Discriminative Geometric-Structure-Based Deep Hashing for Large-Scale Image Retrieval

Guohua Dong, Xiang Zhang, Xiaobo Shen, Long Lan, Zhigang Luo, and Xiaomin Ying

***Abstract*—Deep hashing reaps the benefits of deep learning and hashing technology, and has become the mainstream of large-scale image retrieval. It generally encodes image into hash code with feature similarity preserving, that is, geometric-structure preservation, and achieves promising retrieval results. In this article, we find that existing geometric-structure preservation manner inadequately ensures feature discrimination, while improving feature discrimination of hash code essentially determines hash learning retrieval performance. This fact principally spurs us to propose a discriminative geometric-structure-based deep hashing method (DGDH), which investigates three novel loss terms based on class centers to induce the so-called *discriminative* geometrical structure. In detail, the margin-aware center loss assembles samples in the same class to the corresponding class centers for intraclass compactness, then a linear classifier based on class center serves to boost interclass separability, and the radius loss further puts different class centers on a hypersphere to tentatively reduce quantization errors. An efficient alternate optimization algorithm with guaranteed desirable convergence is proposed to optimize DGDH. We theoretically analyze the robustness and generalization of the proposed method. The experiments on five popular benchmark datasets demonstrate superior image retrieval performance of the proposed DGDH over several state of the arts.**

***Index Terms*—Deep supervised hashing (DSH), discriminative ability, geometric structure, image retrieval.**

I. Introduction

LARGE-SCALE image retrieval has witnessed great advances in the efficacy of approximate nearest neighbors (ANNs) search methods. This success can be attributed to the merits of hashing technology, for example, little memory space and near real-time search. Importantly, hashing technology maintains the underlying neighborhood relationships between data samples. Traditional nondeep hashing methods [1]–[6] learn hash code based on features that have been extracted in advance, and their performance has a lot to do with those features, which may limit the representation capability of hash code.

The powerful representation capability surged in the era of deep learning has been enjoyed in deep hashing methods [8]–[14], where both feature representation and hashing function are jointly learned in an end-to-end manner. This desirable advantage enables deep supervised hashing (DSH) methods to significantly boost the image retrieval performance. Among them, a wealth of attempts, including deep pairwise supervised hashing (DPSH) [8], deep discrete supervised hashing (DDSH) [9], and improved deep hashing networks (IDHNs) [12], introduce the pairwise similarity in deep networks for learning hash code and have achieved impressive performance. Overall, such deep hashing methods considers the asymmetric similarity learning loss to preserve pairwise similarity, that is, geometric-structure preservation, but this could not induce adequate feature discrimination, which essentially determines the efficacy of hash codes.

To clarify this point, a toy example is conducted on the CIFAR-10 dataset to illustrate the 2-D deep features of samples from four classes before the quantization process into Fig. 1. It is not difficult to find that the features of samples from four distinct classes are highly overlapped in Fig. 1(a). As a result, most of the corresponding binary codes are encoded incorrectly, according to the 2-D coordinates of real-value deep features in Fig. 1(c). This hints that the mere geometric-structure preservation, that is, similarity preservation, might be insufficient. In contrast, incorporating feature discrimination with geometric-structure preservation can induce high-quality binary codes, as shown in Fig. 1(b) and (d). For ease, we call the above insight the *discriminative* geometric-structure preservation.

From the above analysis, three aspects should be emphasized to support our motivation for discriminative geometric-structure preservation: 1) different binary codes in themselves are essentially separable in Hamming space, thus we make it rather necessary to distinguish their real-valued cousins. For instance, two different binary codes "10" and "01" have fixed margin value of $\sqrt{2}$ in Euclidean distance; 2) discriminative features are derived before quantization process.

Manuscript received July 10, 2021; revised January 20, 2022 and April 28, 2022; accepted May 4, 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 62176126, Grant 61906091, and Grant 61806213; in part by the Natural Science Foundation of Jiangsu Province, China, (Youth Fund Project) under Grant BK20190440; and in part by the Fundamental Research Funds for the Central Universities under Grant 30921011210. This article was recommended by Associate Editor L. Rutkowski. *(Corresponding authors: Xiang Zhang; Xiaomin Ying.)*

Guohua Dong and Xiaomin Ying are with the Center for Computational Biology, Beijing Institute of Basic Medical Sciences, Beijing 100850, China (e-mail: dgh1991.learn@gmail.com; yingxmbio@foxmail.com).

Xiang Zhang, Long Lan, and Zhigang Luo are with the Institute for Quantum Information & State Key Laboratory of High Performance Computing, National University of Defense Technology, Changsha 410073, China, and also with the College of Computer, National University of Defense Technology, Changsha 410073, Hunan, China (e-mail: zhangxiang08@nudt.edu.cn; long.lan@nudt.edu.cn; zgluo@nudt.edu.cn).

Xiaobo Shen is with the School of Computer and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: njust.shenxiaobo@gmail.com).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TCYB.2022.3173315.

Digital Object Identifier 10.1109/TCYB.2022.3173315

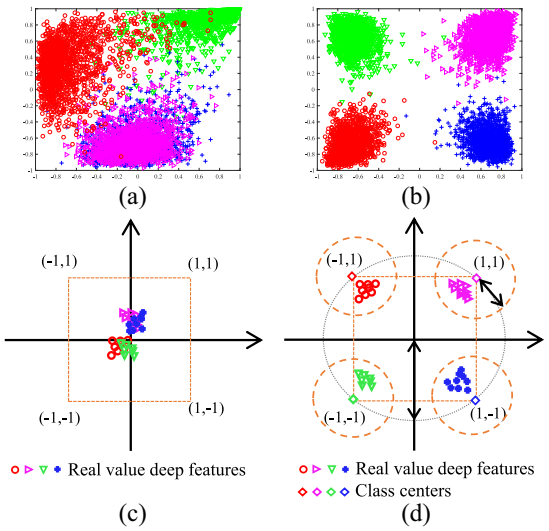


Fig. 1. Toy example on CIFAR-10 [7], where four classes are randomly selected and then 1500 samples selected from each class as the training set, to illustrate the motivation of the discriminative geometric-structure. (a) 2-bit real-value deep features learned by asymmetric similarity learning loss, and (c) spirit of the corresponding loss, that is, similarity preservation. In contrast, the analogue of DGDH is shown, respectively, to emphasize (b) *feature discrimination* but to still implicitly and (d) reduce quantization errors.

Thus, real-valued feature discrimination plays a core role in learning binary codes. As Fig. 1(c) and (d) shows, discriminative features can induce good binary codes, even though the quantization process is not advanced enough. This could weaken the effect of quantization process on the quality of binary codes. Accordingly, the general quantization process available could be straightforwardly used; and 3) the quantization loss in most existing hashing methods serves to make real-value features approach the corresponding binary codes. To some degree, the quantization loss indeed makes real-value features separable, if such real-value features of different samples are very far from each other. But, for the lack of global class information, this still induces incorrect binary codes. Hence, feature discrimination almost effects the entire pipeline of hashing learning.

Inspired by the above analysis, we propose a discriminative geometric-structure-based deep hashing method (DGDH), which consists of asymmetric similarity learning loss and discriminative geometric-structure learning loss. To instantly yield the hash codes of gallery set, we slightly modify the asymmetric similarity learning loss with adding an averaging factor to eliminate the effect of large sample size. On the basis of our modified asymmetric similarity learning loss, three novel loss functions are further explored to jointly preserve the previous mentioned discriminative geometrical structure. In detail, the margin-aware center loss serves to ensure intraclass compactness by making intraclass samples close to their corresponding class center. Besides, the liner classifier based on the class centers is investigated to boost interclass separability. For good quality of binary codes, the radius loss imposes deep features on a hypersphere with a specific radius to reduce the quantization error of their pertinent hash codes. As a result, we yield the learned discriminative geometrical structure that the learned class centers corresponding to the binary codes are distributed on a hypersphere, and meanwhile, intraclass samples surround the class centers with a fixed certain margin. More detailed explanations about our loss are deferred to the sequel text.

In summary, our contributions are threefold.

1) We first discuss the importance of feature discrimination on the entire pipeline of hashing learning, and then propose a DGDH that can learn discriminative real-value deep features to improve retrieval accuracy and efficiency. In DGDH, three lightweighted loss terms are jointly proposed for feature discrimination.
2) We develop an efficient alternate algorithm guaranteed with desirable convergence to optimize hash code. Moreover, we present a solid theoretical validation on the robustness and generalization of the proposed method (supplementary material).
3) Experiments of image retrieval on five popular datasets including the very large Cloth-1M [15] demonstrate the superior performance of the proposed DGDH against the state-of-the-art hashing methods.

The remainder of this article is organized as follows. Section II reviews the most related methods. Section III details the proposed method. Experimental analyses are presented in Section IV. Section V concludes this article.

II. Related Work

This article combines the joint merits of hashing [16]–[18] and deep learning for single-modal retrieval, so we introduce single-modal works of the above two aspects and related multimodal works.

1) Nondeep Hashing Methods: According to previous literatures [5], [6], nondeep hashing methods can be divided into data-independent and data-dependent ones. As early efforts in this field, locality sensitive hashing (LSH) [1] and followup works [19], [20] endeavor to adopt random projection to produce hash code, but the random fashion is data independent and these methods need longer bits to achieve competitive performance to the data dependent.

To induce compact hash code and keep efficiency, data-dependent hashing methods [2]–[5], [21]–[29] have attracted a tremendous amount of attention in machine learning and become the mainstream of large-scale image retrieval. Roughly, they can be grouped into three branches: 1) unsupervised [2]–[4]; 2) supervised [5], [24]–[29]; and 3) semisupervised hashing. Unsupervised hashing mainly focuses on preserving the intrinsic data structure. For instance, [2] and [3] model the manifold structure with graph to preserve the neighborhood relationship between hash codes. Gong *et al.* [4] learned hash code through PCA reduction and rotation matrix learning. In contrast, supervised counterparts [5], [24]–[29] exploit supervised information like the data label or pairwise similarity to produce hash code. For example, [5], [25]–[28] introduce pairwise label to maintain similarity, and [24] and [29] use class label to keep semantic information. Besides absolute supervised or unsupervised hashing, semisupervised hashing [30]–[32], such as MLAGH [31] and SGDH [32],

can simultaneously explore both the supervised information and the underlying data structures in one framework, and thus is a good choice to deal with large scale dataset with a small number of supervised samples. One shortcoming of the above nondeep hashing methods is the limited representation capability of sample features.

2) Deep Hashing Methods: The earliest deep hashing method is semantic hashing [33], which, proposed in 2009, adopts restricted Boltzmann machine to pretrain the network and uses a multilayer autoencoder structure to fine-tune it.

Since 2012, when AlexNet [34] was proposed and showed encouraging performance in object recognition task, convolutional neural networks (CNNs) have drawn extensive attention in many computer vision tasks which a fortiori include image retrieval. For instance, Xia *et al.* [35] proposed the first CNN-based hashing method (CNNH), which consists of two-stage learning processes. Later, the network in network hashing (NINH) [36] and deep semantic ranking-based hashing (DSRH) [37] are proposed to jointly combine two stages in an end-to-end fashion. Particularly, NINH learns deep representation using triplet loss and designs a divide-and-encode module to encode features into binary hashing codes. DSRH endeavors to introduce multilevel similarity of a ranking list into the deep hashing network (DHN). No lunch is free. Constructing the ranking list would entail extra computation costs. Recently, DSH [38] is proposed to design a discriminative loss to pull the network outputs of similar images together as well as to push the dissimilar ones far away. Besides, DHN [39] simultaneously addresses both the similarity preservation and the quality of binary coding. Afterward, deep priority hashing (DPH) [40] further mines more difficult-to-learn samples to boost performance both in similarity preservation and binary quantization. Different from them, DPSH [8] designs a discrete hashing optimization method to train the networks with a quantization loss function for the quality of binary codes, guided by the pairwise labels. The spirit in deep regularized similarity comparison hashing (DRSCH) [41] treats both the ranking information and pairwise labels as the supervised information. Nonetheless, most previous methods use diverse expensive symmetry schemes to construct the ranking or pairwise information. This might displease retrieval performance in efficiency.

To address the above issue, asymmetric deep hashing methods are in efforts to construct the lightweight pairwise similarity for learning binary codes in an asymmetric manner. Among them, many efforts [9], [42], [43] leverage the pairwise information to perform feature learning as well as to learn binary codes. To be specific, DDSH [9] directly constructs two asymmetric datasets to guide deep networks to capture two types of information: the similarity between deep features from one dataset and binary codes from the other dataset, and that of binary codes of different datasets. Recently, asymmetric DSH (ADSH) [43] improves traditional asymmetric graph hashing [44] by using the pairwise similarity of the entire gallery dataset and its random subset.

As for other existing deep hashing methods, they mostly consist of either the similarity preservation or local ranking or quantization error reduction; thus, the effect of feature discrimination over hashing code is still underexplored. Although central similarity quantization (CSQ) [13] pulls all similar data points close to the corresponding hash centers for the compactness of hash codes but ignores the interclass relationship of hashing centers. Besides, it only works in the special scenario where the number of hash bits is a power of 2.

3) Related Multimodal Hashing Works: In contrast to single modal hashing technique, multimodal hashing may be relatively hard as it not only accounts for intermodality relations [45]–[51] but also handles intramodality problem. If putting aside the differences between them, both hashing techniques still have some common issues. For instance, DCMH [45] deliberately devises the single-modal deep hashing module without discrete relaxation, and then extends it to multimodal model by introducing intermodality similarity. SRLCH [49] exploits the projection subspace of the labels to guide each modality for cross-modal semantic consistency. Similar to our motivation, DCH [46] also emphasizes the significance of feature discrimination in cross-modal retrieval. CPAH [50] chiefly leverages adversarial learning to keep modality-common representation consistency, thereby achieving appealing performance, yet it is still built on asymmetric similarity learning. DSMHN [51] adopts 2-D CNN to capture the spatial information for image-text retrieval and 3-D CNN to capture the spatial and temporal information for video-text retrieval.

III. Discriminative Geometric-Structure-Based Deep Hashing

Deep hashing methods have been shown efficient in image retrieval. However, few methods focus on discrimination of hash code, which is helpful to improve retrieval performance. In this section, we propose DGDH, which jointly learns discriminative real-value deep feature and compact hash code through constructing a discriminative geometrical structure. In what follows, we first give some definitions and the network architecture of DGDH. Then, the loss functions of DGDH to train the network and their motivations are illustrated in detail. An algorithm is then proposed to solve DGDH. In the supplementary material, we analyze the robustness and generalization ability of DGDH in theory.

A. Problem Definition and Notations

Given an image $x_i \in R^{d_1 \times d_2 \times 3}$ and the corresponding image label set $y_i \subseteq \{1, \ldots, \bar{c}\}$, where 3 indicates that the image has three channels and both d_1 and d_2 denote the width and height of each image, respectively. $\bar{c}$ is the total number of classes in a dataset. Note that the number of elements in y_i depends on how many classes x_i belongs to. DGDH aims to generate the corresponding discriminative hash code $b_i = \text{sign}(F(x_i; \phi)) \in R^{r \times 1}$, where r is the hash code length and ϕ denotes the network parameters.

B. Network Architecture

The basic leaning framework of DGDH is illustrated in Fig. 2. The proposed method is an end-to-end deep model, and its backbone network structure is based on CNN-F [52],

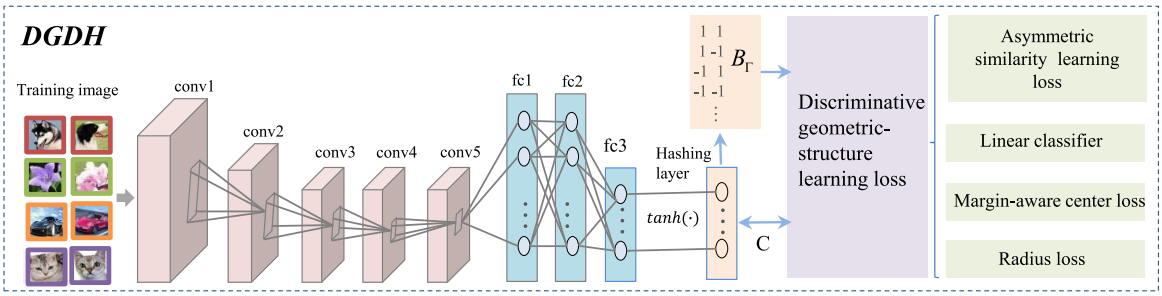


Fig. 2. Overall learning framework of DGDH. It has five convolutional layers, three full connected layers, and a hashing layer. The network parameters are optimized by minimizing the proposed discriminative geometric-structure learning loss and the asymmetric similarity learning loss. See more details in the text.

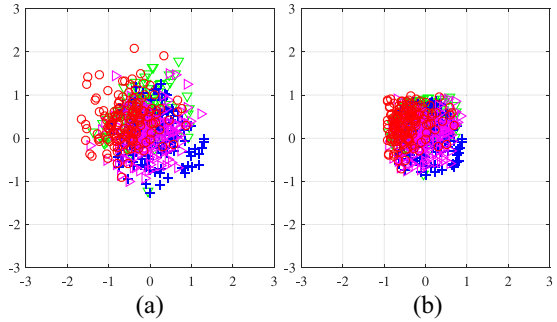


Fig. 3. Distribution of the network outputs before training. Here, we take four classes as an example to illustrate the distribution. (a) Deep features without tanh(·). (b) Real-value deep features with tanh(·) operation.

which has five convolutional layers and three full-connected layers. More details about CNN-F are in [52]. In this article, we replace the last full-connected layer (Fc3) of CNN-F with a r-node full-connected layer to generate real-value deep feature, where r is the hash codes length. After this layer, a hashing layer is added to generate hash code. We minimize two kinds of loss to optimize the network parameters of DGDH: 1) the proposed discriminative geometric-structure learning loss and 2) the asymmetric similarity learning loss. The former, as the main contribution of DGDH, defines an effective discriminative geometric structure, which involves three components: 1) the margin-structure center loss; 2) the radius loss; and 3) the linear classifier-based discriminative loss. The latter is to learn hash code of gallery set directly in this article. Details about these losses are in the next section.

C. Loss Function

The purpose of DGDH is to learn discriminative hash code, which may determine the retrieval performance. To attain this goal, we first introduce asymmetric similarity learning loss to learn hash code of gallery set directly. Then, we design an ingenious discriminative geometric structure, which consists of three parts: 1) the margin-aware center loss; 2) the linear classifier; and 3) the radius loss. In what follows, we detail each loss of DGDH and their aims and interpretations one step by step.

Samples that pass through a pretrained network *without the guidance of our loss* could obtain 2-D deep features as in Fig. 3(a). To make the deep features hash like, we can resort to a sign(·) constraint to the deep feature $F(x_i;\phi)$ of each sample x_i, that is, $\text{sign}(F(x_i;\phi))$. But as we all know, on the basis of deep neural networks, $\text{sign}(F(x_i,\phi))$ with respect to x_i is not differentiable, and thus, it is not easy to optimize the network parameters with the gradient descend algorithm. Following previous works [43], [53], here "sign(·)" is replaced with "tanh(·)," that is, $\tilde{x}_i = \tanh(F(x_i;\phi)) \in R^{r\times 1}$. Then, the distribution of real-value deep features with tanh(·) operation is shown in Fig. 3(b). Note that the real-value deep feature of sample x_i in this manuscript is $\tanh(F(x_i;\phi))$ and $F(x_i;\phi)$ is dubbed as deep feature of x_i.

1) Asymmetric Similarity Learning Loss: In order to avoid passing large-scale samples in gallery set through the network, we also use asymmetric similarity learning loss to connect the real-value deep features of training samples and the hash codes of gallery samples. The formulation is

$$L_A = \min_{\phi,b_j} \frac{1}{n_\Omega n_\Gamma} \sum_{i=1}^{n_\Omega}\sum_{j=1}^{n_\Gamma} \left(\tilde{x}_i^T b_j - rS_{ij}\right)^2. \quad \text{s.t.} \quad b_j \in \{-1,1\}^r \tag{1}$$

where n_Ω and n_Γ are the cardinality of training set and gallery set, respectively. $[1/(n_\Omega n_\Gamma)]$ is added to eliminate the effect of large sample size. Let $\emptyset$ denote the empty set, and the similarity matrix $S \in R^{n_\Omega \times n_\Gamma}$ is defined as

$$S_{ij} = \begin{cases} 1, & y_i \cap y_j \neq \emptyset \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

Actually, the asymmetric similarity learning loss can maintain similarity/dissimilarity among samples. However, it is a local constraint, as shown in Fig. 1(a), with which dragging samples close/apart from each other may not make different classes separable enough. Fortunately, the proposed discriminative geometric structure as below, composed of margin-aware center loss, center-based linear classifier, and radius loss, will endeavor to improve feature discrimination from different aspects.

2) Margin-Aware Center Loss: The ral-value deep features learned by (1) shown in Fig. 1(b) are not separable enough. To address this issue, we introduce center loss [54], which is in efforts to enhance intraclass compactness and has the simple form

$$\min_{\phi} \frac{1}{2} \sum_{i=1}^{n_b} \|\tilde{x}_i - c_k\|_2^2 \tag{3}$$

where n_b is the minibatch size, and $c_k \in R^{r\times 1}$ denotes the class center of the example x_i. The original center loss is

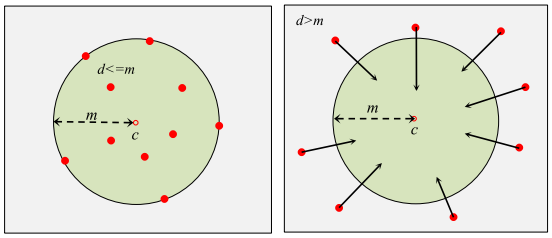


Fig. 4. Illustration of the margin-aware center loss, the distance $d = \|\tilde{x}_i - c_k\|_2$ of samples with their corresponding centers are (a) less than or equal to the margin m and (b) greater than the margin m, respectively. The distance in green space is ignored and that in gray space is punished.

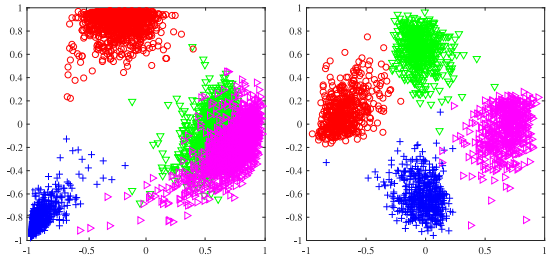


Fig. 5. Illustration of the effect of linear classifier. (a) Real-value deep features after using margin-aware center loss. (b) Real-value deep features after adding linear classifier.

designed for single label classification. In image retrieval, the images often involve multiple labels. Thus, we extend it to the multilabel case and obtain the margin-aware center loss as follows:

$$L_M = \min_{\phi, c_k} \frac{1}{n_\Omega} \sum_{i=1}^{n_\Omega} \sum_{k=1}^{\bar{c}} \omega_{ik} \max(\|\tilde{x}_i - c_k\|_2 - m, 0)^2 \quad (4)$$

where the weight

$$\omega_{ik} = \frac{\delta(k \in y_i)}{\tau + \sum_{k=1}^{\bar{c}} \delta(k \in y_i)\delta(\|\tilde{x}_i - c_k\|_2 > m)} \quad (5)$$

wherein the parameter τ is to prevent the denominator from being zero, and m is the margin of real-value deep features to the corresponding class centers.

The indicator function is

$$\delta(\text{condition}) = \begin{cases} 1, & \text{if condition is true} \\ 0, & \text{otherwise.} \end{cases} \quad (6)$$

The matrix $C = [c_1, \ldots, c_k] \in R^{r \times \bar{c}}$ denotes the class center matrix where each column represents a class center. We ever, respectively, initialize them with the random values, the class means, and k-means centers of real-value deep features and find that such initialization ways almost make no difference on performance; thus, we opt to initialize them randomly.

Interpretation: By minimizing (4), each sample will be close to the corresponding class center within a certain margin. To clearly understand the mechanism of margin-aware center loss, we draw an illustration in Fig. 4. As shown in this figure, the margin-aware center loss punishes the distances between samples and their corresponding class centers when the distances are larger than the margin m, and otherwise, reckons that the samples are close to their class centers. Thus, this loss could drag intraclass samples into a circle with the radius m where the circle center corresponds to the class center. As the tolerable margin is introduced, the interclass relationships are not devastatingly hurt. It paves the safe way to use the interclass discrimination. By comparing the 2-D features of Figs. 1(a) and 5(a), the margin-aware center loss can make the samples from the same class more compact, on the basis of asymmetric similarity learning loss.

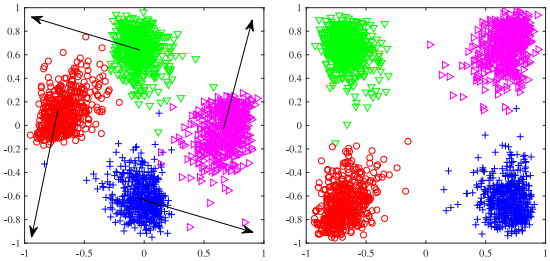
3) Center-Based Linear Classifier: The margin-aware center loss only pulls real value deep features to the corresponding class centers, but the class centers are not necessarily separable. We expect that the class centers should be as discriminative as possible. This induces diverse semantic information. For concise, we learn a linear classifier based on the class centers to boost interclass separability. Then, the linear classifier is

$$L_L = \min_{c_k, W} \frac{1}{\bar{c}} \sum_{k=1}^{\bar{c}} \left\| {c_k}^T W - \bar{Y}_k \right\|_2^2 + \eta \|W\|_F^2 \quad (7)$$

where $W \in R^{r \times \bar{c}}$ is the transformation matrix to project the class centers to the label space. The last term $\|W\|_F^2$ is to avoid trivial solution of W, and η is the corresponding penalty parameter. $\bar{Y}_k \in R^{1 \times \bar{c}}$ is a one-hot label vector, that is, the kth column of the label matrix $\bar{Y}$. This label matrix is class-agnostic and purely pushes different centers far away from each other with the help of the projection matrix W. In other words, (7) always behaves identically as long as the label matrix has different columns of one-hot vectors; thus, it is a permutation of the identity matrix. In general, for simplicity, we directly treat the identity matrix as the label matrix, that is, the kth entry of $\bar{Y}_k$ is "1," while the others are "0."

Interpretation: We also draw a diagram to explain the effect of linear classifier. Benefitting from margin-aware center loss, the center-based linear classifier will push away different centers of classes, while the samples within the same class will be dragging toward the class center. As a result, the samples from different classes are also far from each other. As shown in Fig. 5(a) and (b), the linear classifier makes the samples of different classes distinguishable from each other.

4) Radius Loss: So far, we have obtained the real-value deep features with sufficient separability. Is that enough for DGDH? The answer is no. We need to further reduce the quantization error between the real-value deep features and the binary hash codes. How to make it with the loss? As each entry of binary hash codes is either 1 or -1, the 2-norm of the r-bit binary hash code vectors is $\sqrt{r}$. Since real-value deep features are the surrogate of binary hash codes, the norm of real-value deep features should be also $\sqrt{r}$. Note that through the $\tanh(\cdot)$ function, the norm of real-value deep features has been already restricted within the intersection between a hypersphere and the square region, which is bounded by the corresponding binary codes [Fig. 1(d)]. Thus, their norms near $\sqrt{r}$ means pushing them to the binary codes. This could be beneficial for reducing the quantization error. This spirit can



(a) (b)

Fig. 6. Illustration of the effect of radius loss. (a) Real-value deep features before introducing radius loss. (b) Real-value deep features after introducing radius loss. The black arrows indicate radius loss pulls class centers to binary hash codes.

be formulated as

$$\min_{\phi} \frac{1}{n_\Omega} \sum_{i=1}^{n_\Omega} \left(\|\tilde{x}_i\|_2 - \sqrt{r}\right)^2. \tag{8}$$

As in real datasets, the amount of training samples n_Ω is far larger than the amount of class centers k; thus, in contrast to constraining amounts of samples in (8), constraining the class centers can greatly reduce the computation overhead. In addition, as in (4), we have constrained the samples to be close to the corresponding class center in the margin-aware center loss. Then, making the class centers locate on the hypersphere with the radius of $\sqrt{r}$ could also drag the corresponding samples close to this hypersphere. Taking these into account, we project the class centers instead of real-value deep features of samples to the hypersphere with the radius of $\sqrt{r}$. Thus, the insight can be formulated as

$$L_R = \min_{c_k} \frac{1}{\bar{c}} \sum_{k=1}^{\bar{c}} \left(\|c_k\|_2 - \sqrt{r}\right)^2. \tag{9}$$

Interpretation: In Fig. 6, since the activation function works there, the real value of deep features should range from -1 to 1. Thus, the entire samples are not beyond the above range. According to (9), the class centers will be dragged toward the hypersphere with the radius $\sqrt{2}$. Due to the above facts, the real-value features will approach to the binary codes $+1/-1$. Fig. 6(b) verifies the spirit of radius loss, that is, the radius loss indeed reduces the binary quantization error by dragging the class centers to the vertex of the hypersphere.

5) Overall Loss of DGDH: By combining (1), (4), (7), and (9) together, we obtain the overall loss of DGDH as follows:

$$L = \min_{\phi, C, W} L_A + \alpha L_M + \beta L_R + \gamma L_L \tag{10}$$

where α, β, and γ are, respectively, the balance parameters of each loss in (10). Since the proposed discriminative geometric-structure learning loss is based on small sample size of class centers, they are lightweighted and have low complexity. This point will be further verified in experiments.

D. Optimization

When training DGDH, we need to determine all the parameters in (10). In this section, we propose an efficient alternate algorithm to solve (10). More specifically, we update one variable with the others fixed at each round iteration.

Update ϕ With c_k, b_j, and W Fixed: When c_k, b_j, and W are fixed, we derive the following gradient of (10) with respect to the network weights ϕ through the chain rule:

$$\frac{\partial L}{\partial \phi} = \frac{\partial L}{\tilde{x}_i} \frac{\partial \tilde{x}_i}{\partial F(x_i;\phi)} \frac{\partial F(x_i;\phi)}{\phi}. \tag{11}$$

The first two terms of (11) are computed by

$$\begin{aligned} \frac{\partial L}{\tilde{x}_i} \frac{\partial \tilde{x}_i}{\partial F(x_i;\phi)} = \Bigg\{ & \frac{2}{n_\Gamma} \sum_{j \in \Gamma} \left(\tilde{x}_i^T b_j - r S_{ij}\right) b_j{}^T \\ & + 2\alpha \sum_{k=1}^{\bar{c}} \varpi_{ik} (\tilde{x}_i - c_k) \delta(\|\tilde{x}_i - c_k\|_2 > m) \\ & \times \left(1 - \frac{m}{\|\tilde{x}_i - c_k\|_2}\right) \Bigg\} \\ & \odot \left(1 - \tilde{x}_i^2\right) \end{aligned} \tag{12}$$

where $\odot$ denotes the Hadamard product. Based on the gradient of (12), the backpropagation (BP) algorithm is used to update ϕ.

Update c_k With ϕ, b_j, and W Fixed: When ϕ, b_j, and W are fixed, (10) can be recast as

$$\begin{aligned} \min_{c_k} \ & \frac{\alpha}{n_\Omega} \sum_{i=1}^{n_\Omega} \sum_{k=1}^{\bar{c}} \varpi_{ik} \max(\|\tilde{x}_i - c_k\|_2 - m, 0)^2 \\ & + \frac{\beta}{\bar{c}} \sum_{k=1}^{\bar{c}} \left(\|c_k\|_2 - \sqrt{r}\right)^2 + \frac{\gamma}{\bar{c}} \sum_{k=1}^{\bar{c}} \left\|c_k{}^T W - \bar{Y}_k\right\|_2^2. \end{aligned} \tag{13}$$

To solve (13), we compute the gradient of (10) with respect to c_k

$$\begin{aligned} \frac{\partial L}{\partial c_k} = & \frac{2\alpha}{n_\Omega} \frac{\sum_{i=1}^{n_\Omega} \delta(k \in y_i) \cdot \delta(\|\tilde{x}_i - c_k\|_2 > m) \left(1 - \frac{m}{\|\tilde{x}_i - c_k\|_2}\right)(c_k - \tilde{x}_i)}{\tau + \sum_{i=1}^{n_\Omega} \delta(k \in y_i) \delta(\|\tilde{x}_i - c_k\|_2 > m)} \\ & + \frac{2\beta}{\bar{c}} \left(1 - \frac{\sqrt{r}}{\|c_k\|_2}\right) c_k + \frac{2\gamma}{\bar{c}} \left(W W^T c_k - W \bar{Y}_k^T\right). \end{aligned} \tag{14}$$

Then, c_k can be updated by

$$c_k^{t+1} = c_k^t - \mu \frac{\partial L}{\partial c_k^t} \tag{15}$$

where c_k^t denotes c_k in the tth iteration, and μ is the learning rate of updating c_k.

Update b_j With ϕ, c_k, and W Fixed: Let $B_\Gamma \in R^{n_\Gamma \times r}$ denote the gallery hashing matrix. When ϕ, c_k, and W are fixed, we recast (10) as

$$\begin{aligned} \min_{B_\Gamma} \ & \left\|\tilde{X}_\Omega^T B_\Gamma{}^T - rS\right\|_F^2 \\ & = \left\|B_\Gamma \tilde{X}_\Omega\right\|_F^2 - 2rtr\left(B_\Gamma \tilde{X}_\Omega S\right) + \text{const} \\ \text{s.t.} \ & B_\Gamma \{-1, 1\}^{n_\Gamma \times r} \end{aligned} \tag{16}$$

where $\tilde{X}_\Omega = \tanh(F(X_\Omega;\phi)) \in R^{r \times n_\Omega}$ denotes the real-value deep features for training set. Then, we can solve B_Γ to update b_j, a transposition of a row in B_Γ. According to [43], we solve B_Γ in a bit-by-bit manner. That is, each bit in turn is

Algorithm 1: DGDH

Input: Training set X_Ω, class labels Y, parameters α, β, γ, η, τ, pre-fined network CNN-F. Network iteration N_{net}, center updating iteration N_{center} and algorithm iteration N_{iter}. The network learning rate l, the class center updating rate μ and the batch size n_b.

Output: Network parameters ϕ and gallery hashing matrix B_Γ.

Initialize network parameters ϕ, randomly initialize transformation matrix W, class center matrix C and gallery hashing matrix B_Γ, compute similarity matrix S and class label matrix $\bar{Y}$ based on Y.
for i = 1:N_{iter}
  for j = 1:N_{net}
    Update network parameters ϕ via Eqs. (11)-(12).
  end
  for j = 1 : N_{center}
    Update class centers via Eqs. (14)-(15).
  end
  Update transformation matrix W via Eq. (20).
  Update gallery hashing matrix B_Γ via Eq. (18).
end

updated. Let b_q, $\tilde{x}_q$, and r_q denote the qth column of B_Γ, $\tilde{X}_\Omega^T$, and R, respectively, wherein $R = rS^T\tilde{X}_\Omega^T$. Besides, $\hat{B}_\Gamma$, $\hat{\tilde{X}}_\Omega^T$, and $\hat{R}$ indicate to remove the qth column of B_Γ, $\tilde{X}_\Omega^T$, and R, respectively. To solve b_q, (16) becomes

$$\min_{b_q} tr\Big(b_q\Big(2\tilde{x}_q^T\hat{\tilde{X}}_\Omega^T\hat{B}_\Gamma^T - 2r_q^T\Big)\Big), \text{s.t. } b_q \in \{-1, 1\}^r. \quad (17)$$

The solution to (17) is

$$b_q = -\text{sign}\Big(2\hat{B}_\Gamma\hat{\tilde{X}}_\Omega\tilde{x}_q - 2r_q\Big). \quad (18)$$

Update W With b_j, ϕ, and c_k Fixed: When b_j, ϕ, and c_k are fixed, we optimize the following formula:

$$\min_W \left\|C^TW - \bar{Y}\right\|_F^2 + \eta\|W\|_F^2 = tr\big(C^TWW^TC - 2C^TW\bar{Y}^T + \eta W^TW\big). \quad (19)$$

By setting the gradient of (19) about W to be zero, we obtain the solution of W

$$W = \big(CC^T + \eta I\big)^{-1}C\bar{Y} \quad (20)$$

where $\bar{Y} \in R^{\bar{c}\times\bar{c}}$ denotes the label matrix of class centers C, wherein each row represents the label vector corresponding to each individual class center.

The entire training procedure of DGDH is listed in Algorithm 1. We theoretically analyze the robustness and generalization of DGDH. For space limitation, the theoretical analysis is left in the supplementary material (Section I).

IV. Experiments

This section mainly verifies the efficacy of DGDH by conducting image retrieval experiments on five popular datasets, including CIFAR-10 [7], MS-COCO [55], Flickr25K [56], NUS-WIDE [57], and large-scale dataset Clothing 1M (Cloth-1M) [15]. Extensive analyses of efficiency, ablation study, and feature visualization further show the effectiveness of DGDH.

For CIFAR-10, MS-COCO, and NUS-WIDE datasets, we use the same data split as DMUH [58]. For Flickr25K and Cloth-1M, the protocols are as follows.

TABLE I
Dataset Statistic

Datasets	#Samples	# For experiments	#Training	#Query	#Classes
CIFAR-10	60,000	60,000	5,000	1,000	10
MS-COCO	82,783+40,504	87,081	10,000	5,000	91
Flickr25K	25,000	24,581	5,000	1,140	38
NUS-WIDE	269,648	195,834	10,500	2,100	21
Cloth-1M	1,034,912	1,034,912	7,000	1,400	14

1) The Flickr25K dataset [56] is a subset of one million Flickr1M dataset [59]. It consists of 25 000 multilabel images from 38 categories. We also remove the images with no labels in this dataset and keep the rest 24 581 images for experiment. Then, we randomly select 30 images from each category as query set and randomly select 5000 images as the training set. The rest images form the gallery set for image retrieval.
2) The Cloth-1M dataset [15] contains 1 034 912 clothing images from the Internet. Most of the images are annotated into 14 categories according to their surrounding texts; thus, 1 000 000 images have noisy labels. Another 34 912 images are manually annotated with clean labels. Based on the images with clean labels, we randomly select 100 and 500 images each category as query and training sets, respectively. The rest part of the manually annotated images and the images with noisy labels constitute the gallery set.

The dataset statistic is shown in Table I. For all the datasets, the pairwise similarity construction is based on the class labels, where the similarity is 1, if a pair of examples shares at least one label, and otherwise 0.

A. Settings

We compare DGDH with two kinds of well-established baselines.

Traditional nondeep hashing methods involve three representative methods, which correspond to LSH [1], ITQ [4], and SDH [5]. For such methods, we extract the second Fc layer features with 4096-dimension of the pretrained CNN-F network as the input features, and then run the source codes provided by the authors to conduct experimental comparison.

Deep supervised hashing methods include some state-of-the art methods, such as DPSH [8], DDSH [9], IDHN [12], CSQ [13], OrthoH [60], and DMUH [58]. All of the compared deep methods are in frame of CNN-F. The network input images are consistently resized as a 224×224×3-pixel tensor, that is, center crop and resize each channel of image to 224×224. For single-channel image, copy it three times to derive the input with the size of 224×224×3. For fairness, all the compared methods use identical training and query set to train and test deep models, respectively. All deep methods are evaluated on a TITAN-RTX GPU, the training batch size of DGDH on CIFAR-10, MS-COCO, Flickr25K, and NUS-WIDE dataset is 64 and 128 on Cloth-1M dataset, and the encoding batch size is 128 on all datasets. For other deep methods, the parameters are consistent with their published paper or released source codes. For the datasets that has not appeared in their original papers, we tune as the better

TABLE II
MAP RESULTS (%) ON CIFAR-10, MS-COCO, FLICKR25K, AND CLOTH-1M DATASETS, AND THE MAP@5000 RESULTS ON NUS-WIDE DATASET FOR 16, 32, AND 64 BITS IMAGE RETRIEVAL, RESPECTIVELY

Methods	Dataset														
	CIFAR-10			MS-COCO			Flickr25K			NUS-WIDE			Cloth-1M		
	16-bit	32-bit	64-bit	16-bit	32-bit	64-bit	16-bit	32-bit	64-bit	16-bit	32-bit	64-bit	16-bit	32-bit	64-bit
LSH	13.38	16.47	17.60	51.09	52.07	52.66	61.96	63.37	66.23	51.98	57.15	61.01	8.63	9.23	9.42
ITQ	26.27	25.45	26.51	64.10	63.53	63.98	67.98	69.39	69.65	69.60	71.93	74.01	9.45	10.34	10.67
SDH	59.06	62.49	64.90	70.69	71.60	72.13	75.52	79.58	80.05	77.84	79.41	80.78	20.88	24.63	26.86
DPSH	71.50	82.02	83.47	70.80	76.16	**80.30**	78.89	83.44	84.00	79.45	83.22	85.35	36.80	36.96	36.76
DDSH	82.00	84.14	81.41	67.91	68.93	66.90	80.70	80.06	80.28	74.59	78.53	78.16	27.23	33.79	33.39
IDHN	72.49	72.17	74.60	73.13	73.57	74.10	80.74	80.79	81.35	79.25	80.62	81.05	28.99	32.71	34.21
CSQ	74.74	76.08	74.28	62.37	67.76	69.57	75.92	80.93	81.42	79.70	82.55	82.51	35.15	35.98	35.66
OrthoH	66.94	74.97	76.76	64.36	69.62	73.15	77.35	77.50	80.97	77.37	81.56	83.47	21.01	24.98	28.52
DMUH	77.90	82.20	83.00	**76.50**	78.50	79.20	77.03	77.39	78.66	80.30	82.50	83.40	22.40	33.06	35.29
DGDH$_{onlyTrain}$	**82.06**	**84.95**	**85.14**	74.79	**78.62**	79.96	**82.31**	**83.90**	**84.17**	**81.25**	**83.69**	**85.56**	**39.26**	**41.09**	**43.87**
DGDH	**90.14**	**91.11**	**91.28**	**79.02**	**81.31**	**82.83**	**88.62**	**89.96**	**88.40**	**86.67**	**88.32**	**89.17**	**53.94**	**61.03**	**63.28**

parameters for them as possible in our best. For DPSH, DDSH, IDHN, CSQ, and OrthoH, we run the released source code by the authors. For DMUH, we reproduce its code and obtain similar results with the published paper [58]. Since the data division of CIFAR-10, MS-COCO, and NUS-WIDE datasets in this article is the same with DMUH, we directly cite the mean average precision (MAP) results of DMUH on these three datasets.

For DGDH, we fix the involved parameters: $\alpha = 100$, $\beta = 100$, $\gamma = 100$, $\eta = 10$, and $\tau = 0.001$. In addition, we set the iteration numbers of updating the network to $N_{\text{net}} = 4$, the iteration numbers of updating centers to $N_{center} = 1$, the algorithmic iteration number to $N_{\text{iter}} = 10$, and the weight decay to 5×10^{-4}, respectively. For all datasets, we tune the class center updating rate μ ranging from $[10^{-5}, 10^{-6}]$ and the network learning rate l from $[10^{-4.5}, 10^{-6}]$, respectively.

To validate DGDH and the compared methods, we use the evaluation metrics: Hamming ranking in MAP, precision–recall performance, and precision of top 50 retrieved images (precision@top50).

B. Results

Hamming Ranking Performance: We report the MAP results for CIFAR-10, MS-COCO, Flickr25K, and Cloth-1M datasets, and the MAP@5000 results for NUS-WIDE dataset in Table II. By comparing ITQ and SDH with LSH, we can draw the conclusion that data-dependent methods gain better performance than data-independent methods; this can attribute to the advantage of information inherent in the data. Benefiting from supervisory signal, supervised deep hashing methods and SDH outperform unsupervised counterparts. In most cases, deep hashing methods can outperform traditional nondeep methods, as deep methods enjoy powerful feature representation. It is not difficult to find that DGDH achieves more performance gains as compared to other baselines. In particular, DGDH, respectively, exceeds DDSH by 6.97% and SDH by 28.62% for 32-bit results on CIFAR-10. For the MS-COCO dataset, taking 64 bits as example, DGDH achieves the increase of 2.53%–15.93% against deep hashing counterparts, and exceeds the nondeep methods by 10.70%–30.17%. On the Flickr25K dataset, taking 16-bit image retrieval as example, DGDH exceeds deep counterparts by 7.88%–12.70% and surpasses the traditional nondeep methods by 13.10%–26.56%, respectively. DGDH achieves the sound performance again on NUS-WIDE and Cloth-1M datasets. The main reason for sound performance gains might be that DGDH can learn more discriminative features through the discriminative geometric-structure learning loss, thereby inducing compact and well-performed hash code. Besides, we find that DGDH can achieve better performance with the rise of the number of bits on CIFAR-10, MS-COCO, NUS-WIDE, and cloth-1M datasets. This is because more bits deliver more information under a certain length range. For instance, on the Flickr25K dataset, the above trends are kept on 16–32 bits, but the performance declines with 64 bit; this is also reasonable as Flickr25K contains relatively less samples and the longer bits may induce information redundancy, thus decreasing the performance. To remove the effect of the gallery set on performance, we train DGDH with another setting where the gallery set is discarded while the mere training set is left (we call this model DGDH$_{\text{onlyTrain}}$). According to Table II, DGDH$_{\text{onlyTrain}}$ still achieves powerful retrieval performance.

The precision–recall curves on CIFAR-10, MS-COCO, Flickr25K, NUS-WIDE, and Cloth-1M datasets are shown in Fig. 7. We can easily find that DGDH shows the best retrieval performance compared with all the baselines. On CIFAR-10, DGDH remains high precision even if the recall increases greatly. This may be because the CIFAR-10 dataset is relatively easy to retrieve for DGDH, so that the retrieval results are always correct during retrieval. By comparison, supervised methods show better performance than unsupervised ones. Besides, deep hashing methods mostly outperform nondeep methods. For another single-label dataset Cloth-1M, DGDH gains better performance than other methods by a large margin. This shows the superiority of DGDH in retrieving large-scale dataset when the training sets are relatively small. For multilabel datasets, such as MS-COCO, Flickr25K, and NUS-WIDE, DGDH also achieves the decent performance against the compared baselines. These results show the effectiveness of DGDH on both single-label and multilabel datasets.

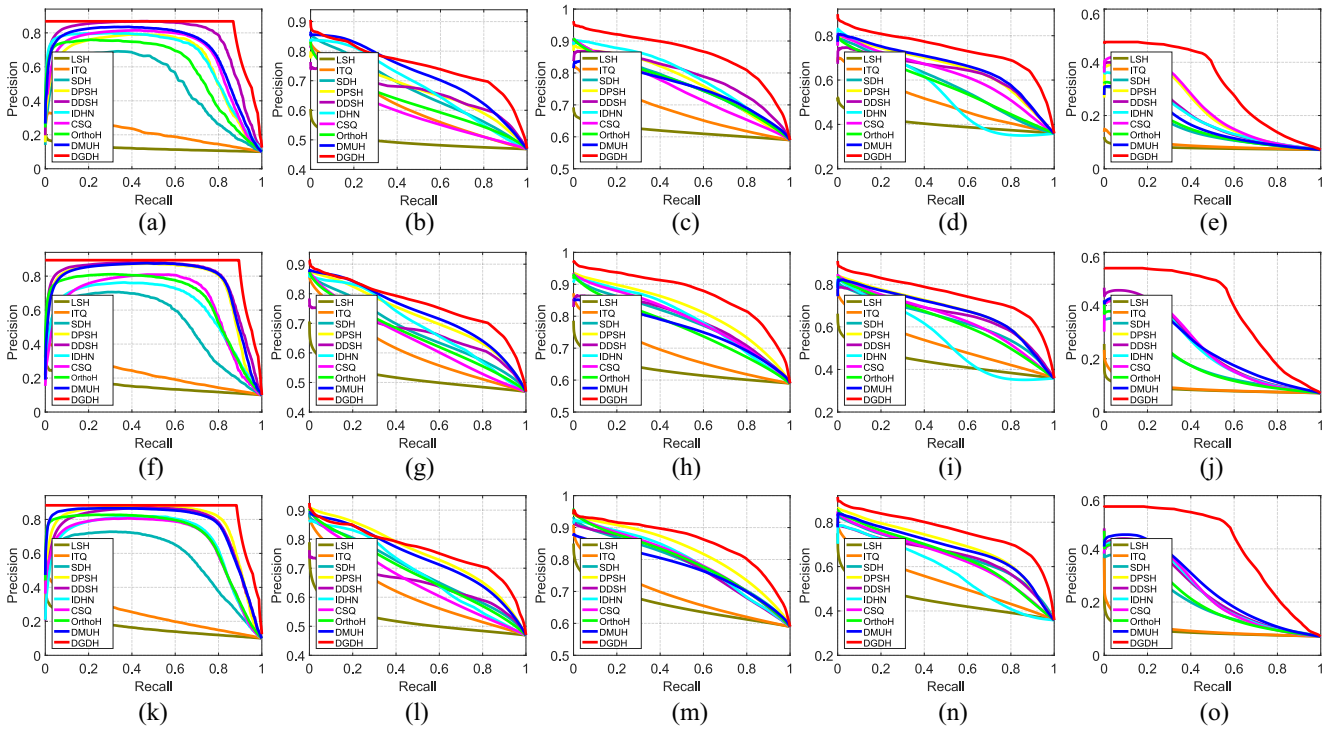


Fig. 7. Hamming ranking performance (in precision–recall curve) for 16 bits on (a) CIFAR-10, (b) MS-COCO, (c) Flickr25K, (d) NUS-WIDE, and (e) Cloth-1M datasets, 32 bits on (f) CIFAR-10, (g) MS-COCO, (h) Flickr25K, (i) NUS-WIDE, and (j) Cloth-1M datasets, and 64 bits on (k) CIFAR-10, (l) MS-COCO, (m) Flickr25K, (n) NUS-WIDE, and (o) Cloth-1M datasets, respectively.

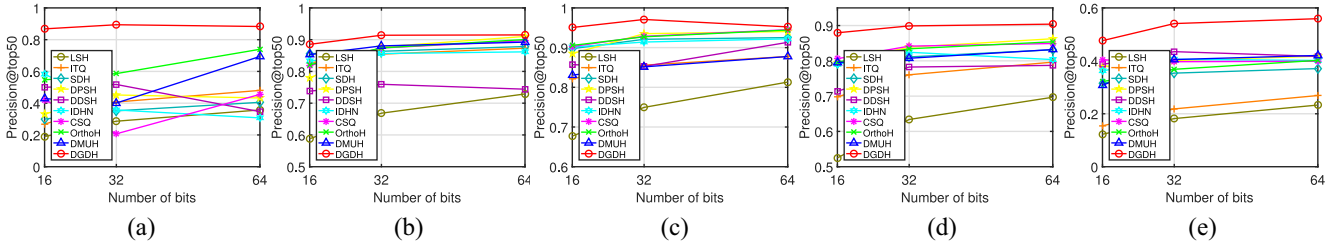


Fig. 8. Hamming ranking performance (in precision@top50) on (a) CIFAR-10, (b) MS-COCO, (c) Flickr25K, (d) NUS-WIDE, and (e) Cloth-1M datasets, respectively.

Retrieving top-ranked images is widely used in real life, so we also report the precision of top 50 retrieved images versus different hashing bits on five datasets in Fig. 8. For this evaluation, DGDH also yields the salient performance as compared to the well-behaved baselines. This implies the promising potential of the discriminative geometric-structure learning loss again.

Time Analysis: To evaluate the efficiency of DGDH, we test the training and encoding time of DGDH compared with other deep hashing methods. The comparison results on CIFAR-10, MS-COCO, Flickr25K, and NUS-WIDE datasets are shown in Table III, and results of Cloth-1M dataset are shown in Table IV. From these two tables, DGDH spends less training and encoding time than other compared methods on CIFAR-10, MS-COCO, and Flickr25K datasets. For NUS-WIDE and Cloth-1M datasets, DGDH spends competitive training time comparing with other methods, and uses less encoding time than the compared methods. The time comparison results verify the efficiency of DGDH.

TABLE III
TRAINING TIME AND ENCODING TIME (MIN) ON CIFAR-10, MS-COCO, FLICKR25K, AND NUS-WIDE DATASETS FOR 32-BIT IMAGE RETRIEVAL, RESPECTIVELY (GPU: TITAN-RTX)

Methods	CIFAR-10		MS-COCO		Flickr25K		NUS-WIDE	
	Train	Encode	Train	Encode	Train	Encode	Train	Encode
DPSH	22.14	1.01	46.92	3.39	21.60	0.54	**47.96**	5.62
DDSH	24.16	1.08	50.29	3.06	24.42	0.50	53.63	5.04
IDHN	34.05	0.94	54.70	2.87	33.97	0.91	56.04	3.29
CSQ	22.69	0.96	40.21	1.73	26.02	0.79	48.84	6.10
OrthoH	20.83	1.08	64.17	5.15	32.83	1.70	52.17	2.67
DMUH	24.50	2.17	62.50	1.75	26.67	0.75	65.83	5.32
DGDH	**9.60**	**0.19**	**36.81**	**1.15**	**11.72**	**0.09**	55.67	**1.09**

Impact of Network Backbone: Extra experiments are evaluated to analyze the impact of network backbone on DGDH. The comparison results of DGDH with DPSH, IDHN, CSQ, OrthoH, and DMUH are shown in Table IV. We observe that DGDH has better MAP performance than the compared

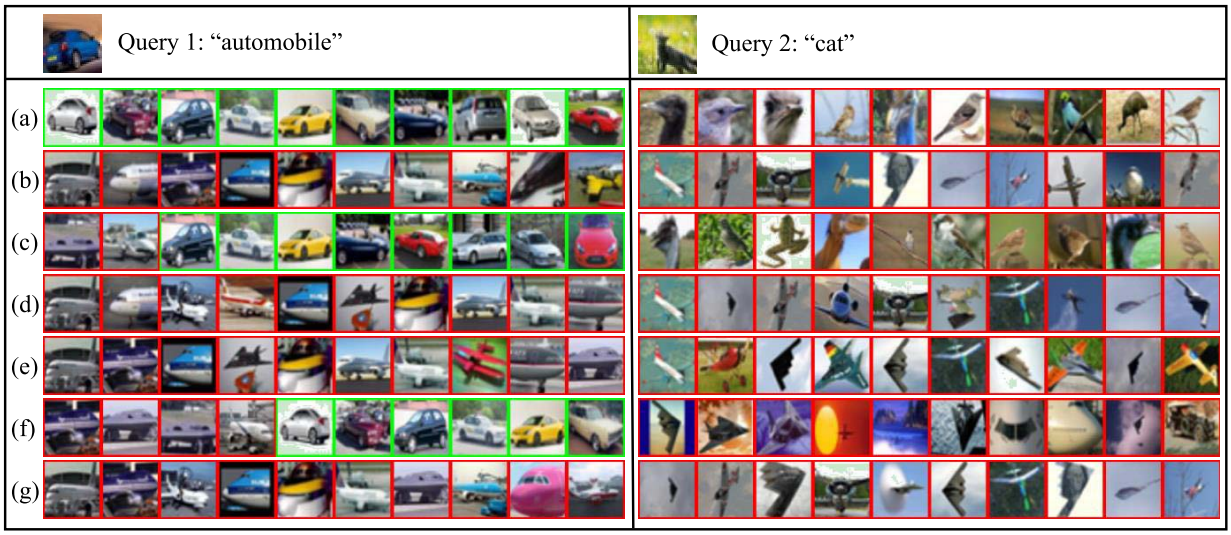


Fig. 9. Visualization of top-10 retrieving results for 64-bit on CIFAR-10 dataset. (a) DGDH. (b) DMUH. (c) OrthoH. (d) CSQ. (e) IDHN. (f) DDSH. (g) DPSH. The green rectangles indicate correct retrieved images and the red ones are inaccurate.

TABLE IV
MAP RESULTS VERSUS DIFFERENT NETWORKS ON LARGE-SCALE CLOTH-1M DATASET FOR 16, 32, AND 64 BITS IMAGE RETRIEVAL, AND THE TRAINING AND ENCODING TIME (MIN) ON CLOTH-1M DATASET FOR 32-BIT IMAGE RETRIEVAL, RESPECTIVELY

Methods	Cloth-1M			Time-32 bits	
	16-bit	32-bit	64-bit	Train	Encode
DPSH-CNNF	36.80	36.96	36.76	21.67	19.33
IDHN-CNNF	28.99	32.71	34.21	23.67	20.85
CSQ-CNNF	35.15	35.98	35.66	20.59	19.22
OrthoH-CNNF	21.01	24.98	28.52	24.17	24.75
DMUH-CNNF	22.40	33.06	35.29	**19.67**	19.60
DGDH-CNNF	**53.94**	**61.03**	**63.28**	20.14	**0.12**
DPSH-ResNet50	39.05	46.13	44.86	50.00	31.02
IDHN-ResNet50	38.22	37.45	36.58	55.00	29.12
CSQ-ResNet50	44.77	47.22	45.47	48.00	28.50
OrthoH-ResNet50	40.68	43.78	45.99	52.00	27.63
DMUH-ResNet50	45.78	45.00	44.74	55.89	30.25
DGDH-ResNet50	**66.55**	**70.03**	**69.85**	**31.98**	**0.38**

TABLE V
MAPS OF ABLATION STUDY ON (a) CIFAR-10, (b) MS-COCO, (c) FLICKR25K, (d) NUS-WIDE, AND (e) CLOTH-1M DATASETS

L_A	L_M	L_L	L_R	(a)	(b)	(c)	(d)	(e)
✓				89.80	80.11	89.16	87.59	59.45
✓	✓			90.37	80.78	89.70	87.96	60.30
✓	✓	✓		90.65	80.92	89.79	88.08	60.96
✓	✓	✓	✓	**91.11**	**81.31**	**89.96**	**88.32**	**61.03**

methods, regardless of using CNN-F or ResNet50. Moreover, when the network goes deeper, most methods achieve obvious performance improvement.

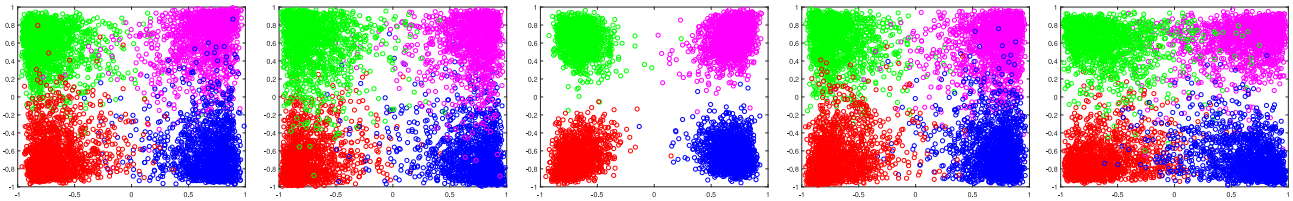
Ablation Study: We conduct ablation study to investigate the effects of different components in DGDH, that is, asymmetric learning loss L_A, margin-aware center loss L_M, center-based linear classifier L_L, and radius loss L_R. Table V shows that when we add L_M, L_L, and L_R to DGDH, the retrieval performance increases, thus verifying the effectiveness of geometric-structure learning loss again.

Visualization of Image Retrieval Results: To intuitively see retrieval performance, we conduct image retrieval for 64-bit on the CIFAR-10 dataset and visualize the retrieval images ranked by seven deep supervised methods in Fig. 9. The retrieval results of query 1 and query 2 are the success and failure cases, respectively. For query 1, the top-10 images of DGDH are all correct. In the second retrieval exemplar, it is difficult for query 2 to be discerned by humans, thus it is allowable that all retrieval images are all incorrect.

Visualization of the Discriminative Geometric-Structure: To clearly understand how the discriminative geometric-structure DGDH learns, we do experiments on CIFAR-10 [7] to show the learned deep features with different margin values m. As m describes the margin between the class centers and the real value features, and the 2-norm of both the class centers and the real value features have correlation with the hash bit r, so we make the margin value m ranges from $(0, 0.01, 0.1, 1, 10) \times r$. In detail, we randomly select four classes and select 1500 images from each class on CIFAR-10 to train DGDH. Then, we display the learned 2-D deep features in Fig. 10. As Fig. 10(a)–(e) shows, the discriminative geometric-structure of 2-D features is that *samples from four classes locate on four circle regions around the corresponding binary hash codes with tolerable margins*, and this is consistent with our goal in Fig. 1(d). Since the different distributions of dataset, amounts of intraclass samples are usually hard to approach to the corresponding class centers. To this, preserving a proper margin for most intraclass samples can significantly reduce the cumulative loss and thus could achieve more compact representation for intraclass samples. As shown in Fig. 10(c), the margin $0.1 \times r$ achieves the best discriminative ability among all the given margins.

V. CONCLUSION

This article proposes a DGDH, which can learn discriminative and compact deep features and hash codes. In detail,



(a) (b) (c) (d) (e)

Fig. 10. Learned discriminative geometric-structure of 2-bit deep features $\tanh(F(x_i;\phi))$ on CIFAR-10 dataset versus different margin values. (a) $m = 0 \times r$. (b) $m = 0.01 \times r$. (c) $m = 0.1 \times r$. (d) $m = 1 \times r$. (e) $m = 10 \times r$.

DGDH learns an attractive discriminative geometric structure where class centers are located on a hypersphere and samples surround the corresponding class centers with a certain margin. The experimental results on toy and real-world datasets verify the effectiveness of this discriminative geometric structure. Moreover, DGDH can reduce quantization errors in a new manner, which connects the samples, class centers, and the binary hash codes through the proposed geometrical structure. Theoretical analysis verifies robustness and generalization ability of DGDH. Extensive image retrieval experiments on five popular datasets demonstrate the superiority of DGDH over several state-of-the-art methods. In future work, it is hopeful to further enhance image retrieval performance by integrating the intriguing insights of recent studies [61], [62] into a unified hashing learning framework to enjoy the strengths of the context information.

REFERENCES

[1] A. Andoni and P. Indyk, "Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions," in *Proc. IEEE Symp. Found. Comput. Sci.*, 2006, pp. 459–468.

[2] Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2008, pp. 1753–1760.

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

[4] 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.

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

[6] G. Dong, X. Zhang, L. Lan, X. Huang, and L. Zhigang, "Discrete graph hashing via affine transformation," in *Proc. IEEE Int. Conf. Multimedia Expo*, 2018, pp. 1–6.

[7] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, Rep. TR-2009, 2009.

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

[9] Q. Jiang, X. Cui, and W. Li, "Deep discrete supervised hashing," *IEEE Trans. Image Process.*, vol. 27, pp. 5996–6009, 2018.

[10] S. Su, Z. Zhong, and C. Zhang, "Deep joint-semantics reconstructing hashing for large-scale unsupervised cross-modal retrieval," in *Proc. Int. Conf. Comput. Vis.*, 2019, pp. 3027–3035.

[11] D. Wu, Q. Dai, J. Liu, B. Li, and W. Wang, "Deep incremental hashing network for efficient image retrieval," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 9069–9077.

[12] Z. Zhang, Q. Zou, Y. Lin, L. Chen, and S. Wang, "Improved deep hashing with soft pairwise similarity for multi-label image retrieval," *IEEE Trans. Multimedia*, vol. 22, no. 2, pp. 540–553, Feb. 2020.

[13] L. Yuan *et al.*, "Central similarity quantization for efficient image and video retrieval," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 3083–3092.

[14] X. Shen, G. Dong, Y. Zheng, L. Lan, I. W. Tsang, and Q.-S. Sun, "Deep co-image-label hashing for multi-label image retrieval," *IEEE Trans. Multimedia*, vol. 24, pp. 1116–1126, 2022.

[15] T. Xiao, T. Xia, Y. Yang, C. Huang, and X. Wang, "Learning from massive noisy Labeled data for image classification," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 2691–2699.

[16] E. Yang, T. Liu, C. Deng, and D. Tao, "Adversarial examples for hamming space search," *IEEE Trans. Cybern.*, vol. 50, no. 4, pp. 1473–1484, Apr. 2020.

[17] X. Zhou *et al.*, "Graph convolutional network hashing," *IEEE Trans. Cybern.*, vol. 50, no. 4, pp. 1460–1472, Apr. 2020.

[18] C. Ma, I. W. Tsang, F. Shen, and C. Liu, "Error correcting input and output hashing," *IEEE Trans. Cybern.*, vol. 49, no. 3, pp. 781–791, Mar. 2019.

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

[20] B. Kulis and K. Grauman, "Kernelized locality-sensitive hashing for scalable image search," in *Proc. Int. Conf. Comput. Vis.*, 2009, pp. 2130–2137.

[21] Y. Yang, Y. Luo, W. Chen, F. Shen, J. Shao, and H. T. Shen, "Zero-shot hashing via transferring supervised knowledge," in *Proc. ACM Int. Conf. Multimedia*, 2016, pp. 1286–1295.

[22] L. Zhu, Z. Huang, X. Chang, J. Song, and H. T. Shen, "Exploring consistent preferences: Discrete hashing with pair-exemplar for scalable landmark search," in *Proc. ACM Int. Conf. Multimedia*, 2017, pp. 726–734.

[23] L. Zhu, Z. Huang, Z. Li, L. Xie, and H. T. Shen, "Exploring auxiliary context: Discrete semantic transfer hashing for scalable image retrieval," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 11, pp. 5264–5276, Nov. 2018.

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

[25] B. Neyshabur, N. Srebro, R. Salakhutdinov, Y. Makarychev, and P. Yadollahpour, "The power of asymmetry in binary hashing," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2013, pp. 2823–2831.

[26] 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. Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1971–1978.

[27] W. Kang, W. Li, and Z. Zhou, "Column sampling based discrete supervised hashing," in *Proc. AAAI Conf. Artif. Intell.*, 2016, pp. 1230–1236.

[28] C. Da, S. Xu, K. Ding, G. Meng, S. Xiang, and C. Pan, "AMVH: Asymmetric multi-valued hashing," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 898–906.

[29] 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.

[30] J. Wang, S. Kumar, and S.-F. Chang, "Semi-supervised hashing for large-scale search," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 12, pp. 2393–2406, Dec. 2012.

[31] H. Hu, K. Wang, C. Lv, J. Wu, and Z. Yang, "Semi-supervised metric learning-based anchor graph hashing for large-scale image retrieval," *IEEE Trans. Image Process.*, vol. 28, pp. 739–754, 2019.

[32] Z. Li, J. Tang, L. Zhang, and J. Yang, "Weakly-supervised semantic guided hashing for social image retrieval," *Int. J. Comput. Vis.*, vol. 128, no. 8, pp. 2265–2278, 2020.

[33] R. Salakhutdinov and G. E. Hinton, "Semantic hashing," *Int. J. Approx. Reason.*, vol. 50, no. 7, pp. 969–978, 2009.

[34] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2012, pp. 1106–1114.

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

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

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

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

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

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

[41] 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, pp. 4766–4779, 2015.

[42] J. Li, B. Zhang, G. Lu, and D. Zhang, "Dual asymmetric deep hashing learning," 2018, *arXiv:1801.08360*.

[43] Q. Jiang and W. Li, "Asymmetric deep supervised hashing," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 3342–3349.

[44] X. Shi, F. Xing, K. Xu, M. Sapkota, and L. Yang, "Asymmetric discrete graph hashing," in *Proc. AAAI Conf. Artif. Intell.*, 2017, pp. 2541–2547.

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

[46] X. Xu, F. Shen, Y. Yang, H. T. Shen, and X. Li, "Learning discriminative binary codes for large-scale cross-modal retrieval," *IEEE Trans. Image Process.*, vol. 26, pp. 2494–2507, 2017.

[47] X. Zhang, G. Dong, Y. Du, C. Wu, Z. Luo, and C. Yang, "Collaborative subspace graph hashing for cross-modal retrieval," in *Proc. Int. Conf. Multimedia Retrieval*, 2018, pp. 213–221.

[48] G. Dong, X. Zhang, L. Lan, S. Wang, and Z. Luo, "Label guided correlation hashing for large-scale cross-modal retrieval," *Multimedia Tools Appl.*, vol. 78, pp. 30895–30922, Feb. 2019.

[49] H. T. Shen *et al.*, "Exploiting subspace relation in semantic labels for cross-modal hashing," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 10, pp. 3351–3365, Oct. 2021.

[50] D. Xie, C. Deng, C. Li, X. Liu, and D. Tao, "Multi-task consistency-preserving adversarial hashing for cross-modal retrieval," *IEEE Trans. Image Process.*, vol. 29, pp. 3626–3637, 2020.

[51] L. Jin, Z. Li, and J. Tang, "Deep semantic multimodal hashing network for scalable image-text and video-text retrievals," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jun. 5, 2020, doi: 10.1109/TNNLS.2020.2997020.

[52] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," in *Proc. Brit. Mach. Vis. Conf.*, 2014, pp. 1–11.

[53] C. Deng, E. Yang, T. Liu, and D. Tao, "Two-stream deep hashing with class-specific Centers for supervised image search," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 31, no. 6, pp. 2189–2201, Jun. 2020.

[54] Y. Wen, K. Zhang, Z. Li, and Y. Qiao, "A discriminative feature learning approach for deep face recognition," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 499–515.

[55] T. Lin *et al.*, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.

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

[57] T. 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. Int. Conf. Image Video Retrieval*, 2009, p. 48.

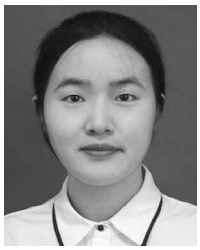
[58] C. Fu, G. Wang, X. Wu, Q. Zhang, and R. He, "Deep momentum uncertainty hashing," *Pattern Recognit.*, vol. 22, Feb. 2022, Art. no. 108264.

[59] B. Thomee, M. J. Huiskes, and M. S. Lew, "New trends and ideas in visual concept detection: The MIR Flickr retrieval evaluation initiative," in *Proc. Int. Conf. Multimedia Inf. Retrieval*, 2010, pp. 527–536.

[60] J. T. Hoe, K. W. Ng, T. Zhang, C. S. Chan, Y.-Z. Song, and T. Xiang, "One loss for all: Deep hashing with a single cosine similarity based learning objective," in *Advances in Neural Information Processing Systems*. Red Hook, NY, USA: Curran Assoc., 2021.

[61] S. Minaee, Y. Y. Boykov, F. Porikli, A. J. Plaza, N. Kehtarnavaz, and D. Terzopoulos, "Image segmentation using deep learning: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Feb. 17, 2021, doi: 10.1109/TPAMI.2021.3059968.

[62] Z. Li, Y. Sun, L. Zhang, and J. Tang, "CTNet: Context-based tandem network for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Dec. 2, 2021, doi: 10.1109/TPAMI.2021.3132068.

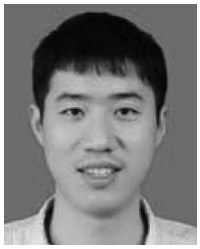


Guohua Dong received the Ph.D. degree from the College of Computer, National University of Defense Technology, Changsha, China, in 2019.

She is currently a Research Assistant with the Beijing Institute of Basic Medical Sciences, Beijing, China. Her current research interests include information retrieval, hashing, graph models, network embedding, and bioinformatics.

Xiang Zhang received the M.S. and Ph.D. degrees from the National University of Defense Technology, Changsha, China, in 2010 and 2015, respectively.

He is currently a Research Assistant with the Institute for Quantum Information & State Key Laboratory of High Performance Computing, College of Computer, National University of Defense Technology. His current research interests include computer vision and machine learning.



Xiaobo Shen received the B.Sc. and Ph.D. degrees from the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China, in 2011 and 2017, respectively.

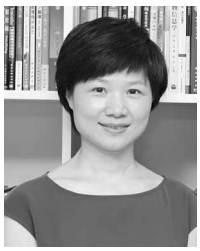
He is currently a Professor with the School of Computer Science and Engineering, Nanjing University of Science and Technology. His primary research interests are machine learning and pattern recognition.

Long Lan received the Ph.D. degree in computer science from National University of Defense Technology, Changsha, China, in 2017.

He is currently a Lecturer with the College of Computer, National University of Defense Technology. He was a visiting Ph.D. student with the University of Technology Sydney, Sydney, NSW, Australia, from 2015 to 2017. His research interests include multiobject tracking, computer vision, and discrete optimization.

Zhigang Luo received the B.S., M.S., and Ph.D. degrees from the National University of Defense Technology, Changsha, China, in 1981, 1993, and 2000, respectively.

He is currently a Professor with the College of Computer, National University of Defense Technology. His current research interests include machine learning, computer vision, and bioinformatics.



Xiaomin Ying received the B.S. and Ph.D. degrees from the National University of Defense Technology, Changsha, China, in 1997, and 2003, respectively.

She is currently a Professor of Bioinformatics with the Beijing Institute of Basic Medical Sciences, Beijing, China. Her current research interests include machine learning and interdisciplinary researches of AI and biology.