbol{b}_{jn}\| = 0.28$. Obviously, the inner-product operation generates a greater error in similarity measure. This verifies that the uncontrolled quantization error results in a greater approximate error in paired similarity.

In this work, to effectively maintain the well-learned paired similarity, namely, making $\boldsymbol{h}_i^T \cdot \boldsymbol{h}_j$ much more close to $\boldsymbol{b}_i^T \cdot \boldsymbol{b}_j$, we develop a pairwise similarity-preserving quantization constraint based on the exponential manner:

$$Q = \sum_{i,j} \exp\ (\frac{1}{k}||\boldsymbol{h}_i^T \cdot \boldsymbol{h}_j - \boldsymbol{b}_i^T \cdot \boldsymbol{b}_j||_1), \tag{8}$$

where $\boldsymbol{h}_i^T \cdot \boldsymbol{h}_j$ is the inner-product of real-valued codes, and $\boldsymbol{b}_i^T \cdot \boldsymbol{b}_j$ is the inner-product of discrete codes. As stated in the previous analysis, it is reasonable that we employ the inner-product of two data to describe their similarity. The above-proposed quantization loss focuses on maintaining the paired similarity, and enforces the inner-product of real-valued codes close to that of discrete codes. The final generated binary codes are more favorable for similarity retrieval.

C. Overall Objective

Integrating the Equation. (7) with Equation. (8), the final objective of the proposed DSRH is formulated as follows:

$$\begin{aligned} \min J &= \sum_{i,j}(L+\lambda Q) \\ &= \sum_{i,j}\{\exp(||\frac{\boldsymbol{h}_i^T \cdot \boldsymbol{h}_j}{k} - \hat{s}_{ij}||_1) \\ &\quad +\lambda\exp(\frac{1}{k}||\boldsymbol{h}_i^T \cdot \boldsymbol{h}_j - \boldsymbol{b}_i^T \cdot \boldsymbol{b}_j||_1)\}, \end{aligned} \tag{9}$$

where the hyper-parameter λ is used to balance the pairwise quantization constraint. On the basis of the Equation. (9), the binary codes can be obtained by jointly reconstructing high-level semantic similarity and maintaining the well-learned semantic similarity.

In training stage, we adopt a mini-batch-based strategy for updating the DSRH model. More specifically, in each iteration, we sample a mini-batch of data points from the whole training set to calculate their feed-forward loss and back-forward gradient information. Since the absolute value $|\cdot|$ is a non-differential whose derivative is difficult to compute, we adopt a smooth surrogate of the absolute function $|x| \approx \log\cosh(x)$ [12]. Then the Equation. (9) can be further formulated as:

$$\min J = \sum_{i,j}\{\exp(\log\cosh(\frac{\boldsymbol{h}_i^T \cdot \boldsymbol{h}_j}{k} - \hat{s}_{ij})) + \lambda \exp(\log\cosh(\frac{1}{k}(\boldsymbol{h}_i^T \cdot \boldsymbol{h}_j - \boldsymbol{b}_i^T \cdot \boldsymbol{b}_j)))\}. \quad (10)$$

The gradient of J with respect to $\boldsymbol{h}_i$ can be calculated as:

$$\frac{\partial J}{\partial \boldsymbol{h}_i} = \sum_{j:\hat{s}_{ij}\in S}\{\exp(U_{ij})\tanh(u_{ij})\boldsymbol{h}_j + \lambda\exp(V_{ij})\tanh(v_{ij})\boldsymbol{h}_j\} + \sum_{j:\hat{s}_{ji}\in S}\{\exp(U_{ji})\tanh(u_{ji})\boldsymbol{h}_j + \lambda\exp(V_{ji})\tanh(v_{ji})\boldsymbol{h}_j\}, \quad (11)$$

where $u_{ij} = \frac{\boldsymbol{h}_i^T \cdot \boldsymbol{h}_j}{k} - \hat{s}_{ij}$, $v_{ij} = \boldsymbol{h}_i^T \cdot \boldsymbol{h}_j - \boldsymbol{b}_i^T \cdot \boldsymbol{b}_j$, $U_{ij} = \log\cosh(u_{ij})$ and $V_{ij} = \log\cosh(v_{ij})$. As the model parameter and the feature variable before the loss function module do not directly join in the loss J calculation, we don't give their back-propagation gradient information. But we can obtain the gradient by the chain rule [16] based on the gradient information in Equation. (11). We implement the proposed approach via the open-source deep framework Caffe [14]. The standard stochastic gradient descent method (SGD) [16] is used to train the proposed method.

D. Out-of-Sample Extension

After we have completed the learning procedure, we can only get the binary codes for points in the training data. We still need to perform an out-of-sample extension to predict the binary codes for the points which do not appear in the training set. The deep hashing framework of DSRH can be naturally applied for the out-of-sample extension. For any data point x_q, we can predict its binary codes just by forward propagation:

$$\boldsymbol{b}_q = sign(\boldsymbol{h}_q). \quad (12)$$

IV. Experiments and Analysis

To evaluate the effectiveness of the proposed DSRH, extensive experiments are conducted on four benchmarks against the state-of-the-art hashing methods.

Fig. 4. Exemplar images from CIFAR-10, ImageNet, NUS-WIDE and MS-COCO datasets.

A. Datasets

CIFAR-10 is a benchmark image dataset, including 60,000 color images in 10 classes. Each class has 6,000 images in size 32×32. Following the evaluation protocol in [60], we sample 100 images per class as the query set, and the remaining images are used as the database. In addition, we sample 5,000 images (500 images per class) from the database as the training set.

ImageNet is a benchmark image dataset in 1,000 classes for Large Scale Visual Recognition Challenge (ILSVRC 2015) [41]. It contains over 1.2M images in the training set and 50K images in the validation set. Follow a slightly different evaluation protocol as HashNet [3], 100 categories with the most images are selected for the experimental evaluation. The images of these 100 categories in the training set are used as the database, and the images in the validation set are used as the query set. We further sample 100 images per class from this database as the training set.

NUS-WIDE is an image dataset containing 269,648 images from Flickr.com. Each image is associated with one or more semantics among 81 semantic concepts. Following the setting in DHN [60], the 21 most frequent concepts with 195,834 images are used for experimental evaluation. We sample 2,100 images (100 images per class) as the query set, and the remaining images are used as the database. In addition, 500 images per class are selected from the remaining images as the training set.

MS-COCO is a large-scale image dataset for recognition, segmentation and captioning task. It contains 82,783 training images and 40,504 validation images, which belong to 80 semantic concepts. In our experiment, we retain only those images which belong to the 20 most frequent concepts and remove the others, leaving 86,199 images available. We sample 100 images per class as the query set, and the remaining images are used as the database. 500 images per class are further selected from the remaining images as the training set. Figure 4 shows exemplar images of the experimental dataset.

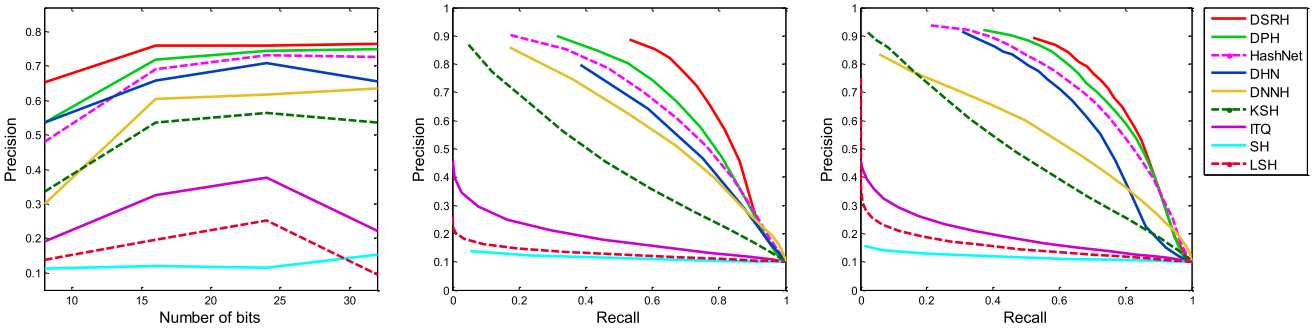
B. Experimental Setting and Protocols

The redefined paired similarity set $S = \{\hat{s}_{ij}\}$ obtained by the Equation. (5) is used as the ground truth. If $\hat{s}_{ij}$ is greater than 0, it indicates the target data pair is similar

TABLE I

COMPARISON OF RETRIEVAL MAP@ALL SCORES AND PRECISION@1,000 OF DIFFERENT METHODS ON THE CIFAR-10 DATASET

Method	CIFAR-10							
	MAP@all				Precision@1,000			
#Bits	$k = 8$	$k = 16$	$k = 24$	$k = 32$	$k = 8$	$k = 16$	$k = 24$	$k = 32$
LSH [7]	0.1280	0.1368	0.1474	0.1637	0.1584	0.1785	0.2020	0.2337
SH [50]	0.1200	0.1254	0.1215	0.1277	0.1628	0.1613	0.1553	0.1672
ITQ [9]	0.1834	0.1997	0.2035	0.2087	0.2531	0.2948	0.3065	0.3171
KSH [17]	0.3860	0.4551	0.4701	0.4914	0.4684	0.5542	0.5650	0.5853
FastH [22]	0.4190	0.5006	0.5353	0.5436	0.5088	0.6047	0.6338	0.6444
SDH [42]	0.3192	0.5026	0.5318	0.5458	0.3615	0.5918	0.6132	0.6277
CNNH [51]	0.4214	0.4381	0.4597	0.4923	0.3853	0.4474	0.4676	0.5025
DNNH [18]	0.5561	0.6041	0.5876	0.5857	0.6113	0.6701	0.6576	0.6564
DSH [25]	0.5842	0.6396	0.6545	0.6654	0.6541	0.6642	0.7289	0.7442
DHN [60]	0.5918	0.6554	0.6586	0.6601	0.6622	0.6707	0.7336	0.7382
HashNet [3]	0.6568	0.6925	0.7234	0.7401	0.6485	0.7239	0.7587	0.7777
DCH [4]	0.6595	0.6965	0.7189	0.7426	0.6642	0.7386	0.7658	0.7825
DPH [2]	0.6672	0.6922	0.7243	0.7448	0.6850	0.7504	0.7755	0.7910
DSRH	**0.7140**	**0.7464**	**0.7552**	**0.7602**	**0.7642**	**0.7890**	**0.7896**	**0.7940**



(a) Precision within Hamming radius 2 (b) Precision-recall curve @ 16 bits (c) Precision-recall curve @ 32 bits

Fig. 3. Comparative evaluation of different algorithms on the CIFAR-10 dataset. (a) Precision within Hamming radius 2 curves w.r.t. different number of hash bits. (b) Precision-recall curves @ 16-bit. (c) Precision-recall curves @ 32-bit.

and otherwise dissimilar, where the label information from the single-label dataset is coded in the form of one-hot. In addition, to avoid the effect caused by a class-imbalance problem between similar and dissimilar similarity information, we empirically set the weight of the similar pair as the ratio between the number of dissimilar pairs and the number of similar pairs in each batch.

To demonstrate the superiority of DSRH, several state-of-the-art hashing methods are used for comparisons, including the traditional hashing (**LSH** [7], **SH** [50], **ITQ** [9], **KSH** [28], **FastH** [22] and **SDH** [42]) and the deep learning based hashing(**CNNH** [51], **DNNH** [18], **DSH** [25], **DHN** [60], **HashNet** [3], **DCH** [2] and **DPH** [4]). Most of these methods obtains similarity-preserving binary codes according to the hard-assigned similarity (i.e., -1 or 1), such as DPH, DCH, HashNet, DHN, DSH, CNNH, FastH and SH. However, the proposed DSRH redefines the similarity affinity, then it reconstructs the high-level semantic similarity within two binary codes towards the redefined similarity.

In the proposed DSRH, we adopt the CNN-F network [41] as our basic network, and the semantic classification layer is replaced with the hashing layer fch. We initialize the basic network based on the pre-trained weight on the ImageNet 2012, and train the semantic hashing layer. The initial learning rate is set to 10^{-5}. Considering the hashing layer being trained from scratch, we set its learning rate to be 10 times that of the lower layers. The weight decay parameter is set to be 0.0005, and the mini-batch size is fixed to be 200. In our work, the value of parameter m is set to 2, and the hyper-parameters λ for each dataset is selected from the range [0.05, 1]. The two parameters yield the best performance by cross-validation. For the above non-deep hashing methods, each image is represented by a 4096-dim deep feature as the input, which is extracted by the CNN-F network architecture.

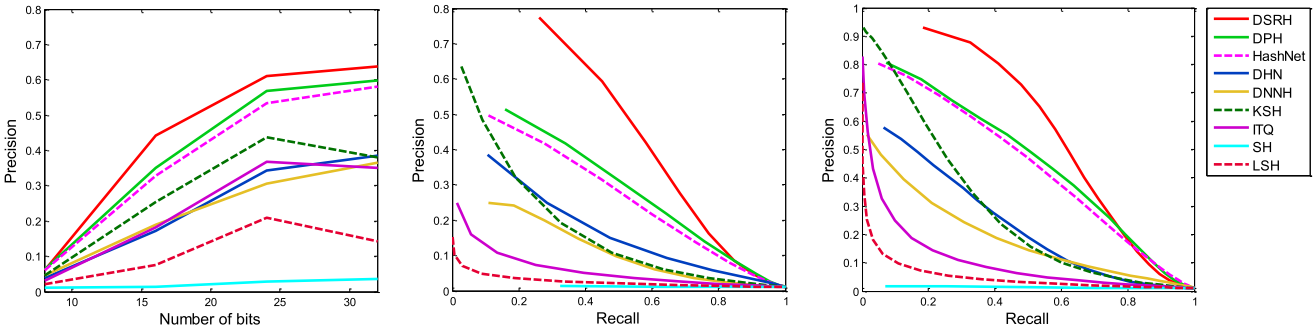
In the evaluation, several metrics are adopted to measure the quantitative performance under four different bits (8-bit, 16-bit, 24-bit and 32-bit), including Mean Average Precision (**MAP**), Precision-Recall curves (**PR**), Precision curves within Hamming distance 2 (**P@H=2**), and Precision with respect to different numbers of top returned samples (**P@N**). The top N images are selected from the ranked list in terms of the Hamming distance.

C. Results and Analysis

1) Retrieval Results on CIFAR-10: Table I shows the MAP scores and Precision@1,000 on the CIFAR-10. It is observed

TABLE II
COMPARISON OF RETRIEVAL MAP@ALL SCORES AND PRECISION@1,000 OF DIFFERENT METHODS ON THE IMAGENET DATASET

Method	ImageNet							
	MAP@all				Precision@1,000			
#Bits	$k = 8$	$k = 16$	$k = 24$	$k = 32$	$k = 8$	$k = 16$	$k = 24$	$k = 32$
LSH [7]	0.0251	0.0346	0.0571	0.0697	0.0406	0.0615	0.0973	0.1147
SH [50]	0.0164	0.0215	0.0252	0.0295	0.0232	0.0340	0.0424	0.0497
ITQ [8]	0.0556	0.1068	0.1451	0.1777	0.0923	0.1609	0.2050	0.2404
KSH [17]	0.1016	0.1842	0.2307	0.2841	0.1416	0.2418	0.2912	0.3471
FastH [22]	0.1601	0.3319	0.4174	0.4620	0.2197	0.3928	0.4840	0.5285
SDH [42]	0.2242	0.3878	0.4590	0.4951	0.2987	0.4572	0.5304	0.5626
CNNH [51]	0.1008	0.1013	0.1172	0.1466	0.1245	0.1362	0.1578	0.1662
DNNH [18]	0.1556	0.2210	0.2304	0.2439	0.2011	0.2094	0.2781	0.2920
DSH [25]	0.1427	0.2376	0.3281	0.3864	0.1672	0.2582	0.3478	0.3856
DHN [60]	0.1126	0.2525	0.3499	0.4097	0.1033	0.2691	0.2893	0.3442
HashNet [3]	0.2014	0.3602	0.4468	0.5093	0.2255	0.3693	0.4687	0.5484
DCH [2]	0.2096	0.3725	0.4576	0.5269	0.2412	0.4059	0.4895	0.5576
DPH [4]	0.2269	0.3757	0.4661	0.5446	0.2558	0.4112	0.4990	0.5630
DSRH	**0.2838**	**0.5068**	**0.5730**	**0.5984**	**0.3412**	**0.5618**	**0.6226**	**0.6475**



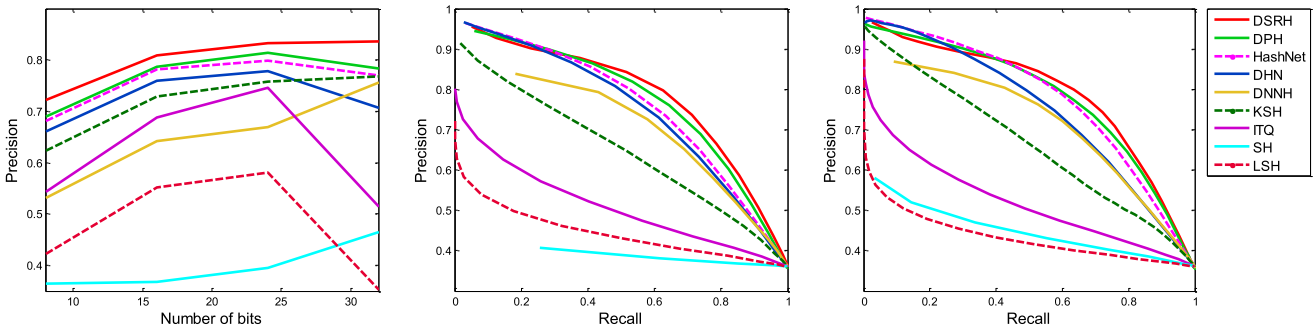
(a) Precision within Hamming radius 2 (b) Precision-recall curve @ 16 bits (c) Precision-recall curve @ 32 bits

Fig. 5. Comparative evaluation of different algorithms on the ImageNet dataset. (a) Precision within Hamming radius 2 curves w.r.t. different number of hash bits. (b) Precision-recall curves @ 16-bit. (c) Precision-recall curves @ 32-bit.

that the proposed DSRH constantly outperforms the baselines, including traditional non-deep hashing methods, (e.g., SDH and FastH) and deep hashing methods (e.g., DPH, DCH and HashNet). In addition, we can observe that the MAP and precision@1,000 of deep hashing methods distinctly outperform that of the traditional hashing methods. The behind reason is that deep networks enable joint performing feature learning and hash coding in an end-to-end way, and the two processes can promote each other for improving the quality of binary codes.

Specifically, the average MAP absolute increase can be up to 26.91% compared to the traditional hashing method SDH [42] for different bits. Compared to the state-of-the-art deep hashing methods, the proposed DSRH method improves the average MAP from 64.15%(DHN), 70.29%(HashNet), 70.43%(DCH) and 70.71%(DPH) to 74.39%. The reason is that those baselines simply define dissimilar data pairs' similarity affinity as the discrete value -1. Next, they employ the defined affinity to maximize the Hamming distance of dissimilar pairs, and it would result in two unrelated data having similar codes, compromising the retrieval performance. However, the proposed DSRH redefines data pairs' similarity affinity for dissimilar pairs, and the affinity is a local variable within a batch. The new affinity encourages the generated binary codes to be compatible when dealing with two or more dissimilar data pairs related to an anchor sample.

The performance in terms of Precision within Hamming radius 2 (P@H=2) is very important for efficient retrieval, since the search time is a constant for each query. Figure 3(a) shows the P@H=2 result on the CIFAR-10, and it is clear that DSRH consistently obtains the best precision. With the length of codes becoming longer, some baselines show a decreasing tendency, and the possible explanation is that the Hamming space will become sparse and few data points fall within the Hamming ball with radius 2. However, our P@H=2 can still show a steady result under longer codes, and it further validates the effectiveness of the proposed DSRH for P@H=2. In addition, Figure 3(b-c) shows the precision-recall curves within 16-bit and 32-bit. Compared to the state-of-the-art methods, it is clear that the proposed DSRH consistently works the best.



(a) Precision within Hamming radius 2 (b) Precision-recall curve @ 16 bits (c) Precision-recall curve @ 32 bits

Fig. 6. Comparative evaluation of different algorithms on the NUS-WIDE dataset. (a) Precision within Hamming radius 2 curves w.r.t. different number of hash bits. (b) Precision-recall curves @ 16-bit. (c) Precision-recall curves @ 32-bit.

TABLE III
COMPARISON OF RETRIEVAL MAP@ALL SCORES ON THE NUS-WIDE DATASET

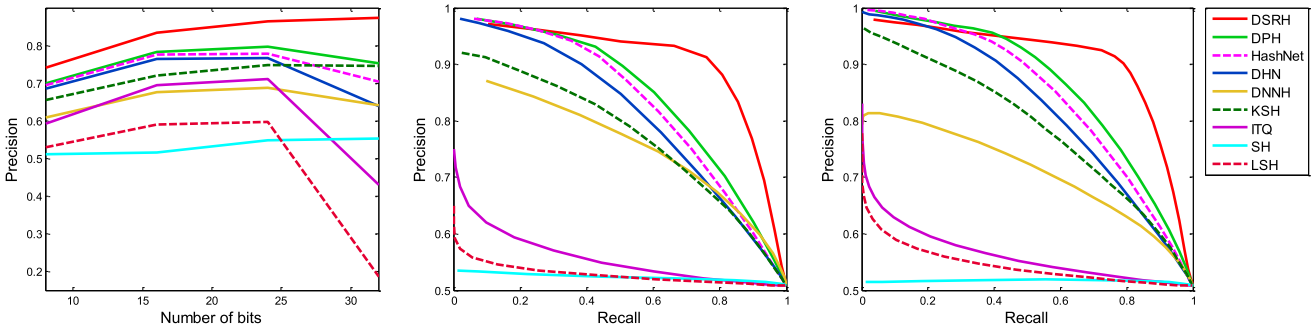
Method	NUS-WIDE(bits)			
	8	16	24	32
LSH [7]	0.1658	0.1867	0.2127	0.2494
SH [50]	0.1684	0.1694	0.1653	0.1765
ITQ [8]	0.2649	0.3142	0.3289	0.3407
KSH [17]	0.4696	0.5564	0.5684	0.5855
FastH [22]	0.5054	0.5962	0.6257	0.6386
SDH [42]	0.3608	0.5876	0.6080	0.6212
CNNH [51]	0.5282	0.5221	0.5289	0.5266
DNNH [18]	0.6121	0.6456	0.6574	0.6586
DSH [25]	0.6706	0.6789	0.6802	0.6846
DHN [60]	0.6713	0.6823	0.6835	0.6871
HashNet [3]	0.6772	0.7001	0.7122	0.7239
DCH [2]	0.6815	0.7048	0.7165	0.7258
DPH [4]	0.6852	0.7121	0.7199	0.7265
DSRH	**0.6979**	**0.7242**	**0.7341**	**0.7403**

TABLE IV
COMPARISON OF RETRIEVAL MAP@ALL SCORES ON THE MS-COCO DATASET

Method	MS-COCO(bits)			
	8	16	24	32
LSH [7]	0.5225	0.5333	0.5401	0.5528
SH [50]	0.5139	0.5176	0.5290	0.5269
ITQ [8]	0.5720	0.5809	0.5843	0.5868
KSH [17]	0.6484	0.6622	0.6674	0.6711
FastH [22]	0.6252	0.6577	0.6729	0.6825
SDH [42]	0.6186	0.6394	0.6464	0.6536
CNNH [51]	0.5084	0.5123	0.5196	0.5284
DNNH [18]	0.5895	0.6004	0.6176	0.6213
DSH [25]	0.6473	0.6586	0.6652	0.6692
DHN [60]	0.6713	0.6712	0.6776	0.6794
HashNet [3]	0.6807	0.6937	0.7021	0.7084
DCH [2]	0.6832	0.6986	0.7045	0.7125
DPH [4]	0.6864	0.7033	0.7112	0.7168
DSRH	**0.7052**	**0.7329**	**0.7423**	**0.7475**

2) Retrieval Results on ImageNet: ImageNet includes more detailed information and is a more challenging dataset. Table II shows the MAP scores and precision@1,000 results on the ImageNet dataset. We can observe that the proposed DSRH achieves the best results among all state-of-the-art hashing methods. Besides, the DSRH shows a significant MAP improvement compared to those baselines. For example, the DSRH can imporve the MAP scores from 24.39% (DNNH), 40.97% (DHN), 50.93% (HashNet), 52.69%(DCH) and 54.46% (DPH) to 59.84% in 32-bit binary codes. This reason is the DSRH redefines the similarity affinity among dissimilar data pairs, and reconstruct the semantic similarity towards the redefined similarity, rather than maximizing the Hamming distance among dissimilar pairs. Meanwhile, the DRSH adopts a pairwise similarity-preserving quantization constraint to reduce the similarity loss when performing quantization. Figure 5(a) shows the precision curves within Hamming radius 2 for different lengths of codes. Figure 5(b−c) shows the precision-recall curves. It is clear that the proposed DSRH approach gets the best search accuracy in different lengths of codes.

Compared to the CIFAR-10 dataset, the proposed DSRH shows a larger performance improvement over the baselines on the ImagNet. For example, the MAP absolute increase w.r.t. 16-bit can be up to 13.31% compared to the state-of-the-art method DPH on the ImageNet, and the corresponding MAP absolute increase is 5.42 % on the CIFAR-10. The reason is that the ImageNet has in total of 100 class concepts for performance evaluation and the structure information is more complicated among data pairs. For an anchor sample, it has more dissimilar data pairs from the other 99 class sample to preserve a certain distance. Maximizing the Hamming distance of dissimilar pairs would lead to the binary codes being badly incompatible when it adopts the hard-assigned similarity -1 as its ground truth. The data pairs' similarity suffers from more severe disruption. However, the proposed DSRH redefines a high-level similarity affinity to guide data pairs preserving their high-level semantics, facilitating generating



(a) Precision within Hamming radius 2 (b) Precision-recall curve @ 16 bits (c) Precision-recall curve @ 32 bits

Fig. 7. Comparative evaluation of different algorithms on the MS-COCO dataset. (a) Precision within Hamming radius 2 curves w.r.t. different number of hash bits. (b) Precision-recall curves @ 16-bit. (c) Precision-recall curves @ 32-bit.

TABLE V
MEAN AVERAGE PRECISION (MAP) OF DSRH AND ITS VARIANTS, DSRH-C, DSRH-S, DSRH-T AND DSRH-P ON THREE DATASETS

Method	ImageNet				NUS-WIDE				MS-COCO			
	8 bits	16 bits	24 bits	32 bits	8 bits	16 bits	24 bits	32 bits	8 bits	16 bits	24 bits	32 bits
DSRH-C	0.2598	**0.5152**	**0.5866**	**0.6101**	**0.7230**	**0.7469**	**0.7527**	**0.7549**	**0.7316**	**0.7564**	**0.7645**	**0.7686**
DSRH	**0.2838**	0.5068	0.5730	0.5984	0.6979	0.7242	0.7341	0.7403	0.7052	0.7329	0.7423	0.7475
DSRH-S	0.2500	0.4631	0.5428	0.5662	0.6709	0.7069	0.7077	0.7155	0.6755	0.6883	0.6942	0.6890
DSRH-T	0.2800	0.4981	0.5474	0.5712	0.6910	0.7182	0.7127	0.7269	0.6753	0.6945	0.6984	0.7017
DSRH-P	0.2821	0.5016	0.5586	0.5805	0.6936	0.7214	0.7194	0.7301	0.6824	0.7084	0.7136	0.7194

Hamming-compatible binary codes. The corresponding performance improvement is more obvious.

3) Retrieval Results on NUS-WIDE: NUS-WIDE is a multi-label dataset. To verify the effectiveness of the proposed DSRH, we compare it with several state-of-the-art hashing algorithms on the NUS-WIDE. Table III shows the MAP scores of those methods, and we can observe that our approach works the best compared to the baselines. For example, on the 8-bit codes, the DRSH can improve the MAP scores from 68.15%(DCH) and 68.52%(DPH) to 69.79%.

Figure 6(a) shows the precisions within Hamming distance of 2. Figure 6(b−c) shows the precision-recall curves w.r.t. 16-bit and 32-bit. Under those evaluation metrics, our method outperforms other state-of-the-art hashing methods, which further demonstrates the benefits of redefining data pairs' similarity to achieve their high-level semantic reconstruction.

4) Retrieval Results on MS-COCO: To further demonstrate the superiority of the DSRH, we compare it with existing methods on the MS-COCO dataset. The MAP scores of different methods are shown in Table IV. We can observe that the DSRH shows a clear MAP gain over these baselines. To be specific, the absolute average MAP under different codes can be up to 9.24% and 2.75% compared to the state-of-the-art hashing SDH and DPH, respectively.

Figure 7(a−c) shows the precisions within the Hamming distance of 2 and the precision-recall curves. It is clear that the precision obviously outperforms other compared methods. In low or high recall ratio, our method obtains a higher precision, which is desirable for precision-first practical retrieval systems. These obtained best results demonstrate the effectiveness of the proposed method, where we redefine the semantic similarity of data pairs for high-level semantic reconstruction, and adopt a pairwise similarity-preserving quantization constraint.

D. Empirical Analysis

1) Ablation Study: To further validate the efficacy of the proposed DSRH, we investigate four variants of DSRH: DSRH-C, DSRH-S, DSRH-T and DSRH-P for comparison. DSRH-C is the DSRH variant without binarization, where $\text{sign}(\boldsymbol{h}_i)$ is not performed. DSRH-S adopts directly the hard-assigned similarity (i.e., $s_{ij} = 1$ or -1) as the ground truth for hashing learning. DSRH-T means that we don't utilize the proposed pairwise quantization constraint (i.e., $\lambda = 0$), and uses the $tanh()$ as the activation of hashing layer for outputting approximate -1 or 1 codes. DSRH-P is the DSRH variant with the L_1-norm based point-wise quantization constraint, namely, it adopts the conventional point-wise constraint for narrowing quantization error. The Map scores about these variants are shown in Table V.

As expected, the DSRH shows superior results compared to its variants DSRH-S, DSRH-T and DSRH-P. To be specific, compared to the DSRH-S, the best absolute MAP increase of DSRH can be up to 4.37%, 2.70% and 5.85% on ImageNet, NUS-WIDE and MS-COCO dataset, respectively. The explanation is that DSRH-S employs the hard-assigned similarity for hash learning, and enforces the similarity within two binary codes towards this semantic similarity. The unexpected result is that two unrelated dissimilar samples would have similar codes when both of them are most dissimilar with an anchor sample. Besides, On the multi-label dataset MS-COCO

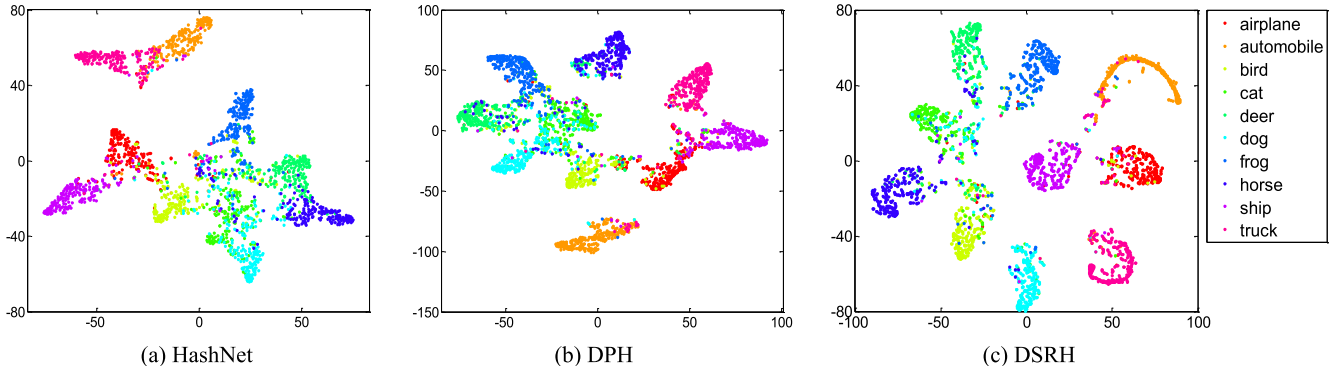


Fig. 8. The t-SNE visualization of binary codes learned by HashNet, DPH and DSRH.

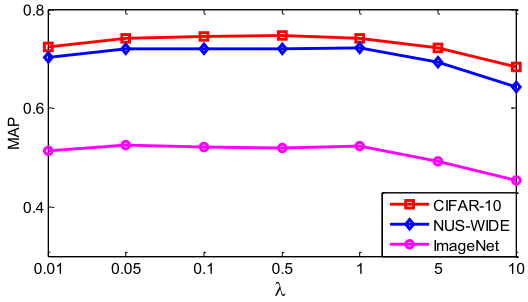


Fig. 9. Sensitivity analysis of λ for DSRH w.r.t. 16-bit codes on three datasets.

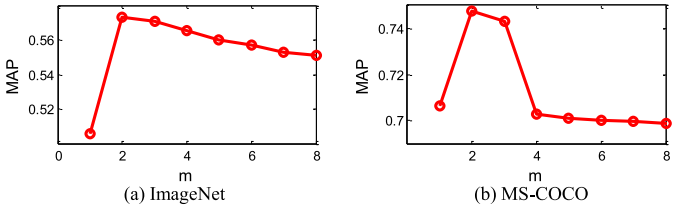


Fig. 10. The MAP scores comparison w.r.t. different value of m on the ImageNet(24-bit) and MS-COCO(32-bit) dataset.

and NUS-WIDE, the binary codes of similar pairs would be unified, even for two dissimilar samples when both of them are most similar with an anchor sample. However, the DSRH redefines data pairs' similarity affinity to learn high-level semantic similarity between a pair of codes.

Compared to the DSRH-T, the best absolute MAP increase of the DSRH can be up to 2.66%(24-bit), 2.14%(24-bit) and 4.58%(32-bit) on ImageNet, NUS-WIDE and MS-COCO dataset, respectively. The corresponding MAP increase of the DSRH-P is 1.12%(24-bit), 0.67%(24-bit) and 1.77%(32-bit), and the MAP increase is inferior to that of the DSRH. What's more, the whole MAP increase of the DSRH is better than DSRH-P in different lengths of bits. This indicates that the proposed pairwise quantization constraint can maintain the well-learned paired similarity, enabling more effective similarity retrieval.

For the DSRH-C, it shows a relatively better retrieval result compared to the DSRH. The truth is that the DSRH-C doesn't perform binary quantization, and there is no information loss in the retrieval task.

2) Parameter Sensitivity: We further investigate the sensitivity of the tradeoff hyper-parameter λ in Equation. (9). Figure 9 shows the MAP scores about different λ on the CIFAR-10, NUS-WDIE and ImageNet datasets. We can observe that the MAP results are steady when λ is in a range of [0.05, 1]. This demonstrates that the DSRH is not sensitive to the scale of the tradeoff hyper-parameter. When λ being greater than 1, the MAP scores show a decreasing trend with λ increasing. The reason is that the hash model takes more effort on preserving the paired similarity, but such similarity (under a larger λ) cannot fully show the semantic similarity of a pair of codes. The corresponding retrieval result is not satisfactory.

In our experiment, the parameter m is used to constrain the redefined similarity affinity. Figure 10 shows the MAP scores w.r.t. different values of m on the ImageNet and MS-COCO dataset. When $m = 2$, the MAP scores achieve the best result. Wherein, $m = 1$ means that the $\hat{s}_{ij}$ is decided by the cosine similarity of two labels or the ratio of similar data pairs. If we set m to be a relatively larger value, it means that the redefined similarity turns back to the hard-assigned similarity, i.e., 1 or -1. According to the MAP scores, we can observe that the MAP shows a decreasing trend with the value of m increasing. Therefore, we set the m to be 2 in our experiment for an optimal retrieval result.

3) Similarity Change by Quantization: To verify the degree of similarity-preserving by the proposed pairwise quantization constraint, we make comparisons with the point-wise quantization constraint to state the change of similarity. The specific results are shown in Figure 12, where we sample 50 data pairs and use $\frac{|\boldsymbol{h}_i^T \cdot \boldsymbol{h}_j - \boldsymbol{b}_i^T \cdot \boldsymbol{b}_j|}{k}$ to quantitatively describe this change. It is clear that the error from pairwise quantization schema is less than that of point-wise constraint, and it demonstrates that the proposed similarity-preserving schema can better maintain the well-learned paired similarity.

4) Visualization: In order to observe intuitively the deep binary representation, we visualize the t-SNE [35] of binary codes generated by HashNet, DPH and DSRH in Figure 8. It is observed that the binary codes generated by DSRH show a clear discriminative boundary because the samples of different

Fig. 11. Top 10 retrieved results from the ImageNet dataset with 32 bits. The first column shows the queries, and other columns show the top 10 retrieved images.

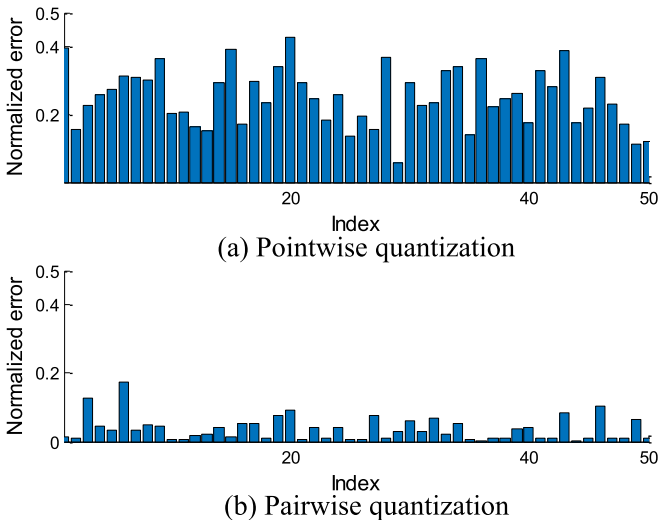


Fig. 12. The change degree of data pairs' similarity before and after quantification by (a) pointwise error-minimization constraint (b) pairwise simialrity-preserving constraint. We use the normalized error $\frac{|\boldsymbol{h}_i^T \cdot \boldsymbol{h}_j - \boldsymbol{b}_i^T \cdot \boldsymbol{b}_j|}{k}$ to measure the change degree of paired similarity.

categories are well separated, while the codes generated by DPH and HashNet do not show such clear and separable boundary.

In addition, to acquire qualitatively visual result, Figure 11 shows the top 10 retrieved images of the DSRH given a query image on the ImageNet dataset.

TABLE VI
THE AVERAGE RETRIEVAL TIME OF EACH TESTING SAMPLE ON THE IMANGENET DATASET

time (s)	$T_{top-100}$	$T_{top-1000}$	$T_{H=2}$
DSRH@8-bit	2.43e-5	9.46e-5	2.20e-3
DPH@8-bit	2.57e-5	1.00e-4	2.10e-3
DSRH@32-bit	2.81e-5	1.10e-4	9.27e-4
DPH@32-bit	2.87e-5	1.13e-4	8.52e-4

5) Efficiency Analysis: In addition, Table VI gives the average retrieval time of each testing sample on the ImageNet dataset. The experimental hardware condition is based on Windows 7 OS. The used CPU is the Core i7-4790@3.60 GHz with 8 processors, and the memory size is 12GB. The experimental software platform is based on Matlab 2014a. We compare the state-of-the-art hashing DPH with the proposed DRSH under three time indicators: $T_{top-100}$, $T_{top-100}$ and $T_{H=2}$. Wherein $T_{top-100}$ and $T_{top-100}$ denote the time of returning the top 100 and 1000 data point, respectively. $T_{H=2}$ denotes the time of returning data within Hamming distance 2.

We can observe that the time cost of DSRH is superior to DPH is consistent under $T_{top-100}$ or $T_{top-100}$. This reason is that they perform the same XOR operation as well as return the top 100 or 1,000 data, but the model parameter of DSRH is less than that of DPH. Thus the DSRH is more efficient in time cost. In both of DSRH and DPH, $T_{top-1000}$ is greater than

$T_{top-100}$, because they need to return more samples under the indicator $T_{top-1000}$. For the indicator $T_{H=2}$, it is observed that the time $T_{H=2}$@8-bit is greater than $T_{H=2}$@32-bit in both of DSRH and DPH. The behind truth is that the Hamming space will become sparse and few data points fall within the Hamming ball with radius 2 when using longer codes. Therefore, when using 8-bit codes for retrieval, there being more targets are retrieved and returned in the database, and it naturally increases the retrieval time cost. In addition, the time $T_{H=2}$ of DSRH is greater than that of DPH. This is because DSRH can enforce more data points fall into the Hamming ball with radius 2, and there being more data points to be retrieved increases the time cost.

V. Conclusion

This paper studies deep learning to hash approaches by redefining the similarity of data pairs to support efficient image retrieval. The proposed deep reconstructive hashing method with pairwise quantization, i.e., DSRH, can generate more compact binary codes based on two contributions: (1) it reconstructs the high-level semantic within a pair of codes by the redefined similarity; (2) it can maintain the well-learned semantic similarity by the proposed pairwise quantization constraint when performing binarization. Extensive experimental results have shown the effectiveness of the proposed DSRH on four widely-used image retrieval datasets compared with state-of-the-art methods. In the future, we further exploit the high-level semantic similarity to learn compact binary codes, especially in partial labels setting. We also plan to exploit the high-level semantic similarity learning for similarity retrieval on the cross-modal dataset.

References

[1] A. Araujo and B. Girod, "Large-scale video retrieval using image queries," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 6, pp. 1406–1420, Jun. 2018.

[2] Y. Cao, M. Long, B. Liu, and J. Wang, "Deep cauchy hashing for Hamming space retrieval," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1229–1237.

[3] Z. Cao, M. Long, J. Wang, and P. S. Yu, "hashNet: Deep learning to hash by continuation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5609–5618.

[4] Z. Cao, Z. Sun, M. Long, J. Wang, and P. S. Yu, "Deep priority hashing," in *Proc. ACM Multimedia Conf. Multimedia Conf. (MM)*, 2018, pp. 1653–1661.

[5] Z. Chen, J. Lu, J. Feng, and J. Zhou, "Nonlinear structural hashing for scalable video search," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 6, pp. 1421–1433, Jun. 2018.

[6] Z. Chen, X. Yuan, J. Lu, Q. Tian, and J. Zhou, "Deep hashing via discrepancy minimization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1129–1137.

[7] A. Gionis *et al.*, "Similarity search in high dimensions via hashing," in *Proc. Int. Conf. Very Large Data Bases*, 1999, pp. 518–529.

[8] Y. Gong and S. Lazebnik, "Iterative quantization: A procrustean approach to learning binary codes," in *Proc. CVPR*, Jun. 2011, pp. 817–824.

[9] 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.

[10] J. Gu *et al.*, "Recent advances in convolutional neural networks," *Pattern Recognit.*, vol. 77, pp. 354–377, May 2018.

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[12] A. Hyvrinen, J. Hurri, and P. O. Hoyer, *Natural Image Statistics: A Probabilistic Approach to Early Computational Vision*, 1st ed. London, U.K.: Springer, 2009.

[13] R. Ji, H. Liu, L. Cao, D. Liu, Y. Wu, and F. Huang, "Toward optimal manifold hashing via discrete locally linear embedding," *IEEE Trans. Image Process.*, vol. 26, no. 11, pp. 5411–5420, Nov. 2017.

[14] Y. Jia *et al.*, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. ACM Int. Conf. Multimedia (MM)*, 2014, pp. 675–678.

[15] F. Khelifi and A. Bouridane, "Perceptual video hashing for content identification and authentication," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 1, pp. 50–67, Jan. 2019.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.

[17] B. Kulis and K. Grauman, "Kernelized locality-sensitive hashing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 6, pp. 1092–1104, Jun. 2012.

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

[19] N. Li, C. Li, C. Deng, X. Liu, and X. Gao, "Deep joint semantic-embedding hashing," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 2397–2403.

[20] Q. Li, Z. Sun, R. He, and T. Tan, "Deep supervised discrete hashing," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 2482–2491.

[21] W. J. Li, S. Wang, and W. C. Kang, "Feature learning based deep supervised hashing with pairwise labels," in *Proc. 25th Int. Joint Conf. Artif. Intell.*, 2016, pp. 1711–1717.

[22] G. Lin, C. Shen, Q. Shi, A. van den Hengel, and D. Suter, "Fast supervised hashing with decision trees for high-dimensional data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1963–1970.

[23] K. Lin, J. Lu, C.-S. Chen, and J. Zhou, "Learning compact binary descriptors with unsupervised deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1183–1192.

[24] H. Liu, M. Lin, S. Zhang, Y. Wu, F. Huang, and R. Ji, "Dense auto-encoder hashing for robust cross-modality retrieval," in *Proc. ACM Multimedia Conf. Multimedia Conf. (MM)*, 2018, pp. 1589–1597.

[25] H. Liu, R. Wang, S. Shan, and X. Chen, "Deep supervised hashing for fast image retrieval," *Int. J. Comput. Vis.*, vol. 127, no. 9, pp. 1217–1234, Sep. 2019.

[26] J. Liu, S. Zhang, W. Liu, C. Deng, Y. Zheng, and D. N. Metaxas, "Scalable mammogram retrieval using composite anchor graph hashing with iterative quantization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 11, pp. 2450–2460, Nov. 2017.

[27] W. Liu, C. Mu, S. Kumar, and S.-F. Chang, "Discrete graph hashing," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3419–3427.

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

[29] F. Long, T. Yao, Q. Dai, X. Tian, J. Luo, and T. Mei, "Deep domain adaptation hashing with adversarial learning," in *Proc. 41st Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, 2018, pp. 725–734.

[30] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 640–651, Apr. 2017.

[31] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[32] J. Lu, V. E. Liong, and J. Zhou, "Deep hashing for scalable image search," *IEEE Trans. Image Process.*, vol. 26, no. 5, pp. 2352–2367, May 2017.

[33] X. Lu, Y. Chen, and X. Li, "Hierarchical recurrent neural hashing for image retrieval with hierarchical convolutional features," *IEEE Trans. Image Process.*, vol. 27, no. 1, pp. 106–120, Jan. 2018.

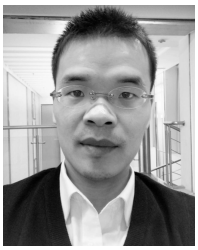
[34] X. Luo, L. Nie, X. He, Y. Wu, Z.-D. Chen, and X.-S. Xu, "Fast scalable supervised hashing," in *Proc. 41st Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, 2018, pp. 735–744.

[35] L. V. D. Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.

[36] H. Nam and B. Han, "Learning multi-domain convolutional neural networks for visual tracking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4293–4302.

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

[38] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *Int. J. Comput. Vis.*, vol. 42, no. 3, pp. 145–175, 2001.
[39] Z. Qiu, Y. Pan, T. Yao, and T. Mei, "Deep semantic hashing with generative adversarial networks," in *Proc. 40th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, 2017, pp. 225–234.
[40] M. Raginsky and S. Lazebnik, "Locality-sensitive binary codes from shift-invariant kernels," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 1509–1517.
[41] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.
[42] F. Shen, C. Shen, W. Liu, and H. T. Shen, "Supervised discrete hashing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 37–45.
[43] F. Shen, W. Liu, S. Zhang, Y. Yang, and H. T. Shen, "Learning binary codes for maximum inner product search," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4148–4156.
[44] J. Tang and Z. Li, "Weakly supervised multimodal hashing for scalable social image retrieval," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 10, pp. 2730–2741, Oct. 2018.
[45] D. Wang, Q. Wang, and X. Gao, "Robust and flexible discrete hashing for cross-modal similarity search," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 10, pp. 2703–2715, Oct. 2018.
[46] J. Wang, T. Zhang, J. Song, N. Sebe, and H. T. Shen, "A survey on learning to hash," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 769–790, Apr. 2018.
[47] Y. Wang, D. Cao, and Z. Sun, "Target code guided binary hashing representations with deep neural network," in *Proc. 4th IAPR Asian Conf. Pattern Recognit. (ACPR)*, Nov. 2017, pp. 530–535.
[48] Y. Wang, J. Liang, D. Cao, and Z. Sun, "Local semantic-aware deep hashing with Hamming quantization," *IEEE Trans. Image Process.*, vol. 28, no. 6, pp. 2665–2679, Jun. 2019.
[49] Y. Wang and Z. Sun, "Towards joint multiply semantics hashing for visual search," in *Proc. Int. Conf. Image Graph.* Cham, Switzerland: Springer, 2019, pp. 47–58.
[50] Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 1753–1760.
[51] R. Xia, Y. Pan, H. Lai, C. Liu, and S. Yan, "Supervised hashing for image retrieval via image representation learning," in *Proc. 28th AAAI Conf. Artif. Intell.*, vol. 1, 2014, pp. 2156–2162.
[52] E. Yang, C. Deng, T. Liu, W. Liu, and D. Tao, "Semantic structure-based unsupervised deep hashing," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 1064–1070.
[53] H.-F. Yang, K. Lin, and C.-S. Chen, "Supervised learning of semantics-preserving hash via deep convolutional neural networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 2, pp. 437–451, Feb. 2018.
[54] T. Yao, F. Long, T. Mei, and Y. Rui, "Deep semantic-preserving and ranking-based hashing for image retrieval," in *Proc. 25th Int. Joint Conf. Artif. Intell.*, 2016, pp. 3931–3937.
[55] D. Zhang, F. Wang, and L. Si, "Composite hashing with multiple information sources," in *Proc. 34th Int. ACM SIGIR Conf. Res. Develop. Inf. (SIGIR)*, 2011, pp. 225–234.
[56] J. Zhang and Y. Peng, "query image retrieval by deep-weighted hashing," *IEEE Trans. Multimedia*, vol. 20, no. 9, pp. 2400–2414, Sep. 2018.
[57] J. Zhang and Y. Peng, "SSDH: Semi-supervised deep hashing for large scale image retrieval," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 1, pp. 212–225, Jan. 2019.
[58] 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.
[59] S. Zhang, J. Li, M. Jiang, P. Yuan, and B. Zhang, "Scalable discrete supervised multimedia hash learning with clustering," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 10, pp. 2716–2729, Oct. 2018.
[60] H. Zhu, M. Long, J. Wang, and Y. Cao, "Deep hashing network for efficient similarity retrieval," in *Proc. 30th AAAI Conf. Artif. Intell.*, 2016, pp. 2415–2421.



Yunbo Wang received the B.E. and M.S. degrees from the School of Electronics and Information Engineering, Sichuan University, Chengdu, China, in 2012 and 2015, respectively, and the Ph.D. degree in pattern recognition and intelligent systems from the Center for Research on Intelligent Perception and Computing, National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, China, in January 2020. His main research interests include image/video retrieval, pattern recognition, and computer vision.

Xianfeng Ou received the B.S. degree in electronic information science and technology and the M.S. degree in communication and information system from Xinjiang University, Urumchi, China, in 2006 and 2009, respectively, and the Ph.D. degree in communication and information system from Sichuan University, Chengdu, China, in 2015. He was a Visiting Researcher with the Internet Media Group, Polytechnic di Torino, Turin, Italy, from January 2014 to April 2014. He is currently an Associate Professor with the School of Information and Communication Engineering, Hunan Institute of Science and Technology. His main research interests include machine vision and artificial intelligence, object detection, and image and video coding process technologies.

Jian Liang received the B.E. degree in electronic information and technology from Xi'an Jiaotong University in July 2013 and the Ph.D. degree in pattern recognition and intelligent systems from NLPR, CASIA, in January 2019. He is currently a Research Fellow with the National University of Singapore. His research interests focus on machine learning, pattern recognition, and computer vision.

Zhenan Sun (Senior Member, IEEE) received the B.E. degree in industrial automation from the Dalian University of Technology, China, in 1999, the M.S. degree in system engineering from the Huazhong University of Science and Technology, China, in 2002, and the Ph.D. degree in pattern recognition and intelligent systems from the Institute of Automation, Chinese Academy of Sciences, China, in 2006. He is currently a Professor with the Center for Research on Intelligent Perception and Computing, National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences. He has authored/coauthored over 200 technical articles. His research interests include biometrics, pattern recognition, and computer vision. He is also an IAPR Fellow. He is also the Chair of the Technical Committee on Biometrics and International Association for Pattern Recognition (IAPR). He serves as an Associate Editor for the IEEE TRANSACTIONS ON BIOMETRICS, BEHAVIOR, AND IDENTITY SCIENCE.