Local Semantic-Aware Deep Hashing With Hamming-Isometric Quantization

Yunbo Wang, Jian Liang, Dong Cao, and Zhenan Sun, *Member, IEEE*

***Abstract*— Hashing is a promising approach for compact storage and efficient retrieval of big data. Compared to the conventional hashing methods using handcrafted features, emerging deep hashing approaches employ deep neural networks to learn both feature representations and hash functions, which have been proven to be more powerful and robust in real-world applications. Currently, most of the existing deep hashing methods construct pairwise or triplet-wise constraints to obtain similar binary codes between a pair of similar data points or relatively similar binary codes within a triplet. However, we argue that some critical local structures have not been fully exploited. So, this paper proposes a novel deep hashing method named local semantic-aware deep hashing with Hamming-isometric quantization (LSDH), aiming to make full use of local similarity in hash function learning. Specifically, the potential semantic relation is exploited to robustly preserve local similarity of data in the Hamming space. In addition to reducing the error introduced by binary quantizing, a Hamming-isometric objective is designed to maximize the consistency of similarity between the pairwise binary-like features and corresponding binary codes pair, which is shown to be able to improve the quality of binary codes. Extensive experimental results on several benchmark datasets, including three single-label datasets and one multi-label dataset, demonstrate that the proposed LSDH achieves better performance than the latest state-of-the-art hashing methods.**

***Index Terms*— Image retrieval, deep hashing, similarity-preserving, local structures, Hamming-isometric.**

I. Introduction

With the explosive growth of visual data on the web and from video surveillance, pursuing an efficient solution to retrieve similar images becomes the spotlight of research. For example, given a query image of a cat, it is desirable to return similar images with a cat as fast and accurate as possible for a search engine. Similarity-preserving hashing [8], [10], [41], [57], [59] is a popular nearest neighbor search technique for large-scale image retrieval, which has shown superior potentials for applications with millions or even billions of images. Due to the appealing efficiency in both search speed and data storage, more and more hashing works are proposed in recent years [3], [31], [33], [64], [66].

Generally, hashing methods could be divided into two categories based on the type of hash functions employed: data-dependent (also known as learning-based) hashing methods [11], [27], [31], [58] and data-independent hashing methods [8], [21]. Since data-independent hashing methods always require long codes to achieve a satisfying retrieval performance, data-dependent hashing methods are proposed to learn more compact binary codes by utilizing a batch of training data. In this paper, we will focus on learning-based hashing with application to image retrieval.

Basically, learning-based hashing methods [15], [35], [58], [59] aim to learn a set of hash functions for coding each data point into low-dimensional binary codes, meanwhile enforcing semantically similar data pair to have small Hamming distance [36], [44], [53], [68]. By encoding each data point into binary codes, the similarity between the query and database can be efficiently computed and the storage cost can be distinctly decreased. According to whether the supervised information is available, the learning-based hashing approaches can be roughly grouped into unsupervised and supervised approaches. In contrast to unsupervised hashing methods [8], [9], [29], [39] where no supervision information is provided, supervised hashing methods [25], [33], [51], [52], [64] mainly leverage supervision information (e.g., pointwise semantic labels, pairwise semantic similarity) to obtain compact binary codes. Among these hashing methods, the input data is usually represented by hand-crafted feature descriptors such as SIFT [38] and GIST [46], followed by separate projection and quantization steps to encode these descriptors into binary codes. Since such descriptors cannot effectively represent the raw data and the coding process cannot make feedback to feature descriptors, the retrieval accuracy is not good in practical applications.

Recently, a number of hashing methods [22], [26], [62] explore Convolutional Neural Networks (CNN) [13], [20] to learn effective feature representations and hash functions, which have shown much better performance than the traditional hashing methods with handcrafted feature. Specifically, deep hashing methods with pairwise labels [2], [3], [32], [70] generally exploit data pairs' semantic similarity

Manuscript received December 17, 2017; revised June 27, 2018 and October 7, 2018; accepted December 6, 2018. Date of publication December 21, 2018; date of current version March 21, 2019. This work was supported in part by the National Natural Science Foundation of China under Grant U1836217, Grant 61427811, and Grant 61573360, and in part by the National Key Research and Development Program of China under Grant 2017YFC0821602 and Grant 2016YFB1001000. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Wen Gao. *(Corresponding author: Zhenan Sun.)*

Y. Wang and J. Liang are with the Center for Research on Intelligent Perception and Computing, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China, and also with the University of Chinese Academy of Sciences, Beijing 100190, China (e-mail: yunbo.wang@cripac.ia.ac.cn; jian.liang@nlpr.ia.ac.cn).

D. Cao is with the Center for Research on Intelligent Perception and Computing, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: dong.cao@nlpr.ia.ac.cn).

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

Digital Object Identifier 10.1109/TIP.2018.2889269

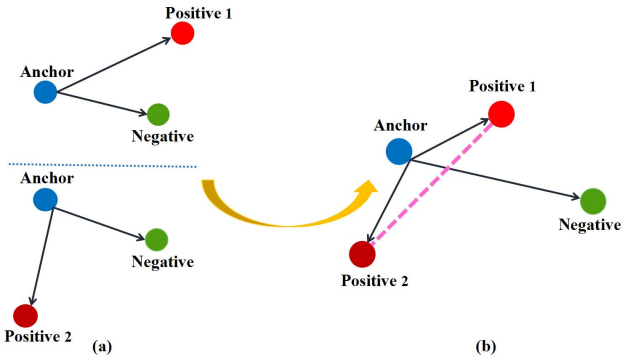


Fig. 1. The illustration of the proposed LSDH. Two ordinary triplets, sharing the anchor and the negative sample. We first work on a single triplet to get the ranking order, then we further exploit the underlying semantic relation to preserve the similarity of the two positive samples.

to obtain similar/dissimilar codes between similar/dissimilar data pair. Besides, the triplet-wise labels based deep hashing methods [22], [23], [65], [67], [69] maximize the margin between similar pair and dissimilar pair to obtain relative similar binary codes within a triplet. Although these methods utilize the typical pairwise or triplet-wise constraints to perform hash learning, the underlying local structure of the data is not exploited and the generated binary codes can not robustly preserve the similarity with its neighbors.

In this paper, we propose a novel deep hashing method named local semantic-aware deep hashing with Hamming-isometric quantization (LSDH), where we exploit the potential local structure of the data for hash learning. The proposed LSDH encourages the generated binary codes to preserve their local similarity based on their semantic relation, and Fig. 1 illustrates the main idea of the LSDH. We use a quadruplet as an input unit, where each quadruplet consists of an anchor sample termed A, two positive samples termed P_1 and P_2, a negative sample termed N. We first expect a triplet embedding to satisfy such ranking constraint $dist(A, P_1)<dist(A, N)$ from triplet $T(A, P_1, N)$ and $dist(A, P_2)<dist(A, N)$ from triplet $T(A, P_2, N)$, that is to say, we employ ranking constraint to increase the distance between anchor-positive data pair and anchor-negative data pair. However, the distance of data pair (P_1, P_2) is unknown about the anchor-negative pair (A, N). Studies [4], [5] have shown that triplet-based methods still have a relatively large intra-class variation on the testing set. Therefore, we further consider the local similarity among these data pairs and preserve their similarity by introducing a novel constraint $dist(P_1, P_2) < dist(A, N)$, where data point P_1 and P_2 come from the same class. According to these constraints, a semantic-aware loss is defined to formulate a novel optimization problem over these data pairs, aiming to generate more discriminative binary codes.

In order to guarantee the generated binary codes to be as discriminative after quantization, most of the existing deep hashing methods minimize the error introduced by binary quantizing (quantization error) based on the pointwise quantization strategy [3], [9], [22], [40], [51], [70]. However, they ignore a crucial issue that how to efficiently maintain the well learned paired similarity after binary quantizing. Benefiting from the isometric mapping [47], we further develop a Hamming-isometric quantization strategy to ensure the well learned paired similarity as unchanged as possible, where a novel quantization loss is proposed to improve the quality of binary codes.

In addition, to fit the proposed algorithm into a multi-label image retrieval setting, an extension could be naturally adopted to LSDH so that it could preserve the neighbors' similarity in multi-label image retrieval tasks. Experimental results on four benchmark datasets demonstrate the effectiveness of the proposed method.

The contributions of this work are summarized as follows:

1) We propose a novel learning-based hashing method named LSDH to effectively perform feature learning and hash learning with CNN, where we exploit the underlying local structure of the data to preserve their local similarity.
2) We develop a Hamming-isometric quantization loss for enhancing the quality of binary codes, in which we aim to maintain the well learned paired similarity when binary quantizing is performed.
3) We extend the proposed LSDH to map multi-label images into compact binary codes according to their semantic similarity in a multi-label manner, so that the LSDH is capable of improving the performance for the multi-label images retrieval.
4) We evaluate the proposed method on several benchmark datasets. Experimental results show that the LSDH outperforms state-of-the-art hashing methods.

The rest of this paper is organized as follows. Section II gives a brief review of related work about deep hashing. Section III presents the procedure of the proposed LSDH. Section IV shows the details, results, and analysis of the experiment. Section V concludes the paper.

II. Related Work

Due to the efficiency in both search speed and storage cost, hashing has become the most popular technique for preserving similarity in large-scale image retrieval. In recent years, deep learning [14], [16], [20] has made great successes in image classification [54], [55], object detection [48], semantic segmentation [37] and visual tracking [43]. Emerging deep hashing methods [22], [62] also show great competitiveness in image retrieval. Generally, many hashing methods use two-step learning strategies [3], [9], [18], [40], [70]: metric learning [1] and binary quantizing [51]. Metric learning is applied to the dimensionality reduction of original spatial samples for obtaining low dimensional binary-like embedding, and binary quantizing transforms the binary-like embedding into binary codes. In this section, we mainly make a brief review of typical deep hashing methods.

A. Deep Hashing With Pairwise Samples

A series of deep hashing methods adopt feature learning strategies with pairwise labels for coding images into

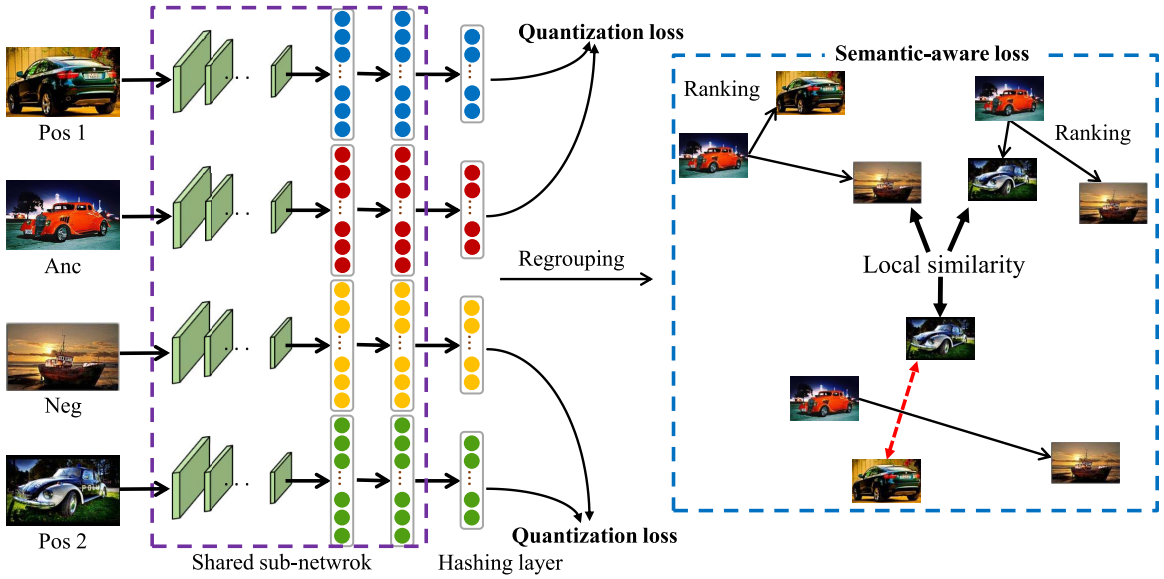


Fig. 2. An overview of the proposed local semantic-aware deep hashing framework, which includes a shared sub-network, a hashing layer, a quantization loss and a semantic-aware loss.

binary codes, aiming to preserve the similarity between a pair of data samples.

CNNH: Convolutional Neural Network Hashing (CNNH) [62] is a supervised hashing method, which utilizes a CNN to perform hash learning with pairwise labels. Specifically, it decomposes this process into two stages: a hash codes learning stage and a hash functions learning stage. Given n images $I = \{I_1, I_2, ...I_n\}$, in stage 1 (the hash code learning stage), CNNH performs the approximated hash codes learning for each raw image by optimizing the following loss function:

$$\min_{s} ||S - \frac{1}{q} H H^T||_F^2, \tag{1}$$

where $||\cdot||_F$ denotes the Frobenius norm; $H \in \{-1, 1\}^{n \times q}$ denotes the approximate hash codes matrix where each row is a q-dimensional hash codes; $S \in \{0, 1\}^{n \times n}$ denotes the semantic similarity of image pairs in I, in which $S_{ij} = 1$ when image I_i and I_j are similar, otherwise $S_{ij} = 0$. The codes inner product $H_{i.} H_{j.}^T$ is divided by codes length of q in order to fit $S_{ij} \in \{-1, 1\}$, thus optimizing Eq.(1) is equivalent to minimize the distance between $H_{i.} H_{j.}^T$ and qS. In the process of optimization, CNNH firstly relaxes the integer constraints of H and randomly initializes $H \in [-1, 1]^{n \times q}$, then optimizes the objective using a coordinate descent algorithm with newton directions. Thus, Eq.(1) can be re-formulated as:

$$\min ||H_{.j} H_{.j}^T - (qS - \sum_{c \neq j} H_{.c} H_{.c}^T)||_F^2, \tag{2}$$

where $H_{.j}$ and $H_{.c}$ denote the j-th and the c-th column of H respectively. In stage 2 (the hash functions learning stage), CNNH utilizes the CNN to simultaneously learn image feature and hash functions with the supervision of binary codes, where it adopts the LeNet [24] as its basic network framework, and constructs a latent layer with q bits as its output. During the training procedure, CNNH takes the hash codes learned in stage 1 as the ground-truth, and it also uses the provided label information to guide the hash functions learning when label information is available. Although CNNH can learn both feature representations and hash functions, its two-stage framework is suboptimal for hash learning as the deep feature representations learned in stage 2 cannot make feedback to the binary codes learned in stage 1.

B. Deep Hashing With Triplet-Wise Samples

Deep hashing with triplet labels is mainly designed to maximize the margin between positive sample pair and negative sample pair to preserve their relative similarity within a triplet.

DNNH: Different from the CNNH, Deep Neural Network Hashing (DNNH) [22] is a supervised hashing method employing an end-to-end deep hashing framework with triplet-wise constraints. DNNH adopts the Network in Network architecture [30] as its basic framework, where a shared sub-network with multiple convolution-pooling layers captures image representations, and it further adopts a divide-encode module encouraged by sigmoid activation function and a piece-wise threshold function for hash learning. Unlike CNNH [62] that exploits the similarity between image pairs, DNNH develops a triplet ranking loss [50] to maximize the margin between positive sample pairs and negative sample pairs when generating binary codes:

$$\begin{aligned} &l_{triplet}(\boldsymbol{b}(I),\ \boldsymbol{b}(I^+),\ \boldsymbol{b}(I^-)) \\ &\quad = max\left(0, 1 + \|\boldsymbol{b}(I) - \boldsymbol{b}(I^+)\|_H - |\boldsymbol{b}(I) - \boldsymbol{b}(I^-)\|_H\right) \\ &\qquad s.t.\ \boldsymbol{b}(I),\ \boldsymbol{b}(I^+),\ \boldsymbol{b}(I^-) \in \{-1, 1\}^k, \end{aligned} \tag{3}$$

where I, I^+, and I^- denote the anchor, the positive, and the negative sample in each triplet respectively; $\boldsymbol{b}(\cdot)$ denotes the discrete binary codes; $\|\cdot\|_H$ denotes the Hamming distance. Considering the integer constraints and non-differential

property in Eq.(3), binary codes are relaxed by using the range constraints and the Hamming distance is replaced by the Euclidean distance for facilitating loss computation and gradient updating.

Like the CNNH and DNNH, many other hashing methods [3], [32], [65] directly construct pairwise or triplet-wise constraint to obtain similar binary codes between a similar data pair or relative similar codes within a triplet. However, the proposed LSDH is different from them in motivation. In addition, many recent metric learning methods widely adopt the pairwise or triplet-wise constraint to preserve data pairs' similarity. For example, the Siamese network [6] learned contrastive embedding to reduce (increase) the distance between a positive (negative) pair. The FaceNet [50] proposed an online strategy by associating each positive pair to obtain the relative similarity within a triplet. Song *et al.* [45] proposed a lifted structure feature embedding, which takes full advantage of the training batches by lifting the vector of pairwise distances to obtain a relative similarity. Chen *et al.* [4] obtained the relative similarity by further reducing intra-class variation. Wang *et al.* [60] attempted to constrain the angle of the negative point within a triplet to obtain the relative similarity. Different from these methods, our method exploits the potential local structure of the data for hash learning, and employs the semantic relation to facilitate the generated binary codes preserving their local similarity, rather than the above-mentioned similarity or relative similarity.

In the following, we will detailedly discuss our two improvements: 1) exploring the local structure of the data to preserve their local similarity, 2) maintaining the well learned paired similarity after binary quantizing.

III. The Proposed Method

Most existing deep hashing methods are proposed to learn the similarity-preserving binary codes between a data pair and the potential local structure of the data is always overlooked. In quantization, they usually adopt the pointwise quantization scheme, e.g., L_1-norm constraint, L_2-norm constraint and smooth approximation function $tanh()$, to control the pointwise quantization error, which can not ensure the well learned paired similarity unchanged after quantization [70]. In this paper, the proposed LSDH takes full consideration of the local structure of the data distribution to perform feature learning and hash learning in a unified framework. Meanwhile, we present a Hamming-isometric quantization schema to maintain the well learned paired similarity, thus improving the retrieval performance.

The framework of LSDH is shown in Fig. 2. This architecture takes a quadruplet of image (an anchor, a negative sample, and two positive instances) as its input and mainly consists of four components: 1) a shared sub-network with multiple convolution-pooling layers and two fully-connected layers for extracting feature; 2) a hashing layer followed after the second fully-connected layer for coding each image into K-bit representations; 3) a semantic-aware loss being used to similarity learning; 4) a quantization loss being presented to enhance the quality of binary codes.

A. Semantic-Aware Hashing and Optimization

Given a image set consisting of n data points $\boldsymbol{I} = [I_1, I_2, ..., I_n]$, our goal is to map the data $\boldsymbol{I}$ to a Hamming space to obtain the corresponding compact representations. Suppose the output of the shared sub-network in LSDH is denoted by feature matrix $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n] \in \mathbb{R}^{d\times n}$ consisting of d-dimensional feature $\boldsymbol{x}_i$ and K hashing functions are learned to project the feature matrix into K-bit binary representation $\boldsymbol{B} = [\boldsymbol{b}_1, \boldsymbol{b}_2, ..., \boldsymbol{b}_n] \in \mathbb{R}^{K\times n}$. As in [2], [22], and [40], we use linear projections followed by an element-wise transformation as our hashing functions. Firstly, we can obtain the output of the hashing layer by linear projections, and the specific output is listed as follows:

$$\boldsymbol{h}_i = \boldsymbol{W}_H^T \boldsymbol{x}_i + \nu_H, \tag{4}$$

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:

$$\boldsymbol{b}_i = sign(\boldsymbol{h}_i), \tag{5}$$

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 LSDH in the next part, where we use one of the existing representative networks as our basic network.

The triplet-based input [22] used in learning to rank consists of an anchor image I, a positive image I^+ (similar) and a negative image I^-(dissimilar), and it is not enough to exploit the potential structure of data. In this paper, we define a quadruplet $Q(I_i, I_j, I_k, I_n)$ for hash learning, in which the quadruplet includes an anchor point I_i, two similar points I_j and I_k, and a dissimilar point I_n. For facilitating loss computation and fast convergence in training, a novel loss function termed semantic-aware loss is proposed to preserve their similarity over the generated binary codes of the quadruplet $Q(I_i, I_j, I_k, I_n)$. The specific loss function can be described as:

$$\begin{aligned} L(\boldsymbol{b}_i, \boldsymbol{b}_j, \boldsymbol{b}_k, \boldsymbol{b}_n) = & \, max(0, 1+\|\boldsymbol{b}_i-\boldsymbol{b}_j\|_H-\|\boldsymbol{b}_i-\boldsymbol{b}_n\|_H) \\ & + max(0, 1+\|\boldsymbol{b}_i-\boldsymbol{b}_k\|_H-\|\boldsymbol{b}_i-\boldsymbol{b}_n\|_H) \\ & + max(0, 1+\|\boldsymbol{b}_j-\boldsymbol{b}_k\|_H-\|\boldsymbol{b}_i-\boldsymbol{b}_n\|_H), \end{aligned} \tag{6}$$

where $\|\cdot\|_H$ represents the Hamming distance. The first two terms are designed for ranking to learn according to two different triplets $T(I_i, I_j, I_n)$ and $T(I_i, I_k, I_n)$, respectively. The third term takes consideration of the margin between the positive-positive data pair (I_j, I_k) and anchor-negative data pair (I_i, I_n), aiming to preserve their local similarity. The three terms jointly contribute to exploiting the potential local structure of the data for hash learning.

With the help of the three constraints, the semantic relation can be potentially employed to facilitate the binary codes preserving the similarity with its neighbors. Different from the pairwise or triplet-based hashing methods for obtaining similar codes between a data pair or relative similar codes

within a triplet, our method encourages the generated binary codes to preserve the local similarity. In consideration of robustness, we construct multiple such quadruplets for each anchor sample, in which we expect more similar data pairs to have similar codes.

Due to the non-differential property of the integer constraints, we relax Eq.(6) by replacing the Hamming distance with the Euclidean distance and replacing the integer constraints with the range constraints. Considering the error [32], [51] introduced by the range constraints as well as the data pairs' similarity being changed in two different spaces, we define a quantization loss to maintain the well learned paired similarity. Therefore, the final objective can be listed as follows:

$$min\ \hat{L}(\boldsymbol{h}_i, \boldsymbol{h}_j, \boldsymbol{h}_k, \boldsymbol{h}_n) + \lambda Q(\boldsymbol{h}_i, \boldsymbol{h}_j, \boldsymbol{h}_k, \boldsymbol{h}_n), \tag{7}$$

where $\hat{L}$ contributes to learning preserving-similarity binary codes; Q is used to maintain the paired similarity after binary quantizing; λ is a hyper-parameter for balancing the importance of the overall quantization part Q. In the subsection, we mainly concentrate on the similarity learning part $\hat{L}$, and the quantization part Q will be discussed detailedly in the next subsection, noting that the quantization part is an essential component of hashing. Then, the $\hat{L}$ could be specifically written as:

$$\begin{aligned}\hat{L}(\boldsymbol{h}_i, \boldsymbol{h}_j, \boldsymbol{h}_k, \boldsymbol{h}_n) = &\ max(0, 1+\|\boldsymbol{h}_i - \boldsymbol{h}_j\|_2^2 - \|\boldsymbol{h}_i - \boldsymbol{h}_n\|_2^2) \\ &+ max(0, 1+\|\boldsymbol{h}_i - \boldsymbol{h}_k\|_2^2 - \|\boldsymbol{h}_i - \boldsymbol{h}_n\|_2^2) \\ &+ max(0, 1+\|\boldsymbol{h}_j - \boldsymbol{h}_k\|_2^2 - \|\boldsymbol{h}_i - \boldsymbol{h}_n\|_2^2),\end{aligned} \tag{8}$$

and it is continuous and differentiable. According to the back-propagation algorithm, the gradients of $\hat{L}$ with respect to $\boldsymbol{h}_i$ are computed as:

$$\begin{aligned}\frac{\partial \hat{L}}{\partial \boldsymbol{h}_i} &= 2(\boldsymbol{h}_n - \boldsymbol{h}_j)\tau[\boldsymbol{h}_i, \boldsymbol{h}_j, \boldsymbol{h}_n]_+ + 2(\boldsymbol{h}_n - \boldsymbol{h}_k)\tau[\boldsymbol{h}_i, \boldsymbol{h}_k, \boldsymbol{h}_n]_+ \\ &\quad -2(\boldsymbol{h}_i - \boldsymbol{h}_n)\tau[\boldsymbol{h}_i, \boldsymbol{h}_j, \boldsymbol{h}_k, \boldsymbol{h}_n]_+ \\ \frac{\partial \hat{L}}{\partial \boldsymbol{h}_j} &= -2(\boldsymbol{h}_i - \boldsymbol{h}_j)\tau[\boldsymbol{h}_i, \boldsymbol{h}_j, \boldsymbol{h}_n]_+ + 2(\boldsymbol{h}_j - \boldsymbol{h}_k)\tau[\boldsymbol{h}_i, \boldsymbol{h}_j, \boldsymbol{h}_k, \boldsymbol{h}_n]_+ \\ \frac{\partial \hat{L}}{\partial \boldsymbol{h}_k} &= -2(\boldsymbol{h}_i - \boldsymbol{h}_k)\tau[\boldsymbol{h}_i, \boldsymbol{h}_k, \boldsymbol{h}_n]_+ - 2(\boldsymbol{h}_j - \boldsymbol{h}_k)\tau[\boldsymbol{h}_i, \boldsymbol{h}_j, \boldsymbol{h}_k, \boldsymbol{h}_n]_+ \\ \frac{\partial \hat{L}}{\partial \boldsymbol{h}_n} &= 2(\boldsymbol{h}_i - \boldsymbol{h}_n)\tau[\boldsymbol{h}_i, \boldsymbol{h}_j, \boldsymbol{h}_n]_+ + 2(\boldsymbol{h}_i - \boldsymbol{h}_n)\tau[\boldsymbol{h}_i, \boldsymbol{h}_k, \boldsymbol{h}_n]_+ \\ &\quad + 2(\boldsymbol{h}_i - \boldsymbol{h}_n)\tau[\boldsymbol{h}_i, \boldsymbol{h}_j, \boldsymbol{h}_k, \boldsymbol{h}_n]_+ ,\end{aligned} \tag{9}$$

where $\tau[\cdot]_+ = 1$ if the expression $[\cdot]_+$ is true and $\tau[\cdot]_+ = 0$ otherwise. $\tau[\boldsymbol{h}_i, \boldsymbol{h}_j, \boldsymbol{h}_n]_+$ is equivalent to $\tau[1 + \|\boldsymbol{h}_i - \boldsymbol{h}_j\|_2^2 > \|\boldsymbol{h}_i - \boldsymbol{h}_n\|_2^2]$, the same to $\tau[\boldsymbol{h}_i, \boldsymbol{h}_k, \boldsymbol{h}_n]_+$, and $\tau[\boldsymbol{h}_i, \boldsymbol{h}_j, \boldsymbol{h}_k, \boldsymbol{h}_n]_+$ is equivalent to $\tau[1 + \|\boldsymbol{h}_j - \boldsymbol{h}_k\|_2^2 > \|\boldsymbol{h}_i - \boldsymbol{h}_n\|_2^2]$. Thus, the gradients can be easily integrated into the back propagation of CNN. It is observed that the semantic-aware loss provides informative gradient signal for the positive-positive pair that is beneficial to preserve their similarity in a local neighborhood.

B. Hamming-Isometric Quantization

In quantization, existing hashing approaches use binary quantizing to transform binary-like embedding h_i into binary codes $\boldsymbol{b}_i$. For similarity-preserving hash learning, it is necessary to keep discriminability invariance of the feature after quantization. Therefore, in training stage, a regularizer with L_1/L_2-norm constraint [9], [18], [26], [34], [40], [51] is widely used to control the quantization error. Following [32], [40], we use the L_1-norm imposed on this error:

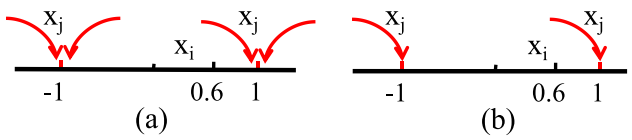


Fig. 3. The impact on the solver x_j by L_1-norm discrete constraint quantization loss (a) and Hamming-isometric quantization loss (b), when a component or a bit of feature x_i is 0.6. The red arrow represents all possible approximation direction about the solver of x_j.

$$Q_{L_1}(x_i) = \|\boldsymbol{h}_i - \boldsymbol{b}_i\|_1, \tag{10}$$

Although the L_1-norm based pointwise quantization schema can reduce the quantization error, it can't generate high-quality binary codes for efficient retrieval. In image retrieval stage, we always weight the similarity of image pairs by the Hamming distance, while the similarity of image pairs is weighted by the Euclidean distance $\|\boldsymbol{h}_i - \boldsymbol{h}_i\|_2^2$ in training stage. Therefore, apart from enforcing the pointwise quantization error as small as possible, we also expect data pairs' similarity should be maintained after quantization. According to the way of balancing similarity, the data pair should be an isometric mapping [47] from the Euclidean space to the Hamming space, that is to say, the distance should be consistent between the pairwise binary codes and the corresponding binary-like embedding pairs.

Since the distance in Hamming space and Euclidean space are computed in a different manner, we unify the distance calculation method of binary codes with $L2$-norm, i.e., the Euclidean distance, and the proposed Hamming-isometric quantization loss can be naturally described as follows:

$$\begin{aligned}Q_{Ham}(h_i, h_j) = &\ \|\boldsymbol{h}_i - \boldsymbol{b}_i\|_1 + \|\boldsymbol{h}_j - \boldsymbol{b}_j\|_1 \\ &+\mu(|\ \|\boldsymbol{h}_i - \boldsymbol{h}_j\|_2^2 - \|\boldsymbol{b}_i - \boldsymbol{b}_j\|_2^2\ |),\end{aligned} \tag{11}$$

where μ is a hyper-parameter to weight the importance of Hamming-isometric term. Since the absolute value operation in the objective function is non-differentiable at some certain points, we use unit sub-gradient instead in those cases. Therefore, the gradient of Q_{Ham} with respect to h_i can be written as:

$$\begin{aligned}\frac{\partial Q_{Ham}}{\partial \boldsymbol{h}_i} &= \delta(\boldsymbol{h}_i) + 2\mu(\boldsymbol{h}_i - \boldsymbol{h}_j)sign(\|\boldsymbol{h}_i - \boldsymbol{h}_j\|_2^2 - \|\boldsymbol{b}_i - \boldsymbol{b}_j\|_2^2) \\ \frac{\partial Q_{Ham}}{\partial \boldsymbol{h}_j} &= \delta(\boldsymbol{h}_j) - 2\mu(\boldsymbol{h}_i - \boldsymbol{h}_j)sign(\|\boldsymbol{h}_i - \boldsymbol{h}_j\|_2^2 - \|\boldsymbol{b}_i - \boldsymbol{b}_j\|_2^2),\end{aligned} \tag{12}$$

where

$$\delta(x) = \begin{cases} 1, & -1 \le x \le 0\ or\ x \ge 1 \\ -1, & otherwise \end{cases}$$

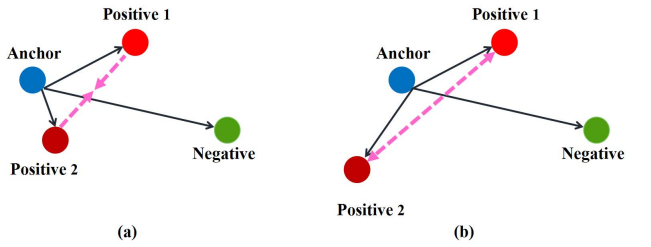


Fig. 4. Local semantic-aware hashing for multi-label images. We first work on two triplets to get the correct ranking, then we further compute the similarity of the positive-positive sample pair in a multi-label way, aiming to drag them closer when they share semantic labels as shown in (a), otherwise push them far away as shown in (b).

From the proposed Hamming-isometric quantization loss, we can observe that the novel schema preserves both the discriminability of real-valued feature and the consistency of image pairs between two different spaces. On the other hand, it is also beneficial to guide the network to search for the optimal solver. For example, when the value of a component in feature x_i is 0.6, the corresponding bit value of feature x_j will be close to 1 or -1 from four directions by L_1-norm quantization, as is shown in Fig. 3 (a). In the Hamming-isometric quantization schema, when x_j and x_i have the same sign, x_j will be close to 1 from the left of 1 endpoint, otherwise x_j will be close to -1 from the left of -1 endpoint, as is shown in Fig. 3 (b).

C. Multi-Label Based Hashing

The multi-label image has semantic information of multiple domains. In this subsection, we focus on exploiting the local structure of data in a multi-label setting. In the above-defined quadruplet $Q(I_i, I_j, I_k, I_n)$, the positive-positive sample pair (I_j, I_k) may not share the same semantic information for all labels, so we make a slight modification on the previous defined semantic-aware loss. Specifically, we define the semantic similarity between I_j and I_k as s_{jk} in a multi-label manner, where $s_{jk} = 1$ if they share at least one same semantic label, and $s_{jk} = 0$ otherwise. According to the quadruplet $Q(I_i, I_j, I_k, I_n)$, the semantic-aware loss in the multi-label setting can be described as:

$$\begin{aligned} L(\boldsymbol{b}_i, \boldsymbol{b}_j, \boldsymbol{b}_k, \boldsymbol{b}_n) = & \, max(0, 1+\|\boldsymbol{b}_i - \boldsymbol{b}_j\|_H - \|\boldsymbol{b}_i - \boldsymbol{b}_n\|_H) \\ & + max(0, 1+\|\boldsymbol{b}_i - \boldsymbol{b}_k\|_H - \|\boldsymbol{b}_i - \boldsymbol{b}_n\|_H) \\ & + s_{jk} max(0, 1+\|\boldsymbol{b}_j - \boldsymbol{b}_k\|_H - \|\boldsymbol{b}_i - \boldsymbol{b}_n\|_H) \\ & + (1 - s_{jk}) max(0, \ 1 - \|\boldsymbol{b}_j - \boldsymbol{b}_k\|_H) \end{aligned} \tag{13}$$

Fig. 4 demonstrates the overview of multi-label image hash learning. Different from the Eq.(6), we dray the positive-positive sample pair closer to each other when $s_{jk} = 1$, as shown in Fig. 4(a), otherwise we push the positive-positive sample pair far away, as shown in Fig. 4(b). In addition, we also relax the objective function by replacing the Hamming distance with the Euclidean distance and replacing the integer constraints with the range constraints. Following the single-label hash learning on section B, we simultaneously adopt a quantization loss for maintaining the well learned paired similarity after quantization. Therefore, The new objective can be short for $\hat{L}(\boldsymbol{h}_i, \boldsymbol{h}_j, \boldsymbol{h}_k, \boldsymbol{h}_n) + \lambda Q(\boldsymbol{h}_i, \boldsymbol{h}_j, \boldsymbol{h}_k, \boldsymbol{h}_n)$, like that in Eq. (7), and the quantization part is the same to section B. The gradients of $\hat{L}$ with respect to h_i are computed as:

$$\begin{aligned} \frac{\partial \hat{L}}{\partial \boldsymbol{h}_i} = & \, 2(\boldsymbol{h}_n - \boldsymbol{h}_j)\tau[\boldsymbol{h}_i, \boldsymbol{h}_j, \boldsymbol{h}_n]_+ + 2(\boldsymbol{h}_n - \boldsymbol{h}_k)\tau[\boldsymbol{h}_i, \boldsymbol{h}_k, \boldsymbol{h}_n]_+ \\ & - 2s_{jk}(\boldsymbol{h}_i - \boldsymbol{h}_n)\tau[\boldsymbol{h}_i, \boldsymbol{h}_j, \boldsymbol{h}_k, \boldsymbol{h}_n]_+ \\ \frac{\partial \hat{L}}{\partial \boldsymbol{h}_j} = & -2(\boldsymbol{h}_i - \boldsymbol{h}_j)\tau[\boldsymbol{h}_i, \boldsymbol{h}_j, \boldsymbol{h}_n]_+ \\ & + 2s_{jk}(\boldsymbol{h}_j - \boldsymbol{h}_k)\tau[\boldsymbol{h}_i, \boldsymbol{h}_j, \boldsymbol{h}_k, \boldsymbol{h}_n]_+ \\ & - 2(1 - s_{jk})(\boldsymbol{h}_j - \boldsymbol{h}_k)\tau[\boldsymbol{h}_j, \boldsymbol{h}_k]_+ \\ \frac{\partial \hat{L}}{\partial \boldsymbol{h}_k} = & -2(\boldsymbol{h}_i - \boldsymbol{h}_k)\tau[\boldsymbol{h}_i, \boldsymbol{h}_k, \boldsymbol{h}_n]_+ \\ & - 2s_{jk}(\boldsymbol{h}_j - \boldsymbol{h}_k)\tau[\boldsymbol{h}_i, \boldsymbol{h}_j, \boldsymbol{h}_k, \boldsymbol{h}_n]_+ \\ & + 2(1 - s_{jk})(\boldsymbol{h}_j - \boldsymbol{h}_k)\tau[\boldsymbol{h}_j, \boldsymbol{h}_k]_+ \\ \frac{\partial \hat{L}}{\partial \boldsymbol{h}_n} = & \, 2(\boldsymbol{h}_i - \boldsymbol{h}_n)\tau[\boldsymbol{h}_i, \boldsymbol{h}_j, \boldsymbol{h}_n]_+ + 2(\boldsymbol{h}_i - \boldsymbol{h}_n)\tau[\boldsymbol{h}_i, \boldsymbol{h}_k, \boldsymbol{h}_n]_+ \\ & + 2s_{jk}(\boldsymbol{h}_i - \boldsymbol{h}_n)\tau[\boldsymbol{h}_i, \boldsymbol{h}_j, \boldsymbol{h}_k, \boldsymbol{h}_n]_+, \end{aligned} \tag{14}$$

where we use $\tau[\boldsymbol{h}_j, \boldsymbol{h}_k]_+$ to represent $\tau[1 - \|\boldsymbol{h}_j - \boldsymbol{h}_k\|_2^2 > 0]$, while $\tau[\boldsymbol{h}_i, \boldsymbol{h}_j, \boldsymbol{h}_n]_+$, $\tau[\boldsymbol{h}_i, \boldsymbol{h}_k, \boldsymbol{h}_n]_+$ and $\tau[\boldsymbol{h}_i, \boldsymbol{h}_j, \boldsymbol{h}_k, \boldsymbol{h}_n]_+$ being the same as in Eq.(9).

D. Overall Objective

The overall objective function of the proposed local semantic-aware deep hashing with Hamming-isometric quantization is given as:

$$min \sum_{i,j,k,l} \hat{L}\ (\boldsymbol{h}_i, \boldsymbol{h}_j, \boldsymbol{h}_k, \boldsymbol{h}_n) + \lambda \sum_{i,j} Q_{Ham}(\boldsymbol{h}_i, \boldsymbol{h}_j), \tag{15}$$

where λ is a hyper-parameter to balance the presented quantization loss, and the data pair of Q_{Ham} are from the similarity learning $\hat{L}$. It is observed that $\lambda\mu$ is used to weight the importance of Hamming-isometric term. In parameters updating, we adopt stochastic gradient descent algorithm [20] to update all parameters until convergence.

IV. Experiments

To evaluate the effectiveness of the proposed LSDH, extensive experiments are conducted on three single-label image datasets and a multi-label dataset. Besides, a variant of our framework is implemented, namely LSDHL_1, where we only use the L_1-norm discrete constraint as the quantization loss. To better show the advantages of the LSDH, several state-of-the-art hashing methods are compared under several retrieval evaluation metrics.

A. Datasets

Experiments are conducted on three large-scale single-label datasets, i.e., **CIFAR-10**[1] [19], **SUN397**[2] [63],

[1] http://www.cs.toronto.edu/ riz/cifar.html
[2] http://vision.princeton.edu/projects/2010/SUN/

CIFAR-20[3] [19] and a large-scale multi-label datasets **NUS-WIDE**[4] [7]. These datasets are introduced in details as follows:

- CIFAR-10 is a benchmark image dataset for similarity retrieval, consisting of 60,000 color images. Each image belongs to one of the ten categories, and the size of each image is 32×32. Following the same setting in [70], we randomly sampled 1,000 images per class as the query images. For the unsupervised methods, all the rest of the images are used as the training set. For the supervised methods, 5,000 images (500 images per class) are randomly selected from the rest of images for training.
- SUN397 consists of 108,754 images from 397 scene categories. Following the similar setting in [34], we use the subset of 32,099 images that are associated with the 29 largest categories, in which each category consists of at least 600 images. In our experiments, we randomly sample 1,000 images as the query set. For the unsupervised methods, all the rest of the images are used as the training set. For the supervised methods, 5,000 images are further randomly selected from the rest images as the training set.
- CIFAR-20 is another famous dataset for object recognition and image retrieval, which includes 20 superclasses grouped from the CIFAR-100 dataset.[5] and each class contains 3,000 images of size 32×32. Following the similar setting in [70], we randomly sample 100 images per class as a test query set. For the unsupervised methods, all the rest of the images are used for training. For the supervised methods, 500 images per class are further randomly selected from the rest images for training.
- NUS-WIDE is a public web image dataset downloaded from Flickr.com, and it contains nearly 270,000 images with one or multiple labels of 81 semantic concepts. Following the setting [22] and [39] the subset of 195,834 images that are associated with the 21 most frequent concepts are used, where each concept consists of at least 5,000 images. We also randomly sample 100 images per class as a query set. For the unsupervised methods, all the rest of the images are used for training. For the supervised methods, 500 images per class are further randomly selected from the rest images for training.

B. Compared Algorithms

To demonstrate the performance of the proposed LSDH method, we compare it with several popular hashing methods:

- LSH [8]: It generates a set of random linear projections as hash functions.
- SH [61]: It learns hash functions by keeping the neighbors' consistency in both the input space and the Hamming space.
- ITQ [9]: It minimizes the quantization loss by a projection matrix and identifies an orthogonal rotation matrix to refine the initial projection function learned by PCA/CCA.
- KSH [21]: It minimizes the Hamming distances of similar pairs and simultaneously maximizes the Hamming distances of dissimilar pairs based on a kernel-based supervised hashing model.
- FastH [28]: It alternately seeks binary codes and learns hash functions in two steps. Binary codes learning is formulated as binary quadratic problems while hash functions are accomplished by training a standard binary classifier.
- SDH [51]: It learns hash functions and one linear classifier by obtained hashing codes.
- CNNH [62]: It first solves for the hashing codes based on the pair-wise similarities on the training set, and utilizes the hashing codes and semantic labels to learn hash functions.
- DNNH [22]: It learns the hash functions by maximizing the margin between positive sample pair and negative sample pair within a triplet.
- DHN [70]: It obtains similarity-preserving binary codes by jointly learning robust image representations tailored to hash coding and formally control the quantization error.
- DSH [32]: It takes pairs of images (similar/dissimilar) as training inputs and encourages the output of each image to approximate discrete values and simultaneously imposing regularization on the real-valued outputs to approximate the desired discrete values.
- HashNet [3]: It learns hash functions by a continuation method with convergence guarantees, and take the imbalanced similarity data into consideration.

C. Experimental Settings and Protocols

We implement the proposed method based on the open-source Caffe [17] framework. The VGG-16 [56] network is adopted as our basic networks, which has been pre-trained on the ImageNet2012 dataset [49]. For the hashing layer, we set its learning rate to be 10 times of that in the preceding layers. The initial learning rate is 0.0005 and the weight decay parameter is 0.0005. For the two hyper-parameter parameters in the quantization loss, we set $\lambda = 0.8$, $\mu = 0.25$ in the single-label dataset and $\lambda = 0.8$, $\mu = 0.75$ in the multi-label dataset, respectively.

In order to comprehensively compare the performance of different methods, we utilize two search procedures, i.e. Hamming ranking and hash lookup [66]. For Hamming ranking, we use three evaluation criterias: 1) precision@500, i.e., the average precision of the first 500 returned images, 2) precision@k, i.e., the top k closest images in the Hamming space and 3) precision-recall curves. Regarding the hash lookup, we adopt precision@R=2 for evaluation, i.e., precision within a Hamming distance of 2. Since the Mean Average Precision (MAP) is an aggregative indicator of the overall performance, we also utilize the MAP to evaluate the retrieval performance. Considering the calculation of MAP being not inefficient in a large-scale dataset, we report the results of top 5,000 returned neighbors for the NUS-WIDE dataset.

[3]https://www.cs.toronto.edu/ kriz/cifar.html
[4]http://lms.comp.nus.edu.sg/research/NUS-WIDE.htm
[5]https://www.cs.toronto.edu/ kriz/cifar.html

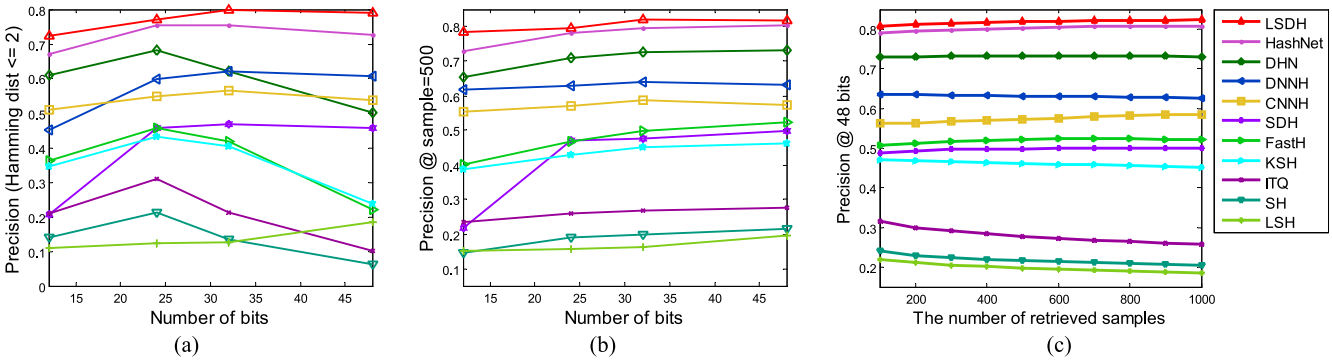


Fig. 5. Comparative evaluations on the CIFAR-10 dataset. (a) precision curves with Hamming radius r = 2. (b) precision curves with top 500 retrieved samples. (c) precision curves with 48 hashing bits.

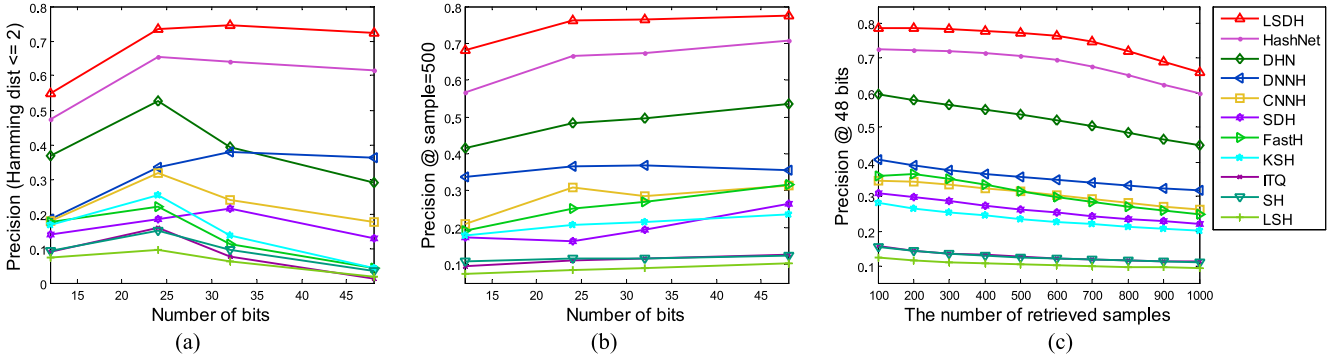


Fig. 6. Comparative evaluations on the SUN397 dataset. (a) precision curves with Hamming radius r = 2. (b) precision curves with top 500 retrieved samples. (c) precision curves with 48 hashing bits.

TABLE I

MAP COMPARISON OF DIFFERENT HASHING ALGORITHMS ON THE CIFAR-10 DATASET

Method	CIFAR-10(bits)			
	12	24	32	48
LSDH	**0.746**	**0.752**	**0.779**	**0.782**
LSDHL_1	0.722	0.738	0.766	0.771
HashNet [3]	0.668	0.737	0.755	0.766
DHN [70]	0.620	0.633	0.644	0.657
DSP [32]	0.601	0.629	0.662	0.689
DNNH [22]	0.552	0.566	0.558	0.581
CNNH [62]	0.465	0.521	0.521	0.532
SDH-CNN	0.280	0.550	0.567	0.569
FastH-CNN	0.515	0.583	0.597	0.610
KSH-CNN	0.470	0.524	0.539	0.547
SDH [51]	0.203	0.340	0.354	0.351
FastH [28]	0.293	0.345	0.365	0.391
KSH [21]	0.303	0.337	0.346	0.356
SH [61]	0.131	0.135	0.133	0.130
ITQ [9]	0.162	0.169	0.172	0.175
LSH [8]	0.121	0.126	0.120	0.120

To guarantee fair comparisons, all deep hashing methods mentioned above are implemented using the Caffe framework [17], and the source codes are provided by the corresponding authors. In the experiments, all the methods use identical training and testing sets. For the deep hashing methods, we use the raw image pixels as input. For traditional hashing method, images are represented by the 512-dimensional handcrafted descriptor GIST [46]. In addition, in order to show the accuracy by the CNN feature, we also represent each image by a 4096-dimensional CNN feature that is extracted from the VGG-16 pre-trained on the ImageNet, where we use '$-CNN$' to distinguish them in our experimental comparisons.

D. Experiment Results and Analysis

1) Results on CIFAR-10: Table I shows the MAP scores with all the returned results on the CIFAR-10 dataset for different lengths codes. Among various methods compared, it is clear that most of the deep hashing approaches constantly outperform the traditional hashing methods both in handcrafted feature and CNN feature, e.g., LSH, KSH, and SDH, This could be attributed to the fact that deep networks enable joint learning of feature representations and hash functions from raw pixels, and the two processes can promote each other for improving the generation ability of hash coding. In addition, the MAP scores of traditional hashing methods with CNN feature distinctly outperform that with the handcrafted feature, and it indicates the CNN feature have more powerful representation ability for raw images.

Compared to the state-of-the-art deep hashing methods, the proposed LSDH method improves the average MAP from 46.6%(CNNH), 55.2% (DNNH), 62.0%(DHN), 66.8%(HashNet) to 74.6% in 12-bit codes, this is because that the CNN, DHN, HashNet mainly focus on the similar

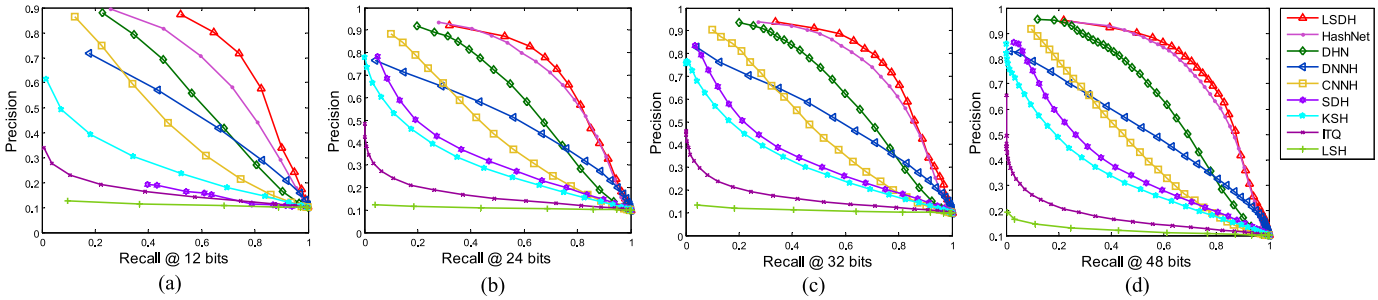


Fig. 7. Precision recall curves with different lengths codes on the CIFAR-10 dataset.

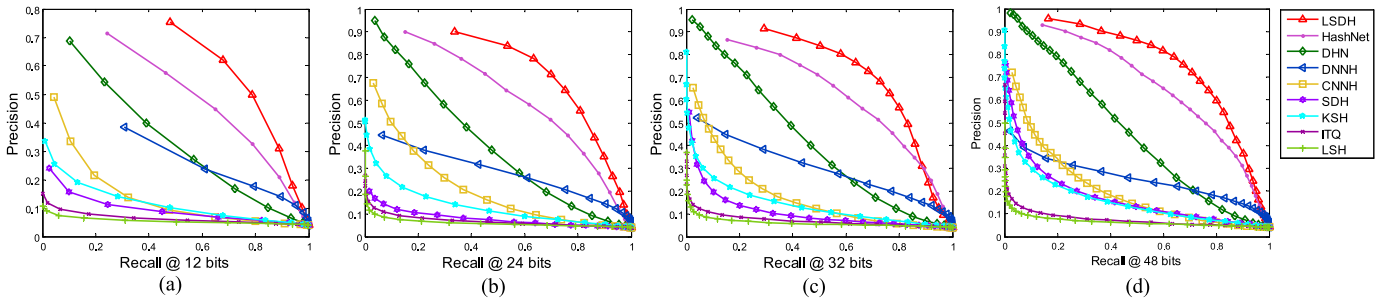


Fig. 8. Precision recall curves with different lengths codes on the SUN397 dataset.

TABLE II
MAP COMPARISON OF DIFFERENT HASHING ALGORITHMS ON THE SUN397 DATASET

Method	SUN397(bits)			
	12	24	32	48
LSDH	**0.646**	**0.728**	**0.730**	**0.742**
LSDHL_1	0.645	0.717	0.725	0.738
HashNet[3]	0.513	0.619	0.636	0.671
DHN [70]	0.344	0.401	0.414	0.453
DSP [32]	0.361	0.416	0.442	0.495
DNNH [22]	0.294	0.315	0.320	0.303
CNNH [62]	0.155	0.236	0.214	0.243
SDH-CNN	0.580	0.581	0.562	0.694
FastH-CNN	0.580	0.690	0.712	0.721
KSH-CNN	0.462	0.563	0.582	0.603
SDH [51]	0.124	0.115	0.137	0.200
FastH [28]	0.140	0.183	0.200	0.238
KSH [21]	0.126	0.150	0.157	0.177
ITQ [9]	0.079	0.085	0.085	0.090
SH [61]	0.070	0.078	0.080	0.087
LSH [8]	0.056	0.066	0.069	0.076

feature of image pair, and DNNH mainly utilizes the ranking loss to maximize the margin between positive image pair and negative image pair to obtain the relative similarity. However, the proposed LSDH takes into consideration the local structure of the data for hash learning and encourages the generated binary codes to preserve their local similarity based on their semantic relation. Besides, the LSDH method shows some performance gain against the LSDHL_1, and it demonstrates that Hamming-isometric quantization schema is favorable to generate more compact binary codes for image retrieval.

More comprehensive results are demonstrated in Fig. 5. Fig. 5(a) shows the precision curves within Hamming radius 2 for different lengths codes. Fig. 5(b) illustrates the precision curves within the top 500 retrieved neighbors with various numbers of bits. Fig. 5(c) gives the precision curves within different numbers of top retrieved neighbors in 48 bits. In addition, the precision-recall curves with different numbers of bits are shown in Fig. 7. Compared to the state-of-the-art methods, we can observe that our method consistently obtain the best performance under these metrics, because our method exploits the intrinsic structure of the data and the generated binary codes by our model are more discriminative yet compact.

2) Results on SUN397: SUN397 includes more detailed information and is more challenging than CIFAR-10. Table II shows the retrieval MAP results on the SUN397 dataset for various bits. We can observe that the proposed LSDH achieves the best results among all state-of-the-art hashing methods. The LSDH outperforms the traditional hashing method with the hand-crafted feature in a large margin. Although the MAP scores of traditional hashing method with CNN feature have achieved improvements greatly and even outperform some deep hashing method, our method still surpasses them by a clear advantage. Besides, we improve the retrieval MAP from 15.5% (CNNH), 29.4% (DNNH), 34.4% (DHN), 51.3% (HashNet) to 64.6% in 12-bit binary codes. This is because the LSDH employs the semantic relation to preserve their local similarity, rather than the similarity between a data pair or the relative similarity within a triplet. Meanwhile, the LSDH maintains the well learned paired similarity as much as possible after binary quantizing, thus the binary codes are more effective for retrieval.

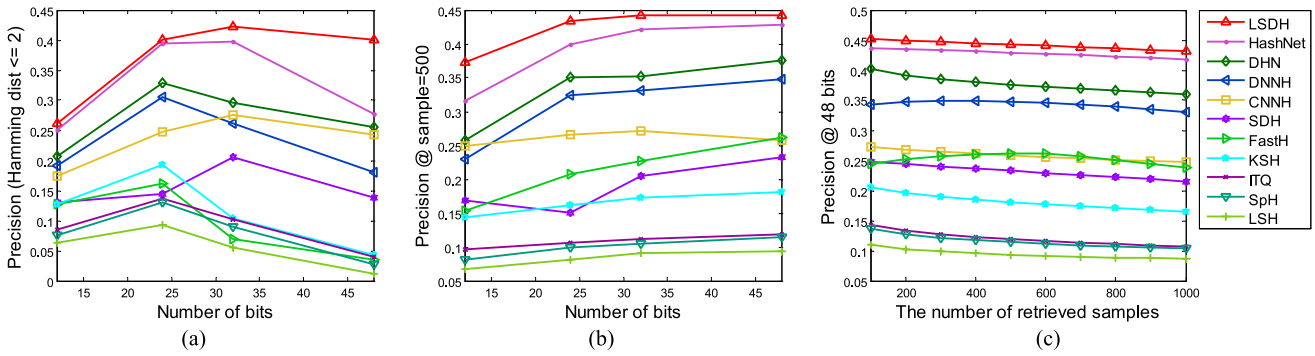


Fig. 9. Comparative evaluations on the CIFAR-20 dataset. (a) precision curves with Hamming radius r = 2. (b) precision curves with top 500 retrieved samples. (c) precision curves with 48 hashing bits.

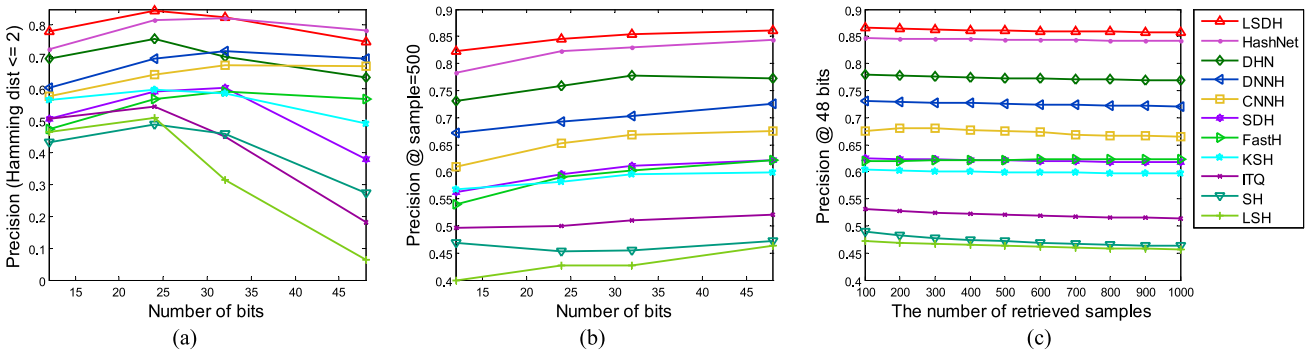


Fig. 10. Comparative evaluations on the NUS-WIDE dataset. (a) precision curves with Hamming radius r = 2. (b) precision curves with top 500 retrieved samples. (c) precision curves with 48 hashing bits.

TABLE III
MAP COMPARISON OF DIFFERENT HASHING ALGORITHMS ON THE CIFAR-20 DATASET

Method	CIFAR-20(bits)			
	12	24	32	48
LSDH	**0.310**	**0.354**	**0.358**	**0.360**
LSDHL_1	0.296	0.338	0.339	0.338
HashNet [3]	0.241	0.320	0.340	0.345
DHN [70]	0.192	0.218	0.226	0.233
DSP [32]	0.130	0.157	0.168	0.173
DNNH [22]	0.191	0.204	0.205	0.196
CNNH [62]	0.125	0.132	0.127	0.132
SDH-CNN	0.194	0.205	0.261	0.267
FastH-CNN	0.200	0.260	0.276	0.309
KSH-CNN	0.161	0.192	0.200	0.218
SDH [51]	0.108	0.108	0.132	0.151
FastH [28]	0.105	0.134	0.143	0.162
KSH [21]	0.095	0.102	0.108	0.113
ITQ [9]	0.068	0.071	0.072	0.075
SH [61]	0.063	0.067	0.070	0.072
LSH [8]	0.057	0.062	0.065	0.065

Fig. 6(a) shows the precision curves within Hamming radius 2 for different lengths codes. it is clear that the proposed LSDH approach gets the best search accuracy on all codes lengths (over 70.0 % search precision in 24 bits, 32 bits, 48 bits). Fig. 6(b) illustrates the precision curves within the top 500 retrieved neighbors with various numbers of bits (over 70.0 % search precision except for 12 bits). Fig. 6(c) gives the precision curves within different numbers of top retrieved neighbors with 48-bit codes (the search precision is consistently the best). In addition, Fig. 8 shows the precision-recall curves with respect to different lengths codes, and the proposed method consistently has obtained the best precision.

3) Results on CIFAR-20: To further verify the retrieval performance of the proposed method, we compare LSDH with the state-of-the-art hashing algorithms on the CIFAR-20 dataset. CIFAR-20 is another famous dataset for image retrieval and includes 20 super-classes grouped from the CIFAR-100 dataset.[6] Table III shows the MAP results against the state-of-the-art hashing methods, and it is observed that the LSDH works the best. Due to more detail information in this dataset, the overall retrieval results of the CIFAR-20 are inferior to that of the CIFAR-10. In addition, with the codes becoming longer, e.g., from 12 to 48, the gain of the MAP become smaller in all methods, one reasonable explanation is that the manifold distribution in more classes setting is much more complicated to estimate.

Fig. 9(a) shows the precision within a Hamming distance of 2. Fig. 9(b) shows the precision with the top 500 samples. Fig. 9(c) illustrates the precision@k with 48 bits, and k ranges from 100 to 1,000. It is observed that LSDH works the best compared to all other methods. For the precision within Hamming distance of 2, LSDH could keep a relatively good

[6]https://www.cs.toronto.edu/ kriz/cifar.html

TABLE IV
MAP COMPARISON OF DIFFERENT HASHING ALGORITHMS ON THE NUS-WIDE DATASET

Method	NUS-WIDE(bits)			
	12	24	32	48
LSDH	**0.813**	**0.832**	**0.844**	**0.852**
LSDHL_1	0.800	0.825	0.838	0.846
HashNet [3]	0.782	0.817	0.825	0.836
DHN [70]	0.708	0.735	0.748	0.758
DSP [32]	0.709	0.720	0.731	0.735
DNNH [22]	0.674	0.697	0.713	0.715
CNNH [62]	0.611	0.618	0.625	0.608
SDH-CNN	0.798	0.825	0.825	0.829
FastH-CNN	0.767	0.809	0.826	0.831
KSH-CNN	0.760	0.788	0.807	0.817
SDH [51]	0.530	0.546	0.536	0.582
FastH [28]	0.496	0.568	0.596	0.613
KSH [21]	0.556	0.572	0.581	0.588
SH [61]	0.33	0.426	0.426	0.423
ITQ [9]	0.452	0.468	0.472	0.477
LSH [8]	0.403	0.421	0.426	0.441

result in various numbers of bits, and the performance of other methods drops drastically when more bits are generated. The reason for this is that the LSDH takes consideration of the local structure of the data for hash coding, and it is prone to preserve the similarity with its neighbors.

4) Results on NUS-WIDE: To verify the performance of LSDH in multi-label image retrieval, we compare it with several state-of-the-art hashing algorithms on the NUS-WIDE dataset. In all experiments, the similarity between image pairs is defined according to whether they share semantic labels. Table IV shows the MAP scores of all compared methods, we can observe that our approach consistently outperforms these methods. For example, on the 12-bit codes, the LSDH first exceeds the MAP score 81.3% while the state-of-the-art MAP value is 78.1% [3]. In addition, it is observed that the traditional hashing methods with CNN feature achieve more than the 20% absolute gain of MAP scores compared to the hand-crafted feature, and this indicates CNN feature is more suitable to act as a representation for raw images.

The precisions within Hamming distance of 2 are shown in Fig. 10(a), and LSDH achieves about 80.0% precision on all code lengths. The precision curves within the top 500 retrieved samples are shown in Fig. 10(b), and it is observed that LSDH achieves a steady and high precision over 81.0%. The precision curves at 48-bit binary codes of different numbers of top retrieved samples are illustrated in Fig. 10(c), and LSDH achieves over 85.0% accuracy for different numbers of top retrieved samples. Under the three evaluation metrics, our method outperforms other state-of-the-art supervised hashing methods, which further demonstrates the benefits of exploiting the local structure of the data in hash learning.

5) LSDH Versus Metric Learning: Hash learning consists of metric learning and quantization. In the field of metric learning, it generally utilizes contrastive loss [12], triplet loss [50] and their variants for similarity learning.

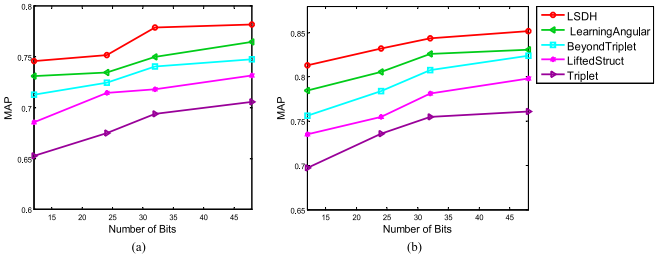


Fig. 11. The MAP scores of binary codes of different methods based on the CIFAR-10 and NUS-WIDE datasets.

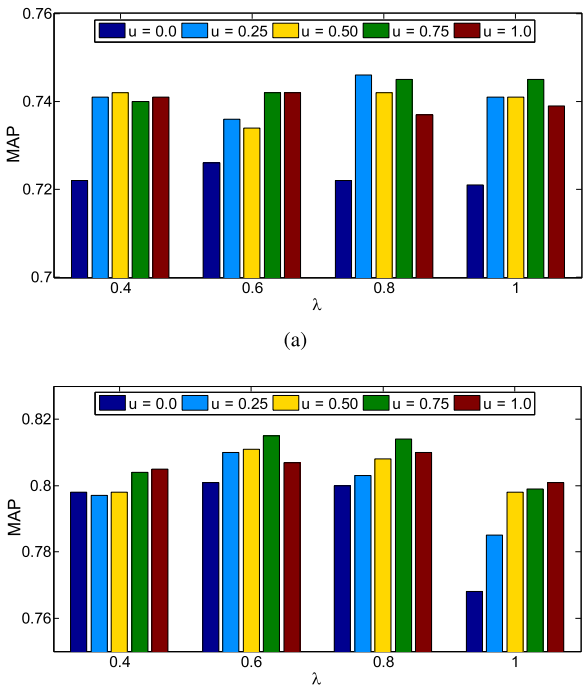


Fig. 12. The MAP of LSDH @ 12 bits w.r.t. tradeoff parameter $\lambda \in [0, 1]$ and $\mu \in [0, 1]$. (a) CIFAR-10. (b) NUS-WIDE.

In order to further validate the effectiveness of the proposed LSDH, we make comparisons with these typical metric learning method for the application of image retrieval, including LiftedStruct [45], BeyondTriplet [4], LearningAngular [60] and the baseline triplet loss [50]. Noting that triplet loss in [50] uses its assigned second network. Considering the error between the real-valued output and the discrete binary codes, we employ the proposed quantization technique after metric learning for fair comparisons.

Based on the mean of ten-times running results, the MAP scores of these metric learning methods on dataset CIFAR-10 and NUS-WIDE are shown in Fig. 11. It is clear that our approach obviously surpasses the baseline triplet loss, and shows competitive results over other three metric learning methods on different lengths codes. Specifically, we can achieve an average absolute increase of 5.21% against LiftedStruct, 3.30% against BeyondTriplet and 1.95% against LearningAngular on the CIFAR-10 dataset. On the NUS-WIDE, we can achieve an average absolute increase of

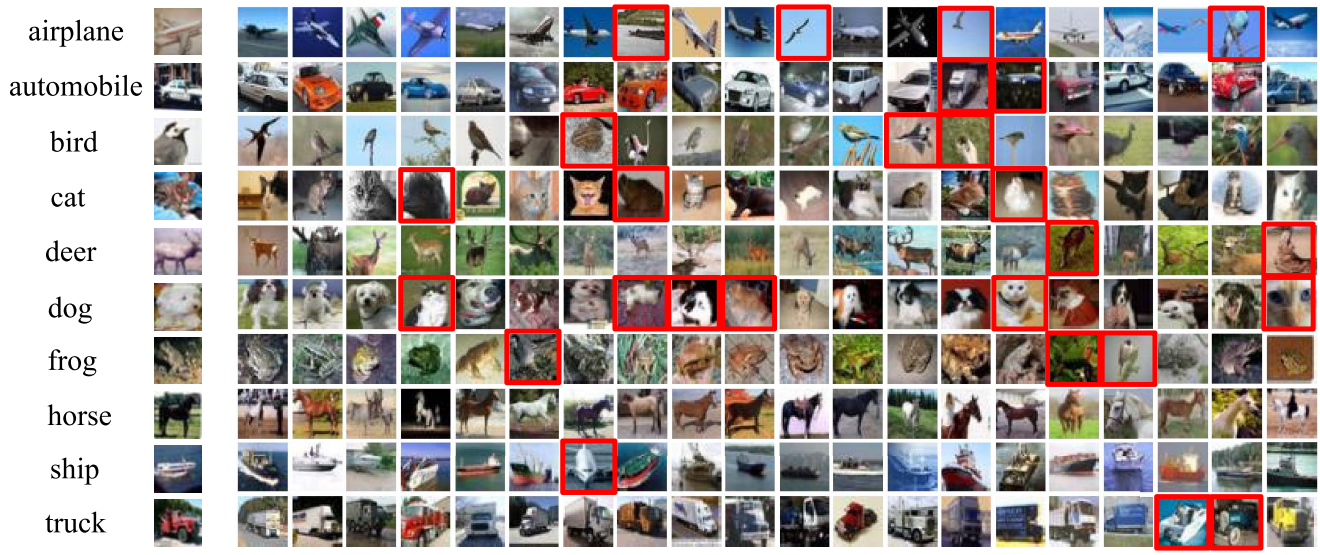


Fig. 13. Top 20 retrieved results from the CIFAR-10 dataset by LSDH with 12 bits. The first column shows the query examples, and other columns show the top-20 retrieval results of LSDH. The red rectangles indicate wrong retrieval results.

6.80% against LiftedStruct, 4.15% against BeyondTriplet and 2.72% against LearningAngular.

The reason behind of getting better results is that the proposed LSDH exploits the potential local structure for hash learning, where we attempt to employ the semantic relation to facilitate the binary codes preserving their local similarity, rather than the similarity between a data pair or the relative similarity within a triplet. Nevertheless, these metric learning methods mainly learn a relative similarity within a triplet, and it is not enough to obtain compact binary codes. The LiftedStruct adopts an effective sampling to construct the triple for obtaining a relative similarity. The BeyondTriplet further reduces the anchor-positive sample pair variation to get a robust relative similarity. The LearningAngular constrains the angle of the negative point for obtaining a relative similarity. In addition, the retrieval task aims to return the top k nearest neighbors given an image, and we think our method is more prone to find the nearest neighbors compared to these metric learning methods.

E. Empirical Analysis

1) Sensitivity: We use the hyper-parameters λ and μ to balance the importance of the proposed quantization loss. In our schema, $\lambda\mu$ is jointly used to balance the importance of Hamming-isometric term.

Fig. 12 shows the effect on the MAP for different λ and μ on the CIFAR-10 and NUS-WIDE datasets. We can see that our model is stable for different λ. In addition, when μ is non-zero, the precision becomes better to some degree, which validates the effectiveness of the proposed Hamming-isometric quantization loss.

2) Visualization: In order to observe intuitively the deep representation, Fig. 14 visualizes the binary representation learned by DHN [70], HashNet [3], LSDHL_1 and LSDH based on the visualization tool t-SNE [42], which is a non-linear dimensionality reduction algorithm for exploring high dimensional data and maps multidimensional data into two or more dimensions for intuitive visual observation. In the Fig. 14, different colors denote different category distribution and a single point denotes a single sample, where we randomly choose 1,000 samples of each category from the CIFAR-10 dataset (10 classes) for visualization.

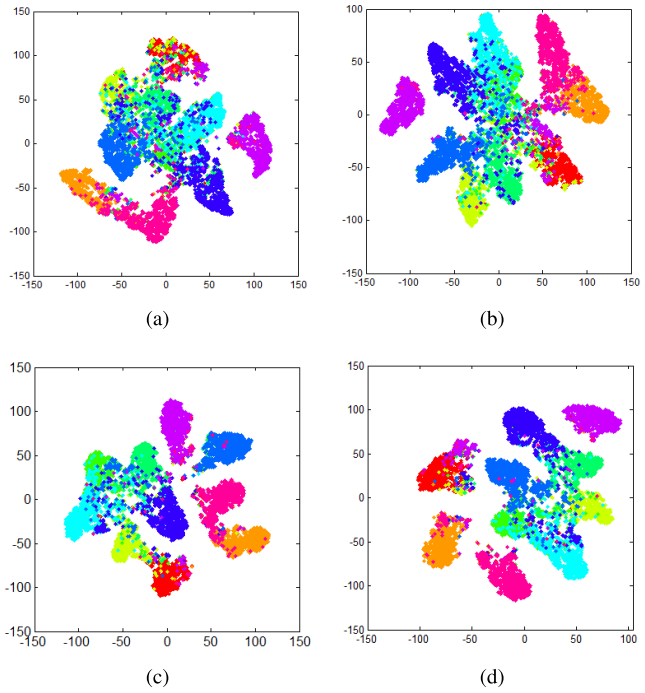


Fig. 14. The t-SNE of binary codes learned by DHN [70], HashNet [3], LSDHL_1 and LSDH on the CIFAR-10 dataset. (a) DHN. (b) HashNet. (c) LSDHL_1. (d) LSDH.

We can observe that the DHN and HashNet fail to show clear boundaries in Fig. 14(a-b), because their models only learn pairwise similarity relationships in feature representation, whereas LSDHL_1 fully exploits the underlying structure of

TABLE V
MAP SCORES OF LSDH AND ITS VARIANTS, LSDH-C, LSDHL_1, LSDHL_1-C ON THREE DATASETS

Method	CIFAR-10				SUN397				NUS-WIDE			
	12 bits	24 bits	32 bits	48 bits	12 bits	24 bits	32 bits	48 bits	12 bits	24 bits	32 bits	48 bits
LSDH-C	**0.753**	**0.762**	0.769	0.765	0.686	**0.746**	**0.749**	**0.743**	**0.861**	0.873	**0.873**	**0.870**
LSDH	0.746	0.752	**0.779**	**0.782**	0.646	0.728	0.730	0.742	0.813	0.832	0.844	0.852
LSDHL_1-C	0.749	0.760	0.767	0.761	**0.690**	0.721	0.734	0.723	0.861	**0.874**	0.872	0.870
LSDHL_1	0.722	0.738	0.766	0.771	0.646	0.717	0.725	0.738	0.800	0.825	0.838	0.846

data to preserve the similarity and learns more discriminative deep representation, as shown in Fig. 14(c). LSDH further takes into consideration Hamming-isometric quantization for enhancing the quality of binary codes, so the boundaries of deep representation are clearer in Fig. 14. In addition, to acquire qualitative visual results, Fig. 13 shows the top 20 image retrieval results on CIFAR-10 with 12-bit binary codes given a query image.

3) Ablation Study: We further investigate a variant of LSDH: LSDH-C and a variant of LSDHL_1: LSDHL_1-C. '-C' indicates that binarization ($sign(x) \rightarrow b$) is not performed in testing and we directly use the binary-like embedding x for similarity retrieval.

We present the results of the MAP in Table V. We can observe that LSDHL_1-C gives superior results compared to LSDHL_1, due to the loss of discriminative power caused by quantizing binary-like embedding into binary codes. LSDH performs better than LSDHL_1 with different lengths codes, because LSDH can maintain the well learned paired similarity after binary quantizing, thus LSDH can obtain more discriminative and compact binary codes. In addition, LSDH-C is superior to LSDHL_1-C in the majority of cases, which further demonstrates that the Hamming-isometric quantization schema is beneficial to image retrieval.

V. CONCLUSION

In this paper, we propose a novel hashing method named local semantic-aware deep hashing with Hamming-isometric quantization to learn compact binary codes. We fully consider the local structure of data distribution to perform hash learning, and a semantic-aware loss is defined on multiple sample pairs to preserve their local similarity. Moreover, we develop a Hamming-isometric quantization loss to maintain the well learned paired similarity after binary quantizing, which is proved to be helpful in improving the quality of binary codes. In addition, we make an extension of our model for coding the multi-label image so that our model is adaptable to multi-label image retrieval. Experimental results have shown the effectiveness of our method compared with eleven state-of-the-art methods on four widely-used image retrieval datasets. In the future, we will further explore the underlying structure of data to perform effective hash learning. We also plan to investigate the quantization technique and figure out its impact on the retrieval performance.

REFERENCES

[1] A. Bellet, A. Habrard, and M. Sebban. (2013). "A survey on metric learning for feature vectors and structured data." [Online]. Available: https://arxiv.org/abs/1306.6709

[2] Y. Cao, M. Long, J. Wang, H. Zhu, and Q. Wen, "Deep quantization network for efficient image retrieval," in *Proc. 13th AAAI Conf. Artif. Intell.*, 2016, pp. 3457–3463.

[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.*, Jun. 2017, pp. 5609–5618.

[4] W. Chen, X. Chen, J. Zhang, and K. Huang, "Beyond triplet loss: A deep quadruplet network for person re-identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 1320–1329.

[5] D. Cheng, Y. Gong, S. Zhou, J. Wang, and N. Zheng, "Person re-identification by multi-channel parts-based CNN with improved triplet loss function," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1335–1344.

[6] S. Chopra, R. Hadsell, and Y. LeCun, "Learning a similarity metric discriminatively, with application to face verification," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, vol. 1, Jun. 2005, pp. 539–546.

[7] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y. Zheng, "NUS-WIDE: A real-world Web image database from National University of Singapore," in *Proc. ACM Int. Comput. Vis. Pattern Recognit.*, 2009, p. 48.

[8] A. Gionis, P. Indyk, and R. Motwani, "Similarity search in high dimensions via hashing," in *Proc. Int. Conf. Very Large Data Bases*, 1999, pp. 518–529.

[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. Gui, T. Liu, Z. Sun, D. Tao, and T. Tan, "Fast supervised discrete hashing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 2, pp. 490–496, Feb. 2018.

[11] Y. Guo, G. Ding, L. Liu, J. Han, and L. Shao, "Learning to hash with optimized anchor embedding for scalable retrieval," *IEEE Trans. Image Process.*, vol. 26, no. 3, pp. 1344–1354, Mar. 2017.

[12] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2006, pp. 1735–1742.

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

[14] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural Comput.*, vol. 18, no. 7, pp. 1527–1554, 2006.

[15] L.-K. Huang, Q. Yang, and W.-S. Zheng, "Online hashing," in *Proc. 23rd Int. Joint Conf. Artif. Intell.*, 2013, pp. 1422–1428.

[16] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[17] Y. Jia *et al.*, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. 22nd ACM Int. Conf. Multimedia*, 2014, pp. 675–678.

[18] G. Jie, T. Liu, Z. Sun, D. Tao, and T. Tan, "Fast supervised discrete hashing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 2, pp. 490–496, Feb. 2018.

[19] A. Krizhevsky and G. E. Hinton, "Learning multiple layers of features from tiny images," Univ. Toronto, Toronto, ON, Canada, Tech. Rep., 2009. [Online]. Available: https://www.cs.toronto.edu/~kriz/cifar.html

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

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

[22] 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.*, Jun. 2015, pp. 3270–3278.

[23] 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.

[24] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

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

[26] 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.

[27] X. Li, G. Lin, C. Shen, A. van den Hengel, and A. Dick, "Learning hash functions using column generation," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 142–150.

[28] 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.

[29] 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.*, Jun. 2016, pp. 1183–1192.

[30] M. Lin, Q. Chen, and S. Yan. (2013). "Network in network." [Online]. Available: https://arxiv.org/abs/1312.4400

[31] H. Liu, R. Ji, J. Wang, and C. Shen, "Ordinal constraint binary coding for approximate nearest neighbor search," *IEEE Trans. Pattern Anal. Mach. Intell.*, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8326558/citations?tabFilter=papers#citations

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

[33] L. Liu, Z. Lin, L. Shao, F. Shen, G. Ding, and J. Han, "Sequential discrete hashing for scalable cross-modality similarity retrieval," *IEEE Trans. Image Process.*, vol. 26, no. 1, pp. 107–118, 2017.

[34] L. Liu, L. Shao, F. Shen, and M. Yu, "Discretely coding semantic rank orders for supervised image hashing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2017, pp. 1425–1434.

[35] Q. Liu, G. Liu, L. Li, X.-T. Yuan, M. Wang, and W. Liu, "Reversed spectral hashing," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 6, pp. 2441–2449, Jun. 2018.

[36] X. Liu, C. Deng, B. Lang, D. Tao, and X. Li, "Query-adaptive reciprocal hash tables for nearest neighbor search," *IEEE Trans. Image Process.*, vol. 25, no. 2, pp. 907–919, Feb. 2016.

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

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

[39] 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.

[40] 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.

[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. van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, Nov. 2008.

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

[44] M. Norouzi, D. J. Fleet, and R. R. Salakhutdinov, "Hamming distance metric learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1061–1069.

[45] H. O. Song, Y. Xiang, S. Jegelka, and S. Savarese, "Deep metric learning via lifted structured feature embedding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 4004–4012.

[46] 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.

[47] T. M. Rassias, "Properties of isometric mappings," *J. Math. Anal. Appl.*, vol. 235, no. 1, pp. 108–121, Jul. 1999.

[48] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[49] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

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

[51] F. Shen, C. Shen, W. Liu, and H. T. Shen, "Supervised discrete hashing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 37–45.

[52] F. Shen, C. Shen, Q. Shi, A. van den Hengel, Z. Tang, and H. T. Shen, "Hashing on nonlinear manifolds," *IEEE Trans. Image Process.*, vol. 24, no. 6, pp. 1839–1851, Jun. 2015.

[53] F. Shen, X. Zhou, Y. Yang, J. Song, H. T. Shen, and D. Tao, "A fast optimization method for general binary code learning," *IEEE Trans. Image Process.*, vol. 25, no. 12, pp. 5610–5621, 2016.

[54] W. Shi, Y. Gong, X. Tao, J. Wang, and N. Zheng, "Improving CNN performance accuracies with min–max objective," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 7, pp. 2872–2885, Jul. 2017.

[55] W. Shi, Y. Gong, X. Tao, and N. Zheng, "Training DCNN by combining max-margin, max-correlation objectives, and correntropy loss for multilabel image classification," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 7, pp. 2896–2908, Jul. 2018.

[56] K. Simonyan and A. Zisserman. (2014). "Very deep convolutional networks for large-scale image recognition." [Online]. Available: https://arxiv.org/abs/1409.1556

[57] J. Tang, Z. Li, M. Wang, and R. Zhao, "Neighborhood discriminant hashing for large-scale image retrieval," *IEEE Trans. Image Process.*, vol. 24, no. 9, pp. 2827–2840, Sep. 2015.

[58] J. Wang, W. Liu, S. Kumar, and S.-F. Chang, "Learning to hash for indexing big data—A survey," *Proc. IEEE*, vol. 104, no. 1, pp. 34–57, Jan. 2016.

[59] 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.

[60] J. Wang, F. Zhou, S. Wen, X. Liu, and Y. Lin, "Deep metric learning with angular loss," in *Proc. IEEE Int. Conf. Comput. Vis.*, Jun. 2017, pp. 2612–2620.

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

[62] 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.*, 2014, pp. 2156–2162.

[63] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba, "SUN database: Large-scale scene recognition from abbey to zoo," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 3485–3492.

[64] 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.

[65] 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.

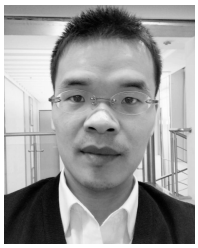
[66] L. Zhang, Y. Zhang, X. Gu, J. Tang, and Q. Tian, "Scalable similarity search with topology preserving hashing," *IEEE Trans. Image Process.*, vol. 23, no. 7, pp. 3025–3039, Jul. 2014.

[67] 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.

[68] S. Zhang, J. Liang, R. He, and Z. Sun, "Code consistent hashing based on information-theoretic criterion," *IEEE Trans. Big Data*, vol. 1, no. 3, pp. 84–94, Sep. 2015.

[69] Z. Zhang, Y. Chen, and V. Saligrama, "Efficient training of very deep neural networks for supervised hashing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1487–1495.

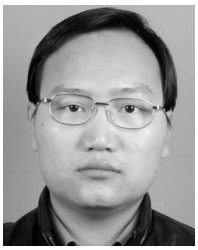
[70] H. Zhu, M. Long, J. Wang, and Y. Cao, "Deep hashing network for efficient similarity retrieval," in *Proc. 13th AAAI Conf. Artif. Intell.*, 2016, pp. 2415–2421.



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

Jian Liang received the B.E. degree in electronic information and technology from Xi'an Jiaotong University, Xi'an, China, in 2013. He is currently pursuing the Ph.D. degree in pattern recognition and intelligent systems with the National Laboratory of Pattern Recognition, Center for Research on Intelligent Perception and Computing, Institute of Automation, Chinese Academy of Sciences. His research interests focus on machine learning, pattern recognition, and computer vision.

Dong Cao received the B.S. degree in computer science and technology from the Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2010, the M.S. degree in computer science and technology from Hohai University, Nanjing, in 2013, and the Ph.D. degree in pattern recognition and intelligent systems from the Chinese Academy of Sciences, Beijing, China, in 2016. His research interests are in computer vision, pattern recognition, deep learning, and in particular, face recognition.



Zhenan Sun received the B.E. degree in industrial automation from the Dalian University of Technology, China, the M.S. degree in system engineering from the Huazhong University of Science and Technology, China, and the Ph.D. degree in pattern recognition and intelligent systems from the Institute of Automation, Chinese Academy of Sciences (CASIA), China, in 1999, 2002, and 2006, respectively. He is currently a Professor with the National Laboratory of Pattern Recognition, Center for Research on Intelligent Perception and Computing, CASIA. His research interests include biometrics, pattern recognition, and computer vision. He is a fellow of the IAPR.