Deep Ordinal Hashing With Spatial Attention

Lu Jin, Xiangbo Shu, Kai Li, Zechao Li, Guo-Jun Qi, and Jinhui Tang, *Senior Member, IEEE*

***Abstract*— Hashing has attracted increasing research attention in recent years due to its high efficiency of computation and storage in image retrieval. Recent works have demonstrated the superiority of simultaneous feature representations and hash functions learning with deep neural networks. However, most existing deep hashing methods directly learn the hash functions by encoding the global semantic information, while ignoring the local spatial information of images. The loss of local spatial structure makes the performance bottleneck of hash functions, therefore limiting its application for accurate similarity retrieval. In this paper, we propose a novel deep ordinal hashing (DOH) method, which learns ordinal representations to generate ranking-based hash codes by leveraging the ranking structure of feature space from both local and global views. In particular, to effectively build the ranking structure, we propose to learn the rank correlation space by exploiting the local spatial information from fully convolutional network and the global semantic information from the convolutional neural network simultaneously. More specifically, an effective spatial attention model is designed to capture the local spatial information by selectively learning well-specified locations closely related to target objects. In such hashing framework, the local spatial and global semantic nature of images is captured in an end-to-end ranking-to-hashing manner. Experimental results conducted on three widely used datasets demonstrate that the proposed DOH method significantly outperforms the state-of-the-art hashing methods.**

***Index Terms*— Hashing, image retrieval, ranking structure, fully convolutional network, convolutional neural network, local spatial, global semantic information.**

I. Introduction

RECENTLY, large-scale image retrieval has gained wide attention in the field of computer vision due to the rapid advancement of information techniques. With the explosive growth of multimedia data including images and videos, hashing has received a great deal of attention in large-scale visual retrieval for its capability in storage and computation efficiency [1]–[5]. Hashing is to construct a set of hash functions by projecting the high dimensional data in the visual space into compact binary codes in the Hamming space. Due to the amazing performance of deep features in computer vision tasks, such as image classification [6]–[8], image captioning [9]–[11] and image retrieval [12]–[14], many deep hashing methods have been proposed to learn feature representations and hash functions simultaneously [15], [16].

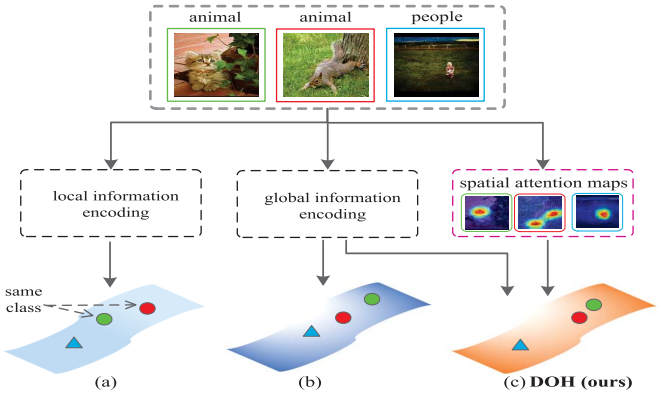


Fig. 1. Illustration of the idea of the proposed hashing framework. (a) and (b) show the hash codes generated by encoding the local information with FCN network and the global semantic information with CNN network, respectively. (c) shows the proposed hashing method that jointly captures the global semantic structure and the local spatial information with spatial attention maps. The resultant hash codes of the proposed method can achieve accurate matching.

As the outputs of the fully-connected layer of CNN have much richer semantic information than hand-crafted features [6], [7], most deep hashing methods directly use the outputs of the fully-connected layer to approximate binary hash codes [15], [17]. Unfortunately, it may be unsuitable to directly utilize the feature representation from the fully-connected layer due to the spatial information loss [18], which may lead to the suboptimal hash codes. Besides, recent works on image localization [19], [20] and object detection [21], [22] have shown the importance of the inherent spatial information from 2-dimensional feature maps of the convolutional layer. Therefore, it is necessary and suitable to explore the local spatial information in the deep hashing framework. By exploring the local spatial and global semantic information simultaneously, the resultant hash codes can preserve the similarities well, as shown in Figure 1. On the other hand, as we all know, the binary quantization functions $sign(\cdot)$ and $threshold(\cdot)$ are sensitive to the perturbations in numeric values caused by noise and variations [23]. Meanwhile, the ranking-based function that encodes the relative ordering correlation benefits from excellent properties of rank correlation measures such as scale-invariance, numeric stability, and high nonlinearity [3], [23]–[25]. Thus, we propose

Manuscript received May 14, 2018; revised October 3, 2018; accepted November 5, 2018. Date of publication November 28, 2018; date of current version January 16, 2019. This work was supported in part by the National Key Research and Development Program of China under Grant 2016YFB1001001 and in part by the National Natural Science Foundation of China under Grants 61732007, 61522203, and 61772275. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Husrev T. Sencar. *(Corresponding author: Jinhui Tang.)*

L. Jin, X. Shu, Z. Li, and J. Tang are with the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China (e-mail: lujin505@gmail.com; shuxb@njust.edu.cn; zechao.li@njust.edu.cn; jinhuitang@njust.edu.cn).

K. Li is with Facebook, Menlo Park, CA 94025 USA (e-mail: kailee88@fb.com).

G.-J. Qi is with the Department of Computer Science, University of Central Florida, Orlando, FL 32816 USA (e-mail: guojun.qi@ucf.edu).

Digital Object Identifier 10.1109/TIP.2018.2883522

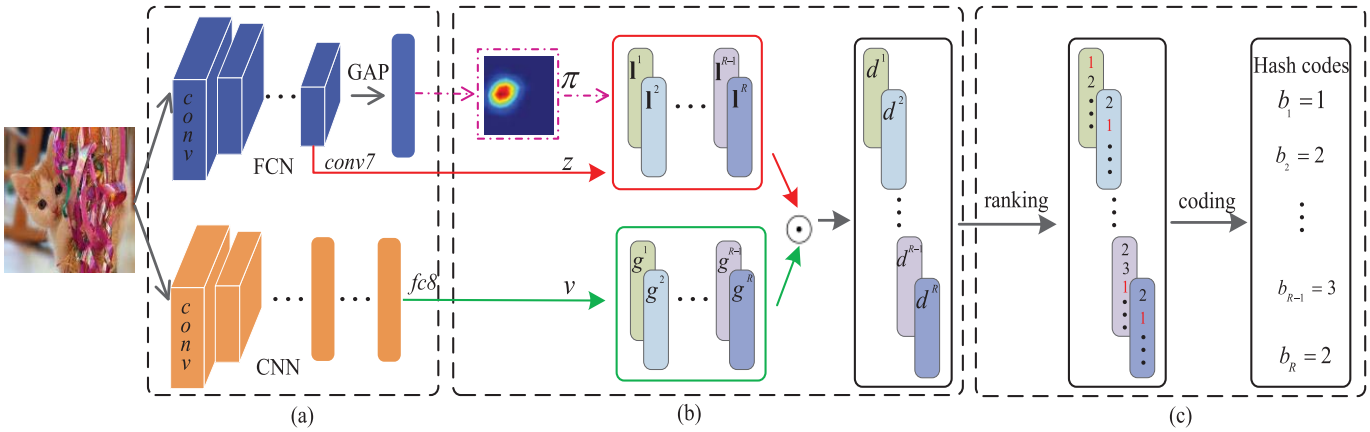


Fig. 2. Framework of the proposed Deep Ordinal Hashing method, which consists of three components: (a) the feature representation learning which jointly learns the local spatial and global semantic information from FCN and CNN networks, where GAP indicates the global average pooling layer for FCN. (b) a subnetwork to learn the local and global-aware representations $[\mathbf{d}^1, \cdots, \mathbf{d}^R]$, where $[\mathbf{l}^1, \cdots, \mathbf{l}^R]$ defines the local-aware representations of the FCN network, $[\mathbf{g}^1, \cdots, \mathbf{g}^R]$ defines the global-aware representations of the CNN network, R defines the length of the K-ary hash code, and $\odot$ denotes element-wise Hadamard product. In order to capture the local discriminativity, the spatial attention map π and the channel-wise representation $\mathbf{z}$ of $conv7$ layer are both leveraged to learn the local-aware representation $[\mathbf{l}^1, \cdots, \mathbf{l}^R]$. Additionally, the feature representation $\mathbf{v}$ of $fc8$ layer is used to generate the global-aware representation $[\mathbf{g}^1, \cdots, \mathbf{g}^R]$. (c) the ordinal representation learning to approximate the vectorized representation of ranking-based hash functions which are used to generate the hash codes.

to jointly explore the inherent spatial structure and global semantic information to learn the rank correlation space under the deep hashing framework.

In this work, we propose a novel hashing method to explore the global semantic information, the local spatial information and the relative ordering correlation based on the deep learning framework. Here we introduce the Convolutional Neural Network (CNN) [6], [7] to learn much richer global semantic information than hand-crafted features from images. To explore the local spatial information, an effective spatial attention model is designed to selectively learn well-specified locations closely related to target objects (i.e., aubergine dotted box in Figure 1). To effectively build the ranking structure, the rank correlation space is learned by exploring the local spatial and global semantic information simultaneously. By incorporating the above terms into one unified framework, we propose a novel Deep Ordinal Hashing (DOH) method, as illustrated in Figure 2. Our network architecture contains three major components: (a) the feature representation learning which learns the local spatial and global semantic information from FCN and CNN respectively; (b) a subnetwork to learn the local and global-aware representation by encoding the local spatial and global semantic information simultaneously; (c) the ordinal representation learning to generate compact ranking-based hash codes. In summary, we highlight four contributions of this paper as follows:

- First, we propose a unified framework to learn hash functions by exploring the rank correlation space from both local and global views. To the best of our knowledge, this is the first attempt that learns a group of ranking-based hash functions by jointly exploiting the local information obtained from the spatial attention model and the global semantic structure for image retrieval.
- Second, we design a subnetwork to effectively build the rank structure by jointly exploring the local spatial information from FCN and the global semantic information from CNN.
- Third, we develop an effective spatial attention model to capture the local discriminativity by learning well-specified locations closely related to target objects.
- Finally, we extensively evaluate the proposed method on three widely-used image retrieval benchmarks. The experimental results show that the proposed method significant outperforms the state-of-the-arts, which demonstrates the superiority and effectiveness of the proposed method.

The rest of this paper is organized as follows. In Section II, we briefly review the related works. The proposed method is introduced in Section III. We present the optimization algorithm in Section IV. The extensive experiments and discussions of the experimental results are provided in Section V. Finally, we conclude this work in Section VI.

II. Related Work

A. Attention-Aware Methods

Recently, the attention mechanism has been widely studied in image/video captioning [10], [26], [27], localization [18], [28], [29] and object detection [30], [31] and so on. The attention models selectively learn the salient region of the images for a particular visual task. Xu *et al.* [32] propose the attention model to learn the attention locations for image captioning. Specifically, two different strategies are introduced for the attention model including the "hard" stochastic attention and "soft" deterministic attention. The stochastic attention model decides the attention locations using the "hard" pooling, while the deterministic attention model learns a soft annotation vector by averaging the feature vectors with the attention weights. Spatial Memory Network [33] is proposed to adopt the stacked recurrent neural networks with the spatial attention for Visual Question Answering.

Luong *et al.* [34] propose the "global" and "local" attention models for Neural Machine Translation, in which the global model selects all source words and the local model only selects a subset of source words. Yang *et al.* [35] adopt a multiple-layer stacked attention networks for image question answering. Comparative Attention Network [36] uses the soft attention model to choose the discriminative parts of the person images for person re-identification. Chen *et al.* [9] integrate the channel and spatial-wise attention models in a convolutional neural network for image captioning. Although the attention models have been successfully used for a wide range of visual tasks, it is still not fully exploited for image retrieval.

B. Hash Learning

In this paper, we mainly focus on data-dependent hashing methods that learn hash functions by preserving the data structure. In general, data-dependent hashing methods can be grouped into unsupervised and supervised ones. Unsupervised hashing methods usually learn hash functions by exploiting the intrinsic data structure embedded in the original space. For example, Iterative Quantization (ITQ) [4] attempts to generate zero-centered binary codes by maximizing the variance of each binary bit as well as minimizing the quantization error. Representative methods include Spectral Hashing (SH) [37], Self-Taught Hashing (STH) [38], Anchor Graph Hashing (AGH) [39], Neighborhood Discriminant Hashing (NDH) [40], etc. For supervised methods, the supervised information is incorporated to learn compact binary codes. Existing works have indicated that leveraging the supervised information can produce high-quality hash codes. Supervised Hashing with Kernels (KSH) [41] employs the kernel-based function as the hash function. Supervised Discrete Hashing (DSH) [42] and Fast Hashing (FastH) [43] generate binary hash codes by directly solving the binary programming problem. Other notable methods include Binary Reconstructive Embedding (BRE) [44], Minimal Loss Hashing (MLH) [45], Asymmetry in Binary Hashing [46], Label Preserving Multimedia Hashing (LPMH) [47], etc.

Although the aforementioned hashing methods achieve desirable performance, they usually generate suboptimal hash codes due to independently learning feature representations and hash functions. Motivated by the great success of deep learning in computer version tasks, deep hashing frameworks are starting to receive broad attention recently. Different from conventional hashing methods, deep hashing methods [48]–[51] generate hash codes in such a way that feature representations are optimized during the hash function learning process. The works in [16] and [48] are the earliest attempt in jointly learning feature representations and hash functions. Lu *et al.* [52] construct multiple hierarchical non-linear transformations using a deep network with stacked fully-connected layers to learn binary hash codes. Deep Supervised Hashing (DSH) [12] tries to preserve the similarities and minimize the binarization loss simultaneously. Supervised Semantics-preserving Deep Hashing (SSDH) [53] constructs a latent hash layer to generate hash codes by directly minimizing the classification error on the outputs of the hash layer.

However, most existing deep hashing methods adopt binary quantization functions, which are known to be sensitive with prevalent noises and variations. In comparison, the ranking-based hash function encodes the relative ordering of the projected feature space to generate hash codes. Therefore, it benefits from excellent properties of ordinal measures in scale-invariant, numerically stable, and highly nonlinear [23]. Typical methods contain Min-wise Hash [3], Winner-Take-All Hash [24], Linear Subspace Ranking Hashing [25] and Deep Semantic-Preserving Ordinal Hashing (DSPOH) [54]. By ranking the ordering of feature dimensions represented by the hand-crafted descriptors, these ranking-based hashing methods are inadequate to learn optimal hash functions. Apart from this, the linear transformation of the feature space is insufficient to capture the complex semantic structure of the images, which limits the retrieval performance.

In this work, we employ Convolutional Neural Network and Fully Convolutional Network with the spatial attention model for hash learning. Specifically, we integrate the local spatial attention information and global semantic information to learn the ranking-based hash functions. To our best knowledge, this work is the first attempt to learn ranking-based hash functions with the deep neural networks by exploiting the local information obtained from the spatial attention model and the global semantic information simultaneously.

III. Deep Ordinal Hashing

In this section, we introduce the proposed deep hashing framework in detail. The entire framework is illustrated in Figure 2. The proposed method first adopts the spatial attention model to generate the spatial attention map for the input image by using the FCN network. Then the spatial attention map and deep representations of the deep networks are exploited to generate the local and global-aware representations, which are further used to learn the ranking-based hash function. Finally, the ordinal representation are introduced to approximate the vectorized representation of the ranking-based hash function.

A. Problem Definition

Suppose $\mathcal{I} = \{q_n\}_{n=1}^{N}$ be a set of N images, and $\mathcal{T} = \{\mathbf{t}_n\}_{n=1}^{N}$ be a set of their corresponding binary label vectors, where $\mathbf{t}_n \in \{0, 1\}^C$ and C defines the total number of the categories. The non-zero entry in $\mathbf{t}_n$ indicates that the n^{th} image belongs to the corresponding class. Let $\mathbf{S} = \{s_{ij}\} \in \{0, 1\}^{N \times N}$ be the similarity matrix, where $s_{ij} = 1$ if the image pair (q_i, q_j) share at least one common class, otherwise $s_{ij} = 0$. Our goal is to learn a set of mappings $\mathcal{F} = \{f^r(\cdot)\}_{r=1}^{R} : q \rightarrow \mathbf{b}$, which maps the image q into a R-bit hash code $\mathbf{b}$. Specifically, the hash function $f^r(\cdot)$ is used to generate a K-ary hash bit b_r by encoding the ranking structure of a K-dimensional feature space, where b_r is the r^{th} bit of $\mathbf{b}$. Table I summarizes the definition of the notations used in this work.

B. Network Architecture

The proposed method is a two-stream network where the FCN model aims to capture the local information using

TABLE I
DEFINITION OF THE NOTATIONS

notation	definition
K	the dimension of the feature space
R	the length of K-ary hash code
N_c	the length of the binary hash code
M	the number of the feature maps of the $conv7$ layer of FCN network
$\mathbf{b} \in \mathbb{R}^R$	the K-ary hash code
$\mathbf{z}_{xy} \in \mathbb{R}^M$	the channel-wise representation at the spatial location (x, y) of the $conv7$ layer of FCN network
$\mathbf{v} \in \mathbb{R}^M$	the feature representation of the $fc8$ layer of CNN network
μ_{xy}^c	the object-specific local response at the spatial location (x, y)
p_c	the c^{th} probabilistic output of the classification layer of FCN network
π_{xy}	the local response at the spatial location (x,y)
$\mathbf{l}^r \in \mathbb{R}^K$	the local-aware representation for the r^{th} slice of the hash layer of FCN network
$\mathbf{g}^r \in \mathbb{R}^K$	the global-aware representation for the r^{th} slice of the hash layer of CNN network
$\mathbf{d}^r \in \mathbb{R}^K$	the local and global-aware representation for generating the r^{th} hash bit
$f^r(\cdot)$	the ranking-based hash function for generating the r^{th} hash bit
$\mathbf{W}_s^r$,$\mathbf{c}_s^r$	the transformation matrix and the bias vector for the r^{th} slice of the hash layer of FCN network
$\mathbf{W}_g^r$,$\mathbf{c}_g^r$	the transformation matrix and the bias vector for the r^{th} slice of the hash layer of CNN network
h_k^r	the probability that the k^{th} dimension taking the maximum value in $\mathbf{d}^r$
$\mathbf{h}^r = [h_1^r, \cdots, h_K^r]$	the ordinal representation for generating the r^{th} hash bit
$\Phi_F = \{\mathbf{W}_s^r, \mathbf{c}_s^r\}_{r=1}^R$	all R transformation matrixes and bias vectors of the hash layer of FCN network
$\Phi_C = \{\mathbf{W}_g^r, \mathbf{c}_g^r\}_{r=1}^R$	all R transformation matrixes and bias vectors of the hash layer of CNN network

the spatial attention model while the CNN model explores the global semantic information. We adopt the widely-used Alexnet [6] as our basic network which includes five convolutional layers from $conv1$ to $conv5$, two fully-connected layers from $fc6$ to $fc7$ and a task-specific fully-connected classification layer (i.e., fc-c). However, we make some small modification to Alexnet. For CNN network, we add a fully-connected layer (i.e., $fc8$) followed by fc-c to learn the global visual descriptors by encoding the global semantic information. Additionally, for FCN network, we replace $fc6$ and $fc7$ by two convolutional layers (i.e., $conv6$ and $conv7$) followed by the fc-c layer, where $conv7$ is followed with a global average pooling layer (i.e., $pool7$). Similar with the CNN network, the $conv7$ layer aims to learn the local visual descriptors by encoding the local spatial information.

The detailed configurations of the FCN and CNN networks are described in Table II. For the convolutional layers (e.g. $conv1$ to $conv7$), "Filter" represents the number and the receptive filter size of the convolutional kernels as "$number \times size \times size$", "Pad" specifies the number of the pixels to add to the input, "Stride" specifies the intervals at witch to apply the filter to the input, "Pool" specifies the down-sampling operation, "LRN" denotes the Local Response Normalization. For the fully-connected layer, "number" denotes the number of the outputs of this layer. Besides, "ReLU" denotes the Rectified Linear Unit activation function. "Dropout" specifies whether the layer is regularized by dropout operation. More specifically, except for the $conv7$ layer adopting the global average pooling (GAP), the remaining "Pool"s adopt the max pooling strategy.

Additionally, DOH constructs hash layers (i.e., fc-h) to learn the ranking-based hash functions. Specifically, we set the number of the outputs as $K \times R$ for fc-h layer where K is the dimension of the feature space and R is the length of K-ary hash code. Different from the binary quantization hashing methods, the proposed DOH generates K-ary hash code.

TABLE II
CONFIGURATIONS OF THE TWO-STREAM NETWORK

Network	Layer	Configuration
FCN	$conv1$	Filter: $96 \times 11 \times 11$; Pad: 0; Stride:4; ReLU; Pool; LRN;
	$conv2$	Filter: $256 \times 5 \times 5$; Pad: 2; Stride:1; ReLU; Pool; LRN;
	$conv3$	Filter: $384 \times 3 \times 3$; Pad: 1; Stride:1; ReLU;
	$conv4$	Filter: $384 \times 3 \times 3$; Pad: 1; Stride:1; ReLU;
	$conv5$	Filter: $384 \times 3 \times 3$; Pad: 1; Stride:1; ReLU; Pool;
	$conv6$	Filter: $512 \times 3 \times 3$; Pad: 1; ReLU;
	$conv7$	Filter: $512 \times 3 \times 3$; Pad: 1; ReLU; Pool (GAP); Dropout;
	fc-c	Number: C;
	fc-h	Number: $K \times R$;
CNN	$conv1$	Filter: $96 \times 11 \times 11$; Pad: 0; Stride:4; ReLU; Pool; LRN;
	$conv2$	Filter: $256 \times 5 \times 5$; Pad: 2; Stride:1; ReLU; Pool; LRN;
	$conv3$	Filter: $384 \times 3 \times 3$; Pad: 1; Stride:1; ReLU;
	$conv4$	Filter: $384 \times 3 \times 3$; Pad: 1; Stride:1; ReLU;
	$conv5$	Filter: $384 \times 3 \times 3$; Pad: 1; Stride:1; ReLU; Pool;
	$fc6$	Number: 4096; ReLU; Dropout;
	$fc7$	Number: 4096; ReLU; Dropout;
	$fc8$	Number: 512; ReLU; Dropout;
	fc-c	Number: C;
	fc-h	Number: $K \times R$;

For fair comparison, we set $R = N_c / \log_2 K$ when comparing with other binary quantization hashing methods, where N_c is the length of the binary hash code.

C. Spatial Attention Model

Generally, the target objects are located in different spatial locations of images, as shown in Figure 1. For example,

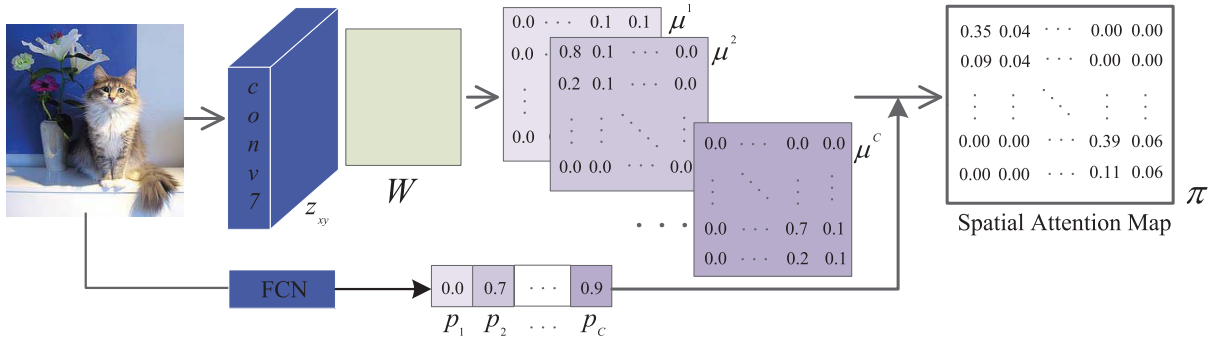


Fig. 3. An imaginary illustration of the spatial attention model for the FCN network. This model aims to capture the local discriminativity by learning well-specified locations closely related to target objects. Firstly, the object-specific local response map μ^c is obtained by projecting the weight matrix $\mathbf{W}$ of the the classification layer on the channel-wise feature representation $\mathbf{z}_{xy}$ of the $conv7$ layer. After that, the probabilistic outputs $\mathbf{P}$ of the classification layer of the FCN network and object-specific local response maps are jointly leveraged to produce the spatial attention map π.

when we use the query to search for relevant images in the database, those spatial locations closely related to the target objects are more useful for accurate matching. Instead of considering each spatial location equally, we expect to detect those discriminative spatial locations that are highly related to the objects and give larger response values to them. Consequently, we introduce a spatial attention model, which aims to generate object-specific spatial attention map. Figure 3 illustrates the generation of the spatial attention map. We also visualize the spatial attention maps of some examples in Figure 8.

Following the work in [18], we employ Fully Convolutional Network with the global average pooling to generate the attention locations of the images. Specifically, we perform the global average pooling on the $conv7$ layer and feed the outputs into the fully-connected classification layer (i.e., fc-c). The outputs of the global average pooling can be regarded as the spatial average of feature maps of the $conv7$ layer. Those spatial average values are used to generate the probabilistic outputs of the fc-c layer. In this section, we introduce an intuitive way to produce the spatial attention map by projecting the weight matrix of the fc-c layer on the feature maps of the $conv7$ layer.

Let $\mathbf{z}_{xy}$ define the channel-wise representation at the spatial location (x, y) of the $conv7$ layer of FCN network, which can be computed as

$$\mathbf{z}_{xy} = \psi_{\mathcal{F}}(q; \Omega_{\mathcal{F}}), \tag{1}$$

where $\mathbf{z}_{xy} \in \mathbb{R}^M$ with M being the number of the feature maps, $x \in \{1, \cdots, X\}$, $y \in \{1, \cdots, Y\}$, X and Y are the width and height of feature maps, q represents the input image, $\psi_{\mathcal{F}}$ defines the non-linear projection function for the FCN network and $\Omega_{\mathcal{F}} = \{\mathbf{W}^d_{\mathcal{F}}, \mathbf{c}^d_{\mathcal{F}}\}_{d=1}^{D_{\mathcal{F}}}$ defines a set of non-linear projection parameters with $D_{\mathcal{F}}$ being the depth of the FCN network.

Consider the weight matrix $\mathbf{W} \in \mathbb{R}^{M \times C}$ performs a mapping of the spatial average values to the semantic class labels. In particular, we define the object-specific local response at the spatial location (x, y) as μ^c_{xy}, which can be obtained by

$$\mu^c_{xy} = \max(\mathbf{w}_c^T \mathbf{z}_{xy}, 0), \quad \text{for } c = 1, \cdots, C, \tag{2}$$

where $\mathbf{w}_c$ is the c^{th} column of $\mathbf{W}$. Obviously, μ^c_{xy} indicates the importance at the spatial location (x, y) that the input image q is classified to the c^{th} class. As each spatial location (x, y) corresponds to different local patch in the original image, the local response μ^c_{xy} indicates the relative similarity of the local image patch to the c^{th} class.

Essentially, we can obtain the local discriminative information at different spatial locations for a particular class. The bigger value of μ^c_{xy} indicates that the image patch at the spatial location (x, y) is more relative to the c^{th} class. Inversely, the smaller value of μ^c_{xy} indicates less relative to the c^{th} class. By integrating μ^c_{xy} together, we can define the spatial attention map π to identify all the object-specific image patches. Specifically, the local response at (x, y), denoted as π_{xy}, is defined as follows

$$\pi_{xy} = \sum_{c=1}^{C} p_c \mu^c_{xy} / \sum_{c=1}^{C} p_c, \tag{3}$$

where p_c is the c^{th} probabilistic output of the classification (i.e., fc-c) layer of the FCN network.

D. Local and Global-Aware Representations

In order to jointly capture the local spatial and global semantic information to learn ranking-based hash functions, we require the ordinal representations which are used to approximate vectorized representations of the hash functions to be local and global aware. As shown in Figure 2, DOH jointly encodes the feature representations of the $conv7$ layer and spatial attention map to generate the local-aware representations which explicitly exploits the local spatial structure of images. In addition, DOH learns global-aware representations by encoding the global semantic information from the feature representations of the $fc8$ layer. Let $\mathbf{l}, \mathbf{g} \in \mathbb{R}^K$ be the local-aware and global-aware representations respectively. To jointly preserve both of local and global information, we further integrate them to define the confident score d_k as follow

$$d_k = l_k g_k, \quad \text{for } k = 1, \cdots, K, \tag{4}$$

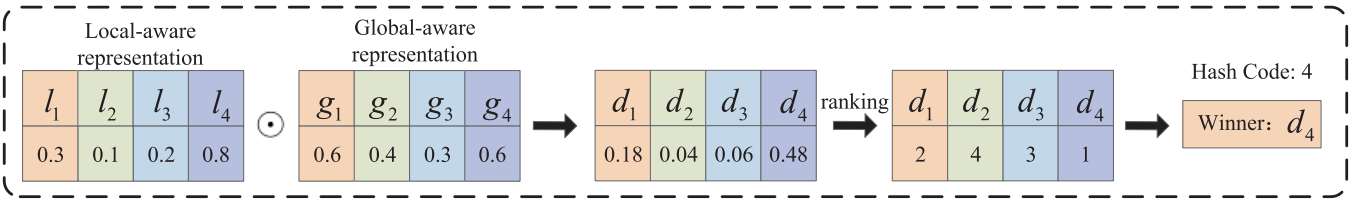


Fig. 4. An example with 4-dimensional feature space to produce 4-ary hash code.

where l_k and g_k are the k^{th} dimension of $\mathbf{l}$ and $\mathbf{g}$ respectively. By connecting d_k together, we can define $\mathbf{d} = [d_1, \cdots, d_K] \in \mathbb{R}^K$ as the local and global-aware representations.

1) Local-Aware Representations: The spatial attention map demonstrates the local discriminativity in identifying partial image regions that are highly related to target objects. When searching relevant images in the database, those discriminative spatial locations are crucial important for accurate matching. We argue for learning above local spatial structure for ordinal representations by jointly encoding the discriminative spatial attention map and convolutional feature maps of the $conv7$ layer. First, we define the transformation matrix and the bias vector as $\mathbf{W}_s = [\mathbf{w}_{s1}, \cdots, \mathbf{w}_{sK}] \in \mathbb{R}^{M\times K}$ and $\mathbf{c}_s \in \mathbb{R}^K$ respectively, which are used to project $\mathbf{z}_{xy}$ into local-ware representations $\mathbf{l}$. For each latent patten $\mathbf{w}_{sk}$,[1] we define ω^k_{xy} as the spatial confident score over the occurrence of $\mathbf{w}_{sk}$ at the spatial location (x, y), which is given by,

$$\omega^k_{xy} = \mathbf{w}^T_{sk}\mathbf{z}_{xy} + c_{sk}, \tag{5}$$

where $\mathbf{w}_{sk}$ is the k^{th} column of $\mathbf{W}_s$, c_{sk} is the k^{th} element of $\mathbf{c}_s$.

To model the local spatial structure, we employ the softmax function to calculate the probability that the latent pattern $\mathbf{w}_{sk}$ appears at the location (x, y) as follow

$$\xi^k_{x,y} = \frac{\exp(\omega^k_{xy})}{\sum_{x',y'=1}^{X,Y}\exp(\omega^k_{x'y'})}. \tag{6}$$

Furthermore, we leverage spatial attention map to estimate the local awareness l_k carried by the latent pattern $\mathbf{w}_{sk}$ for identifying object-specific local regions. In particular, we define l_k as follow

$$l_k = \sum_{x,y}\pi_{xy}\xi^k_{xy}. \tag{7}$$

By concatenating l_k together, we can obtain the local-aware representations as $\mathbf{l} = [l_1, \cdots, l_K]$.

2) Global-Aware Representations: In addition to exploiting the local spatial structure, the ordinal representation is still expected to preserve the global semantic information from the visual descriptors of the $fc8$ layer. First, we define the feature representation extracted from the $fc8$ layer of CNN network as $\mathbf{v}$, computed by

$$\mathbf{v} = \psi_{\mathcal{C}}(q; \Omega_{\mathcal{C}}), \tag{8}$$

where $\mathbf{v} \in \mathbb{R}^M$, $\psi_{\mathcal{C}}$ defines the non-linear projection function for the CNN network and $\Omega_{\mathcal{C}} = \{\mathbf{W}^d_{\mathcal{C}}, \mathbf{c}^d_{\mathcal{C}}\}^{D_{\mathcal{C}}}_{d=1}$ defines a set of non-linear projection parameters with $D_{\mathcal{C}}$ being the depth of the CNN network. Let $\mathbf{W}_g = [\mathbf{w}_{g1}, \cdots, \mathbf{w}_{gK}] \in \mathbb{R}^{M\times K}$ and $\mathbf{c}_g \in \mathbb{R}^K$ be the transformation matrix and the bias vector, which project the feature representation of CNN network into global-aware representations. Similarly, we define the global awareness g_k as follow,

$$g_k = \mathbf{w}^T_{gk}\mathbf{v} + c_{gk}, \tag{9}$$

where $\mathbf{w}_{gk}$ is the k^{th} column of $\mathbf{W}_g$ and c_{gk} is the k^{th} element of $\mathbf{c}_g$. Intuitively, the global awareness g_k carried by the latent pattern $\mathbf{w}^1_{gk}$ measures the relative similarity to the image from the global perspective. By concatenating g_k together, we can obtain the global-aware representations as $\mathbf{g} = [g_1, \cdots, g_K]$.

E. Ranking-Based Hash Functions

The proposed DOH method targets to learn ranking-based hash functions by encoding the local spatial and global semantic information from deep networks. In this work, we develop intuitive ranking-based hash functions by encoding the comparative ordering of the local and global-aware representations. Figure 4 shows an example of a 4-dimensional feature space which is encoded to generate one hash bit. DOH firstly generates the representation $\mathbf{d}$ from the local-aware representation $\mathbf{l}$ and the global-aware representation $\mathbf{g}$. After that, a permutation on $\mathbf{d}$ is performed in descending order. The dimension (i.e., d_4 in Figure 4) which takes the maximum value will win the comparison and its index will be used as the hash code (i.e., 4). In the following, we give the mathematical formulations of ranking-based hashing functions in detail.

For $r = 1, \cdots, R$, let $\mathbf{l}^r$ and $\mathbf{g}^r$ define the local-aware and global-aware representations for the r^{th} slice of the hash layer for the FCN and CNN network respectively. Then, we can define $\mathbf{d}^r = [d^r_1, \cdots, d^r_K]$ as the local and global-aware representation for learning the r^{th} hash bit, which can be calculated as

$$\mathbf{d}^r = \mathbf{l}^r \odot \mathbf{g}^r \tag{10}$$

where $\odot$ denotes element-wise Hadamard product. Specifically, we define $f^r(\cdot)$ as the ranking-based hash function for generating the r^{th} hash bit, which can be calculated as

$$f^r(\mathbf{z}, \mathbf{v}; \mathbf{W}^r_s, \mathbf{c}^r_s, \mathbf{W}^r_g, \mathbf{c}^r_g) = arg\max_{\theta}\ \theta^T\mathbf{d}^r$$
$$s.t.\ \theta \in \{0, 1\}^K, \theta^T\mathbf{1} = 1, \tag{11}$$

[1] As $\mathbf{w}_{sk}$ and $\mathbf{w}_{gk}$ are unlabeled, we consider them as latent.

Algorithm 1 Deep Ordinal Hashing

Input: The input image q, code length R, network parameter sets $\Omega_{\mathcal{F}}$, $\Phi_{\mathcal{F}} = \{\mathbf{W}_s^r, \mathbf{c}_s^r\}_{r=1}^R$, $\Omega_{\mathcal{C}}$, $\Phi_{\mathcal{C}} = \{\mathbf{W}_g^r, \mathbf{c}_g^r\}_{r=1}^R$.
Output: Hash code $\mathbf{b} = [b_1, \cdots, b_r, \cdots, b_R]$.
Compute $\mathbf{z}_{xy} = \psi_{\mathcal{F}}(q; \Omega_{\mathcal{F}})$ and $\mathbf{v} = \psi_{\mathcal{C}}(q; \Omega_{\mathcal{C}})$ by forward propagation.
Calculate the spatial attention map π according to Eq. (3).
for $r = 1, \cdots, R$ **do**
Calculate the local-ware representation $\mathbf{l}^r$ and the global-ware representation $\mathbf{g}^r$ according to Eq. (5)-(7) and Eq. (9), respectively.
Calculate the local and global-aware representation $\mathbf{d}^r = [d_1^r, \cdots, d_K^r]$ according to Eq. (10).
$b_r = \hat{k} \leftarrow \arg\max_{1 \le k \le K} d_k^r$.
end for

where $\mathbf{W}_s^r = [\mathbf{w}_{s1}^r, \cdots, \mathbf{w}_{sK}^r]$ and $\mathbf{c}_s^r = [c_{s1}^r, \cdots, c_{sK}^r]$ define the transformation matrix and the bias vector for the r^{th} slice of the hash layer of FCN network, $\mathbf{W}_g^r = [\mathbf{w}_{g1}^r, \cdots, \mathbf{w}_{gK}^r]$ and $\mathbf{c}_g^r = [c_{g1}^r, \cdots, c_{gK}^r]$ define the transformation matrix and the bias vector for the r^{th} slice of the hash layer of CNN network, $\mathbf{1}$ is a vector with each element being 1. Meanwhile, the constraints in Eq. (11) act as an 1-of-K indicator of the rank ordering of the input representation $\mathbf{d}^r$. **Algorithm 1** summarizes the entire DOH hashing procedure to produce a code sequence $\mathbf{b}$.

It is worth to note that the $argmax$ term in Eq. (11) is non-convex and highly discontinuous, thus making it hard to optimize. To make it tractable, the softmax function is employed to approximate the vectorized representation of the hash function defined in Eq. (11). Therefore, we reformulated Eq. (11) and define the ordinal representation $\mathbf{h}^r$ as follow

$$\mathbf{h}^r(\mathbf{z}, \mathbf{v}; \mathbf{W}_s^r, \mathbf{c}_s^r, \mathbf{W}_g^r, \mathbf{c}_g^r) = \text{softmax}(\mathbf{d}^r), \tag{12}$$

where $\mathbf{h}^r = [h_1^r, \cdots, h_K^r]$ is the probabilistic approximation of the vectorized representation of the hash function $f^r(\cdot)$. In fact, the entry h_k^r of $\mathbf{h}^r$ represents the probability that the k^{th} dimension takes the maximum value in $\mathbf{d}^r$. Specifically, the probability h_k^r can be calculated as

$$h_k^r = \frac{\exp(d_k^r)}{\sum_{k'=1}^{K} \exp(d_{k'}^r)}, \quad \text{for } k = 1, \cdots, K, \tag{13}$$

where h_k^r can be interpreted as the probability that both of the latent patterns $\mathbf{w}_s^r$ and $\mathbf{w}_g^r$ contain the most discriminative information.

Suppose an image pair (q_i, q_j) with their corresponding similarity label $s_{ij} \in \{0, 1\}$ is given, we try to learn a set of hash functions which makes the hash codes of the similar pairs close but dissimilar pairs apart. Let $\mathbf{b}(i)$ and $\mathbf{b}(j)$ be the code sequence for the image q_i and q_j respectively. Formally, ε_{ij} is defined to measure the similarity of $\mathbf{b}(i)$ and $\mathbf{b}(j)$ as follow

$$\begin{aligned}\varepsilon_{ij} &= \sum_{r=1}^{R} \varphi_r^{ij} \Big/ \sum_{r=1}^{R}\sum_{k=1}^{K} h_k^r \\ &= \frac{1}{R}\sum_{r=1}^{R} \varphi_r^{ij},\end{aligned} \tag{14}$$

where

$$\begin{aligned}\varphi_r^{ij} &\equiv P(b_r(i) = b_r(j) | \mathbf{W}_s^r, \mathbf{c}_s^r, \mathbf{W}_g^r, \mathbf{c}_g^r) \\ &= \sum_{k=1}^{K} P(b_r(i) = k | \mathbf{W}_s^r, \mathbf{c}_s^r, \mathbf{W}_g^r, \mathbf{c}_g^r) \\ &\quad P(b_r(j) = k | \mathbf{W}_s^r, \mathbf{c}_s^r, \mathbf{W}_g^r, \mathbf{c}_g^r) \\ &= \sum_{k=1}^{K} h_k^r(i) h_k^r(j) = (\mathbf{h}^r(i))^T \mathbf{h}^r(j).\end{aligned} \tag{15}$$

In Eq. (15), $b_r(i)$ and $b_r(j)$ are the r^{th} bit of $\mathbf{b}(i)$ and $\mathbf{b}(j)$ respectively, $\mathbf{h}^r(i)$ and $\mathbf{h}^r(j)$ are the ordinal representation for generating the r^{th} hash bit for the image q_i and q_j respectively. Besides, φ_r^{ij} calculates the probability that the k^{th} dimension of the feature space is selected as the winner for generating the r^{th} hash bit of both $\mathbf{b}(i)$ and $\mathbf{b}(j)$.

Actually, the larger value of ε_{ij} indicates that the code sequence $\mathbf{b}(i)$ and $\mathbf{b}(j)$ are more similar with each other. But conversely, the smaller value of ε_{ij} indicates less similar of two code sequence. That is, if $s_{ij} = 1$, ε_{ij} should be pushed towards 1, otherwise ε_{ij} should be pushed to 0. Intuitively, our goal is to maximize ε_{ij} when $s_{ij} = 1$ and minimize ε_{ij} when $s_{ij} = 0$. This inspires us to employ the Euclidian loss function to define the following objective function

$$\ell_{ij}(q_i, q_j, s_{ij}) = \frac{1}{2}(\varepsilon_{ij} - s_{ij})^2. \tag{16}$$

Let $\Gamma = \{(q_i, q_j), s_{ij}\}_{i,j=1}^{N_p}$ be the training set. We define the overall loss function over the training set Γ as follow

$$\mathcal{L}(\Gamma; \Omega_{\mathcal{F}}, \Phi_{\mathcal{F}}, \Omega_{\mathcal{C}}, \Phi_{\mathcal{C}}) = \frac{1}{N_p} \sum_{s_{ij} \in \mathbf{S}} \ell_{ij}(q_i, q_j, s_{ij}), \tag{17}$$

where $\Phi_{\mathcal{F}} = \{\mathbf{W}_s^r, \mathbf{c}_s^r\}_{r=1}^R$ and $\Phi_{\mathcal{C}} = \{\mathbf{W}_g^r, \mathbf{c}_g^r\}_{r=1}^R$ denotes all R transformation matrixes and bias vectors for the hash layer of the FCN and CNN network respectively. Finally, the proposed framework can be formulated as

$$\min_{\Omega_{\mathcal{F}}, \Phi_{\mathcal{F}}, \Omega_{\mathcal{C}}, \Phi_{\mathcal{C}}} \mathcal{L}(\Gamma; \Omega_{\mathcal{F}}, \Phi_{\mathcal{F}}, \Omega_{\mathcal{C}}, \Phi_{\mathcal{C}}). \tag{18}$$

IV. Optimization

The proposed DOH algorithm involves four sets of variables, such as the parameter sets $\Omega_{\mathcal{F}}$ and $\Phi_{\mathcal{F}}$ for the FCN network and the parameter sets $\Omega_{\mathcal{C}}$ and $\Phi_{\mathcal{C}}$ for the CNN network. As it is non-convex to simultaneously learn Ω_* and Φ_*, to make it tractable, we adopt an alternating optimization approach to update one variable by fixing the rest variables, where $*$ is a placeholder for $\mathcal{F}$ and $\mathcal{C}$. The proposed optimization algorithm is summarized in **Algorithm 2**. In the following, we discuss the derivatives of $\mathcal{L}$ in detail.

Algorithm 2 The Learning Algorithm for DOH

Input: Training set Γ.
Output: Network parameter sets $\Omega_{\mathcal{F}}$, $\Phi_{\mathcal{F}}$, $\Omega_{\mathcal{C}}$, $\Phi_{\mathcal{C}}$.
Initialization
Initialize Network parameter sets $\Omega_{\mathcal{F}}$, $\Phi_{\mathcal{F}}$, $\Omega_{\mathcal{C}}$, $\Phi_{\mathcal{C}}$, iteration number N_{iter} and mini-batch size N_{batch}.
repeat:
for $iter = 1, \cdots, N_{iter}$ **do**
Randomly select N_{batch} image pairs from Γ.
for $r = 1, \cdots, R$ **do**
For each selected image pair, compute $\mathbf{h}^r$ by forward propagation according to Eq. (13).
Calculate the derivatives $\frac{\partial \mathcal{L}}{\partial \mathbf{W}_s^r}$, $\frac{\partial \mathcal{L}}{\partial \mathbf{c}_s^r}$, $\frac{\partial \mathcal{L}}{\partial \mathbf{W}_g^r}$ and $\frac{\partial \mathcal{L}}{\partial \mathbf{c}_g^r}$ according to Eq. (20), (21), (22), (23), (24), (25), (26), (27).
Update the parameters $\mathbf{W}_s^r$, $\mathbf{c}_s^r$, $\mathbf{W}_g^r$, $\mathbf{c}_g^r$ using the BP algorithm.
end for
Update parameter sets $\Omega_{\mathcal{F}}$ and $\Omega_{\mathcal{C}}$ for the FCN and CNN network, respectively.
end for
until a fixed number of iteration.

Update $\Phi_{\mathcal{F}}$. We first solve $\Phi_{\mathcal{F}}$ with the parameter sets $\Phi_{\mathcal{C}}$, $\Omega_{\mathcal{C}}$ and $\Omega_{\mathcal{F}}$ fixed. As $\mathbf{W}_s^r$ and $\mathbf{c}_s^r$ are independent on $\{\mathbf{W}_s^{r'}, \mathbf{c}_s^{r'}\}_{r' \neq r}$, we can optimize $\Phi_{\mathcal{F}}$ by decomposing it to R independent subproblems, where each subproblem can be formulated as the following optimization problem:

$$\min_{\mathbf{W}_s^r, \mathbf{c}_s^r} \mathcal{L}(\Gamma; \Omega_{\mathcal{F}}, \Phi_{\mathcal{F}}, \Omega_{\mathcal{C}}, \Phi_{\mathcal{C}}). \tag{19}$$

The optimization problem in Eq. (19) can be efficient solved by using the stochastic gradient descent (SGD) with the back-propagation (BP) algorithm. The gradient of the objective function in Eq. (19) w.r.t. $\mathbf{W}_s^r$ and $\mathbf{c}_s^r$ can be computed as follow

$$\frac{\partial \mathcal{L}}{\partial \mathbf{W}_s^r} = \frac{1}{N_p} \sum_{s_{ij} \in \mathbf{S}} \frac{\partial \ell_{ij}}{\partial \mathbf{W}_s^r}, \tag{20}$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{c}_s^r} = \frac{1}{N_p} \sum_{s_{ij} \in \mathbf{S}} \frac{\partial \ell_{ij}}{\partial \mathbf{c}_s^r}. \tag{21}$$

Specifically, the gradient $\frac{\partial \ell_{ij}}{\partial \mathbf{W}_s^r}$ in Eq. (20) can be calculated by using the chain rule of the derivatives on ℓ_{ij} in Eq. (16), which can be calculated as follow

$$\frac{\partial \ell_{ij}}{\partial \mathbf{W}_s^r} = (\varepsilon_{ij} - s_{ij}) \frac{\partial \varepsilon_{ij}}{\partial \mathbf{W}_s^r}, \tag{22}$$

$$\frac{\partial \varepsilon_{ij}}{\partial \mathbf{W}_s^r} = \frac{1}{R} \frac{\partial \varphi_r^{ij}}{\partial \mathbf{W}_s^r}, \tag{23}$$

$$\begin{aligned} \frac{\partial \varphi_r^{ij}}{\partial \mathbf{W}_s^r} = {} & (\mathbf{h}^r(i) \odot \mathbf{h}^r(j) - (\mathbf{h}^r(i))^T \mathbf{h}^r(j) \mathbf{h}^r(i)) \frac{\partial \mathbf{d}^r(i)}{\partial \mathbf{W}_s^r} \\ & + (\mathbf{h}^r(j) \odot \mathbf{h}^r(i) - (\mathbf{h}^r(j))^T \mathbf{h}^r(i) \mathbf{h}^r(j)) \frac{\partial \mathbf{d}^r(j)}{\partial \mathbf{W}_s^r}, \end{aligned} \tag{24}$$

$$\frac{\partial \mathbf{d}^r}{\partial \mathbf{W}_s^r} = [\frac{\partial d_1^r}{\partial \mathbf{w}_{s1}^r}, \cdots, \frac{\partial d_k^r}{\partial \mathbf{w}_{sk}^r}, \cdots, \frac{\partial d_K^r}{\partial \mathbf{w}_{sK}^r}], \tag{25}$$

$$\frac{\partial d_k^r}{\partial \mathbf{w}_{sk}^r} = l_k^r g_k^r [\mathbf{z}_{xy}^T - \sum_{x'y'} \xi_{x'y'}^{kr} \mathbf{z}_{x'y'}^T], \tag{26}$$

where l_k^r and g_k^r denotes the k^{th} entry of $\mathbf{l}^r$ and $\mathbf{g}^r$ respectively and $\xi_{x'y'}^{kr}$ can be calculate by Eq. (6). As the gradient $\frac{\partial \ell_{ij}}{\partial \mathbf{c}_s^r}$ is similar with $\frac{\partial \ell_{ij}}{\partial \mathbf{W}_s^r}$, we do not elaborate on its solution here.

Update $\Phi_{\mathcal{C}}$. Actually, the solution of $\Phi_{\mathcal{C}}$ is similar with that of $\Phi_{\mathcal{F}}$ except for the terms in Eq. (26). Therefore, we only calculate the derivatives of $\frac{\partial d_k^r}{\partial \mathbf{w}_{gk}^r}$ as follow

$$\frac{\partial d_k^r}{\partial \mathbf{w}_{gk}^r} = l_k^r \mathbf{v}^T, \tag{27}$$

where l_k^r is the k^{th} entry of $\mathbf{l}^r$.

Update Ω_*. The parameters $\mathbf{W}_*^d$ and $\mathbf{c}_*^{\mathbf{d}}$ can be automatically updated by applying SGD with BP algorithm in Caffe [55].

V. Experiments

In this section, we conduct extensive experiments to verify the effectiveness of DOH on three widely-used image retrieval datasets including MIRFlickr25k [56], CIFAR-10 [57] and NUS-WIDE [58].

A. Datasets

MIRFlickr25k[2] contains 25,000 images collected from Flickr website. In this dataset, each image is annotated with one or more of the 24 semantic labels. We randomly select 908 images from this dataset as queries and the rest are used to form the database, from which we randomly sample 5000 images to form the training set.

CIFAR-10[3] includes 60,000 real-world tiny images belonging to 10 classes, where each category has 6,000 images. We randomly select 100 images per class as the queries, 500 images per class as the training set and the rest forms the database.

NUS-WIDE[4] consists of 269,648 images crawled from Flickr websit, where each image is associated with one or more of the 81 sematic concepts (labels). For this dataset, we manually select 195,834 images belonging to the top 21 most-frequent concepts. We randomly select 100 images per class as the queries and the rest images form the databese, from which we select 500 images per class as the training set.

B. Baselines and Evaluation Metrics

We have compared the proposed DOH method with several state-of-the-art image hashing methods, including seven non-deep hashing methods (i.e., LSH [1], WTA [24], ITQ [4], KSH [41], SDH [42], FastH [43] and LSRH [25]) and three deep hashing methods (i.e., DNNH [48], DSH [12] and

[2] http://press.liacs.nl/mirflickr/
[3] http://www.cs.toronto.edu/kriz/cifar.html
[4] http://lms.comp.nus.edu.sg/research/NUS-WIDE.htm

TABLE III
MAP RESULTS OF ALL METHODS WITH RESPECT TO DIFFERENT CODE LENGTHS ON THE THREE DATASETS

Method	MIRFLICKR25K				CIFAR-10				NUS-WIDE			
	8 bits	16 bits	24 bits	32 bits	8 bits	16 bits	24 bits	32 bits	8 bits	16 bits	24 bits	32 bits
LSH	0.5721	0.5845	0.5826	0.5863	0.1087	0.1137	0.1172	0.1331	0.3845	0.4047	0.4090	0.4092
WTA	0.5682	0.5766	0.5812	0.5873	0.1996	0.2388	0.2868	0.2989	0.3852	0.3953	0.4031	0.4251
ITQ	0.6448	0.6472	0.6515	0.6517	0.1767	0.1839	0.1858	0.1902	0.4725	0.4835	0.4880	0.4943
KSH	0.6843	0.6968	0.7001	0.7035	0.3096	0.3599	0.3766	0.3892	0.4985	0.5084	0.5160	0.5236
SDH	0.7268	0.7292	0.7347	0.7416	0.3371	0.4779	0.5142	0.5200	0.5221	0.5394	0.5415	0.5488
FastH	0.7453	0.7748	0.7909	0.7970	0.4253	0.4909	0.5151	0.5395	0.5339	0.5570	0.5732	0.5822
LSRH	0.7422	0.7808	0.7846	0.7950	0.3197	0.4497	0.4708	0.4870	0.5431	0.5680	0.5760	0.5862
DNNH	0.7498	0.7622	0.7745	0.7670	0.5803	0.5959	0.6329	0.6358	0.6472	0.6598	0.6747	0.6784
DSH	0.7127	0.7291	0.7304	0.7353	0.7470	0.7623	0.7773	0.8019	0.6160	0.6370	0.6397	0.6384
SSDH	0.7646	0.7761	0.7931	0.7958	0.7638	0.7735	0.8037	0.8182	0.6408	0.6691	0.6811	0.6817
DOH	**0.8607**	**0.8739**	**0.8838**	**0.8863**	**0.8624**	**0.8686**	**0.8732**	**0.8702**	**0.7551**	**0.7883**	**0.7916**	**0.7997**

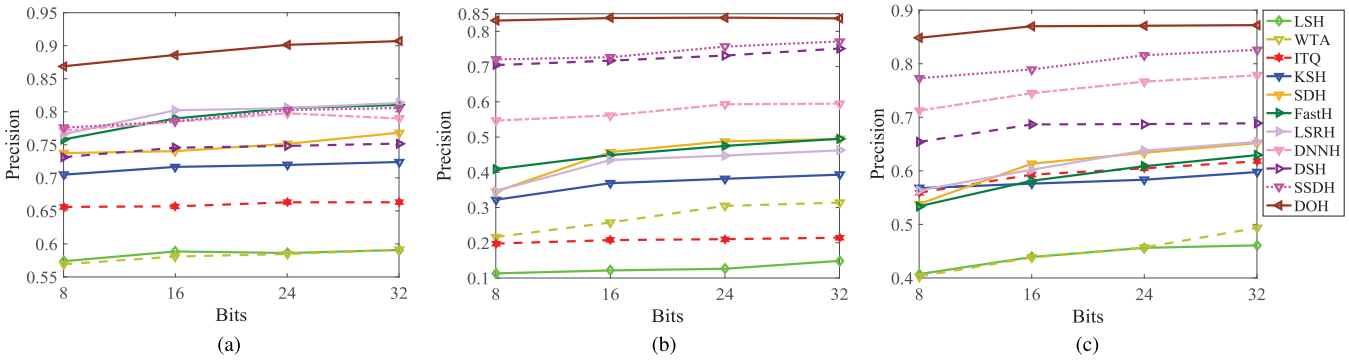


Fig. 5. P@5000 curves with respect to different code lengths for all the methods on the three datasets. (a) MIRFlickr25k (b) CIFAR-10 (c) NUS-WIDE.

SSDH [53]). We have briefly introduced those hashing methods in Section II. For non-deep methods, we extract the 4096-dimensional feature of $fc7$ layer using the Alexnet network. For all deep methods, we adopt the same network (i.e., Alexnet) for fair comparison. Except for DNNH, the source codes of the baselines are kindly provided by their authors. Specifically, we implement DNNH method with the open-source Caffe [55] framework. The parameters for all the compared methods are selected by their default ones. In addition, we evaluate the retrieval performance using three widely-used metrics: mean Average Precision (mAP), top-N Precision (P@N) and Precision-Recall curves (PR).

C. Experimental Settings

We implement the proposed DOH method with the open-source Caffe [55] framework on a NVIDIA K20 GPU server. We initialize the network with "Xavier" initialization except for these layers including $conv1$ to $conv5$ and $fc6$ to $fc7$ that are copied from Alexnet. As the remaining layers are trained from scratch, we set the learning rates as 100 times bigger than that of other layers for fc-h, as well as 10 times bigger for $conv6$, $conv7$, $fc8$ and fc-c. Our network is trained by using the mini-batch stochastic gradient descent with the learning rate setting as 10^{-5}. In all experiments, we fix the size of the min-batch as 64. As DOH involves one hyper-parameter, the dimension K of the feature space, we use linear search in $\{2^1, 2^2, 2^3, 2^4, 2^5\}$ to select K. Specifically, we set K as 2^2 for MIRFlickr25K and CIFAR-10 datasets, and 2^3 for NUS-WIDE dataset respectively.

D. Experimental Results

1) Comparison With the Baselines: We report mAP values for DOH and all the compared baselines in Tabel III. We can observe that DOH significantly outperforms all the compared baselines on different datasets with respect to different code lengths. In fact, compared to the best non-deep method (FastHash), DOH gains average performance increasements of 10.03%, 37.56% and 22.13% for mAP values on MIRFlickr25K, CIFAR-10 and NUS-WIDE datasets respectively. Furthermore, compared to SSDH, the best deep hashing method, DOH can still achieve average performance improvements of 9.42%, 7.85% and 11.48% in terms of mAP values on the three datasets respectively. Such significantly improvements demonstrates the effectiveness of the proposed method.

In addition to mAP values, we also report the performance of P@5000 in terms of different code lengths and the precision curves of the 16-bit hash code with respect to different numbers of top returned samples in Figure 5 and Figure 7, respectively. When we compare DOH with FastH for the retrieval performance in terms of P@5000, the average performance gap between them are 13.24%, 37.58% and 22.03% on the three datasets respectively. Similar, compared to SSDH, DOH grains 9.30%, 7.87% and 11.37% performance improvements in average for P@5000 values on three datasets respectively. From Figure 7, we can observe that DOH also consistently outperforms all the compared methods on different datasets.

We plot the Precision-Recall (PR) curves of 16-bit hash code on three different datasets in Figure 6. Specifically, the PR

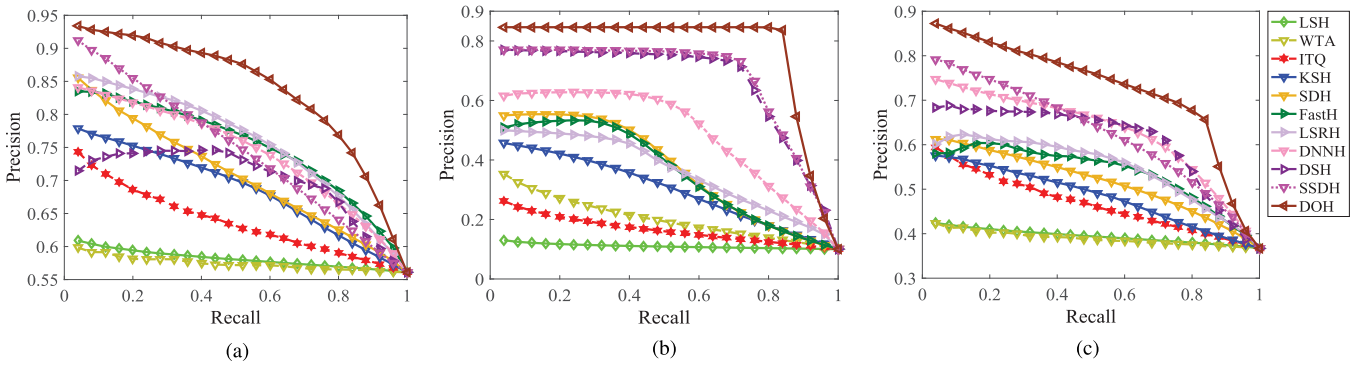


Fig. 6. Precision-recall curves with respect to 16-bit hash code for different methods on the three datasets. (a) MIRFlickr25k (b) CIFAR-10 (c) NUS-WIDE.

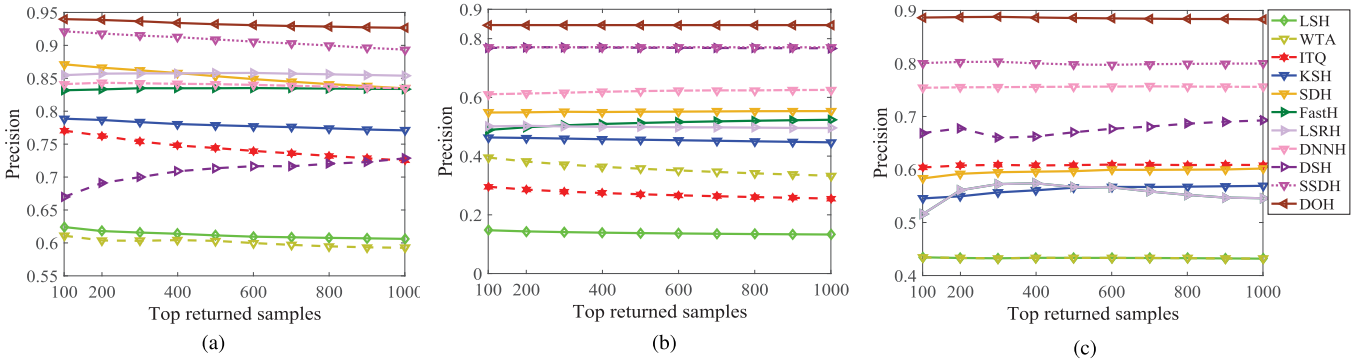


Fig. 7. Precision curves of 16-bit hash code with respect to different numbers of top returned samples on the three datasets. (a) MIRFlickr25k (b) CIFAR-10 (c) NUS-WIDE.

TABLE IV

MAP RESULTS OF DOH AND ITS VARIANTS WITH RESPECT TO DIFFERENT CODE LENGTHS ON THE THREE DATASETS

Method	MIRFLICKR25K				CIFAR-10				NUS-WIDE			
	8 bits	16 bits	24 bits	32 bits	8 bits	16 bits	24 bits	32 bits	8 bits	16 bits	24 bits	32 bits
DOH-F	0.7980	0.8325	0.8337	0.8385	0.8190	0.8267	0.8364	0.8353	0.7014	0.7289	0.7350	0.7401
DOH-C	0.8193	0.8627	0.8735	0.8783	0.8452	0.8585	0.8661	0.8669	0.7382	0.7701	0.7793	0.7870
DOH	**0.8607**	**0.8739**	**0.8838**	**0.8863**	**0.8624**	**0.8686**	**0.8732**	**0.8702**	**0.7551**	**0.7883**	**0.7916**	**0.7997**

curve indicates the overall performance, and the larger value of the area under the PR curve reflects better performance. As can be seen in Figure 6, DOH can achieve superior performance compared with the baselines on all the datasets. More specifically, DOH can yield higher precision at the lower recall points, which is satisfying for practical image retrieval system.

Overall, observed from the experimental results, DOH substantially outperforms all the compared baselines on different datasets in terms of mAP, P@5000 and PR curves. Such significant improvements verify the superiority of the proposed hashing method. We highlight three advantages of DOH in the following. First, DOH can preserve the local discriminativity learned by the spatial attention model with FCN to achieve effective image matching, while the other deep baselines only exploit the global semantic information with CNN. Second, the ranking-based hash functions learned with the deep neural networks can produce discriminative hash codes by exploiting the relative ranking structure of the feature space. Finally, by jointly encoding the local spatial and global semantic information, DOH can capture useful ranking correlation structure to better preserve the similarities.

2) Comparison With Two Variants: In the proposed deep network, we construct a subnetwork followed after both of FCN and CNN to learn the unified ordinal representations $\mathbf{h}$ which are utilized to approximate the vectorized representation of ranking-based hash functions. A possible alternative to this subnetwork is that two independent fully-connected layers are used to learn different ordinal representations for FCN and CNN respectively. Therefore, the ordinal representation learned in this manner can only gain the knowledge from either the local spatial information or the global semantic information. Specifically, we investigate two variants of DOH: (1) **DOH-F**, variant only using the FCN network to learn the ordinal representation that is local-aware; (2) **DOH-C**, variant only using the CNN network to learn the ordinal representation that is global-aware.

The retrieval performance in terms of mAP is illustrated in Table IV. We find that, by exploiting the local spatial and global semantic information simultaneously, DOH can

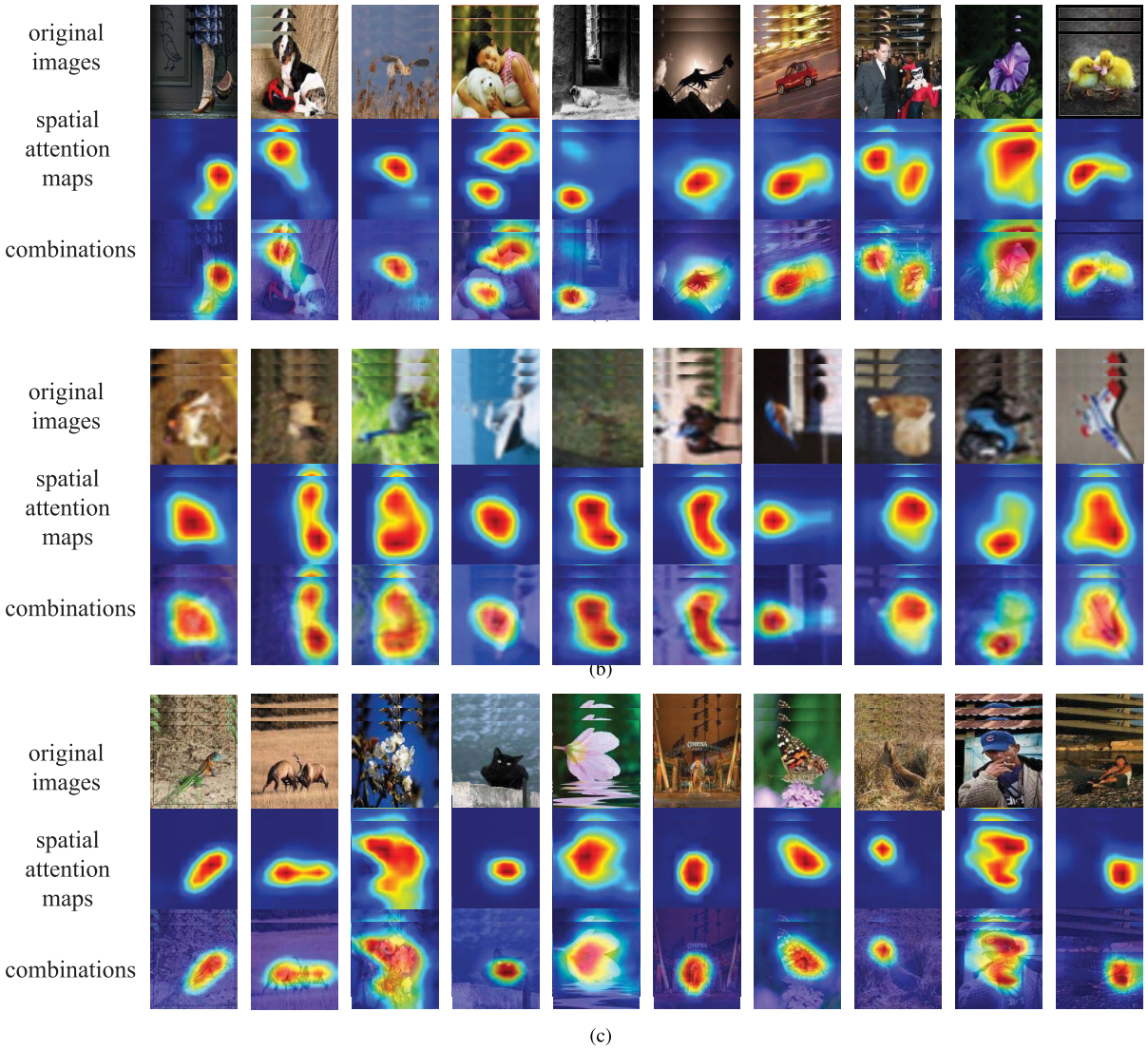


Fig. 8. Some visual examples of spatial attention maps for (a) MIRFlickr25k, (b) CIFAR-10 and (c) NUS-WIDE datasets respectively. The first line shows the original images. The middle line shows the spatial attention maps. And the combination of the original image and spatial attention map is shown in the bottom.

TABLE V

P@5000 RESULTS OF DOH AND ITS VARIANTS WITH RESPECT TO DIFFERENT CODE LENGTHS ON THE THREE DATASETS

Method	MIRFLICKR25K				CIFAR-10				NUS-WIDE			
	8 bits	16 bits	24 bits	32 bits	8 bits	16 bits	24 bits	32 bits	8 bits	16 bits	24 bits	32 bits
DOH-F	0.8188	0.8474	0.8540	0.8585	0.7861	0.7821	0.7925	0.7888	0.7909	0.8173	0.8220	0.8251
DOH-C	0.8401	0.8735	0.8816	0.8820	0.8117	0.8215	0.8259	0.8276	0.8200	0.8343	0.8433	0.8484
DOH	**0.8738**	**0.8852**	**0.8943**	**0.8959**	**0.8322**	**0.8355**	**0.8367**	**0.8325**	**0.8434**	**0.8625**	**0.8641**	**0.8649**

consistently outperform the two variants on the three datasets. For example, compared to DOH-F, DOH can gain average performance improvements of 5.05%, 3.93% and 5.73% on MIRFlickr25K, CIFAR-10 and NUS-WIDE, respectively. Compared to DOH-C, the average performance gap is 1.77%, 0.95% and 1.50% on the three datasets.

In addition to mAP, the performance of P@5000 is shown in Table V. Overall, the proposed method substantially outperforms the two variants. In detail, compared to DOH-F, DOH achieves the average performance improvements of 4.26%, 4.69% and 4.49% for the three datasets. Similarly, compared to DOH-C, DOH gains the average performance improvements of 1.80%, 1.26% and 2.22% on MIRFlickr25K, CIFAR-10 and NUS-WIDE, respectively.

Another interesting finding is that the retrieval performance of DOH and its two variants in terms of mAP and P@5000 substantially outperforms SSDH, the best deep hashing method that adopts the binary quantization function. Actually, DOH and its two variants are similar in the sense that they all involve to exploit rank correlation spaces with

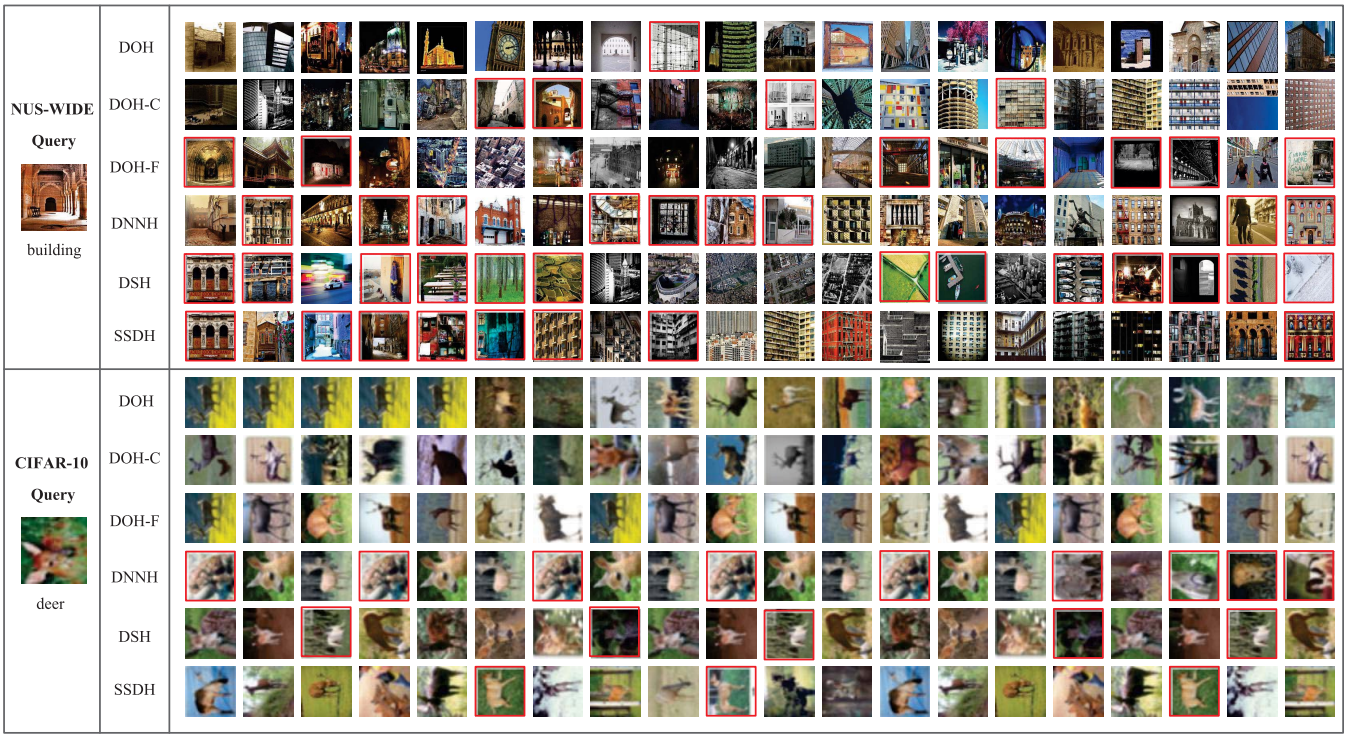


Fig. 9. Examples of retrieval results on the NUS-WIDE and CIFAR-10 datasets. The left column shows the query images. The middle column shows different methods, and their corresponding retrieval results are shown in the right column. The code length of all the methods is set as 32 bits and the top 20 results obtained using Hamming ranking are shown. The red rectangles indicate the wrong retrieval results.

deep networks. Therefore, the ranking structure leveraged by the proposed hashing method is useful to generate discriminative hash code for yielding superior retrieval performance.

Some visual examples of spatial attention maps are shown in Figure 8. Observed from Figure 8, the proposed spatial attention model can learn the discriminative local regions from the original images, which demonstrates the effectiveness of the proposed spatial attention maps. We also show some retrieval results of the top 20 returned samples in terms of Hamming ranking on the NUS-WIDE and CIFAR-10 datasets in Figure 9. We can observe that DOH and its two variants can achieve much better retrieval results than the deep baselines, which indicates the effectiveness of exploiting the ranking structure with the deep network. Specifically, we note that DOH can yield better retrieval results than DOH-F and DOH-C on NUS-WIDE dataset. This indicates that the ranking-based hash function learned by jointly exploiting the local spatial and global semantic information can better preserve the similarity of the image pair.

3) Effect of Parameter: The proposed DOH involves one parameter, the dimension K of the feature space. In order to verify the sensitivity, we conduct experiments to analyze the influence on different datasets by using linear search in $\{2^1, 2^2, 2^3, 2^4, 2^5\}$. Specifically, we set the code length as 60 as it is the least common multiplier of $\log_2 K$. The retrieval performance in terms of mAP and P@5000 are shown in Figure 10. For MIRFlickr25k dataset, there is very small performance influence under different settings of K. However on CIFAR-10 and NUS-WIDE, the performance slightly

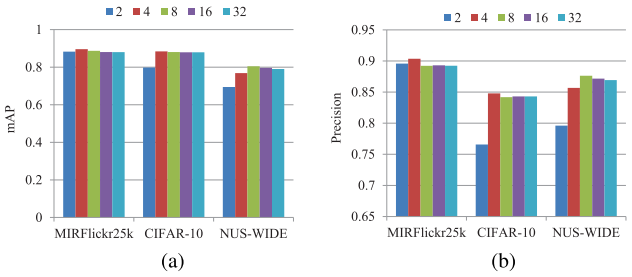


Fig. 10. The mAP and P@5000 with respect to different K for 60-bits hash code on the three datasets. (a) mAP (b) P@5000.

decreases when setting K as 2. As illustrated in Figure 10, we can set K as 4 for MIRFlickr25k and CIFAR-10, and K as 8 for NUS-WIDE, respectively.

VI. Conclusion

In this work, we propose a novel deep hashing method (DOH) to learn the ranking-based hash functions by exploiting the rank correlation space from both the local and global views. Specifically, a two-stream network is designed to learn the unified ordinal representation for approximating the vectorized representation of ranking-based hashing functions by exploiting the local spatial information from the FCN network and the global semantic information from the CNN network simultaneously. More specifically, for the FCN network, an effective spatial attention model is proposed to capture the local discriminativity by learning well-specified

locations closely related to target objects. By jointly leveraging such local information learned with the spatial attention model and the global semantic information, DOH can lean high-quality ordinal representation to produce discriminative hash codes, thus achieving superior performance for image retrieval. Extensive experimental results on three datasets verify the superiority of the proposed DOH method in learning discriminative hash codes for image retrieval.

REFERENCES

[1] A. Gionis, P. Indyk, and R. Motwani, "Similarity search in high dimensions via hashing," in *Proc. VLDB Conf.*, 1999, pp. 518–529.

[2] L. Jin, K. Li, H. Hu, G.-J. Qi, and J. Tang, "Semantic neighbor graph hashing for multimodal retrieval," *IEEE Trans. Image Process.*, vol. 27, no. 3, pp. 1405–1417, Mar. 2018.

[3] A. Z. Broder, M. Charikar, A. M. Frieze, and M. Mitzenmacher, "Min-wise independent permutations," *J. Comput. Syst. Sci.*, vol. 60, no. 3, pp. 630–659, Jun. 2000.

[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] J. Tang and Z. Li, "Weakly supervised multimodal hashing for scalable social image retrieval," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 10, pp. 2730–2741, Oct. 2018.

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

[7] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *Proc. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1717–1724.

[8] J. Lu, J. Hu, and J. Zhou, "Deep metric learning for visual understanding: An overview of recent advances," *IEEE Signal Process. Mag.*, vol. 34, no. 6, pp. 76–84, Nov. 2017.

[9] L. Chen *et al.*, "SCA-CNN: Spatial and channel-wise attention in convolutional networks for image captioning," in *Proc. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 6298–6306.

[10] Q. You, H. Jin, Z. Wang, C. Fang, and J. Luo, "Image captioning with semantic attention," in *Proc. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 4651–4659.

[11] Z. Yang, Y. Yuan, Y. Wu, W. W. Cohen, and R. R. Salakhutdinov, "Review networks for caption generation," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 2361–2369.

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

[13] J. Song, T. He, L. Gao, X. Xu, and H. T. Shen. (2017). "Deep region hashing for efficient large-scale instance search from images." [Online]. Available: https://arxiv.org/abs/1701.07901

[14] J. Tang, Z. Li, and X. Zhu, "Supervised deep hashing for scalable face image retrieval," *Pattern Recognit.*, vol. 75, pp. 25–32, Mar. 2018.

[15] V. E. Liong, J. Lu, G. Wang, P. Moulin, and J. Zhou, "Deep hashing for compact binary codes learning," in *Proc. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 2475–2483.

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

[17] Y. Cao, M. Long, J. Wang, and S. Liu, "Deep visual-semantic quantization for efficient image retrieval," in *Proc. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 1328–1337.

[18] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proc. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 2921–2929.

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

[20] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Is object localization for free?—Weakly-supervised learning with convolutional neural networks," in *Proc. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 685–694.

[21] D. Li, J.-B. Huang, Y. Li, S. Wang, and M.-H. Yang, "Weakly supervised object localization with progressive domain adaptation," in *Proc. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 3512–3520.

[22] A. Diba, V. Sharma, A. Pazandeh, H. Pirsiavash, and L. Van Gool, "Weakly supervised cascaded convolutional networks," in *Proc. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 914–922.

[23] M. Melucci, "On rank correlation in information retrieval evaluation," *ACM SIGIR Forum*, vol. 41, no. 1, pp. 18–33, 2007.

[24] J. Yagnik, D. Strelow, D. A. Ross, and R.-S. Lin, "The power of comparative reasoning," in *Proc. IEEE Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2431–2438.

[25] K. Li, G.-J. Qi, J. Ye, and K. A. Hua, "Linear subspace ranking hashing for cross-modal retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 9, pp. 1825–1838, Sep. 2017.

[26] J. Lu, C. Xiong, D. Parikh, and R. Socher, "Knowing when to look: Adaptive attention via a visual sentinel for image captioning," in *Proc. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 375–383.

[27] Y. Yu, J. Choi, Y. Kim, K. Yoo, S.-H. Lee, and G. Kim, "Supervising neural attention models for video captioning by human gaze data," in *Proc. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 490–498.

[28] X. Xu, X. Chen, C. Liu, A. Rohrbach, T. Darrell, and D. Song, "Fooling vision and language models despite localization and attention mechanism," in *Proc. Comput. Vis. Pattern Recognit.*, Jul. 2018, pp. 4951–4961.

[29] C. Cao *et al.*, "Look and think twice: Capturing top-down visual attention with feedback convolutional neural networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 2956–2964.

[30] Y. Zhu, C. Zhao, H. Guo, J. Wang, X. Zhao, and H. Lu, "Attention couplenet: Fully convolutional attention coupling network for object detection," *IEEE Trans. Image Process.*, vol. 28, no. 1, pp. 113–126, Jan. 2019, doi: 10.1109/TIP.2018.2865280.

[31] X. Wang, A. Shrivastava, and A. Gupta, "A-fast-RCNN: Hard positive generation via adversary for object detection," in *Proc. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 2606–2615.

[32] K. Xu *et al.*, "Show, attend and tell: Neural image caption generation with visual attention," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 2048–2057.

[33] H. Xu and K. Saenko, "Ask, attend and answer: Exploring question-guided spatial attention for visual question answering," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 451–466.

[34] T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2015, pp. 1412–1421.

[35] Z. Yang, X. He, J. Gao, L. Deng, and A. Smola, "Stacked attention networks for image question answering," in *Proc. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 21–29.

[36] H. Liu, J. Feng, M. Qi, J. Jiang, and S. Yan, "End-to-end comparative attention networks for person re-identification," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3492–3506, Jul. 2017.

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

[38] D. Zhang, J. Wang, D. Cai, and J. Lu, "Self-taught hashing for fast similarity search," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2010, pp. 18–25.

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

[40] 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.

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

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

[43] G. Lin, C. Shen, and A. van den Hengel, "Supervised hashing using graph cuts and boosted decision trees," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 11, pp. 2317–2331, Nov. 2015.

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

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

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

[47] K. Li, G.-J. Qi, and K. A. Hua, "Learning label preserving binary codes for multimedia retrieval: A general approach," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 14, no. 1, pp. 1:1–1:23, 2017.

[48] H. Lai, Y. Pan, Y. Liu, and S. Yan, "Simultaneous feature learning and hash coding with deep neural networks," in *Proc. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 3270–3278.
[49] J. Tang, J. Lin, Z. Li, and J. Yang, "Discriminative deep quantization hashing for face image retrieval," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 12, pp. 6154–6162, Dec. 2018.
[50] 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.
[51] T.-T. Do, A.-D. Doan, and N.-M. Cheung, "Learning to hash with binary deep neural network," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 219–234.
[52] 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.
[53] H.-F. Yang, K. Lin, and C.-S. Chen, "Supervised learning of semantics-preserving hash via deep convolutional neural networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 2, pp. 437–451, Feb. 2018.
[54] L. Jin, K. Li, Z. Li, F. Xiao, G.-J. Qi, and J. Tang, "Deep semantic-preserving ordinal hashing for cross-modal similarity search," *IEEE Trans. Neural Netw. Learn. Syst.*, to be published, doi: 10.1109/TNNLS.2018.2869601.
[55] Y. Jia *et al.*, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. ACM Int. Conf. Multimedia*, 2014, pp. 675–678.
[56] M. J. Huiskes and M. S. Lew, "The MIR flickr retrieval evaluation," in *Proc. ACM Int. Conf. Multimedia Inf. Retr.*, 2008, pp. 39–43.
[57] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Univ. Tornoto, Tornoto, ON, Canada, Tech. Rep. 4, 2009, p. 7, vol. 1.
[58] 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. Conf. Image Video Retr.*, 2009, p. 48.

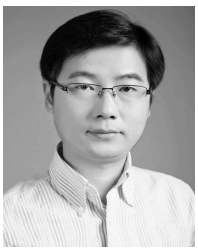
Lu Jin received the B.E. degree in measuring and control technology and instrumentations from Northeastern University at Qinhuangdao, Hebei, China, in 2010. She is currently pursuing the Ph.D. degree in computer science and technology with the Nanjing University of Science and Technology. From 2015 to 2017, she was a Visiting Scholar with the Department of Computer Science, University of Central Florida. Her research interests include multimedia computing, deep learning, and multimedia retrieval. She received the Best Student Paper Award at ICIMCS 2018.

Xiangbo Shu received the Ph.D. degree from the Nanjing University of Science and Technology, China, in 2016. From 2014 to 2015, he was a Visiting Scholar with the Department of Electrical and Computer Engineering, National University of Singapore. He is currently an Assistant Professor with the School of Computer Science and Engineering, Nanjing University of Science and Technology. His research interests include computer vision, multimedia computing, and deep learning. He received the Best Student Paper Award at MMM 2016 and the Best Paper Runner-Up at ACM MM 2015.

Kai Li received the Ph.D. degree from the University of Central Florida in 2017. He is currently a Research Scientist with Facebook. His research interests include deep metric learning, model compression, multimedia retrieval, and the broad areas of image and video understanding. He was a recipient of the Best Paper Award from the IEEE International Symposium on Multimedia 2015.

Zechao Li received the B.E. degree from the University of Science and Technology of China in 2008 and the Ph.D. degree from the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, in 2013. He is currently a Professor with the Nanjing University of Science and Technology. His research interests include intelligent media analysis and computer vision. He was a recipient of the Young Talent Program of China Association for Science and Technology, the Excellent Doctoral Dissertation of Chinese Academy of Sciences, and the Excellent Doctoral Theses of China Computer Federation.

Guo-Jun Qi received the Ph.D. degree from the University of Illinois at Urbana–Champaign in 2013. He is currently a Faculty Member with the Department of Computer Science, University of Central Florida. His research interests include pattern recognition, machine learning, computer vision, multimedia, and data mining. He has served as a program committee member and a reviewer for many academic conferences and journals in the fields of pattern recognition, machine learning, data mining, computer vision, and multimedia. He was a recipient of IBM Ph.D. fellowships for two times and the Microsoft Fellowship. He received the Best Paper Award at the 15th ACM International Conference on Multimedia, Augsburg, Germany, in 2007.



Jinhui Tang received the B.E. and Ph.D. degrees from the University of Science and Technology of China in 2003 and 2008, respectively. From 2008 to 2010, he was a Research Fellow with the School of Computing, National University of Singapore. He is currently a Professor with the School of Computer Science and Engineering, Nanjing University of Science and Technology, China. His current research interests include large-scale multimedia search. He has authored over 150 journal and conference papers in these areas. He was a co-recipient of the best paper awards at ACM MM 2007, PCM 2011, and ICIMCS 2011, and the Best Student Paper Award at MMM 2016.