information

Article

Deep Cross-Dimensional Attention Hashing for Image Retrieval

Zijian Chao and Yongming Li *

College of Information Science and Engineering, Xinjiang University, Urumqi 830046, China
* Correspondence: lym@xju.edu.cn; Tel.: +86-133-2559-7833

Abstract: Nowadays, people's lives are filled with a huge amount of picture information, and image retrieval tasks are widely needed. Deep hashing methods are extensively used to manage such demands due to their retrieval rate and memory consumption. The problem with conventional deep hashing image retrieval techniques, however, is that high dimensional semantic content in the image cannot be effectively articulated due to insufficient and unbalanced feature extraction. This paper offers the deep cross-dimensional attention hashing (DCDAH) method considering the flaws in feature extraction, and the important points of this paper are as follows. This paper proposes a cross-dimensional attention (CDA) module embedded in ResNet18; the module can capture the cross-dimension interaction of feature maps to calculate the attention weight effectively because of its special branch. For a feature map acquired by a convolutional neural network (CNN), each branch takes different rotation measurements and residual transformations to process it. To prevent the DCDAH model from becoming too complex, the CDA module is designed to have the characteristics of low computational overhead. This paper introduces a scheme to reduce the dimension of tensors, which can reduce computation and retain abundant representation. For a dimension of a feature map, the Maxpool and Avgpool are performed, respectively, and the two results are connected. The DCDAH method significantly enhances image retrieval performance, according to studies on the CIFAR10 and NUS-WIDE data sets.

Keywords: image retrieval; deep hashing; zpool; attention

Citation: Chao, Z.; Li, Y. Deep Cross-Dimensional Attention Hashing for Image Retrieval. *Information* **2022**, *13*, 506. https://doi.org/10.3390/info13100506

Academic Editor: Willy Susilo

Received: 5 July 2022
Accepted: 13 October 2022
Published: 20 October 2022

1. Introduction

An abundance of picture-data-processing tasks have emerged in recent years as a result of the Internet and artificial intelligence's rapid development [1–6]. According to Alibaba's data on the number of times that artificial intelligence is used, the processing frequencies of image data are as high as 1 billion times a day. Image retrieval, as a way of extracting similar images from large data sets, is active in a variety of applications [7–10], such as the function of searching for objects by pictures in online shopping, etc. As an important branch of image retrieval algorithms, deep hashing methods are popular because they not only ensure lower storage requirements but also guarantee higher retrieval efficiency [11–15]. Deep hashing methods use deep a convolutional neural network (CNN) to learn the high-dimensional semantic information from an image and convert high-dimensional semantic information into low-dimensional binary code using hash functions [16–20]. Such algorithms can effectively retain the similarity of images, compress the storage cost greatly, and have high-speed retrieval efficiency. Data-independent hashing and data-dependent hashing are two types of hashing algorithms, and they are introduced separately in the following article [20–23].

In order to generate the final hash code, the data-independent hashing methods translate the image to the feature descriptors using the random projection algorithm [9–15]. One of the downsides of such algorithms is instability, in that different hash functions may compute completely different hash codes. At the same time, the performance of such algorithms is poor, and the recall rate is relatively low. In order to ensure accuracy, the length of the hash code needs to be increased [16]. The data-dependent hashing methods

use training data to learn the hash function [17]. That is, compared to data-independent hashing, data-dependent hashing can generate more accurate hash code.

For the deep hashing methods, they usually use a CNN to learn the higher-dimensional semantic features of images; then, the hash function is used to map these feature maps to the hash code [18–25]. However, a CNN has the problem that the feature extraction is insufficient and imbalanced. The accuracy of the hash code is decreased as a result of the inevitable quantization error that occurs during the binary mapping process while utilizing the hash function. In order to compensate for the defects in feature extraction, many scholars introduce attention mechanisms [2,22,26,27].

The attention mechanism can focus on key information in an image to help deep hashing methods achieve a more accurate high-level semantic representation of the image. At present, for this research, most scholars adopt the algorithm of integrating channel attention mechanism or spatial attention mechanism in the CNN. These attention mechanisms assign weight to the feature map by calculating channel or spatial attention. However, apart from the complex structure of these attention mechanisms, the addition of learnable parameters makes the training model too complex, thus causing the problem of over-fitting. Additionally, such attention mechanisms have the problem of imperfect and inaccurate weight learning. Inspired by CBAM [28], this paper finds that the CNN which combines channel and spatial attention mechanisms has significantly improved performance in expressing image semantics. However, CBAM does not consider the importance of cross-dimensional interaction. Therefore, this paper focuses on using a new attention module that can emphasize the importance of the cross-dimensional interaction of image features to achieve better performance with negligible computational overhead [2].

This paper devotes using a cross-dimensional attention mechanism to ensuring low computational overhead and high efficiency. As shown in Figure 1, a cross-dimensional attention module (CDA) is designed which consists of three branches. For input tensors, different branches of the model adopt different rotation schemes. Finally, residual transformations are used to capture the interdependence between different dimensions of the tensor. This module can encode the cross-dimensional information of the channel and spatial with negligible computational cost, and more critical feature representations can be captured by emphasizing the importance of cross-dimensional interaction.

The following information contains a list of this paper's primary contributions:

1. First, this paper designs an end-to-end learning framework: DCDAH, which obtains paired images as inputs and finally generates discrete binary code.
2. Second, this paper proposes the CDA module which is embedded in the ResNet18. The model emphasizes the importance of cross-dimensional dependence while calculating the weights of channel and spatial attention, and improves the accuracy of image feature representation with almost no additional parameters.
3. Third, this paper introduces a new scheme named Zpool to reduce the dimension of the tensor, which can reduce computation and retain abundant representation. A detailed introduction is in the following content. In order to obtain more discernible hash code, the pairwise loss and balanced loss are referenced to minimize quantization error and preserve pairwise similarity.

The rest of the article is framed as follows. Related work is elaborated in Section 2. DCDAH's specifics are described in Section 3. The analysis and results of the experiment are in Section 4. Section 5 serves as the conclusion.

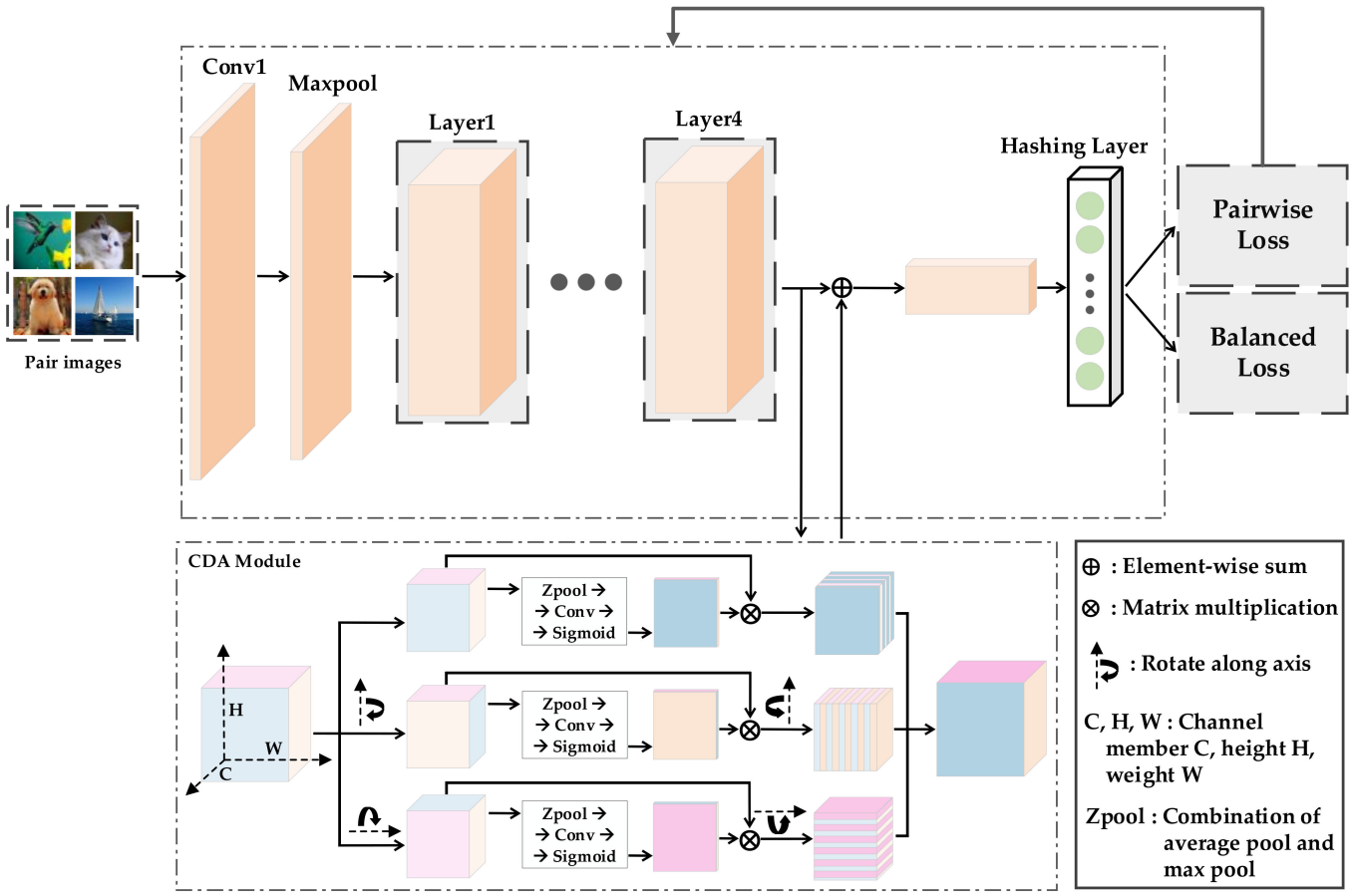


Figure 1. The overall framework of DCDAH is shown below: The framework accepts pair images as the input, and features of images are extracted via ResNet18. For feature maps that are captured after Layer4, DCDAH inputs them to the CDA module, the attention weight is distributed by CDA, and the feature maps are multiplied by weight. Finally, the weighted feature maps generate hash code through the hash layer, and parameters are optimized by two loss functions.

2. Related Work

Deep hashing image retrieval algorithms are one of the deep learning image retrieval methods; these algorithms can compress images into discrete binary codes [18,19,21]. Deep hashing methods have the advantage of low storage and faster image retrieval rate. Therefore, deep hashing methods are widely used in many practical scenes. However, some of the existing deep hashing methods still cannot express semantic features accurately. How to improve the accuracy of image representation is a current research focus. Existing hashing methods include unsupervised hashing and supervised hashing methods; this section introduces several of them.

2.1. Unsupervised Hashing

Without employing picture label information, unsupervised hashing approaches directly use image data to learn hash codes. In the traditional unsupervised hashing, Gong et al. [24] propose iterative quantization (ITQ); the algorithm combines the idea of a multiclass spectral algorithm. This technique significantly lowers the quantization error while quantizing the hash code. However, the traditional algorithms have a limitation in improving the retrieval performance due to the insufficient description of the image's underlying features. The deep unsupervised hashing algorithms use a CNN to capture more abundant semantic features and improve the retrieval property. In the study of deep unsupervised hashing methods, Lin et al. [16] presented the DeepBit algorithm which combines the advantages of deep learning and adds minimization quantization error

loss into the optimization function. DeepBit can make binary descriptors obey uniform distribution so that the original image information is retained better. Shen et al. [29] designed the similarity adaptive deep hashing (SADH) algorithm, which optimized the hash code using the alternating direction of the multiplier method, thereby effectively maintaining the similarity of the data. Zhang et al. [30] proposed deep unsupervised self-evolutionary hashing (DUSH), which generates pseudo-pairs as supervised information by employing a variable threshold. Thus, DUSH can effectively reduce the computational complexity of pseudo-labels. Because of not taking advantage of the data distribution, unsupervised hashing leads to information redundancy in the hash code.

2.2. Supervised Hashing

The supervised hashing approach thoroughly utilizes the label information to improve the similarity matrix calculation and produce more precise retrieval results. Liu et al. [31] proposed supervised hashing with kernels (KSH), where the kernel function is used to deal with linear indivisibility problems. In this class of traditional supervised hashing methods, the learning of the hash function and feature extraction process is usually divided into two steps. These methods obviously ignore the importance of the feedback between the two. Meanwhile, the workload of manual labeling is huge.

The performance of the deep supervised hashing approaches is superior to that of conventional supervised hashing since they train the model using the picture label information. Xia et al. [32] designed supervised hashing based on a CNN (CNNH), which is the first hashing method combined with a neural network. CNNH can consider capturing more delicate features of images and learning hash functions. Liu et al. [33] proposed the deep supervised hashing (DSH) method; DSH designs a loss function to facilitate the output of the image to approximate discrete values. Cao et al. [34] presented a pairwise cross-entropy loss based on Cauchy distribution. When the Hamming distance between a pair of images is greater than the given Hamming radius threshold, this loss function can heavily penalize the similarity of the images. Zheng et al. [35] proposed deep balanced discrete hashing (DBDH); the highlight of this algorithm is that it introduces a straight-through estimator for discrete gradient propagation. It enables the CNN to avoid quantization errors caused by continuous relaxation when learning hash codes. Li et al. [36] designed a deep attention-based hashing (DAH) retrieval method, where the attention is used to construct more recognizable hash code. Jin et al. [37] put forward deep ordinal hashing (DOH), which uses effective spatial attention to emphasize relevant information in local space. Long et al. [26] combined an attention mechanism with a deep hashing retrieval algorithm; the spatial and channel attention models are embedded in the CNN to improve the ability of image feature expression. Yang et al. [38] presented a deep hashing algorithm with parameter-free attention to improve extraction ability of image semantic, where an energy function is used to assign attention weight for feature maps.

Most of these methods achieve performance improvement by integrating an attention mechanism such as the channel or spatial attention algorithm which can carry out the allocation of attention weight. However, most of these methods not only increase the model complexity but also ignore the interaction between different dimensions of the feature map. Inspired by [28], this paper integrates a lightweight attention mechanism CDA into the ResNet18. ResNet18 is used to generate an image's feature map, and the idea of CDA is designing a branch structure to highlight the significance of cross-dimension interaction about the feature map. After completing the weight assignment task, the hash layer is applied to generate discrete binary code. Finally, the pairwise loss and balanced loss are referenced to learn more precise hash codes.

3. Deep Cross-Dimensional Attention Hashing

This section of the paper provides a thorough explanation of DCDAH, including the definition of the formula, the structure of the overall network model, and the attention mechanism part.

3.1. Problem Statement

A given set of data containing N images is denoted as $X = \{x_i\}_{i=1}^{N}$; x_i is the ith image. The corresponding image label information data set is denoted as $Y = \{y_i\}_{i=1}^{M}$, where y_i represents the label information of image x_i. Let M represent all the image categories in the collection. The definition of the similarity matrix is $S = \{s_{ij}\}$. When image x_i and x_j belong to the same category, $s_{ij} = 1$, otherwise $s_{ij} = 0$.

$$s_{ij} = \begin{cases} 1, & x_i \text{ and } x_j \text{ belong to the same category} \\ 0, & \text{others} \end{cases} \quad (1)$$

The hash method requires learning a process of mapping a set of images X to a set of hash codes B, where B is $\{b_n \in (+1, -1)^{L \times N}\}$. The letter L denotes that the hash code is L bits long. Hence, the real-valued variable u_i is generated through the CNN for each image x_i. The sign function $b_i = Sign(u_i)$ is used to calculate the hash code in the end. $b_i = Sign(u_i)$, $Sign(u_i) = 1$ when the $u_i > 0$; otherwise, $Sign(u_i) = 0$.

3.2. Network Architecture

The three main components of the network framework for DCDAH that is suggested in this paper are shown in Figure 1. The backbone network of DCDAH is ResNet18, which can fully acquire image features and avoid the problem of too much parameter calculation. Parameter information of ResNet18 is shown in Table 1. L stands for the length of hash codes, k for convolution kernel size, s for stride, and p for padding.

Table 1. Backbone network settings for ResNet18.

Layer	Configuration
Convolution layer	{64 × 112 × 112, k = 7 × 7, s = 2 × 2, p = 3 × 3, ReLU}
Maxpool	{64 × 54 × 54, k = 3 × 3, s = 2 × 2, p = 1 × 1, ReLU}
Layer1	{64 × 56 × 56, k = 3 × 3, s = 1 × 1, p = 1 × 1, ReLU} × 4
Layer2	{128 × 28 × 28, k = 3 × 3, s = 2 × 2, p = 1 × 1, ReLU} × 4
Layer3	{256 × 14 × 14, k = 3 × 3, s = 2 × 2, p = 1 × 1, ReLU} × 4
Layer4	{512 × 7 x 7, k = 3 × 3, s = 2 × 2, p = 1 × 1, ReLU} × 4
Avgpool	512 × 1 × 1
Hash layer	L

As depicted in Figure 1, this work adds a lightweight attention module CDA based on ResNet18. CDA can effectively capture the cross-dimensional interaction information of images. The attention weight is distributed by CDA which emphasizes the most relevant features more accurately, and the interference of irrelevant information is avoided. The last layer is the hash layer, which uses discrete gradient propagation to learn hash code.

This paper compares the feature distribution results extracted by ResNet18 and DCDAH. Figure 2 shows the visualization of feature activation of the two networks. Taking the picture of the third tag as a bird as an example, it is not difficult to see that the CDA module can better ignore the feature information irrelevant to the tag, and the activation of features is more concentrated near the tag-related information. This verification depends on the effectiveness of the CDA module.

3.3. CDA Model

Recent studies begin to improve the precise expression ability of image semantics by using attention mechanisms. In this paper, a CDA attention model is designed which has three branches. The first branch is like CBAM; spatial attention is built by this. The final two branches are used to collect information about cross-dimension interactions between the spatial dimensions W or H and the channel dimension C.

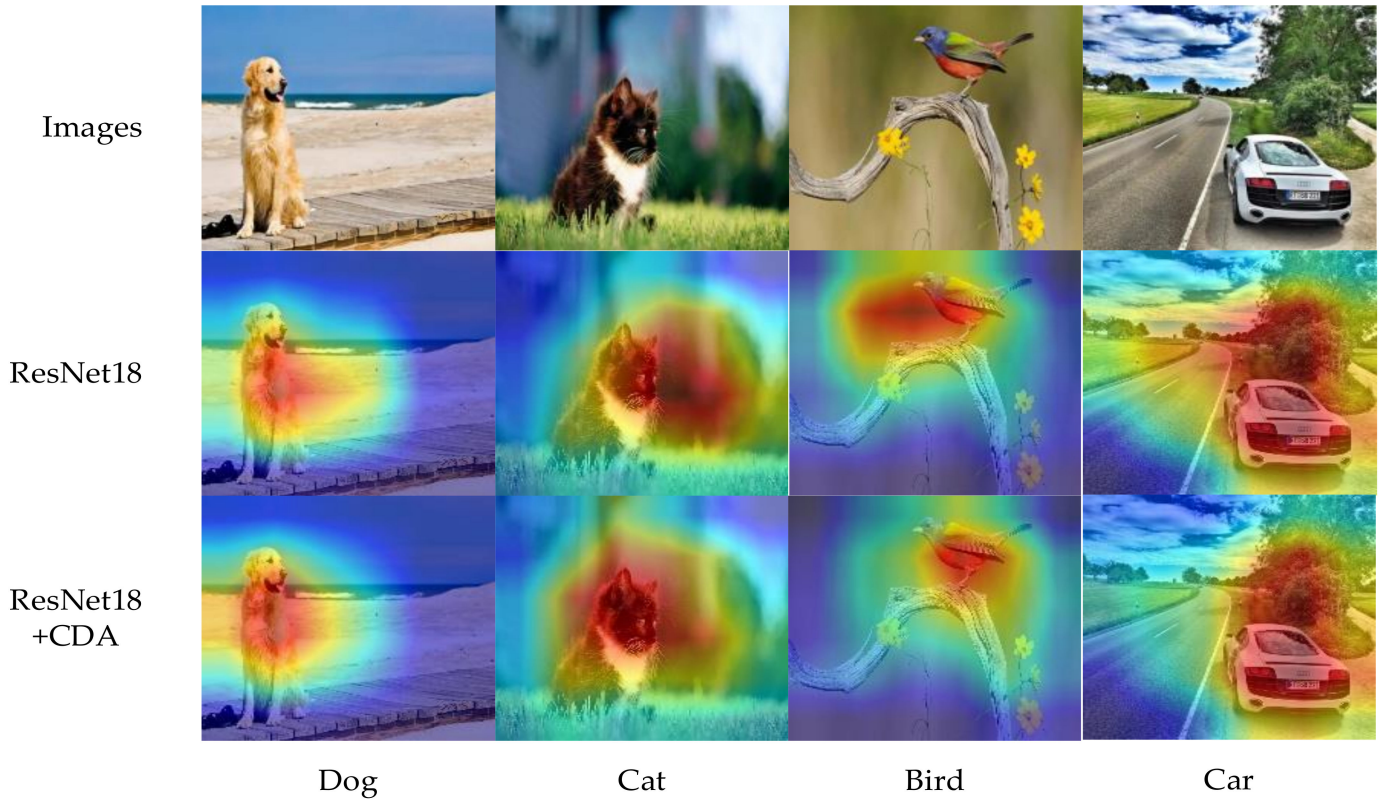


Figure 2. Visualization of feature activation.

To further reduce the computation, CDA introduces Zpool. Zpool is used to connect max-pooled and average-pooled features of a dimension to achieve reduction. Thus, the rich representation of the tensor is well-preserved and the depth of the tensor is compressed. Zpool can be written as:

$$Zpool(x) = [maxpool(x), avgpool(x)] \tag{2}$$

For example, in channel dimension C, the tensor is reduced from the shape of $(C \times H \times W)$ to $(2 \times H \times W)$. Therefore, Zpool can retain the abundant representation of the tensor while keeping computation lightweight.

For a given input tensor $\chi \in R^{C \times H \times W}$, in this paper, it is input into the three branches of the CDA. In the first branch, the channel number of the tensor χ is reduced to 2 by Zpool. At this point, the tensor χ is transformed into a simplified tensor with the shape of $(2 \times H \times W)$. This tensor is further reduced by a convolutional layer with kernel size K and a batch normalization layer follows it. These two layers are denoted as $\Phi_1(x)$. Through the above steps, the shape of the tensor is compressed to $(1 \times H \times W)$. The Sigmoid activation layer $S(x)$ is followed and attention weight is generated when passing the tensor through $S(x)$. Finally, the attention weight is directly applied to the input tensor χ. This branch's output can be expressed as follows:

$$y_1 = \chi S(\Phi_1(Zpool(\chi))) \tag{3}$$

In the second branch, in this paper, a structure is built to capture interactions between the channel and spatial dimension. First, for the input tensor χ, it is counterclockwise-rotated 90° along the H-axis. Here, $\check{\chi}_2$ is used to represent this rotated vector, whose shape is $(W \times H \times C)$. Then, the tensor passes through the ZPool layer, and the tensor is further transformed into a simplified tensor with the shape of $(2 \times H \times C)$, which is denoted as $\check{\chi}_2^Z$, $\check{\chi}_2^Z = ZPool(\check{\chi}_2)$. The batch-normalizing layer is applied after the typical convolution layer with kernel size K on the $\check{\chi}_2^Z$. This process is denoted as $\Phi_2(x)$, which

means: $\Phi_2(ZPool(\check{\chi}_2))$. The shape of the output is $(1 \times H \times C)$. The Sigmoid activation layer $S(x)$ is followed and attention weight is generated when passing the tensor through $S(x)$; the output after this is $S(\Phi_2(ZPool(\check{\chi}_2)))$. The attention weight is applied to $\check{\chi}_2$ and the result is recorded as $\check{\chi}_{2-A}$, $\check{\chi}_{2-A} = \check{\chi}_2 S(\Phi_2(ZPool(\check{\chi}_2)))$. Finally, to preserve the same shape as the initial input tensor χ, $\check{\chi}_{2-A}$ is rotated 90° counterclockwise along the H axis. This step is denoted as $\widetilde{(\check{\chi}_{2-A})}$. The output of this branch can be written as:

$$y_2 = \check{\chi}_2 S(\Phi_2(\widetilde{ZPool}(\check{\chi}_2))) \tag{4}$$

The last branch is like the second except that the input tensor is counterclockwise-rotated 90° along the W-axis, which is denoted as $\check{\chi}_3$ with the shape of $(H \times C \times W)$, Then, the tensor $\check{\chi}_3$ passes through the Zpool and its shape is reduced to $(2 \times C \times W)$. The output of this layer is denoted as $\check{\chi}_3^Z$, $\check{\chi}_3^Z = ZPool(\check{\chi}_3)$. $\check{\chi}_3^Z$ goes via the batch normalization layer after being passed through the ordinary convolution layer with kernel size K. This process is denoted as $\Phi_3(x)$, which means: $\Phi_3(ZPool(\check{\chi}_3))$. The shape of the output is $(1 \times C \times W)$. Through the Sigmoid activation layer $S(x)$, the output after this is $S(\Phi_3(ZPool(\check{\chi}_3)))$. In the following calculation, the input tensor $\check{\chi}_3$ is then given the attention weight, $\check{\chi}_{3-A}$ is recorded for the outcomes, and $\check{\chi}_{3-A} = \check{\chi}_3 S(\Phi_3(ZPool(\check{\chi}_3)))$. In order to keep the same shape as χ, $\check{\chi}_{3-A}$ is turns 90° degrees in the direction of rotation W; this step is denoted as $\widetilde{(\check{\chi}_{3-A})}$. The output of this branch can be written as:

$$y_3 = \check{\chi}_3 S(\Phi_3(\widetilde{ZPool}(\check{\chi}_3))) \tag{5}$$

In summary, for the input tensor $\chi \in R^{C \times H \times W}$, the output tensor y which is allocated attention weight by the CDA can be expressed as:

$$y = \frac{1}{3}(y_1 + y_2 + y_3) \tag{6}$$

3.4. Measure

The Hamming distance between the two images is written as:

$$dist(b_i, b_j) = \frac{1}{2}(L - < b_i, b_j >) \tag{7}$$

L denotes the hash code's length; $< b_i, b_j >$ represents the inner product of b_i and b_j. It is denoted as:

$$I(b_i, b_j) = < b_i, b_j > = b_i^T b_j \tag{8}$$

It can be deduced that when given the hash codes b_i and b_j corresponding to images x_i and x_j, the conditional probability of the corresponding similarity label s_{ij} is:

$$p(s_{ij}|b_i, b_j) = \begin{cases} S(I(b_i, b_j)), & s_{ij} = 1 \\ 1 - S(I(b_i, b_j)), & others \end{cases} \tag{9}$$

where $S(x)$ is the Sigmoid function.

Finally, in the case of given tag information, the expression of the maximum posterior estimate is:

$$\log p(B|S) \propto \log p(S|B)p(B) = \sum_{i,j=1}^{n} \log p(s_{ij}|b_i, b_j)p(b_i)p(b_j) \tag{10}$$

In this paper, a negative logarithmic likelihood loss function is used to learn hash code. Between similar images, this loss function can, as closely as feasibly possible, reduce the Hamming distance. That is, the hash codes of the same semantic images need to be similar. Additionally, for the dissimilar images, this loss function can encourage the increase in

the Hamming distance between them. The learned hash code maximizes the similarity between images. The negative logarithmic likelihood loss function is expressed as L_1:

$$L_1 = -\sum_{s_{ij} \in S} (s_{ij} I(b_i, b_j) - \log(1 + e^{I(b_i, b_j)})) \tag{11}$$

Meanwhile, for the given hash code b_i, the nth element of hash code b_i is denoted as $b_i^{(n)}$, which obeys the discrete probability distribution {+1, −1}. In order to balance hash codes, this paper expects either +1 or −1 to occur with equal probability. Therefore, mean(x) is introduced to realize the calculation of the average value of elements in the vector [17,29]. Through the formula $\sum_{n=1}^{K} \left| mean(b_i^{(n)}) \right|^2$, the learned hash code has the same number of +1 and −1 as possible. Additionally, hash codes become more discernible. The balanced loss function is denoted as:

$$L_2 = \sum_{n=1}^{K} \left| mean(b_i^{(n)}) \right|^2 \tag{12}$$

Thus, the overall loss function can be obtained:

$$\min_{b_i^{(n)}, b_j^{(n)}} L = L_1 + \alpha L_2 \tag{13}$$

where the importance of loss L_2 is measured by the hyperparameter α. It is verified by experiments that $\alpha = 0.1$ has the best effect.

3.5. Learning

After determining the loss function, by computing the gradient of the loss function, backpropagation is carried out, and the network parameters are optimized. Algorithm 1 illustrates the DCDAH model's training procedure. The parameters of each layer of the network are denoted as ξ. The output of the network is expressed as $\psi(x_i; \xi)$, $W^T \in R^{512 \times L}$ is a representation of the weight matrix transposed, and $\tau \in R^{L \times 1}$ symbolizes the deviation vector. Finally, the mapping from feature representation to the hash code is accomplished using the fully connected layer. Let $U = \{u_i\}_{i=1}^{n}$ be the real-value feature variable learned by the CNN; u_i can be expressed as:

$$u_i = W^T \psi(x_i; \xi) + \tau \tag{14}$$

For the DCDAH model, parameters that can be optimized include ξ, τ, W and b_i. For parameter b_i:

$$b_i = sign(u_i) \tag{15}$$

where $sign(x)$ is represented by:

$$sign(u_i) = \begin{cases} 1 & when\ u_i > 0 \\ -1 & others \end{cases} \tag{16}$$

In the process of backpropagation optimization of b_i, this paper selects the $Htanh(x)$ function instead of $sign(x)$ for backpropagation:

$$Htanh(u_i) = \begin{cases} 1, & u_i > 1 \\ u_i, & -1 < u_i < 1 \\ -1, & u_i < -1 \end{cases} \tag{17}$$

The backpropagation by the $Htanh(x)$ can effectively avoid the problem that the $sign(x)$ function cannot be backpropagated. For parameter ξ:

$$\frac{\partial L}{\partial \psi(x_i;\xi)} = W\frac{\partial L}{\partial u_i} \tag{18}$$

where:

$$\frac{\partial L}{\partial u_i} = \frac{1}{2}\sum_{j:s_{ij}\in S}(a_{ij}-s_{ij})u_j + \frac{1}{2}\sum_{j:s_{ji}\in S}(a_{ji}-s_{ji})u_j + \eta\frac{\partial L_2}{\partial u_i} \tag{19}$$

and:

$$\frac{\partial L_2}{\partial u_i} = \frac{\partial \sum_{n=1}^{K}\left|mean(Htanh(u_i^{(n)}))\right|^2}{\partial u_i} = \begin{cases} 2\sum\limits_{n=1}^{K}\left|mean(Htanh(u_i^{(n)}))\right|, & -1 < u_i < 1 \\ 0, & others \end{cases} \tag{20}$$

For parameter W:

$$\frac{\partial L}{\partial W} = \psi(x_i;\xi)\left(\frac{\partial L}{\partial u_i}\right)^T \tag{21}$$

For parameter τ:

$$\frac{\partial L}{\partial \tau} = \frac{\partial L}{\partial u_i} \tag{22}$$

Algorithm 1. DCDAH.

Input:
Data set $X = \{x_i\}_{i=1}^{N}$ and corresponding image label information data set $Y = \{y_i\}_{i=1}^{M}$.
Output:
The updated parameters ξ, τ, W and b_i.
Initialization:
The parameters of the ResNet18 model are initialized using gaussian distribution.
Repeat:
Randomly extract small batches of image data from the input images;
Carry out forward propagation and calculate $\psi(x_i;\xi)$;
Calculate $u_i = W^T\psi(x_i;\xi) + \tau$ and $b_i = sign(u_i)$;
Calculate the partial derivatives according to (18), (21) and (22);
Carry on the back propagation and update the parameters iteratively.
Until:
Complete a certain number of iterations.

4. Experiments

The real performance of the DCDAH model in two public data sets is demonstrated in this portion.

4.1. Data Sets

1. CIFAR-10 data set: This data set contains 60,000 RGB images. This data set is a single-label data set and it contains 10 categories with 6000 images per category. In the experiment of this paper, 500 images were randomly selected from each category to form the training set, and 100 images from each category were selected to form the test set.

2. NUS-WIDE data set: This data set contains 269,648 images and is a multilabel data set. In this paper, 195,834 images in 21 categories were selected. Then, 500 images were randomly selected from each category to build the training set, and 100 images were randomly selected for each category to build the test set.

4.2. Evaluation Index

In this paper, four evaluation indicators were used to assess the effectiveness of the DCDAH model. They are the following: the mean average precision (mAP), the precision

curve within the Hamming distance 2 (P@H = 2), the precision–recall curve (PR), and the curve of the top 1000 search results (P@N). The experiments used mAP@ALL for the CIFAR10 data set and mAP@5000 for the NUS-WIDE data set in order to accurately compare the different approaches.

In this paper, a comparative experiment was conducted under the premise of a ResNet18 and Pytorch framework. Comparative studies demonstrate the superior retrieval performance of DCDAH. In this paper, algorithms such as DFH [39], DCH [34], DBDH [35], DSDH [40], DSH [33], DTSH [41], HashNet [42] and IDHN [43] were selected for performance comparison.

In Table 2, the experimental setup for this paper is displayed. In this paper, the identical training set and test set were used by all methods. Additionally, the learning rate was set to 5×10^{-5}, the mini batch size of pictures was set to 128, and the weight decay parameter was set to 1×10^{-5}. The network model was optimized by the root mean square prop (RMSProp).

Table 2. Experimental environment.

Item	Configuration
OS	Ubuntu 16.04(×4)
GPU	Tesla V100

As shown in Table 3, in this paper, A^2 Attention [44], BAM [45] and CBAM [28] were selected from the experimental comparison algorithms to compare the computational cost with our module. It can be seen from the experimental results that compared with the mainstream network framework, DCDAH had low flops. At the same time, our method introduced the least number of parameters, which verifies the lightweight nature of the CDA attention mechanism.

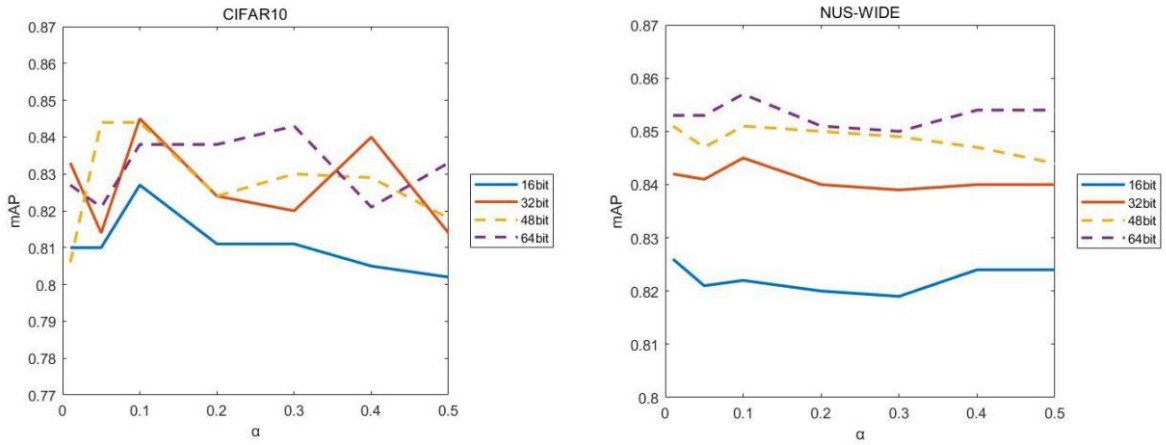
Table 3. Comparison of the complexity of DCDAH with other mainstream models in terms of network parameters and floating-point operations per second (FLOPs).

Model	Parameters	FLOPs
ResNet18	22.36 M	7.29 G
ResNet18 + A^2 Attention	22.62 M	7.31 G
ResNet18 + BAM	22.44 M	7.30 G
ResNet18 + CBAM	22.40 M	7.30 G
ours	22.36 M	7.29 G

4.3. Hyperparameter Analysis

In Equation (13), the importance of loss L_2 is measured by the hyperparameter α. In this section, the CIFAR10 and NUS-WIDE data sets were used in the parameter modification experiment.

Figure 3 shows the effect of adjusting the parameter α on mAP in different data sets. For the experimental results on the CIFAR10 data set, it is obvious that when the value of α is greater than 0.1 and keeps increasing, the mAP of DCDAH decreases with different length hash codes. The same happens when the value of α is less than 0.1. For the NUS-WIDE data set, changes in α have less effect on the mAP of DCDAH. However, in the experimental data from Table 4 with hash code lengths of 32 bits and 64 bits, respectively, when $\alpha = 0.1$, the mAP is in a peak state. To sum up, the value of parameter α is selected as 0.1.



(**a**) mAP on CIFAR10 (**b**) mAP on NUS-WIDE

Figure 3. The corresponding mAP value when α takes different values.

Table 4. mAP of different α.

α	CIFAR10 (mAP@ALL)				NUS-WIDE (mAP@5000)			
	16 bit	**32 bit**	**48 bit**	**64 bit**	**16 bit**	**32 bit**	**48 bit**	**64 bit**
0.01	0.810	0.833	0.806	0.827	0.826	0.842	0.851	0.853
0.05	0.810	0.814	0.844	0.821	0.821	0.841	0.847	0.853
0.1	0.827	0.845	0.844	0.838	0.822	0.845	0.851	0.857
0.2	0.811	0.824	0.824	0.838	0.820	0.840	0.850	0.851
0.3	0.811	0.820	0.830	0.843	0.819	0.839	0.849	0.850
0.4	0.805	0.840	0.829	0.821	0.824	0.840	0.847	0.854
0.5	0.802	0.814	0.818	0.833	0.824	0.840	0.844	0.854

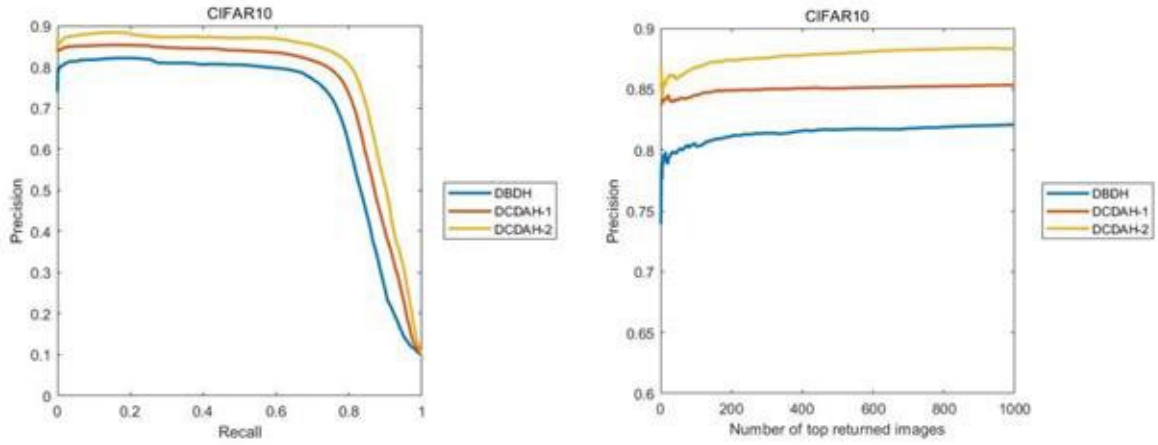
4.4. Ablation Experiments

In this paper, we extended ResNet18 with the CDA attention module to enhance the image representation capability. To verify the validity of the CDA module, we conducted a comparative test on DCDAH and its related variants. DBDH was selected to be the baseline. DCDAH-1: DCDAH chooses ResNet18 as a backbone. DCDAH-2: DCDAH with ResNet18 which adds DCA. In this paper, we carried out a comparative experiment on the CIFAR10 data set and analyzed experimental data about a hash code length of 32 bits. The experimental results are shown in Table 5:

Table 5. Comparative experiment.

Framework	**DBDH**	**DCDAH-1**	**DCDAH-2**
AlexNet	√		
ResNet18		√	√
DCA Module			√
mAP(32 bit)	0.773	0.814	0.845

As shown in Figure 4, this paper shows the PR curve and P@N curve of several variants. The mAP value was increased by 3.1% when the CDA module was added to DCDAH-1. From the PR curves in Figure 4, the PR curve of DCDAH-2 completely wrapped the PR curve of the other two variants. Additionally, it can be proved by the P@N curves that the curve was significantly higher than others after using the CDA model, which further proves the effectiveness of CDA.



(**a**) PR @ 32 bits (**b**) P@N with 32 bits

Figure 4. PR and P@N curves on CIFAR-10 when the hash code length is 32 bits.

4.5. Analysis of Experimental Results

Table 6 displays the experimental findings from the mAP comparison on the CIFAR10 and NUS-WIDE data sets:

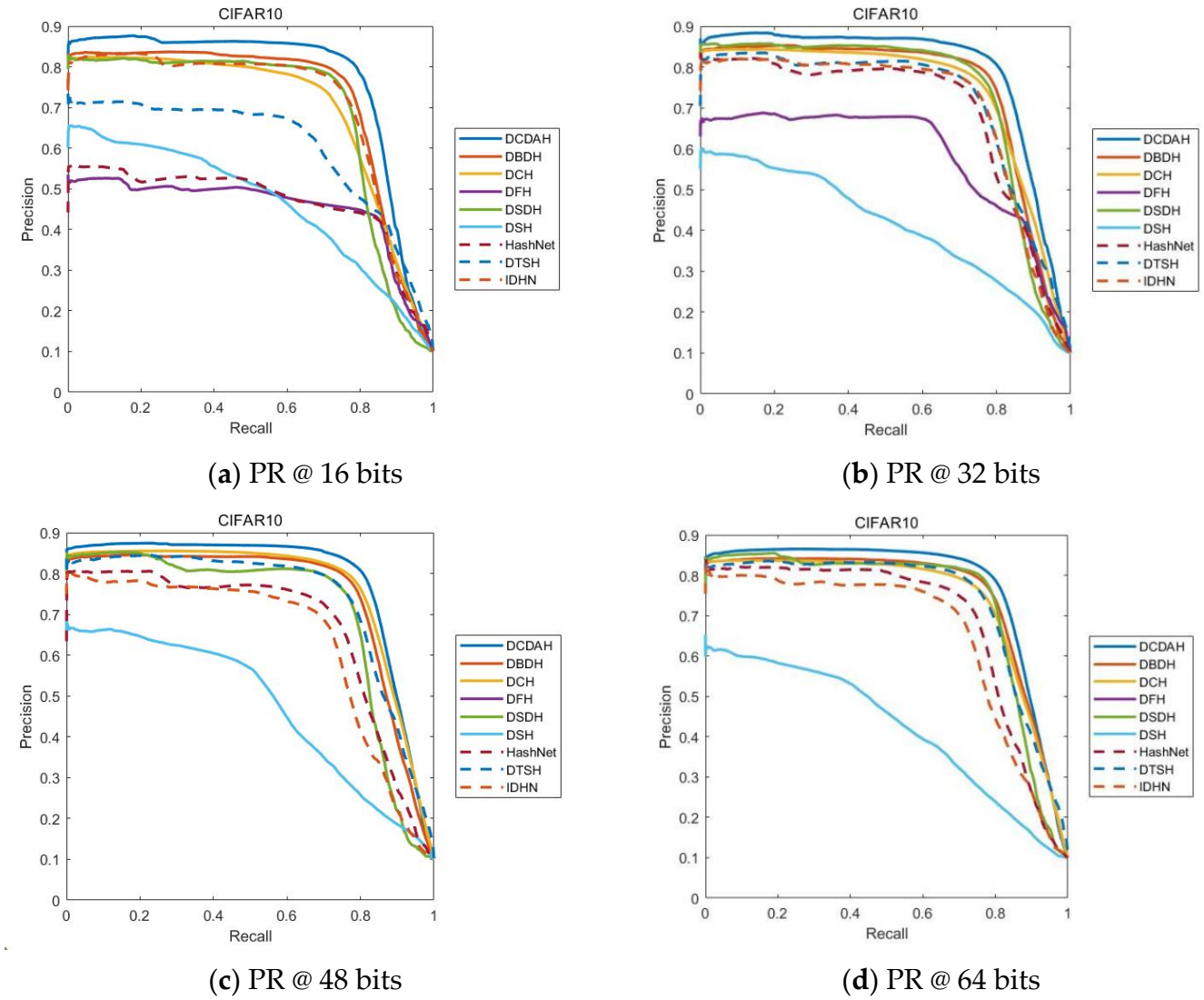
Table 6. Results of comparative experiments.

Method	CIFAR-10 (mAP@ALL)				NUS-WIDE (mAP@5000)			
	16 bit	**32 bit**	**48 bit**	**64 bit**	**16 bit**	**32 bit**	**48 bit**	**64 bit**
DCDAH	0.827	0.845	0.844	0.838	0.822	0.845	0.851	0.857
DBDH	0.798	0.814	0.815	0.820	0.811	0.834	0.839	0.849
DCH	0.756	0.805	0.821	0.800	0.785	0.799	0.807	0.798
DFH	0.584	0.680	0.795	0.822	0.784	0.815	0.806	0.817
DSH	0.555	0.479	0.562	0.499	0.676	0.759	0.785	0.783
DSDH	0.765	0.806	0.812	0.802	0.768	0.782	0.773	0.765
DTSH	0.695	0.784	0.804	0.805	0.815	0.834	0.837	0.832
HashNet	0.543	0.646	0.746	0.764	0.718	0.802	0.809	0.742
IDHN	0.777	0.777	0.728	0.736	0.786	0.766	0.727	0.566

For the CIFAR10 data set, among the hash code of different lengths, compared with DBDH, the mAP of DCDAH could achieve boosts of 2.9%, 3.1%, 2.9% and 1.8%. For the NUS-WIDE data set, compared with DBDH, the mAP of DCDAH in different lengths of hash code could improve by 1.1%, 1.1%, 1.2% and 0.8%. Therefore, compared with the best mAP, DCDAH achieved an increase of 2.7% and 1% in the average mAP for different bits on two data sets. Compared with the DSDH, which is a classic method, for the CIFAR10 data set, DCDAH achieved 6.2%, 3.9%, 3.2% and 3.3% improvement. On the NUS-WIDE data set, the promotion rates of the mAP in the different bits were 5.4%, 6.3%, 7.8% and 9.2%. The experimental findings demonstrate that DCDAH is more robust.

The PR curve takes recall as the abscissa and precision as the ordinate. One can tell which strategy performs better by observing whether the PR curves of the two methods completely encircle one another. Therefore, we chose PR curve to compare the performance of algorithms.

The PR curves are shown in Figures 5 and 6; for the data set CIFAR10, it is evident that the curve of the DCDAH method wraps the other methods in the results of different hash code lengths. Therefore, among all the contrasting methods, DCDAH had the best performance. For the data set NUS-WIDE, due to the complexity of the data set, the improvement achieved by DCDAH was not as obvious as the improvement in the CIFAR10 data set. However, the performance of DCDAH was still the best among these comparison algorithms.



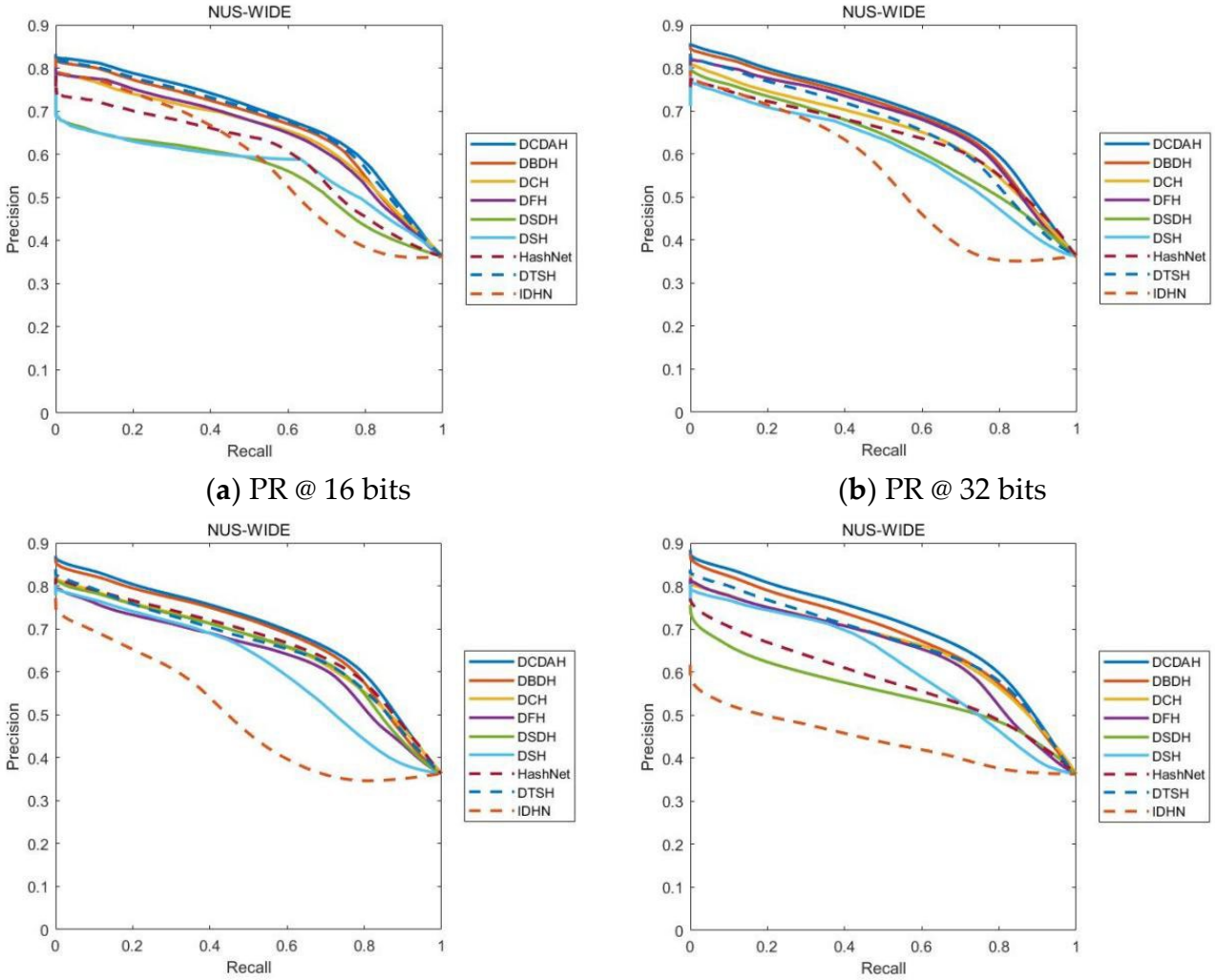
(**a**) PR @ 16 bits (**b**) PR @ 32 bits

(**c**) PR @ 48 bits (**d**) PR @ 64 bits

Figure 5. PR curves of all algorithms on CIFAR10 when the hash code takes 4 different lengths.

To achieve the goal of only O(1) searches for the Hamming rank, P@H = 2 evaluation index is very important for the retrieval of binary space. Figure 7 illustrates that DCDAH achieves the maximum accuracy on the two data sets when comparing the results of P@H = 2 for all methods. In this paper, we selected hash code lengths of 16 bits, 32 bits, 48 bits and 64 bits for comparison experiments, and drew the curves of all comparison algorithms. The results validate that DCDAH can concentrate on more relevant points than all compared methods.

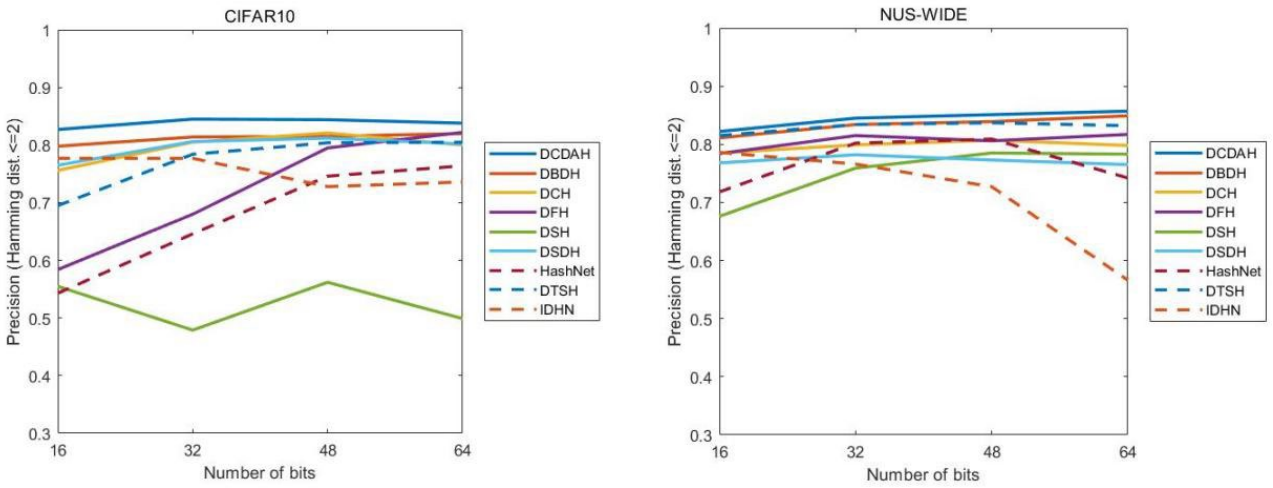
The P@N curve is widely used as another important evaluation index. The experimental choice returns the accuracy of the first 1000 images. Figures 8 and 9 show the P@N curves of the CIFAR-10 and NUS-WIDE data sets, which proves that DCDAH achieves better accuracy than other algorithms. Specifically, in Figure 8a, the P@N curve of DCDAH is significantly higher than DBDH.



(**a**) PR @ 16 bits (**b**) PR @ 32 bits

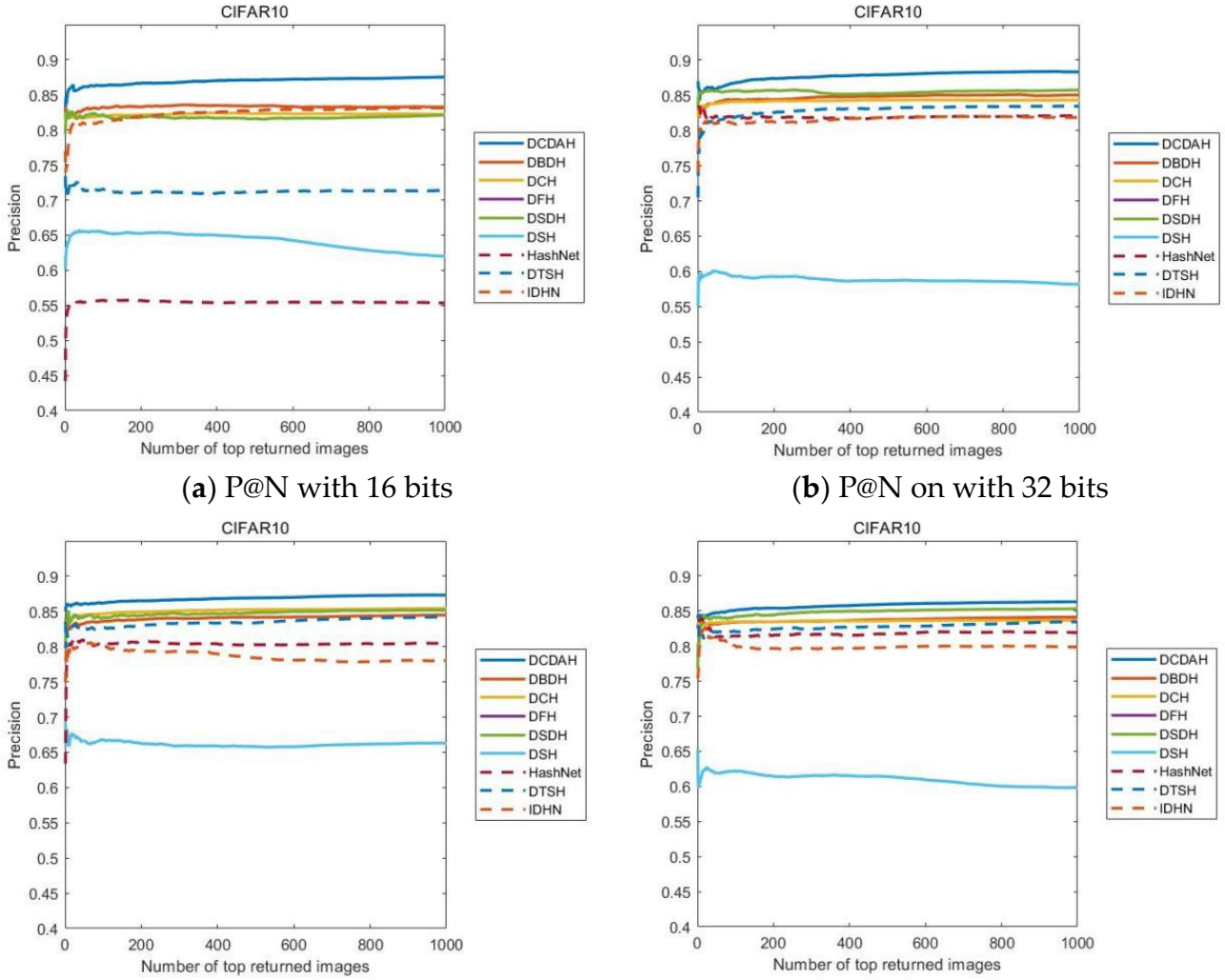
(**c**) PR @ 48 bits (**d**) PR @ 64 bits

Figure 6. PR curves of all algorithms on NUS-WIDE when the hash code takes 4 different lengths.



(**a**) P@H = 2 on CIFAR10 (**b**) P@H = 2 on NUSWIDE

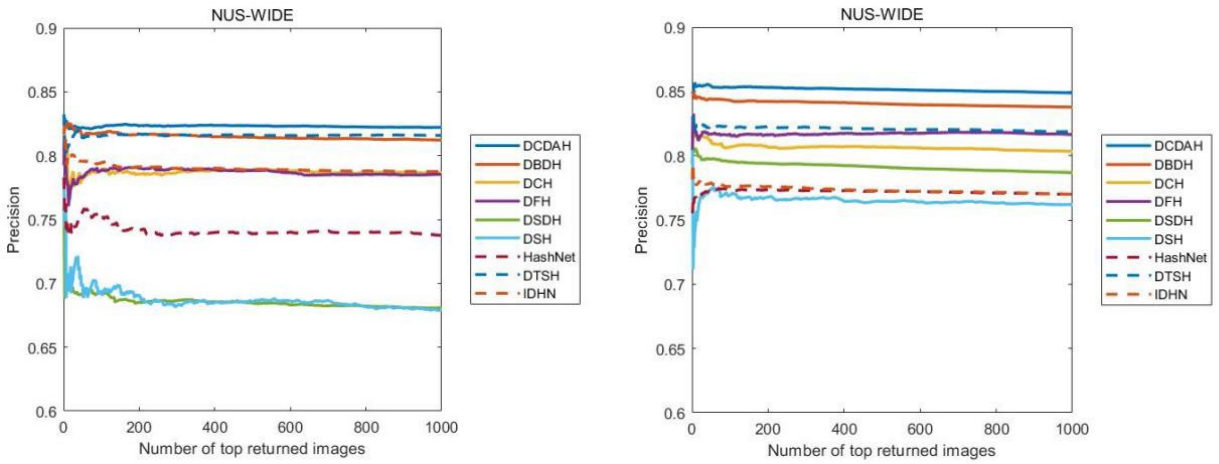
Figure 7. P@H = 2 on two data sets.



(**a**) P@N with 16 bits (**b**) P@N on with 32 bits

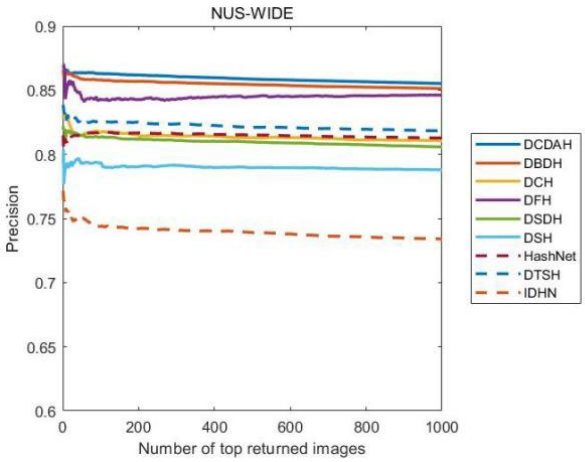
(**c**) P@N with 48 bits (**d**) P@N with 64 bits

Figure 8. The P@N curves of all algorithms on CIFAR10 when the hash code takes 4 different lengths.

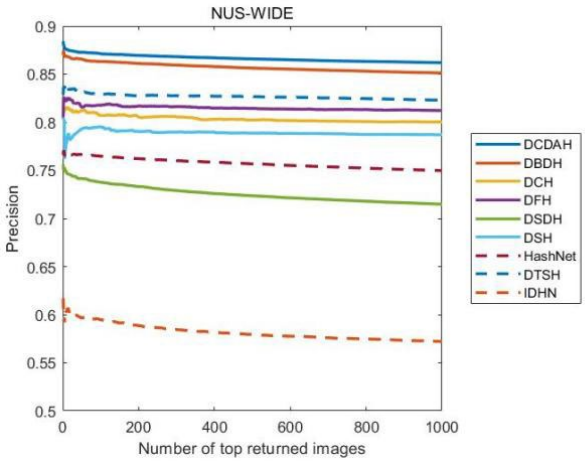


(**a**) P@N with 16 bits (**b**) P@N on with 32 bits

Figure 9. *Cont*.



(**c**) P@N with 48 bits

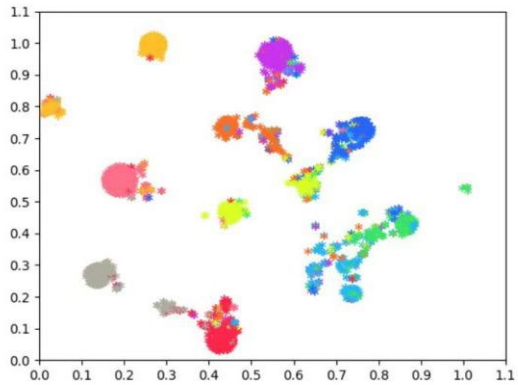


(**d**) P@N on with 64 bits

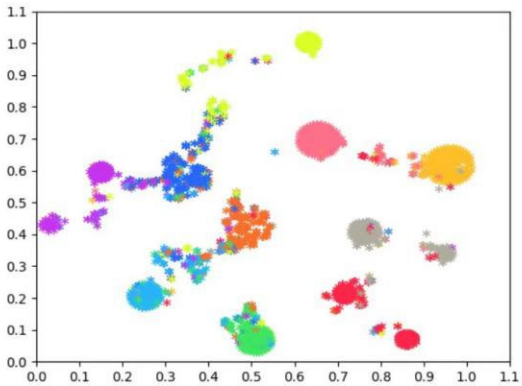
Figure 9. The P@N curves of all algorithms on NUS-WIDE when the hash code takes 4 different lengths.

As shown in Figure 10, this paper makes a visual comparison of the hash codes generated by DCDAH and DBDH on the CIFAR10 data set.

Comparing the results of DBDH, it can be seen from Figure 10 that the hash codes generated by DCDAH are more compact within the same class, and the hash code distance between different classes is also enlarged. This result shows that the hash code generated by the DCDAH is more discriminating and can more accurately represent the image.



(**a**) DCDAH



(**b**) DBDH

Figure 10. Hash code visualization results.

5. Conclusions

How to improve accuracy while ensuring high retrieval speed is a hot research topic in the field of hashing image retrieval. The recently proposed hashing algorithms solve the problem of insufficient feature extraction by integrating an attention module. However, channel and spatial attention currently have limited performance improvement due to imbalanced weight distribution. Additionally, many parameters will be introduced to increase complexity of the model. Therefore, this paper proposes the DCDAH method to improve retrieval performance by integrating CDA. The CDA module effectively captures and emphasizes the interaction information between different dimensions of a feature map through its ingenious branch structure, which can make the image representation more accurate. Moreover, this paper introduces Zpool to reduce the dimension of tensors, which allows the CDA module to accomplish the task of attention weight assignment with negligible computational overhead. This paper proves the validity of the DCDAH

framework through many experiments. Experiments on CIFAR10 and NUS-WIDE showed that DCDAH is superior to the recently proposed hashing methods.

Author Contributions: Conceptualization, Z.C.; methodology, Z.C.; software, Z.C.; validation, Y.L.; investigation, Y.L.; resources, Y.L.; writing—original draft preparation, Z.C.; writing—review and editing, Y.L.; visualization, Z.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Science Foundation of China under Grant U1903213, Tianshan Innovation Team of Xinjiang Uygur Autonomous Region grant number 2020D14044.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available data sets were analyzed in this study. The data sets can be obtained from: [http://www.cs.toronto.edu/~kriz/cifar.html], [https://lms.comp.nus.edu.sg/wp-content/uploads/2019/research/nuswide/NUSWIDE.html] (all accessed on 3 July 2022).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Wei, Y.; Zhao, Y.; Lu, C.; Wei, S.; Liu, L.; Zhu, Z.; Yan, S. Cross-Modal Retrieval With CNN Visual Features: A New Baseline. *IEEE Trans. Cybern.* **2017**, *47*, 449–460. [CrossRef] [PubMed]
2. Chaudhuri, U.; Banerjee, B.; Bhattacharya, A.; Datcu, M. Attention-Driven Cross-Modal Remote Sensing Image Retrieval. In Proceedings of the 2021 IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Brussels, Belgium, 11–16 July 2021; pp. 4783–4786.
3. Misra, M.; Nalamada, T.; Uppili Arasanipalai, A.; Hou, Q. Rotate to Attend: Convolutional Triplet Attention Module. WACV, 2021. pp. 3138–3147. Available online: https://openaccess.thecvf.com/content/WACV2021/html/Misra_Rotate_to_Attend_Convolutional_Triplet_Attention_Module_WACV_2021_paper.html?ref=https://coder.social (accessed on 12 October 2022).
4. Pachori, S.; Deshpande, A.; Raman, S. Hashing in the zero-shot framework with domain adaptation. *Neurocomputing* **2018**, *275*, 2137–2149. [CrossRef]
5. Venkateswara, H.; Eusebio, J.; Chakraborty, S.; Panchanathan, S. Deep Hashing Network for Unsupervised Domain Adaptation. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 5385–5394.
6. Du, A.; Cheng, S.; Wang, L. Low-Rank Semantic Feature Reconstruction Hashing for Remote Sensing Retrieval. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 1–5. [CrossRef]
7. Wang, B.; Lu, X.; Zheng, X.; Li, X. Semantic descriptions of high-resolution remote sensing images. *IEEE Trans. Geosci. Remote Sens. Lett.* **2019**, *16*, 1274–1278. [CrossRef]
8. Guo, Y.; Ding, G.; Liu, L.; Han, J.; Shao, L. Learning to hash with optimized anchor embedding for scalable retrieval. *IEEE Trans. Image Process.* **2017**, *26*, 1344–1354. [CrossRef]
9. Bergamo, A.; Torresani, L.; Fitzgibbon, A. Picodes: Learning a Compact Code for Novel-Category Recognition. In *NIPS*. 2011, pp. 2088–2096. Available online: https://proceedings.neurips.cc/paper/2011/hash/1896a3bf730516dd643ba67b4c447d36-Abstract.html (accessed on 12 October 2022).
10. Liu, D.; Shen, J.; Xia, Z.; Sun, X. A content-based image retrieval scheme using an encrypted difference histogram in cloud computing. *Information* **2017**, *8*, 96. [CrossRef]
11. Bronstein, M.M.; Bronstein, A.M.; Michel, F.; Paragios, N. Data fusion through cross-modality metric learning using similarity-sensitive hashing. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), San Francisco, CA, USA, 13–18 June 2010; pp. 3594–3601.
12. Webb, B.S.; Dhruv, N.T.; Solomon, S.G. Early and late mechanisms of surround suppression in striate cortex of macaque. *Neuroscience* **2005**, *25*, 11666–11675. [CrossRef]
13. Vedaldi, A.; Zisserman, A. Efficient Additive Kernels via Explicit Feature Maps. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *34*, 480–492. [CrossRef]
14. Oquab, M.; Bottou, L.; Laptev, I.; Sivic, J. Learning and Transferring Mid-level Image Representations Using Convolutional Neural Networks. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014; pp. 1717–1724.
15. Li, P.; Han, L. Hashing nets for hashing: A quantized deep learning to hash framework for remote sensing image retrieval. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 7331–7345. [CrossRef]
16. Lin, K.; Lu, J.; Chen, C.; Zhou, J. Learning Compact Binary Descriptors with Unsupervised Deep Neural Networks. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 1183–1192.

17. Deng, C.; Yang, E.; Liu, T.; Li, J.; Liu, W.; Tao, D. Unsupervised Semantic-Preserving Adversarial Hashing for Image Search. *IEEE Trans. Image Process.* **2019**, *28*, 4032–4044. [CrossRef]
18. Zhang, H.; Gu, Y.; Yao, Y.; Zhang, Z.; Liu, L.; Zhang, J.; Shao, L. Deep Unsupervised Self-Evolutionary Hashing for Image Retrieval. *IEEE Trans. Multim.* **2021**, *23*, 3400–3413. [CrossRef]
19. Zhang, J.; Peng, Y. SSDH: Semi-Supervised Deep Hashing for Large Scale Image Retrieval. *IEEE Trans. Circuits Syst. Video Technol.* **2019**, *29*, 212–225. [CrossRef]
20. Zheng, S.; Wang, L.; Du, A. Deep Semantic-Preserving Reconstruction Hashing for Unsupervised Cross-Modal Retrieval. *Entropy* **2020**, *22*, 1266.
21. Zhu, H.; Gao, S. Locality Constrained Deep Supervised Hashing for Image Retrieval. In Proceedings of the 2017 International Joint Conference on Artifificial Intelligence, Melbourne, Australia, 19–25 August 2017; pp. 3567–3573.
22. Liu, C.; Ma, J.; Tang, X.; Liu, F.; Zhang, X.; Jiao, L. Deep Hash Learning for Remote Sensing Image Retrieval. *IEEE Trans. Geosci. Remote. Sens.* **2021**, *59*, 3420–3443. [CrossRef]
23. Yan, C.; Gong, B.; Wei, Y.; Gao, Y. Deep Multi-View Enhancement Hashing for Image Retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 1445–1451. [CrossRef]
24. Gong, Y.; Lazebnik, S.; Gordo, A.; Perronnin, F. Iterative Quantization: A Procrustean Approach to Learning Binary Codes for Large-Scale Image Retrieval. *IEEE Trans. Pattern Anal. Mach Intell.* **2013**, *35*, 2916–2929. [CrossRef] [PubMed]
25. Lu, J.; Hu, J.; Zhou, J. Deep Metric Learning for Visual Understanding: An Overview of Recent Advances. *IEEE Signal Process.* **2017**, *34*, 76–84. [CrossRef]
26. Long, J.; Wei, X.; Qi, Q.; Wang, Y. A deep hashing method based on attention module for image retrieval. In Proceedings of the 2020 13th International Conference on Intelligent Computation Technology and Automation (ICICTA), Xi'an, China, 24–25 October 2020; pp. 284–288.
27. Cheng, S.; Wang, L.; Du, A.; Li, Y. Bidirectional Focused Semantic Alignment Attention Network for Cross-Modal Retrieval. In Proceedings of the ICASSP 2021–2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, 6–11 June 2021; pp. 4340–4344.
28. Woo, S.; Park, J.; Lee, J.; Keweon, I. CBAM: Convolutional Block Attention Module. ECCV. 2018, pp. 3–19. Available online: https://openaccess.thecvf.com/content_ECCV_2018/html/Sanghyun_Woo_Convolutional_Block_Attention_ECCV_2018_paper.html (accessed on 12 October 2022).
29. Shen, F.; Xu, Y.; Liu, L.; Yang, Y.; Huang, Z.; Shen, H.T. Unsupervised Deep Hashing with Similarity-Adaptive and Discrete Optimization. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 3034–3044. [CrossRef]
30. Luo, C.C. A novel web attack detection system for internet of things via ensemble classifcation. *IEEE Trans. Indus.* **2020**, *17*, 5810–5818. [CrossRef]
31. Liu, W.; Wang, J.; Ji, R.; Jiang, Y.; Chang, S. Supervised hashing with kernels. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 16–21 June 2012; pp. 2074–2081.
32. Xia, R.; Pan, Y.; Lai, H.; Liu, C.; Yan, S. Supervised Hashing for Image Retrieval via Image Representation Learning. *In* AAAI. 2014, pp. 2156–2162. Available online: https://web.archive.org/web/*/http://www.aaai.org/ocs/index.php/AAAI/AAAI14/paper/view/8137 (accessed on 12 October 2022).
33. Liu, H.; Wang, R.; Shan, S.; Chen, X. Deep Supervised Hashing for Fast Image Retrieval. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 2064–2072.
34. Cao, Y.; Long, M.; Liu, B.; Wang, J. Deep Cauchy Hashing for Hamming Space Retrieval. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 1229–1237.
35. Zheng, X.; Zhang, Y. Deep balanced discrete hashing for image retrieval. *Neurocomputing* **2020**, *403*, 224–236. [CrossRef]
36. Li, X.; Xu, M.; Xu, J.; Weise, T.; Zou, L.; Sun, F.; Wu, Z. Image Retrieval Using a Deep Attention-Based Hash. *IEEE Access* **2020**, *8*, 142229–142242. [CrossRef]
37. Jin, L.; Shu, X.; Li, K.; Li, Z.; Qi, G.; Tang, J. Deep Ordinal Hashing with Spatial Attention. *IEEE Trans. Image Process.* **2019**, *28*, 2173–2186. [CrossRef] [PubMed]
38. Yang, W.; Wang, L.; Cheng, S. Deep parameter-free attention hashing for image retrieval. *Sci. Rep.* **2022**, *12*, 7082. [CrossRef] [PubMed]
39. Li, Y.; Pei, W.; Zha, Y.; Gemert, J. Push for Quantization: Deep Fisher Hashing. *BMVC* **2019**, *21*. Available online: https://bmvc2019.org/wp-content/uploads/papers/0938-paper.pdf (accessed on 12 October 2022).
40. Li, Q.; Sun, Z.; He, R.; Tan, T. Deep Supervised Discrete Hashing. *Adv. Neural Inf. Processing Syst.* **2017**, *30*, 2482–2491.
41. Wang, X.; Shi, Y.; Kitani, K. Deep Supervised Hashing with Triplet Labels. In *Asian Conference on Computer Vision*; Springer: Cham, Switzerland, 2016; pp. 70–84.
42. Cao, Z.; Long, M.; Wang, J.; Yu, P. HashNet: Deep Learning to Hash by Continuation. *ICCV* **2017**, 5609–5618. Available online: https://www.computer.org/csdl/proceedings-article/iccv/2017/1032f609/12OmNqGA5a7 (accessed on 12 October 2022).
43. Zhang, Z.; Zou, Q.; Lin, Y.; Chen, L.; Wang, S. Improved Deep Hashing with Soft Pairwise Similarity for Multi-label Image Retrieval. *IEEE Trans. Multim.* **2020**, *22*, 540–553. [CrossRef]
44. Chen, Y.; Kalantidis, Y.; Li, J.; Yan, S.; Feng, J. A2-Nets: Double Attention Networks. CORR 2018. Available online: https://proceedings.neurips.cc/paper/2018/hash/e165421110ba03099a1c0393373c5b43-Abstract.html (accessed on 12 October 2022).
45. Woo, S.; Park, J.; Lee, J.; Keweon, I. BAM: Bottleneck Attention Module. *BMVC* **2018**, 147. Available online: http://bmvc2018.org/contents/papers/0092.pdf (accessed on 12 October 2022).