Research Article

Improved Deep Hashing with Scalable Interblock for Tourist Image Retrieval

Jiangfan Feng and Wenzheng Sun

Chongqing University of Posts and Telecommunications, College of Computer Science and Technology, Chongqing, China

Correspondence should be addressed to Jiangfan Feng; fengjf@cqupt.edu.cn

Received 17 March 2021; Revised 16 June 2021; Accepted 5 July 2021; Published 14 July 2021

Academic Editor: Boxiang Dong

Copyright © 2021 Jiangfan Feng and Wenzheng Sun. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Tourist image retrieval has attracted increasing attention from researchers. Mainly, supervised deep hash methods have significantly boosted the retrieval performance, which takes hand-crafted features as inputs and maps the high-dimensional binary feature vector to reduce feature-searching complexity. However, their performance depends on the supervised labels, but few labeled temporal and discriminative information is available in tourist images. This paper proposes an improved deep hash to learn enhanced hash codes for tourist image retrieval. It jointly determines image representations and hash functions with deep neural networks and simultaneously enhances the discriminative capability of tourist image hash codes with refined semantics of the accompanying relationship. Furthermore, we have tuned the CNN to implement end-to-end training hash mapping, calculating the semantic distance between two samples of the obtained binary codes. Experiments on various datasets demonstrate the superiority of the proposed approach compared to state-of-the-art shallow and deep hashing techniques.

1. Introduction

With the rise of cheap sensors, mobile terminals, and social networks, research on tourist images is making good progress, which results in an explosive growth of image retrieval in social networks. This trend imposes great challenges on developing scalable indexing approaches, supporting retrieving relevant images of such massive tourist images. However, current tourist image retrieval mainly relies on manual tags in sensor types, tourist sights, and geographical locations. For example, SIFT [1] uses local descriptors to encode image regions of interest, for example, HOG [2] and BOW [3]. Consequently, it is highly dependent on the availability and quality of tags.

Due to the fast query speed and low storage cost, learning-based hash has been attracting research interests and was applied to applications such as large-scale object retrieval [4], image classification [5], and detection [3]. Recently, deep learning using hash methods has shown promising performance [6, 7]. Due to the high efficiency of binary hash code in the computation of Hamming distance

and the advantage of storage space, it is very efficient in large-scale image retrieval. Convolutional neural network hashing (CNNH) [8] incorporates deep neural networks into hash coding to learn the image representations and hash codes. Network in network hashing (NINH) [9] presents a triplet ranking loss to capture the relative similarities of images. The image representation learning and hash coding can benefit each other within a staged framework. Deep semantic hashing [10] ultimate hash codes produced by the learned hash functions maintain sentiment-level similarity. Other hashing methods have also been proposed [11–13].

Although hashing methods have achieved remarkable performance, they still suffer from the two following problems:

- (1) Existing methods learn binary hash codes with hand-crafted feature representations, which cannot accurately capture the inherent semantic similarities of images
- (2) In most existing hashing methods for images, the semantic similarities are defined at the image level,

and each picture is represented by one piece of hash code

This paper considers large-scale retrieval for multilabel tourist image data, which includes semantic hashing and category-aware hashing. We propose an architecture of deep convolution networks designed for hash learning, which has substantially superior performance on large-scale tourist images by end-to-end learning discriminative short binary code. As a whole, the main contributions of this paper are as follows:

- (1) For binary hash optimization, we propose a discrete hash optimization strategy based on the inner relationship for learning hash codes without relaxing the quantization information loss.
- (2) We provide an improved divide-code layer, substituting for fully connected layers to learn binary hash code to reduce high redundancy and parameters in the retrieval task. Besides, we use an improved triplet loss function to guarantee the feature similarity to the binary code features to improve the algorithmic efficiency while training.
- (3) In terms of applications, the deep hash method is employed for large-scale tourist image retrieval. Consequently, this paper illustrates ways to design and train a deep network of large-scale tourist image retrieval.

2. Related Works

This section briefly reviews two topics: (1) tourist image retrieval models and (2) hashing retrieval models.

2.1. Tourist Image Retrieval Model. Numerous tourist image retrieval methods based on landmark datasets have been proposed. They often use visual descriptors to describe images. The key is how to improve the expressive ability of visual descriptors. For example, Hao et al. [14] and Xiao et al. [15] used multidimensional models to sort in space property and utilized the three-dimensional visual phrase to describe the landmark images. However, these methods have the disadvantages of long-time modeling and high retrieval cost. Recently, to reduce the cost of retrieval, many researchers began to devote themselves to the research of binary images that compose the landmark features of high-dimensional visual words. Ji et al. [16] proposed a Location Discriminative Vocabulary Coding (LDVC) scheme, which achieves deficient bit rate query transmission, discriminative landmark description, and scalable descriptor delivery in a unified framework. Duan et al. [17] combined multiple information, such as image, GPS, and crowd-sourced hotspot Wi-Fi, to extract location discriminative compact image descriptors. Zhou et al. [18] used the scalable cascaded hashing (SCH) method to implement the landmark hashing retrieval. Zhu et al. [19] used a discrete multimodal hash scheme (Cv-Dmh) based on a canonical view to learn binary code through a new three-stage learning process. Jing et al. [20] investigated the spatiotemporal dynamic patterns of

inbound tourism. Cui et al. [21] proposed a Scalable deep hashing (SCADH) to learn enhanced hash codes for social image retrieval.

Furthermore, complex network theory has been used to mine tourism flow patterns [22]. These methods are based on the feature extraction of the image, and then the hashing algorithm is used for iterative computation. However, no method of them is an end-to-end method to learn the hash function. Furthermore, most methods still use hand-craft features to extract image features, which have a weak generalization and migration ability.

Recent examples in which deep learning has made significant advances in tourist image retrieval include positioning the city [23] and tourist photo classification [24]. In addition, many studies have been conducted to analyze the tourist's urban image by modifying the classifier part of the CNN model [25] or considering local characteristics [26]. However, these studies are limited in reflecting the unique landscape or regional characteristics in the area.

2.2. Hashing Retrieval Model. Learning-based hashing retrieval methods can be divided into unsupervised methods and supervised methods. Unsupervised learning has a catalytic effect in reviving interest in hashing retrieval but has been overshadowed by the successes of purely supervised learning. The researchers introduced unsupervised learning procedures that only use the information on image samples without requiring supervision information for hashing. Notable examples in this category include local sensitive hashing (LSH) [27], iterative quantization (ITQ) [28], direct graph hashing (DGH) [29], scalable graph hashing (SGH) [30], and spectral hashing (SH) [31]. Unsupervised training of hashing retrieval is regarded as a "pretraining" phase whose role is to discover good features that model the structure in the input domain. Besides, supervised methods learn hash coding using both feature information and label, including minimum loss hashing [32], kernel-based supervised hashing (KSH) [33], ranking-based supervised hashing (RSH) [34], and column generation hashing (CGH) [35].

New advances in machine learning using deep neural networks enable automated learning of hashing functions. Xia et al. [36] applied deep hashing using a similarity matrix and minimized loss function to discover an approximate hash code. Although it has dramatically improved the retrieval performance, it is still not an accurate end-to-end method. Zhao et al. [37] proposed a deep hashing algorithm for sorting tags. Since image retrieval aims to return an image based on the correlation among the pictures, this approach is optimized for the final evaluation index. Lin et al. [38] proposed a straightforward method to obtain hash values. They added a fixed-length hidden layer to the CNN network that is limited by the activation function. After fine-tuning the CNN network, the hidden layer value is extracted directly. The number of nodes in the hidden layer is the length of the hash code. Although the eigenvalues obtained by this method contain the high-level semantics of the image, the process does not consider the correlation of the

Hamming space features. Therefore, it cannot guarantee the retrieval effect of the elements in the Hamming space.

Later, Lai et al. [9] proposed a training method based on the triplet. Training the objective function is to distance similar images in the Hamming space closer than dissimilar images. Recently, some semisupervised deep hashing models are proposed to utilize unlabeled data to improve retrieval accuracy. Yan et al. [39] proposed the BGDH method to learn embeddings and features simultaneously, as well as hash codes. Zhang and Peng [40] developed a deep hashing method SSDH, which maintains the underlying data structures and the semantic similarity simultaneously to learn hash functions. Both ways use a graph to model unlabeled training samples, which are computationally expensive and memory hog, especially with a large-scale dataset. Shi et al. [41] used the GAN and a discriminative model to learn from both the unlabeled data and labeled data to augment the training dataset, which may not be adapted to semantic representation. Tu et al. developed RDUH [42], which focuses on reducing noisy points by investigating the various input data structures.

Recently, cross-modal hashing methods have provided insight into capturing the intrinsic relationships between various modalities [43] and quantization-based cross-modal similarity [44]. Furthermore, Deng et al. [45] showed that semantic similarity of the training data could perform binary hash codes in an unsupervised manner. However, natural images can have significant intraclass and minor interclass variations. Thus, learning hash codes with class-specific representation centers is required [46]. To further bridge the inherent modality gap, a multitask consistency-preserving adversarial hashing (CPAH) [47] was proposed to fully explore the semantic consistency and correlation between different modalities for efficient cross-modal retrieval.

3. The Proposed Method

In this section, we present the details of our proposed method. We first define the notations used in this paper. Then, we introduce our deep feature learning process, deep hash model training process, and hash codes learning process. Finally, we present a hash optimization solution for solving hash codes and functions and analyzing their convergence and complexity.

3.1. Notations and Problem Definitions. For a tourist image dataset consisting of n images $\{x_i\}_{i=1}^n$ with l user-provided semantic tags, each image is represented by $x_i \in R^d$ and the relationships between the image and tags can be represented as l -dimensional binary-valued vector f_i . The image matrix is denoted as $\Theta \in R^{d \times n}$, and $F = [f_1, \dots, f_n] \in R^{l \times n}$ represents the observed image-tag relation matrix.

We aim to learn a set of hash codes $B = \{b_1, b_2, \dots, b_n\}$ with $b_i = H(x_i)$, $b_i \in \{0, 1\}^c$, where c is the length of binary code and $h(\cdot)$ is the hash function. The binary code should guarantee the similarity of the original data space. Generally, the hash function $H(x)$ satisfies the following:

- (1) b_i and b_j are closer in the Hamming space when $s_{ij} = 1$
- (2) b_i and b_j are far away in the Hamming space when $s_{ij} = 0$

From the view of geographical position semantics, tourist images and the accompanying tags are highly correlated. These tags contain explicit semantics that is complementary to the latent image semantics. Hence, it is promising to exploit the refined auxiliary social tags for the semantic enrichment of image hash codes. To this end, we introduce a semantic correlation matrix $\mathbf{W}$ that directly correlates hash codes with refined social tags. The dynamic semantics can be directly transferred to hash codes. We aim to minimize the difference between the binary hash codes and the mapped semantic vectors from the refined tags.

We propose an architecture of deep convolution networks designed for hash learning, as shown in Figure 1. In detail, we build an end-to-end learning framework that utilizes hash mapping for tourist attraction image retrieval. The method is divided into three parts. The first is a subnetwork with multiple convolutions and pooling layers for learning discriminative image features, pretrained on the Place-2 dataset [48]. The second is the hash layer, which consists of a block coding layer and an activation function. The third is the improved triplet loss function that we use as the objective function to optimize the network. The training process is divided into many minibatches for iterative learning. Each small batch uses multiple images which belong to different categories as input.

3.2. Feature Learning and Deep Convolution Subnetwork Module. Most existing hashing methods adopt hand-crafted features for hash function learning. However, these methods may achieve limited performance because the hand-crafted features might not be optimally compatible with the hash function learning procedure. We propose our deep convolution subnetwork module, which can perform simultaneous feature learning and hash learning in the same framework. The subnetwork is used to learn the image features that can describe the image accurately. After training, the input image is processed through the network to obtain rich semantic descriptors with excellent expressiveness and robustness.

The tags from tourist images are subject to two properties: low rank and error sparsity. In such cases, we use VGG-16 as the subnetwork and transfer the model parameters trained on the Place-365 dataset to the network as the initial parameters. Since the scene recognition task has some similarities with the tourist attraction recognition task, transferring the setting from the network trained on Place-365 to the subnetwork can significantly improve the model's performance. The concrete structure of the network is shown in Table 1, which contains five large convolutional layers, five pooling layers, and two fully connected layers. Each large convolutional layer is followed by a 2×2 maximum pooling of 2 steps, and the detailed network configuration is shown in Table 1.

FIGURE 1: The proposed deep hash retrieval framework consists of the subnetwork hash layer and the improved triplet loss function. subnetwork is a multilayer convolution neural network. This hash layer comprises the divide-encode layer and the concat layer, and the length of the concat layer c represents the hash code length. The parameter transferring process transfers the pretrained parameters on Place-2 to the subnetwork as the initial parameter to train the hash mapping function.

TABLE 1: The subnetwork structure.

Type	Filter size/stride	Output size
Conv1-1	$3 \times 3/1$	$224 \times 224 \times 64$
Conv1-2	$3 \times 3/1$	$224 \times 224 \times 64$
Max pool1	$2 \times 2/2$	$112 \times 112 \times 64$
Conv2-1	$3 \times 3/1$	$112 \times 112 \times 128$
Conv2-2	$3 \times 3/1$	$112 \times 112 \times 128$
Max pool2	$2 \times 2/2$	$56 \times 56 \times 128$
Conv3-1	$3 \times 3/1$	$56 \times 56 \times 256$
Conv3-2	$3 \times 3/1$	$56 \times 56 \times 256$
Conv3-3	$3 \times 3/1$	$56 \times 56 \times 256$
Max pool3	$2 \times 2/2$	$28 \times 28 \times 256$
Conv4-1	$3 \times 3/1$	$28 \times 28 \times 512$
Conv4-2	$3 \times 3/1$	$28 \times 28 \times 512$
Conv4-3	$3 \times 3/1$	$28 \times 28 \times 512$
Max pool4	$2 \times 2/2$	$14 \times 14 \times 512$
Conv5-1	$3 \times 3/1$	$14 \times 14 \times 512$
Conv5-2	$3 \times 3/1$	$14 \times 14 \times 512$
Conv5-3	$3 \times 3/1$	$14 \times 14 \times 512$
Max pool5	$2 \times 2/2$	$7 \times 7 \times 512$
Fc6	—	4096
Fc7	—	4096

3.3. Hash Code Learning. Most existing studies use metric learning to train the positive and negative sample pairs to ensure the binary code similarity relationship [49–52]. However, it is challenging to represent geographic characteristics as a single binary code without losing a significant amount of helpful information. Hence, there is no need to conduct such an evaluation globally, but only among segments with users' geographic information needs. For example, a single tourist image can be represented into multiple binary vectors by treating each block as an image feature.

Tourist images and the accompanying tags are positively correlated with each other. Moreover, these tags contain

explicit semantics, which is complementary to the latent image semantics. Hence, it is promising to exploit the refined auxiliary social tags for the semantic enrichment of image hash codes. To this end, we aim to minimize the difference between the binary hash codes and the mapped semantic vectors from the refined social tags.

This paper uncovered the intrinsic low-rank matrix by decomposing the image-tag relation matrix into its low-rank and sparse components. The low-rank matrix is then taken into Semantic Enhancement as a semantic source to enhance the discriminative capability of the learned hash codes. Therefore, we use a block-coded structure instead of a fully connected layer to implement a hash layer consisting of a block-coded layer, an active layer for each subblock, and a concat layer.

Consider a tourist image dataset consisting of n images $\{x_i\}_{i=1}^n$; we divide the features of fully connected layer $v(x_i)$ into q blocks. q denotes the length of the binary hash code for constructing the block-coded structure. The subfeatures $v(x_i)_j$ are obtained from the j -th slice layer as the input to fully connected layers, $j = 1, 2, \dots, q$, and the output of each fully connected layer is 1-dimensional, which is expressed as follows:

$$g(v(x_i)_j) = \mathbf{W}_j v(x_i)_j, \quad (1)$$

where $\mathbf{W}_j$ is the weight matrix of the j -th subblock, the output of each subblock is the input of the active layer, and the sigmoid function is chosen as the activation function, which is denoted as follows:

$$f_j(x) = \text{Sigmoid}(u^{(j)}) = \frac{1}{1 + e^{u^{(j)}}}, \quad (2)$$

where $u^{(j)} = g(v(x_i)_j)$. After the eigenvalues are converted into the eigenvector, the relaxation of the binary vector is

obtained. To improve the performance, we do not directly map the image into binary values of $\{0, 1\}$. Instead, we use the activation function to limit the eigenvalues among $[0, 1]$ and then use the thresholding to quantize the relaxation binary into binary code.

3.4. Triplet Loss and Optimization. We propose an improved triplet loss function to optimize the network to effectively preserve semantic similarities of images into the binary hash codes.

Let x be an image, the input to the proposed deep architecture is triplets of sample images, that is, $\{x_a, x_p, x_n\}$. $s_{ap} = 1$ and $s_{an} = 0$, where S denotes the similar identity of the images; the optimization of this triplet loss function is to narrow the distance between samples x_a and x_p and to push away the distance between samples x_a and x_n . We use $\|f(x_a) - f(x_p)\|_2$ and $\|f(x_a) - f(x_n)\|_2$ to represent the Euclidean distance between them and for the relaxed binary code obtained from the samples. As Euclidean distance can approximately represent their Hamming distance, the optimization goal is $\|f(x_a) - f(x_p)\|_2 + \sigma < \|f(x_a) - f(x_n)\|_2$. In this way, the objective function can be defined as

$$L = \sum_{\substack{s_{ap}=1 \\ s_{an}=0}} \max\left\{\sigma + \left(\|f(x_a) - f(x_p)\|_2 - \|f(x_a) - f(x_n)\|_2\right), 0\right\}. \quad (3)$$

$$L = \sum_{i=1}^P \sum_{a=1}^K \max\left\{\sigma + \max_{p=1 \dots K} \left(\|f(x_a^i) - f(x_p^i)\|_2\right) - \min_{\substack{j=1 \dots P \\ n=1 \dots K \\ j \neq i}} \left(\|f(x_a^i) - f(x_n^j)\|_2\right), 0\right\}, \quad (4)$$

where P stands for the categories in the batch, K stands for the number of images in the category, x_a^i means the a th picture in the i th class, and σ is the margin parameter.

For fast convergence, it is sensitive to the selection of triplets. Here, we use large mini-batches and only compute the hardest positive and negative samples within a minibatch instead of selecting the hardest triplets in all training data. Furthermore, these functions are differentiable almost everywhere, which means they can be used in models trained by stochastic gradient descent. On the other hand, implementation details make batches of 20–30 exemplars more efficient.

Moreover, by minimizing equation (4), the manual margin parameter σ is designed to enforce a margin between the hard positive and hard negative pairs. Therefore, we optimize the parameter through the training process with the initial value of 0.2, and implementation details make margin parameters of 0.1 to 0.8 of exemplars more efficient. How to automatically determine the margin and incorporate class-specific or sample-specific margins remains challenging.

Because the Euclidean distance is more stable in the training process and the meaning of the function is more consistent with the actual definition [42], we use Euclidean distance $\|\cdot\|_2$ to measure the distance in Hamming space rather than the square of Euclidean distance $\|\cdot\|_2^2$, which is used in the classical triplet loss function. The optimization aims to distinguish between similar samples and the different samples at least margin, which can map semantically equivalent pictures to adjacent locations in the Hamming space. Thus, the semantic features of the images extracted from CNN can be preserved in the hash code.

The basic rule of designing the loss function is to preserve the similarity order, that is, minimize the gap between the approximate nearest neighbor search result computed from the hash codes and the ideal search result obtained from the input space. A widely used solution is to select sample pairs in which the distance between X_a and X_p is greater than the distance between X_a and X_n , in a minibatch. In this work, we choose the hardest positive and negative sample pairs to compute the loss. The function is defined as follows:

3.5. Generate Hash Code. When the network training is completed, the given image will get a K -bit hash code. We define $\text{sgn}(x)$ as a symbolic function for each component.

$$\text{sgn}(x) = \begin{cases} 1, & x \geq 0.5, \\ 0, & x < 0.5. \end{cases} \quad (5)$$

If the eigenvector of image x_i extracted from the network merging layer is x_i , then the hash code of this image x_i can be described as $B_i = \text{sgn}(x_i)$. We can compute all images in the database to build a binary index library. We can use the hash code to do the nearest-neighbor retrieval in the Hamming space during the retrieval process, which is very efficient because the Hamming distance can be calculated using XOR.

The main steps of the proposed method are summarized in Algorithm 1.

4. Experiments

In this section, we conduct extensive experiments on two tourist image datasets to evaluate the efficiency and

```

Input:  $\Theta \in R^{d \times n}$ , the training image matrix
       $q$ , the hash code length
       $j$ , number of sub-layers
       $\mathbf{W}$ , the weight matrix
Output: deep hash functions  $h(x)$ 
(1) Initialize the deep models by the pre-trained VGG-16 Sub-Network
(2) Update  $\mathbf{W}$  in training process according to loss function;
(3) For  $x \in \Theta$  do
(4)   For iter = 1 to  $j$  do
(5)     Compute  $g(v(x_i)_{\text{iter}})$ ;
(6)     Compute  $f_j(x)$ ;
(7)     Quantize the relaxation binary into binary code with  $f_j(x)$ ;
(8)     Return  $h(x_i) = \text{sgn}(x_i)$ 
(9)   End for
(10) End for

```

ALGORITHM 1: Key steps of the approach.

effectiveness of the proposed method. The details of the experiments and the results are described in the following sections.

4.1. Datasets and Experimental Settings

4.1.1. Datasets

(1) *China-60 Dataset*. Most public landmarks such as Oxford5K and Paris6K present unrelated images suitable for classification frameworks. However, images representing views of the same scene are needed. Thus, we developed a dataset called China-60, randomly selected from Flickr and Baidu Images based on the keywords of 60 popular tourist attractions in China. Variability of images comes from different viewing scales, angles, lighting conditions, and image clutter. Therefore, we provide 3–5 tags to describe the image contents, such as name and places. Our research's primary purpose is tourism image retrieval, so we have developed a Chinese image dataset with attraction to verify the method's performance on the image retrieval task.

For each tourist attraction, we crawl 500 to 600 images and remove irrelevant or low-quality photos. The final dataset contains 25,890 images of 60 tourist attractions, including buildings, rivers, forests, mountains, and other types of interests, all photographed under different light, seasons, and angles. We divide the dataset into the training set, test set, and validation set in a ratio of 8:1:1. In evaluation, the images belonging to the same tourist attraction are considered similar. On the contrary, they are deemed dissimilar. Typical images are shown in Figure 2.

(2) *Public Datasets*. For a clear comparison and analysis, we also experiment on the different datasets Cifar-10 and Flickr30k. Cifar-10 contains 60,000 images, which are divided into ten categories, each containing 1,000 images. All photos have a 32×32 resolution. We also divide them into the training set, validation set, and test set according to the proportion of 8:1:1. Flickr30k contains 31,783 images

focusing mainly on people and animals. We select 1000 outdoor images randomly for the testing set and 30783 other for the training set.

4.1.2. Baseline and Evaluating Indicators

(1) *Baseline*. To illustrate the benefits of the proposed method, we compare it with various approaches, including existing traditional hash approaches LSH [27], SH [31], PCAH [53], PCA-ITQ, PCA-RR [28], CBR-rand, CBR-opt [54], and DSH [55]. We also compare it with deep hashing approaches, such as DLBHC [38] and DNNH [9]. Finally, after fine-tuning, the features are extracted from the pre-trained VGG network as the mapping function input instead of handcraft features.

(2) *Evaluating Indicators*. Four evaluation indicators were used to assess the performance of the different methods as follows: (1) precision at N sample curve, where precision is the proportion of the correct samples in the returned images, (2) recall at N samples curve, where recall is the proportion of the accurate results in the query results to all correct results, (3) precision-recall (P-R) curve which is the curve of precision changing with recall, and (4) mean average precision (MAP), which is the area surrounded by the P-R curve.

4.2. *Results and Discussion on China-60*. We first evaluate the effectiveness by comparing each method's performance under different lengths of hash code, which can get a convincing result. Firstly, we assess the performance in terms of MAP, calculated for all returned samples by sorting with the Hamming distance. The MAP value is shown in Table 2, where DNNH, DLBHC, and the proposed method are deep hashing methods, while the other ways are traditional hashing methods. As shown in Table 2, the proposed method's results perform better than other methods, and the MAP values of most practices have a positive correlation with the length of the hash code. The experiments show that

FIGURE 2: Sample images of China-60.

TABLE 2: The value of MAP for different methods on the China-60 dataset.

Method	China-60			
	32 bits	64 bits	128 bits	256 bits
CBE-rand	0.329	0.473	0.618	0.681
CBE-opt	0.338	0.496	0.628	0.694
ITQ	0.681	0.794	0.804	0.813
LSH	0.314	0.483	0.597	0.691
PCAH	0.515	0.614	0.415	0.278
SH	0.102	0.234	0.302	0.293
PCA-RR	0.517	0.652	0.694	0.728
DSH	0.238	0.283	0.397	0.465
DLBHC	0.814	0.841	0.856	0.849
DNNH	0.839	0.864	0.860	0.862
Ours	0.895	0.907	0.912	0.903

traditional hashing methods and the size of the binary feature are often highly correlated.

Figure 3 shows the precision-recall (P-R) curves for different methods on the Cifar-10 dataset. We plotted P-R curves on the hash code of four diverse lengths. It can be seen from the diagram that our approach can always maintain the highest precision rate and smaller curve slope under all-length hash code when the recall rate is low. This means that our policy has better retrieval performance. We can also find the gap between the deep hashing algorithm and the traditional algorithm in the graph. Most traditional hashing algorithms have a concave curve on the short hash code, signifying that they have terrible performance on the short hash code. However, with the increase of the length of hash code, part of the P-R curves of traditional hashing algorithms become convex curves, which signify that an extended hash code is often required to ensure the retrieval

of conventional hashing effect algorithms. This is consistent with what we said before. On the other hand, the deep hashing algorithms have a slight variation in curve radian under different lengths of hash code, showing the stability and superiority of the deep hashing algorithms.

The TOP-K accuracy rate reflects the proportion of the first K returned results from the correct results of the query, which the user can intuitively perceive in the retrieval results. Therefore, the TOP-K precision rate is an important index to evaluate the retrieval algorithm's practical application performance. Figure 4 shows the precision of TOP-K retrieval results in the nearest neighbor retrieval. Similarly, the plot shows the precision curves of 32 bits (a), 64 bits (b), 128 bits (c), and 256 bits (d) lengths of hash code, respectively. The horizontal coordinate of the curve is the number of returned samples, and the vertical coordinate is the precision rate. It can be seen from the diagram that the

FIGURE 3: The precision versus recall curves. The length of hash code is 32 bits (a), 64 bits (b), 128 bits (c), and 256 bits (d).

FIGURE 4: The precision versus the variable number of sample curves. The length of hash code is 32 bits (a), 64 bits (b), 128 bits (c), and 256 bits (d).

FIGURE 5: Recall versus variable number of sample curves. The hash code is 32 bits (a), 64 bits (b), 128 bits (c), and 256 bits (d).

FIGURE 6: On the China-60 dataset, the top 6 results of 4 query images are returned by different hashing methods' retrieval. The pictures of the first column are queries, and we use 128-bit hash codes for retrieval. The image in the red line is considered not similar to the query image.

retrieval precision of our approach is always the best of all cases, and when fewer samples are returned, the retrieval precision can reach the highest value. This reflects that the correct samples can be usually returned preferentially, which makes our method sufficiently meet the requirements of image recognition and retrieval for unknown scenic spot images.

Figure 5 shows the TOP-K relation curve between the recall rate and returned sample. The horizontal coordinate is the number in returned samples, and the ordinate is the recall rate of the sample. The correct sample in the returned sample accounts for all of the correct samples in the database. This is an essential criterion of evaluation that developers and administrators of the retrieval system concern about. In addition, it reflects the retrieval success degree of the algorithm in the database. As shown in the figure, our method achieves the best TOP-K recall for all coding lengths. Figure 6 exhibits some query examples on the China-60 dataset. For each query, each method returns the top 6 query results by using the 128-bit hash code, and red represents the incorrect returned results.

4.3. Generalization to Other Image Data Sources. Although the primary purpose of this article is to explore the effect of retrieval methods on image retrieval tasks in tourist attractions, for demonstrating the universality of the process, we also conducted experiments in public image datasets. Considering that the size of the Cifar-10 dataset image is 32×32 , we shorted the generated hash code length to 12 bits, 24 bits, 32 bits, and 48 bits. Thus, the hash code length is also consistent with the Flickr30 dataset.

Table 3 shows the results of MAP values on the two datasets, where CNNH, DNNH, DLBHC, and the proposed method are deep hashing methods, and the others are the

no-deep methods. It can be seen from the results that our approach has a significant advantage over the no-deep hashing algorithm. The MAP value of most no-deep methods dramatically increases with the length of the hash code. In the best case, compared with the best no-deep hashing method, the deep hashing algorithm still has a significant superiority. For the deep hashing approach, the accuracy of our process has a 4% to 8% enhancement, which shows that the hash code generation strategy proposed in this paper can efficiently improve the retrieval effect.

4.4. Generalization to Cross-Datasets. To verify our method in general, we conduct experiments over the cross-datasets. The aim is to utilize two or more datasets labeled with different classes to train and evaluate a single model. For example, we train the proposed model by various datasets: the Flickr30 dataset and the Cifar-10 dataset, respectively. The performance of the trained model is tested by taking a different dataset, China-60.

The experimental results are shown in Table 4, which shows that the overall precision scores are relatively low, indicating that cross-datasets evaluation is more challenging for the retrieval task. However, it also demonstrates that the proposed method achieves the competitive performance on the cross-datasets tourist images retrieval task, demonstrating the effectiveness of our proposed method.

4.5. Time-Cost Analysis. Besides the effectiveness analysis, we also compare the proposed approach with other methods, deep and no-deep, in terms of the computation time cost. All the experiments are carried out on the same platform with Intel i7 8700K CPU, NVIDIA GTX 2080, and 64G RAM. Table 5 shows the average computation times of different

TABLE 3: The value of MAP for different methods on the public datasets.

Method	Cifar-10				Flickr30			
	12 bits	24 bits	32 bits	48 bits	12 bits	24 bits	32 bits	48 bits
ITQ	0.264	0.282	0.288	0.295	0.577	0.580	0.581	0.580
LSH	0.183	0.164	0.161	0.162	0.557	0.564	0.562	0.569
PCAH	0.157	0.164	0.162	0.170	0.588	0.596	0.604	0.601
SH	0.183	0.164	0.161	0.161	0.561	0.562	0.563	0.562
DSH	0.303	0.337	0.346	0.356	0.678	0.697	0.689	0.685
CNNH	0.439	0.511	0.509	0.522	0.667	0.688	0.654	0.626
DLBHC	0.553	0.580	0.578	0.557	0.692	0.710	0.703	0.707
DNNH	0.571	0.588	0.589	0.595	0.739	0.752	0.753	0.755
Ours	0.613	0.648	0.654	0.663	0.828	0.837	0.835	0.840

TABLE 4: The value of MAP for cross-datasets evaluation.

Method	Cifar-10				Flickr30			
	12 bits	24 bits	32 bits	48 bits	12 bits	24 bits	32 bits	48 bits
ITQ	0.027	0.032	0.035	0.034	0.040	0.051	0.066	0.070
LSH	0.028	0.034	0.041	0.042	0.051	0.064	0.077	0.089
PCAH	0.038	0.045	0.060	0.071	0.057	0.069	0.072	0.078
SH	0.019	0.024	0.027	0.033	0.017	0.018	0.019	0.022
DSH	0.058	0.070	0.073	0.078	0.069	0.092	0.104	0.109
CNNH	0.087	0.092	0.089	0.090	0.095	0.105	0.101	0.104
DLBHC	0.095	0.101	0.112	0.115	0.102	0.107	0.105	0.119
DNNH	0.089	0.105	0.104	0.107	0.107	0.105	0.112	0.116
Ours	0.105	0.113	0.112	0.115	0.134	0.142	0.147	0.145

TABLE 5: Comparison of the average computation time (per image) in different methods.

Method	China-60 (ms)	Cifar-10 (ms)	Flickr30 (ms)
ITQ	7.95	5.42	4.10
SDH	8.05	5.51	4.15
CNNH	6.85	4.62	3.51
DLBHC	6.95	4.65	3.53
DNNH	6.72	4.63	3.47
Ours	6.52	4.45	3.40

methods. The proposed approach is comparable with other methods.

5. Conclusion

In this paper, we proposed a deep hashing method with scalable interblock for large-scale tourist attractions. After end-to-end training of the constructed deep hash network, the network utilizes the triplet loss function to guarantee the hash code's characteristic similarity. To enhance the performance and efficiency of function optimization and the descriptive ability of hash code, we improve the network and triplet loss function. Based on the results, we report the quantitative evaluation of the proposed method to scale hash length. Experimental results on social image datasets validate the superiority of the proposed method. However, the relaxed binary code obtained from the network may cause feature loss in the threshold process. In future work, we will improve the activation function to dispose of these problems.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

The work was supported by the National Natural Science Foundation of China (41971365) and the Chongqing Research Program of Basic Science and Frontier Technology (cstc2019jcyj-msxmX0131).

References

- [1] K. Yan Ke and R. Sukthankar, "PCA-SIFT: a more distinctive representation for local image descriptors," *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision*

- and Pattern Recognition, 2004. CVPR 2004, Washington D. C., USA, June 2004.
- [2] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, pp. 886–893, CA, USA, June 2005.
 - [3] J. Yang, Y. G. Jiang, A. G. Hauptmann, and C. W. Ngo, "Evaluating bag-of-visual-words representations in scene classification," in *Proceedings of the 9th ACM SIGMM International Workshop on Multimedia Information Retrieval*, pp. 197–206, Augsburg, Bavaria, Germany, September 2007.
 - [4] J. Wang, T. Zhang, J. Song, N. Sebe, and H. T. Shen, "A survey on learning to hash," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 4, pp. 769–790, 2018.
 - [5] J. Sanchez and F. Perronnin, "High-dimensional signature compression for large-scale image classification," in *Proceedings of the The 24th IEEE Conference on Computer Vision and Pattern Recognition Cvpr 2011*, pp. 1665–1672, CO, USA, June 2011.
 - [6] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in Neural Information Processing Systems*, vol. 25, no. 2, pp. 1097–1105, 2012.
 - [7] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, <https://arxiv.org/abs/1409.1556>.
 - [8] R. Xia, Y. Pan, H. Lai, C. Liu, and S. Yan, "Supervised hashing for image retrieval via image representation learning," *Proceedings of the National Conference on Artificial Intelligence*, vol. 3pp. 2156–2162, Québec, Canada, July 2014.
 - [9] H. Lai, Y. Pan, and S. Yan, "Simultaneous feature learning and hash coding with deep neural networks," in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 3270–3278, Boston, MA, USA, June 2015.
 - [10] K. Zhou, J. Zeng, Y. Liu, and F. Zou, "Deep sentiment hashing for text retrieval in social CIoT," *Future Generation Computer Systems*, vol. 86, pp. 362–371, 2018.
 - [11] C. Deng, Z. Chen, X. Liu, X. Gao, and D. Tao, "Triplet-based deep hashing network for cross-modal retrieval," *IEEE Transactions on Image Processing*, vol. 27, no. 8, pp. 3893–3903, 2018.
 - [12] Q. Jiang, X. Cui, and W. Li, "Deep discrete supervised hashing," *IEEE Transactions on Image Processing*, vol. 27, no. 12, pp. 5996–6009, 2018.
 - [13] X. Zhe, S. Chen, and H. Yan, "Deep class-wise hashing: semantics-preserving hashing via class-wise loss," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 5, pp. 1681–1695, 2020.
 - [14] Q. Hao, R. Cai, Z. Li, L. Zhang, and F. Wu, "3D visual phrases for landmark recognition," in *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3594–3601, Providence, RI, USA, June 2012.
 - [15] X. Xiao, C. Xu, J. Wang, and M. Xu, "Enhanced 3-D modeling for landmark image classification," *IEEE Transactions on Multimedia*, vol. 14, no. 4, pp. 1246–1258, 2012.
 - [16] R. Ji, L. Y. Duan, and J. Chen, "Location discriminative vocabulary coding for mobile landmark search," *International Journal of Computer Vision*, vol. 96, no. 3, pp. 290–314, 2012.
 - [17] L. Y. Duan, J. Chen, R. Ji, T. Huang, and W. Gao, "Learning compact visual descriptors for low bit rate mobile landmark search," *AI Magazine*, vol. 34, no. 2, p. 67, 2013.
 - [18] W. Zhou, M. Yang, H. Li, X. Wang, Y. Lin, and Q. Tian, "Towards codebook-free: scalable cascaded hashing for mobile image search," *IEEE Transactions on Multimedia*, vol. 16, no. 3, pp. 601–611, 2014.
 - [19] L. Zhu, Z. Huang, X. Liu, X. He, J. Sun, and X. Zhou, "Discrete multimodal hashing with canonical views for robust mobile landmark search," *IEEE Transactions on Multimedia*, vol. 19, no. 9, pp. 2066–2079, 2017.
 - [20] C. Jing, M. Dong, M. Du, Y. Zhu, and J. Fu, "Fine-grained spatiotemporal dynamics of inbound tourists based on geo-tagged photos: a case study in Beijing, China," *IEEE Access*, vol. 8, Article ID 28735, 2020.
 - [21] H. Cui, L. Zhu, J. Li, Y. Yang, and L. Nie, "Scalable deep hashing for large-scale social image retrieval," *IEEE Transactions on Image Processing*, vol. 29, pp. 1271–1284, 2020.
 - [22] N. Mou, R. Yuan, T. Yang, H. Zhang, J. Tang, and T. Makkonen, "Exploring spatio-temporal changes of city inbound tourism flow: the case of Shanghai, China," *Tourism Management*, vol. 76, 2020.
 - [23] A. A. Chugunova, "Soft power digital capabilities in the tourist image construction of a big city (on the example of St. Petersburg)," in *Proceedings of the 2020 IEEE Communication Strategies in Digital Society Seminar (ComSDS)*, pp. 7–13, St. Petersburg, Russia, April 2020.
 - [24] N. D. Payntar, W. L. Hsiao, R. A. Covey, and K. Grauman, "Learning patterns of tourist movement and photography from geotagged photos at archaeological heritage sites in Cuzco, Peru," *Tourism Management*, vol. 82, Article ID 104165, 2020.
 - [25] S. Law, Y. Shen, and C. Seresinhe, "An application of convolutional neural network in street image classification: the case study of London," in *Proceedings of the 1st Workshop on Artificial Intelligence and Deep Learning for Geographic Knowledge Discovery*, pp. 5–9, Redondo Beach, CA, USA, November 2017.
 - [26] Y. Kang, N. Cho, J. Yoon, S. Park, and J. Kim, "Transfer learning of a deep learning model for exploring tourists' urban image using geotagged photos," *ISPRS International Journal of Geo-Information*, vol. 10, no. 3, p. 137, 2021.
 - [27] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, "Locality-sensitive hashing scheme based on p-stable distributions," in *Proceedings of the Twentieth Annual Symposium on Computational Geometry*, pp. 253–262, ACM, NY, USA, June 2004.
 - [28] Y. Gong, S. Lazebnik, A. Gordo, and F. Perronnin, "Iterative quantization: a procrustean approach to learning binary codes for large-scale image retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 12, pp. 2916–2929, 2013.
 - [29] W. Liu, C. Mu, S. Kumar, and S. F. Chang, "Discrete graph hashing," *Advances in Neural Information Processing Systems*, vol. 4, pp. 3419–3427, 2014.
 - [30] Q. Y. Jiang and W. J. Li, "Scalable Graph Hashing with Feature Transformation," in *Proceedings of the 24th International Conference on Artificial Intelligence IJCAI*, pp. 2248–2254, Buenos Aires, Argentina, July 2015.
 - [31] Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing," in *Proceedings of the Advances in Neural Information Processing Systems, Twenty-Second Annual Conference on Neural Information Processing Systems*, pp. 1753–1760, Vancouver, British Columbia, Canada, December 2008.
 - [32] M. Norouzi and D. M. Blei, "Minimal loss hashing for compact binary codes," in *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pp. 353–360, Bellevue, Washington, USA, June 2011.
 - [33] W. Liu, J. Wang, R. Ji, Y. G. Jiang, and S. F. Chang, "Supervised Hashing with Kernels," in *Proceedings of the 2012*

- IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2074–2081, MI, USA, June 2012.
- [34] J. Wang, W. Liu, A. X. Sun, and Y. G. Jiang, “Learning hash codes with listwise supervision,” in *Proceedings of the 2013 IEEE International Conference on Computer Vision (ICCV)*, pp. 3032–3039, IEEE, Sydney, NSW, Australia, December 2013.
 - [35] X. Li, G. Lin, C. Shen, A. Hengel, and A. Dick, “Learning Hash Functions Using Column Generation,” in *Proceedings of the 30th International Conference on Machine Learning*, pp. 142–150, Atlanta, GA, USA, June 2013.
 - [36] R. Xia, Y. Pan, H. Lai et al., “Supervised hashing for image retrieval via image representation learning,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 2156–2162, Qubec, Canada, July 2014.
 - [37] F. Zhao, Y. Huang, and L. Wang, “Deep semantic ranking based hashing for multi-label image retrieval,” in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1556–1564, MA, USA, June 2015.
 - [38] K. Lin, H. Yang, J. Hsiao, and C. Chen, “Deep learning of binary hash codes for fast image retrieval,” in *Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 27–35, MA, USA, June 2015.
 - [39] X. Yan, L. Zhang, and W.-J. Li, “Semi-supervised deep hashing with a bipartite graph,” in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pp. 3238–3244, AAAI Press, Palo Alto, CA, USA, Aug 2017.
 - [40] J. Zhang and Y. Peng, “SSDH: semi-supervised deep hashing for large scale image retrieval,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 1, pp. 212–225, 2019.
 - [41] W. Shi, Y. Gong, B. Chen, and X. Hei, “Transductive semi-supervised deep hashing,” *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–14, 2021.
 - [42] R. C. Tu, X. L. Mao, and W. Wei, “MLS3RDUH: deep unsupervised hashing via manifold based local semantic similarity structure reconstructing,” in *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 3466–3472, Yokohama, Japan, July 2020.
 - [43] E. Yang, C. Deng, W. Liu, X. Liu, D. Tao, and X. Gao, “Pairwise relationship guided deep hashing for cross-modal retrieval,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 1618–1625, CA, USA, February 2017.
 - [44] E. Yang, C. Deng, C. Li, W. Liu, J. Li, and D. Tao, “Shared predictive cross-modal deep quantization,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 11, pp. 5292–5303, 2018.
 - [45] C. Deng, E. Yang, T. Liu, J. Li, W. Liu, and D. Tao, “Unsupervised semantic-preserving adversarial hashing for image search,” *IEEE Transactions on Image Processing*, vol. 28, no. 8, pp. 4032–4044, 2019.
 - [46] C. Yang, E. Yang and T. Liu, “Two-stream deep hashing with class-specific centers for supervised image search,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 6, pp. 2189–2201, 2020.
 - [47] D. Xie, C. Deng, C. Li, X. Liu, and D. Tao, “Multi-task consistency-preserving adversarial hashing for cross-modal retrieval,” *IEEE Transactions on Image Processing*, vol. 29, pp. 3626–3637, 2020.
 - [48] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, “Places: a 10 million image database for scene recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 6, pp. 1452–1464, 2018.
 - [49] H. Liu, R. Wang, S. Shan, and X. Chen, “Deep supervised hashing for fast image retrieval,” in *Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2064–2072, HI, USA, July 2016.
 - [50] H. Zhu, M. Long, J. Wang, and Y. Cao, “Deep Hashing Network for Efficient Similarity Retrieval,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 2415–2421, AZ, USA, February 2016.
 - [51] J. Lin, Z. Li, and J. Tang, “Discriminative deep hashing for scalable face image retrieval,” in *Proceedings of International Joint Conference on Artificial Intelligence*, Melbourne, Australia, August 2017.
 - [52] Y. Zhai, X. Guo, Y. Lu, and H. Li, “In defense of the classification loss for person Re-identification,” in *Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 1526–1535, CA, USA, June 2019.
 - [53] J. Wang, S. Kumar, and S. Chang, “Semi-supervised hashing for scalable image retrieval,” in *Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 3424–3431, CA, USA, June 2010.
 - [54] F. Yu, S. Kumar, Y. Gong, and S. F. Chang, “Circulant binary embedding,” in *Proceedings of the 31st International Conference on Machine Learning, PMLR*, vol. 32, no. 2, pp. 946–954, Beijing, China, June 2014.
 - [55] Z. Jin, C. Li, Y. Lin, and D. Cai, “Density sensitive hashing,” *IEEE Transactions on Cybernetics*, vol. 44, no. 8, pp. 1362–1371, 2014.