frontiers in Neurorobotics

ORIGINAL RESEARCH
published: 18 October 2021
doi: 10.3389/fnbot.2021.728161

Dual Attention Triplet Hashing Network for Image Retrieval

Zhukai Jiang, Zhichao Lian and Jinping Wang*

School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China

In recent years, learning-based hashing techniques have proven to be efficient for large-scale image retrieval. However, since most of the hash codes learned by deep hashing methods contain repetitive and correlated information, there are some limitations. In this paper, we propose a Dual Attention Triplet Hashing Network (DATH). DATH is implemented with two-stream ConvNet architecture. Specifically, the first neural network focuses on the spatial semantic relevance, and the second neural network focuses on the channel semantic correlation. These two neural networks are incorporated to create an end-to-end trainable framework. At the same time, in order to make better use of label information, DATH combines triplet likelihood loss and classification loss to optimize the network. Experimental results show that DATH has achieved the state-of-the-art performance on benchmark datasets.

Keywords: supervised deep hashing, dual network, attention mechanism, image retrieval, loss function

Edited by:
Hang Su,
Fondazione Politecnico di Milano, Italy

Reviewed by:
Emmanuel Dauce,
Centrale Marseille, France
Zhiyong Cheng,
Qilu University of Technology, China
Song Weiwei,
Peng Cheng Laboratory, China

****Correspondence:***
Zhichao Lian
lzcts@163.com

Received: *21 June 2021*
Accepted: *23 September 2021*
Published: *18 October 2021*

Citation:
Jiang Z, Lian Z and Wang J (2021) Dual Attention Triplet Hashing Network for Image Retrieval. Front. Neurorobot. 15:728161. doi: 10.3389/fnbot.2021.728161

INTRODUCTION

Image retrieval is a popular problem of image matching, where the similar images are retrieved from a database with respect to a given query image. Basically, the similarity between the query image and the database images is used to rank the database images in decreasing order of similarity (Dubey, 2021). The traditional content-based image retrieval technology uses the nearest neighbor retrieval to achieve better results when facing small data sets. However, in the context of large-scale image data, considering the data storage space, query speed and other retrieval problems, content-based image retrieval technology can no longer meet the requirements. Because of its small storage space and fast query speed, hashing method is quickly applied to image retrieval. Traditional hash methods, such as KSH (Liu et al., 2012), ITQ (Gong and Lazebnik, 2011), and DSH (Jin et al., 2014), use manually extracted features and separate the feature extraction step from the learning step of hash function. Not only is the process cumbersome, but also the retrieval accuracy of the obtained hash code is generally low.

As deep learning has shown its superior performance in computer vision applications, researchers try to introduce the technique into image retrieval tasks (Oquab et al., 2014; You et al., 2016; Chen et al., 2017). Deep hashing methods are gradually proposed, such as DHN (Zhu et al., 2016), HashNet, ADSH (Jiang and Li, 2018), GAH (Huang et al., 2019), DAGH (Chen et al., 2019), and SCADH (Cui et al., 2020), which have been proved to significantly improve the retrieval speed and accuracy for large-scale multimedia retrieval.

Although deep hashing methods have made great progress, these methods have some limitations on generating short hash codes. Similar images may contain completely different background images, and different images may contain the same image background. Thus, the learned hash codes may not contain important information used to describe the key features of the image.

In the training process of supervised deep hashing algorithm, supervised information is given in the form of pairwise labels or triplet labels, a special case of ranking labels. In recent years, researchers think that triplet labels inherently contain richer information than pairwise labels (Wang et al., 2017). Therefore, current supervised methods are mostly trained using a triplet loss function made up of three images as: (i) an anchor image; (ii) a positive image that is similar to the anchor; and (iii) a negative image that is dissimilar to the anchor, such as Zhou et al. (2019), Fang et al. (2021), and (Zhu et al., 2021). Each triplet label can be naturally decomposed into two pairwise labels. A triplet label ensures that in the learned hash code space, the query image is close to the positive image and far from the negative image simultaneously. However, a pairwise label can only ensure that one constraint is observed. Triplet labels explicitly provide a notion of relative similarities between images while pairwise labels can only encode that implicitly (Wang et al., 2017). At the same time, the classification information only plays a role in deep neural network image representation, and seldom directly classifies the hash code. Therefore, in order to make full use of the label information to learn the hash code, combining the triplet label loss and classification loss is worthy of attention.

In addition, attentional mechanisms have been widely used in natural language processing and some aspects of computer vision, such as semantic segmentation. Attention mechanism can focus on the main information of the object and restrain the useless information of the object. In the field of deep hash retrieval, we also need attention mechanism to enhance the feature representation ability of deep networks, so as to reduce the interference of image useless information on generating hash code.

In order to solve the above problems, this paper proposes a Dual Attention Triplet Hashing Network (DATH), and extensive experimental results on benchmark datasets show that DATH outperforms the state-of-the-art supervised hashing methods. The contributions of this work are summarized as follows:

1. We propose a novel Dual Attention Triplet Hashing Network (DATH). The two neural networks focus on spatial semantic relevance and channel semantic relevance respectively, and then combine the two neural networks to create a unified framework for end-to-end training. To the best of our knowledge, this is the first deep hashing method that utilizes dual attention mechanism to learn the hash codes.
2. In order to guarantee the quality of the final hash codes and fully utilize the supervised information, DATH combines the classification loss function with the triplet likelihood loss function to optimize the generation of hash codes.
3. Extensive experiments on widely-used benchmark datasets have been conducted. The results demonstrate that our method outperforms current state-of-the-art methods for image retrieval, which indicates the effectiveness of the proposed method.

RELATED WORK

Existing hashing methods can be divided into two categories: data-independent (Andoni and Indyk, 2006) and data-dependent methods (Xie et al., 2017). Locality Sensitive Hashing (LSH) (Gionis et al., 1999) is one of the most representative data-independent hashing methods. LSH is unstable and needs longer hash codes to achieve better performance. Due to the limitations of the data-independent methods, current researchers focus on data-dependent methods, which enable the learned hash function to maintain the semantic relationship between images based on a given data set.

Data-dependent methods can be further divided into unsupervised and supervised methods. Unsupervised hashing methods train unlabeled data to learn hash functions that encode data into binary codes. Spectral Hashing (SH) (Weiss et al., 2009), Iterative Quantization (ITQ) (Gong and Lazebnik, 2011) and Principle Component Analysis Hashing (PCAH) (Wang et al., 2012) are traditional unsupervised linear methods. Supervised hashing methods make full use of label information to obtain better performance than unsupervised hashing methods. Canonical Correlation Analysis with Iterative Quantization (CCA-ITQ) (Gong and Lazebnik, 2011) is a supervised version of ITQ, which uses CCA to reduce the dimension of data and map the data to the vertex of binary hypercube to reduce the quantization error. Supervised Discrete Hashing (SDH) (Shen et al., 2015) introduces an auxiliary variable to reformulate the objective function so that it can be solved effectively by regularization algorithm.

Recently, deep convolutional neural networks have yielded remarkable results on many computer vision tasks, and hashing methods based CNN have made great progress. Deep Hashing Network (DHN) (Zhu et al., 2016) uses AlexNet (Krizhevsky et al., 2017) to learn the image representation of hash codes, and uses pairwise cross entropy loss function to maintain similarity learning and pairwise quantization loss function to control the quality of hash codes. Deep Triplet Supervised Hashing (DTSH) (Wang et al., 2017) uses triplet likelihood loss to learn image features and hash codes. DHCNN (Song et al., 2019) and Deep Uniqueness-Aware Hashing (DUAH) (Wu et al., 2018) combine contrastive loss and classification loss to solve the problem of large-scale remote sensing image retrieval and fine-grained multi-label image retrieval, respectively. Deep learning to hash by continuation (HashNet) (Cao et al., 2017) can effectively learn binary hash codes from unbalanced similarity data. Gradient Attention Hashing (GAH) (Huang et al., 2019) proposes a gradient attention mechanism, which is integrated in a deep hashing architecture to address the aforementioned learning issue, and thus accelerate the learning process. Asymmetric

Deep Supervised Hashing (ADSH) (Jiang and Li, 2018) treats the query points and database points in an asymmetric way, learns a deep hash function only for query points while the hash codes for database points are directly learned. Although the above methods achieve good retrieval performance, they do not deal with irrelevant features in the image. Deep Ordinal Hashing with Spatial Attention (DOH) (Jin et al., 2019) designs a subnetwork to build rank structure by jointly exploring the local spatial information from FCN and the global semantic information from CNN. Here the spatial attention model is designed to capture the local spatial information by selectively learning well-specified locations closely related to target objects. In terms of practical application, Scalable Deep Hashing (SCADH) (Cui et al., 2020) formulate a unified scalable deep hash learning framework which explores the weak but free supervision of discriminative user tags that are commonly accompanied with social images. As an important branch of hashing methods, Deep Collaborative Multi-view Hashing (DCMVH) (Zhu et al., 2020) associates different layers with instances and paired semantic tags to solve the multi-view hashing problem.

APPROACH

Network Architecture

To address the limitations of previous learning-based hashing methods, we propose a novel deep hashing method. For fair comparison with other deep hashing methods, we use AlexNet

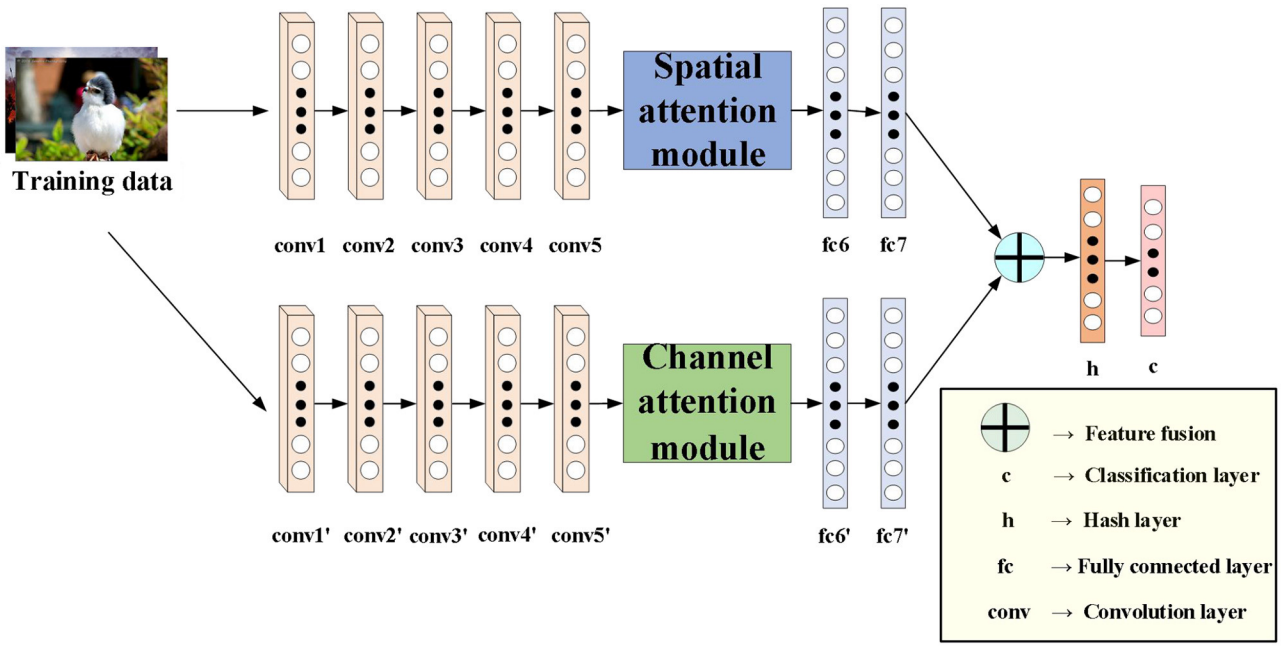

FIGURE 1 | The network architecture of DATH.

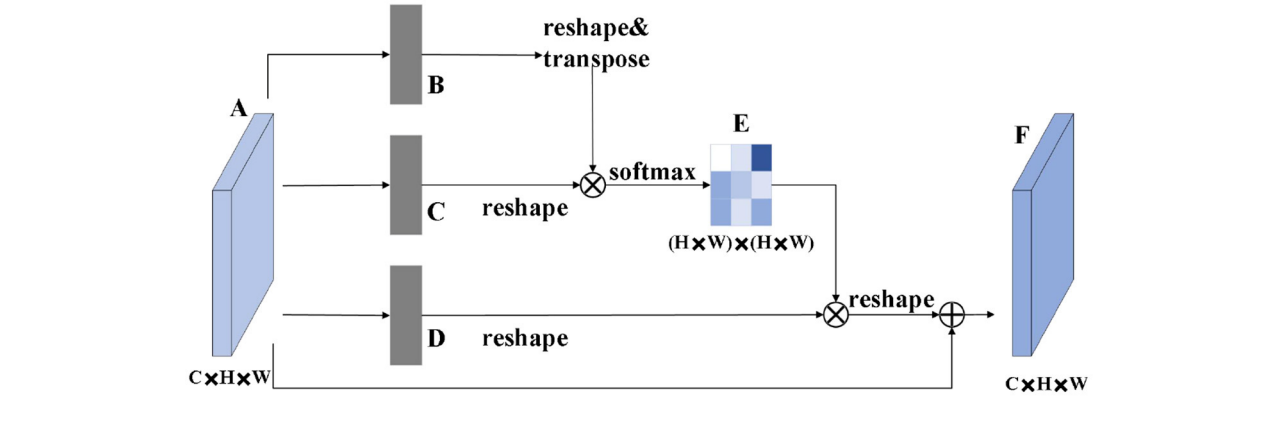

FIGURE 2 | Spatial attention module.

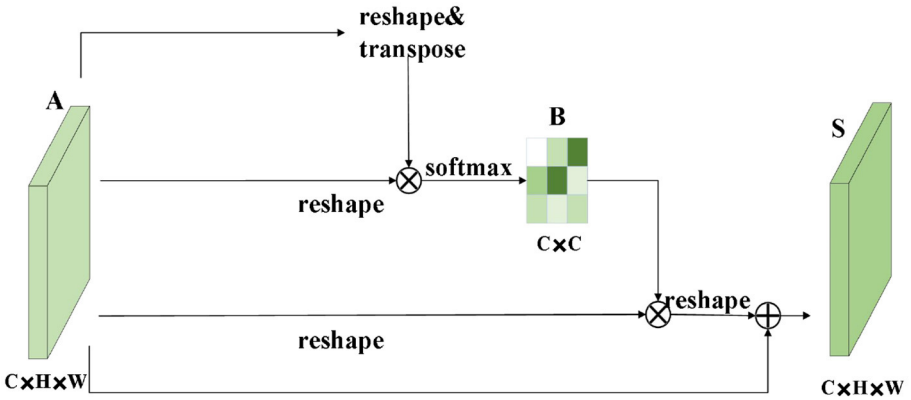

FIGURE 3 | Channel attention module.

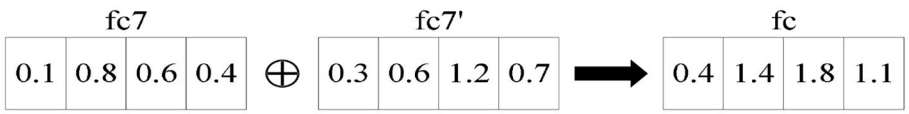

FIGURE 4 | Feature fusion process.

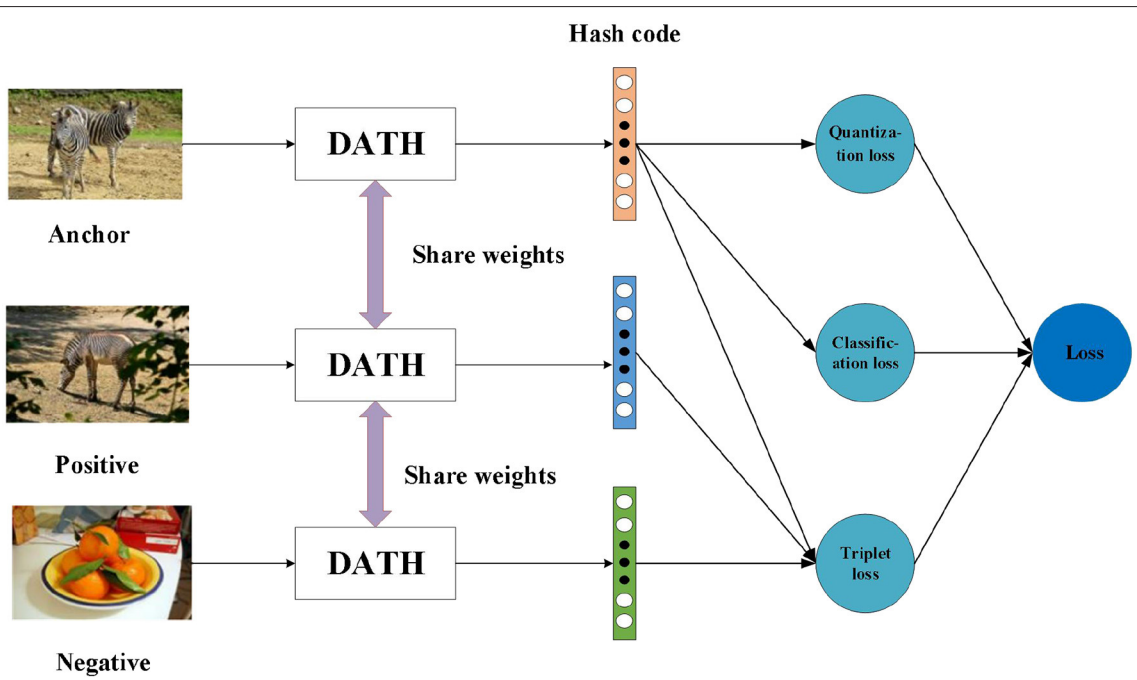

FIGURE 5 | The training process of triplets.

network as the basic architecture of our algorithm. **Figure 1** shows the proposed DATH model. Our method includes two-stream ConvNet architecture. The first stream is embedded with spatial attention module. The second stream is embedded with channel attention module. For hash function learning, we replace the fc8 layer of the softmax classifier in the original AlexNet with a new h layer of k hidden units, which transforms the fc7 representation to k-dimensional hash codes by $b_i = sgn(h_i)$, $sgn(x)$ is the sign function. h_i is the hidden representation of the h layer, and we squash its output to be within $[-1,1]$ by utilizing the tanh activation. And c is the classification layer.

TABLE 1 | Data set allocation.

Data set	NUS-WIDE	MS-COCO
Train set	10000	10000
Test set	5000	5000
Retrieval set	168692	112218
Number of labels	81	80

The role of the attention module is to find salient regions in the original image that need to get attention. Inspired by the attention mechanism in the field of image semantic segmentation (Fu et al., 2019), which is different from the purpose of classification. It combines spatial attention and channel attention to classify each pixel. The spatial attention mechanism mainly sum the features of all pixel positions with different weights. If the features are similar, they will be related to each other. Channel attention mechanism is related to the features in the channel graph selectively and acts on the interdependent channel features. Dual attention modules are fused to capture important features of objects. **Figures 2**, **3** show spatial attention module and channel attention module, respectively.

The sum between fc7 and fc7′ of the fully connected layer is the sum of elements. It is assumed that fc is the fully connected layer after the final fusion. The simple feature fusion process is shown in **Figure 4**.

Triplet Input

As shown in **Figure 5**, it is the training process of triplets, which uses common network parameters for different images. If two images share one same label, we think they are similar, otherwise they are not similar they are dissimilar. With the above definition, an image as the anchor x^a, a positive image x^p that is similar to the anchor and a negative image x^n that is dissimilar to the anchor are used as a triplet input $\left\{x^a, x^p, x^n\right\}$ of the network. For the entire data set, we can define the image triplets set as $T^X = \{x_i^a, x_i^p, x_i^n\}_{i=1}^M$, M is the total number of triplets. Our goal is to learn their compact binary codes $T^B = \{b_i^a, b_i^p, b_i^n\}_{i=1}^M$, and $dist(b_i^a, b_i^p)$ should be much less than $dist(b_i^a, b_i^n)$. *dist()* represents the hamming distance between two hash codes.

However, in a large data set, the workload of constructing triplets is huge. For example, in the NUS-WIDE data set, 10,000 training images are selected before training. The construction of triplets for these 10,000 training images is about $(10000)^3 = 10^{12}$, the number of groups is too large to train. Therefore, we adopt the method of generating triplets online. In a training batch, the image set is denoted as $X = \left\{x_1, x_2, \ldots, x_{b\_s}\right\}$ and the label set is denoted as $L = \left\{l_1, l_2, \ldots, l_{b\_s}\right\}$, b_s means batch size. Every image in a batch will participate in the training and get hash code set $B = \left\{b_1, b_2, \ldots, b_{b\_s}\right\}$. Then use the images that have a similar image and a dissimilar image as the anchor to generate a set of triplets T= $\left\{x_i^a, x_i^p, x_i^n\right\}_{i=1}^m$ $(x_i^* \in X, * \in \{a, p, n\})$, which greatly reduces the training time.

Loss Function

DTSH proposes the triplet likelihood function for hashing coding. Given the likelihood function is:

$$p\left(T|B\right) = \prod_{i=1}^{m} p\left(x_i^a, x_i^p, x_i^n|B\right) \tag{1}$$

with

$$p\left(x_i^a, x_i^p, x_i^n|B\right) = M\left(R_{b_i^a, b_i^p} - R_{b_i^a, b_i^n} - \theta\right) \tag{2}$$

Where m is the number of triplets, $R_{*,*}$ is half of the inner product of two hash codes, such as $R_{b_i^a, b_i^p} = \frac{1}{2} b_i^{aT} b_i^p$. $M(x)$ is the sigmoid function $M(x) = \frac{1}{1+e^{-x}}$, θ is the margin representing the threshold of the similarity difference between a pair of similar images and a pair of dissimilar images (in the subsequent experiments, θ is set to 5) and B is the set of all hash codes. And when the value of $R_{b_i^a, b_i^p}$ is larger and the value of $R_{b_i^a, b_i^n}$ is smaller, triplet likelihood function is larger.

We define triplet loss function as the negative log triplet likelihood as follows:

$$J_1 = -\log p\left(T|B\right) = -\sum_{i=1}^{m} \log p\left(x_i^a, x_i^p, x_i^n|B\right) \tag{3}$$

Where m is the number of triplets which generated from a batch. By taking Equation (2) into Equation (3), we can have:

$$J_1 = -\sum_{i=1}^{m}\left(R_{b_i^a, b_i^p} - R_{b_i^a, b_i^n} - 5 - \log\left(1 + e^{R_{b_i^a, b_i^p} - R_{b_i^a, b_i^n} - 5}\right)\right) \tag{4}$$

In order to fully utilize the label information, we use the joint classification layer to further optimize the hash code. We use the following classification loss function, which can represent the relationship between the learned hash code B and label information L:

$$J_2 = \sum_{i=1}^{b\_s} G\left(l_i, y_i\right) \tag{5}$$

Where y is the output of the classification layer and l is the true label. For single label datasets, $G\left(l_i, y_i\right)$ is formulated as:

$$G_1\left(l_i, y_i\right) = -\sum_{j=1}^{c} l_i[j] \log \frac{e^{y_i[j]}}{\sum_{t=1}^{c} e^{y_i[t]}} \tag{6}$$

Where c is the number of classes and $y[i]$ represents the i-th element of the vector y. If an image contains multiple labels, we refer to this problem as multi-label classification. Cross entropy loss is employed in this case. $G\left(l_i, y_i\right)$ can be calculated as:

$$G_2\left(l_i, y_i\right) = -\sum_{j=1}^{c}\left\{l_i[j] \log \frac{e^{y_i[j]}}{\sum_{t=1}^{c} e^{y_i[t]}} + (1 - l_i[j]) \log(1 - \frac{e^{y_i[j]}}{\sum_{t=1}^{c} e^{y_i[t]}})\right\} \tag{7}$$

TABLE 2 | MAP of different methods on NUS-WIDE and MS-COCO.

Methods	NUS-WIDE				MS-COCO			
	16 bits	32 bits	48 bits	64 bits	16 bits	32 bits	48 bits	64 bits
LSH	0.335	0.348	0.354	0.352	0.382	0.405	0.417	0.417
ITQ-CCA	0.592	0.595	0.590	0.582	0.559	0.590	0.589	0.585
ITQ	0.600	0.622	0.639	0.643	0.566	0.602	0.618	0.627
SDH	0.663	0.710	0.708	0.722	0.618	0.658	0.690	0.693
DHN	0.674	0.703	0.710	0.720	0.643	0.663	0.671	0.672
HashNet	0.681	0.728	0.760	0.767	0.687	0.681	0.706	0.718
ADSH	0.716	0.712	0.682	0.645	0.591	0.550	0.428	0.454
GAH	0.724	0.758	0.766	0.773	0.647	0.689	0.710	0.712
DATH	**0.742**	**0.764**	**0.774**	**0.780**	**0.703**	**0.741**	**0.749**	**0.753**

Bold indicates the best MAP result.

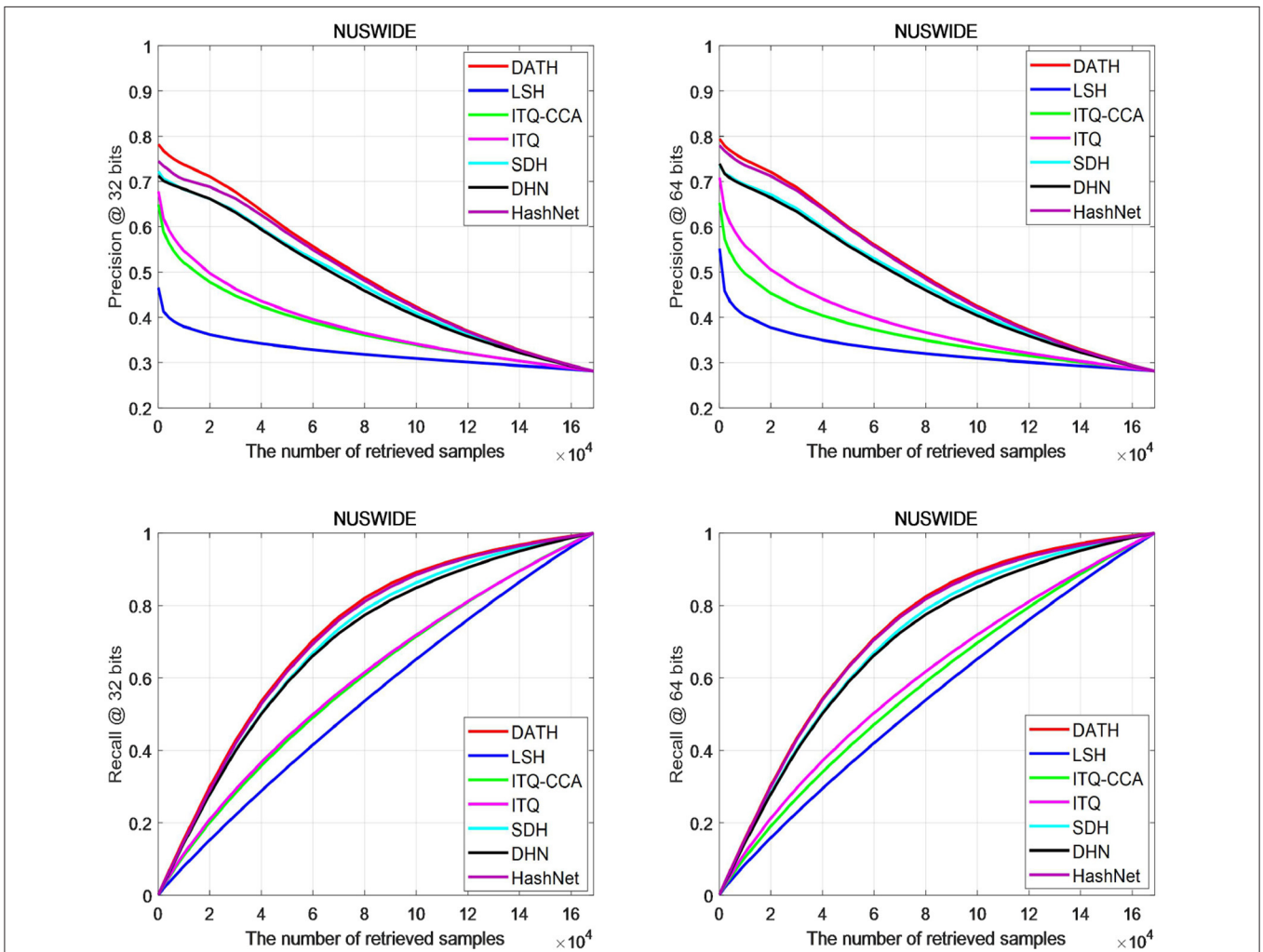

FIGURE 6 | Curves on NUS-WIDE.

In order to get more accurate binary codes, we add the quantization loss function. The loss function is adopted as:

$$J_3 = \sum_{i=1}^{b\_s} \sum_{j=1}^{k} || \, |h_i[j]| - 1 \, ||_1 \tag{8}$$

Where h is the output of hash layer and k is the lengths of hash code. The overall loss function can be written as follows, where β, γ are the hyper-parameters.

$$J = J_1 + \beta J_2 + \gamma J_3 \tag{9}$$

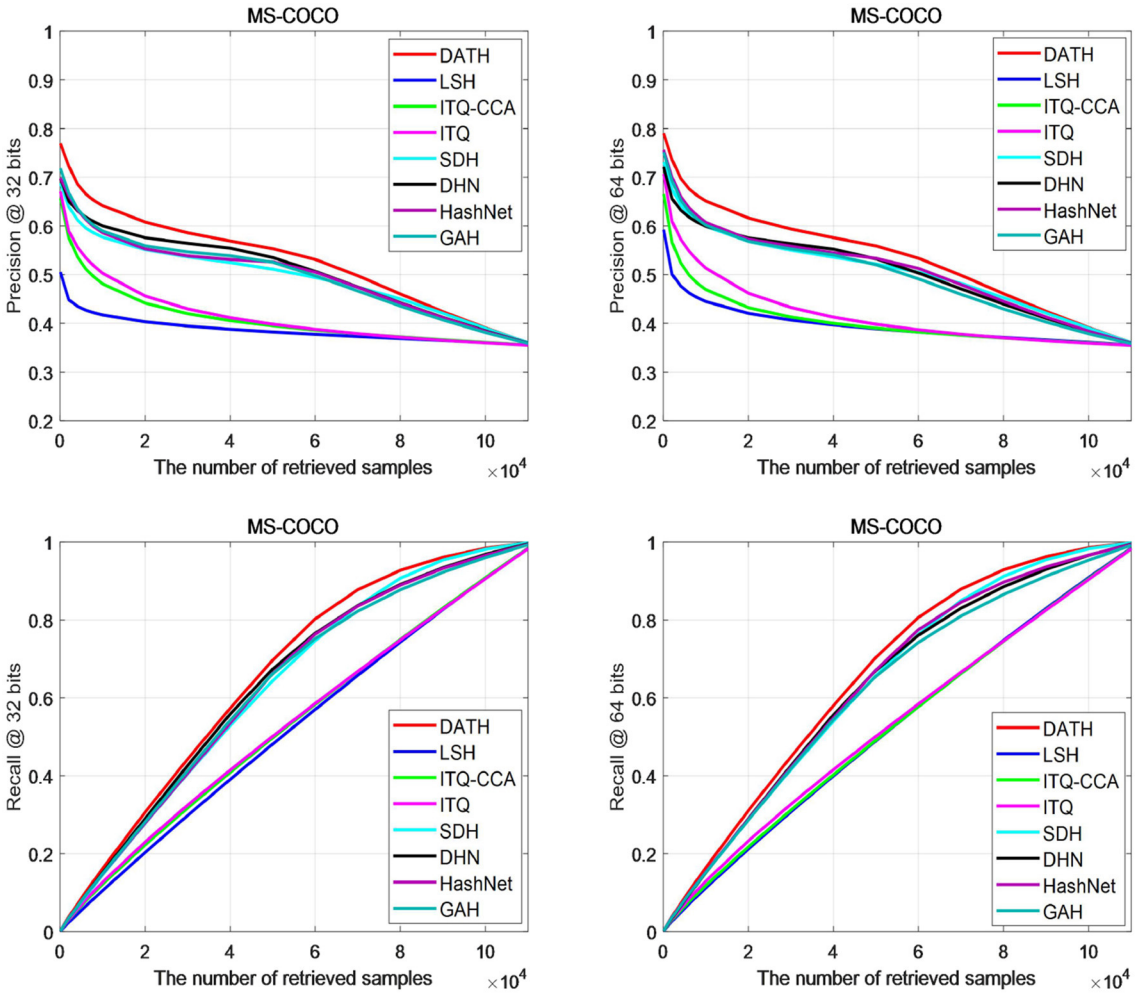

FIGURE 7 | Curves on MS-COCO.

EXPERIMENTS

We compare our method DATH with some classical hashing methods including LSH, ITQ-CCA, ITQ, SDH, DHN, HashNet, ADSH, and GAH. For traditional hashing methods, we feed them DeCAF7 features (Donahue et al., 2014), i.e., the fc7 output of pre-trained AlexNet, as input. For deep hashing methods, we use the same settings in their original papers and re-run their source code with our divided data set. Our two-stream ConvNet architecture use the pre-trained model on ImageNet. The DATH is implemented with Pytorch (Paszke et al., 2019). In the training process, the batch size is 128, the epoch is set to 200, the initial learning rate is set to 1e-5, the optimization algorithm uses RMSProp and weight decay parameter is set to 1e-5. The parameter β and γ are set to 1 and 0.01, respectively.

Datasets and Evaluation Metrics

We evaluate the proposed method on two benchmark datasets: NUS-WIDE (Chua et al., 2009) includes 269,648 images assigned with one or multiple labels under totally 81 concepts. We follow similar experimental protocols as DHN and randomly sample 5,000 images as queries, with the remaining images used as the database, and we randomly sample 10,000 images from the database as training points. MS-COCO (Lin et al., 2014) is an image recognition, segmentation and captioning dataset. The current release contains 82,783 training images and 40,504 validation images, where each image is labeled by some of the 80 categories. After pruning images with no category information, we obtain 122,218 images by combining the training and validation images. We randomly sample 5,000 images as queries, with the rest images used as the database, and we randomly sample 10,000 images from the database as training images.

Table 1 shows the settings of the training set, test set, retrieval set and label number of two data sets. During the training, the size of all images is uniformly changed to 224×224.

We calculate the Mean Average Precision (MAP) values within the top 5000 returned neighbors (MAP@5000) for two datasets, and we draw Precision curves as well as Recall curves

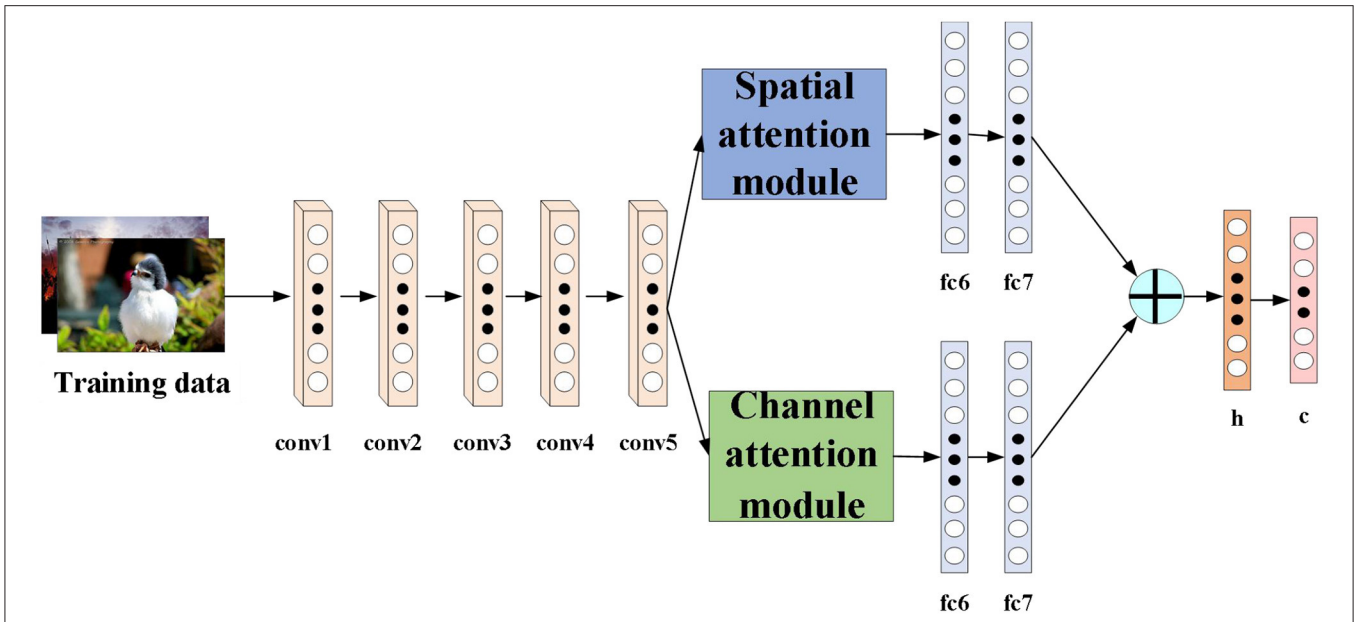

FIGURE 8 | The network architecture of DATH-1.

with respect to different numbers of top returned samples (P@N and R@N). MAP is a measure of the overall performance of image retrieval, which is the mean of Average Precision (AP) for all queries. The definition of AP is as follows:

$$AP = \frac{1}{N}\sum_{i=1}^{M}\frac{i}{R_i} \times rel_i \tag{10}$$

Where N is the number of related images in the database in one query, M is the number of returned images, R_i is the rank of the i-th returned image, and $rel_i = 1$ means the image ranked in the i-th position is similar to the query image, otherwise it is 0. For the NUS-WIDE data set and MS-COCO data set, M is set to 5,000 in the experiments.

Results and Discussion

Mutual Comparison Experiment

Table 2 shows the MAP results for our method DATH. All baseline methods on two datasets with hash code lengths to be 16, 32, 48, 64 bits respectively and bold indicates the best MAP result. From the table, we can observe that DATH outperforms all comparison methods. Specifically, compared to the best traditional hashing method using deep feature as input, we achieve absolute increases of 7.9, 5.4, 6.6, and 5.8% in average MAP for different bits on NUS-WIDE. Compared to deep hashing method GAH, we achieve absolute boosts of 5.6, 5.2, 3.9, and 4.1% in average MAP for different bits on MS-COCO. What needs to be noted is that the original ADSH code uses all the data as the training set. In order to be consistent with other experiments, my ADSH experiment also selects 10,000 images as the training set, and all the data is used during the test. It is noticed that three deep hashing algorithms learn hash codes through pairwise loss function and AlexNet, so the advantage of DATH lies in the use of attention mechanism and the combination of classification loss and triplet loss.

Figure 6 shows the accuracy and recall curves of different retrieved samples on the NUS-WIDE data set. It can be found that the accuracy of the proposed DATH method is higher than other methods on 32 bits, and is closer to the HashNet method on 64 bits. However, it can be seen that the accuracy of DATH can be higher than that of HashNet when there are fewer search samples. For the recall curve, the DATH curve is very close to HashNet.

Figure 7 shows the accuracy and recall curves of different search samples on the MS-COCO data set. It can be clearly seen from the P@N curve that the results of DATH on 32-bit and 64-bit are higher than other curves, especially in retrieval when the number of images is $2\times10\text{^}4$, the gap is more obvious. From the R@N curve, it can be seen that after the number of retrieved images is $>4\times10\text{^}4$, DATH can begin to obtain a clear advantage.

Based on **Figures 6**, **7**, we can find most of the traditional methods cannot achieve better results. The use of attention mechanism and the full use of label information have a significant improvement in the deep hashing method.

Self-Contrast Experiment

We investigate several variants of DATH: DATH-S is the DATH variant without channel attention module. DATH-C is the DATH variant without spatial attention module. DATH-N is the DATH variant without any attention module, only one AlexNet network. DATH-1 changes the network structure mainly in the feature extraction part as shown in **Figure 8**. We compare the results of DATH variants in **Table 3**, bold indicates the best MAP result.

Comparing DATH-N, DATH-S and DATH-C, we can find that adding spatial attention module improves the model more

TABLE 3 | MAP results of DATH and its variants on NUS-WIDE and MS-COCO.

Methods	NUS-WIDE				MS-COCO			
	16 bits	32 bits	48 bits	64 bits	16 bits	32 bits	48 bits	64 bits
DATH-S	0.735	0.759	0.769	0.775	0.697	0.725	0.744	0.750
DATH-C	0.724	0.757	0.767	0.767	0.691	0.732	0.741	0.746
DATH-N	0.722	0.758	0.763	0.768	0.691	0.726	0.740	0.746
DATH-1	0.737	0.763	**0.777**	0.778	0.692	0.737	**0.751**	0.748
DATH	**0.742**	**0.764**	0.774	**0.780**	**0.703**	**0.741**	0.749	**0.753**

Bold indicates the best MAP result.

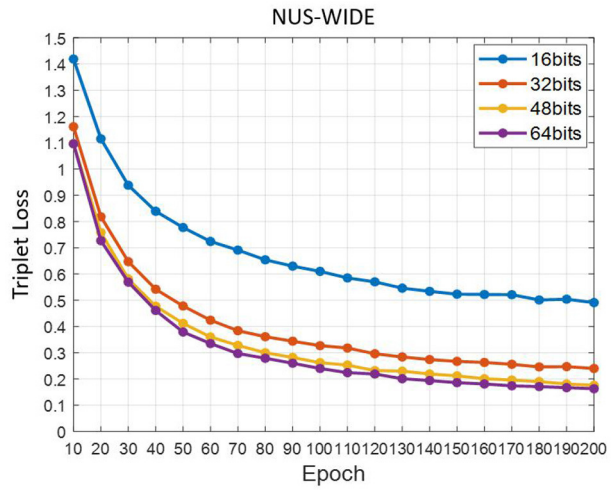

FIGURE 9 | Triplet loss declining curve.

significantly, and spatial attention can indeed focus on important areas to make the hash code more effective. Compared with DATH again, we can find that adding two attention modules will greatly improve the model effect, which shows that the fusion of two attention modules can best extract key features. At the same time, as the length of the hash code increases, the improvement of the model by adding the attention module will decrease. The reason may be that the long hash code itself already contains a lot of useful or useless information, and the attention mechanism is more able to help the short hash code to select critical features from a large amount of image information. Finally, compared with DATH-1, we can find DATH can almost achieve better results, but it is obvious that DATH uses more different convolutional layers and fully connected layers, which will lead to an increase in the amount of network parameters.

Convergence Degree of Triplet Loss

Figure 9 is the trend chart of the triplet loss function with Epoch on the NUS-WIDE data set. We save the loss every 10 Epoch. At different hash code bits, the triplet loss shows a rapid downward trend at the beginning, and gradually tend to a stable value in the later stage. This shows that the triplet loss function plays an important role in the entire training process.

TABLE 4 | MAP results of different hyper-parameter.

γ / β	NUS-WIDE			MS-COCO		
	0	0.1	0.01	0	0.1	0.01
0	0.728	0.719	0.729	0.674	0.664	0.688
1	0.726	0.705	0.742	0.688	0.667	**0.703**
0.1	0.733	0.723	0.740	0.691	0.672	0.696
0.01	0.732	0.717	**0.747**	0.693	0.664	0.698
0.001	0.729	0.723	0.741	0.691	0.672	0.696

Bold indicates the best MAP result.

Hyper-Parameter Analysis

In order to further reveal the impact of classification loss function on the results, we conduct experiments with a 16-bit hash code on two data sets. Except for the values of hyper-parameters, the other experimental parameters remain unchanged. For the classification loss hyper-parameter β, we select five values of 0, 1, 0.1, 0.01, and 0.01, and the quantitative control function hyper-parameter γ we chose 0, 0.1, and 0.01, as shown in **Table 4**, bold indicates the best MAP result.

In the case without the classification loss, that is $\beta = 0$, comparing the best MAP of the two data sets, we can find that not combining the classification loss will cause a significant decrease in retrieval accuracy, which proves the rationality of the joint classification loss. In the cases of weighted quantization loss, we can find that combining different proportions of quantization loss will have different effects on retrieval accuracy. When $\gamma = 0.01$, the retrieval accuracy will be slightly improved. In general, combining classification loss to the network improves retrieval accuracy more obviously.

Efficiency Analysis

In the actual retrieval system, the time efficiency of generating hash codes for new images is also an important part. In order to calculate the encoding time of the DATH network, this section compares the encoding time of the DATH method and other baseline deep hashing methods and all experiments calculate time by NVIDIA GTX1080Ti. **Table 5** shows the average encoding time of images on the MS-COCO dataset with different 16-bit hashing methods. Each method removes the image preprocessing part and only calculates the time for the image to pass the model calculation.

It can be seen from the **Table 5** that our method is close to twice that of other methods. This is mainly because our network structure is more complex and we add the attention module.

TABLE 5 | Encoding time of different methods.

Method	DHN	HashNet	GAH	DATH	DATH-S	DATH-C	DATH-1
Time (ms)	1.995	1.995	1.996	4.015	3.789	3.786	4.013

TABLE 6 | Training time of DATH and DATH-1.

Method	Time (s)			
	16 bits	32 bits	48 bits	64 bits
DATH	40.569	40.634	40.623	40.635
DATH-1	39.954	40.302	40.378	40.377

Table 6 shows the average training time of one epoch on the MS-COCO dataset with two hashing methods DATH and DATH-1.

Although the training time of DATH is longer than DATH-1, the encoding time is the close and the retrieval result of DATH is better. Finally, we choose DATH as the final network.

Summarizing all the above experiments, the retrieval accuracy of DATH is better than other hash methods. However, the image encoding time of DATH is longer and DATH needs time consuming hyperparameter tuning. In practical applications, we may do retrieval on large data sets. DATH-S or DATH-C, which has slightly lower retrieval accuracy but faster speed, is a good choice. As for the selection of parameters, I suggest to make adjustments on part of the data with reference to our experimental results.

CONCLUSION

In this paper, we propose a Dual Attention Triplet Hashing Network. The proposed DATH uses the spatial attention mechanism and the channel attention mechanism to extract the key features of images, and combines the classification loss function with the triplet likelihood loss function to make full use of label information. DATH applies the dual attention structure to image retrieval for the first time, and introduces how to combine classification loss and quantification loss on the basis of triple loss in a batch. Quantitative experiments show our model's successful target-oriented designs. Compared with the highest value of the other methods, we respectively achieve absolute boosts of 3.55% and 0.95% in average MAP for different bits on MS-COCO and NUS-WIDE. In the future, we are looking forward to applying this kind of dual attention network to more Image Retrieval tasks such as Fine-Grained Image Retrieval, and reducing network complexity is also our future goal.

DATA AVAILABILITY STATEMENT

Publicly available datasets were analyzed in this study. This data can be found at: https://cocodataset.org/, https://lms.comp.nus.edu.sg/wp-content/uploads/2019/research/nuswide/NUS-WIDE.html.

AUTHOR CONTRIBUTIONS

ZL: provide the ideas and research concept. ZJ: research concept and design, writing the article, and data analysis and interpretation. JW: data analysis and interpretation and research concept and design. All authors contributed to the article and approved the submitted version.

REFERENCES

Andoni, A., and Indyk, P. (2006). "Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions," in *47th Annual IEEE Symposium on Foundations of Computer Science, Proceedings* (Berkeley, CA).

Cao, Z. J., Long, M. S., Wang, J. M., and Yu, P. S. (2017). "HashNet: deep learning to hash by continuation," in *2017 Ieee International Conference on Computer Vision (Iccv)* (Venice), 5609–5618. doi: 10.1109/ICCV.2017.598

Chen, L., Zhang, H. W., Xiao, J., Nie, L. Q., Shao, J., Liu, W., et al. (2017). "SCA-CNN: Spatial and channel-wise attention in convolutional networks for image captioning," in *30th IEEE Conference on Computer Vision and Pattern Recognition* (Cvpr 2017) (Honolulu, HI), 6298–6306. doi: 10.1109/CVPR.2017.667

Chen, Y. D., Lai, Z. H., Ding, Y. J., Lin, K. Y., and Wong, W. K. (2019). "Deep supervised hashing with anchor graph," in 2019 Ieee/Cvf International Conference on Computer Vision (Iccv 2019) (Seoul), 9795–9803.

Chua, T.S., Tang, J., Hong, R., Li, H., and Luo, Z. (2009). "NUS-WIDE: a real-world web image database from National University of Singapore," in *Acm International Conference on Image and Video Retrieval* (Santorini).

Cui, H., Zhu, L., Li, J. J., Yang, Y., and Nie, L. Q. (2020). Scalable deep hashing for large-scale social image retrieval. *IEEE Trans. Image Process.* 29, 1271–1284. doi: 10.1109/TIP.2019.2940693

Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., et al. (2014). "DeCAF: a deep convolutional activation feature for generic visual recognition," in *Proceedings of the 31st International Conference on Machine Learning*, eds. P. X. Eric and J. Tony (Bejing).

Dubey, S. R. (2021). "A decade survey of content based image retrieval using deep learning," in *IEEE Transactions on Circuits and Systems for Video Technology*, 1–1. doi: 10.1109/tcsvt.2021.3080920

Fang, J., Fu, H., and Liu, J. (2021). Deep triplet hashing network for case-based medical image retrieval. *Med. Image Anal.* 69:101981. doi: 10.1016/j.media.2021.101981

Fu, J., Liu, J., Tian, H. J., Li, Y., Bao, Y. J., Fang, Z. W., et al. (2019). "Dual attention network for scene segmentation," in *2019 IEEE/Cvf Conference on Computer Vision and Pattern Recognition (CVPR 2019)* (Long Beach, CA), 3141–3149.

Gionis, A., Indyk, P., and Motwani, R. (1999). "Similarity search in high dimensions via hashing," in *Proceedings of the Twenty-Fifth International Conference on Very Large Data Bases* (Edinburgh), 518–529

Gong, Y. C., and Lazebnik, S. (2011). "Iterative quantization: a procrustean approach to learning binary codes," in *2011 Ieee Conference on Computer Vision and Pattern Recognition (CVPR)* (Colorado Springs, CO), 817–824.

Huang, L. K., Chen, J., and Pan, S. J. (2019). "Accelerate learning of deep hashing with gradient attention," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV 2019)* (Seoul), 5270–5279.

Jiang, Q. Y., and Li, W. J. (2018). "Asymmetric deep supervised hashing," in *Thirty-Second Aaai Conference on Artificial Intelligence/Thirtieth Innovative Applications of Artificial Intelligence Conference/Eighth Aaai Symposium on Educational Advances in Artificial Intelligence* (New Orleans, LA), 3342–3349.

Jin, L., Shu, X. B., Li, K., Li, Z. C., Qi, G. J., and Tang, J. H. (2019). Deep ordinal hashing with spatial attention. *IEEE Trans. Image Process.* 28, 2173–2186. doi: 10.1109/TIP.2018.2883522

Jin, Z. M., Li, C., Lin, Y., and Cai, D. (2014). Density sensitive hashing. *IEEE Trans. Cybern.* 44, 1362–1371. doi: 10.1109/TCYB.2013.2283497

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2017). ImageNet classification with deep convolutional neural networks. *Commun. ACM* 60, 84–90. doi: 10.1145/3065386

Lin, T.Y., Maire, M., Belongie, S., Hays, J., and Zitnick, C.L. (2014). "Microsoft COCO: common objects in context," in *European Conference on Computer Vision* (Zurich).

Liu, W., Wang, J., Ji, R. R., Jiang, Y. G., and Chang, S. F. (2012). "Supervised hashing with kernels," in *2012 Ieee Conference on Computer Vision and Pattern Recognition (Cvpr)* (Providence, RI), 2074–2081.

Oquab, M., Bottou, L., Laptev, I., and Sivic, J. (2014). "Learning and transferring mid-level image representations using convolutional neural networks," in *2014 IEEE Conference on Computer Vision and Pattern Recognition: 2014 27th IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2014), 23-28 June 2014 Columbus, Ohio* (Columbus, OH).

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., Devito, Z., et al. (2019). "PyTorch: An Imperative Style, High-Performance Deep Learning Library," in *Advances in Neural Information Processing Systems 32* (Curran Associates, Inc.), 8024–8035. Retrieved from: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

Shen, F. M., Shen, C. H., Liu, W., and Shen, H. T. (2015). "Supervised discrete hashing," in *2015 Ieee Conference on Computer Vision and Pattern Recognition (CVPR)* (Boston, MA), 37–45.

Song, W., Li, S., and Benediktsson, J. A. (2019). Deep Hashing Learning for Visual and Semantic Retrieval of Remote Sensing Images. *IEEE Trans. Geosci. Remote Sens.* 1–12. doi: 10.1109/TGRS.2020.3035676

Wang, J., Kumar, S., and Chang, S. F. (2012). Semi-supervised hashing for large-scale search. *IEEE Trans. Pattern Anal. Mach. Intell.* 34, 2393–2406. doi: 10.1109/TPAMI.2012.48

Wang, X. F., Shi, Y., and Kitani, K. M. (2017). Deep supervised hashing with triplet labels. *Comput. Vis.* 10111, 70–84. doi: 10.1007/978-3-319-54181-5_5

Weiss, Y., Torralba, A., and Fergus, R. J. A. I. N. I. P. S. (2009). "Spectral hashing," in *Advances in Neural Information Processing Systems 21 (NIPS 2008), Vol. 282* (Vancouver, BC), 1753–1760.

Wu, D. Y., Lin, Z., Li, B., Liu, J., and Wang, W. P. (2018). "Deep uniqueness-aware hashing for fine-grained multi-label image retrieval," in *2018 Ieee International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (Calgary, AB), 1683–1687.

Xie, L., Shen, J., Han, J., Zhu, L., and Shao, L. (2017). "Dynamic multi-view hashing for online image retrieval," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI*-17 (Melbourne, VIC), 3133–3139.

You, Q. Z., Jin, H. L., Wang, Z. W., Fang, C., and Luo, J. B. (2016). "Image captioning with semantic attention," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Las Vegas, NV), 4651–4659.

Zhou, C., Po, L. M., Liu, M., Yuen, W., Wong, P., Luk, H. T., et al. (2019). "Deep hashing with triplet labels and unification binary code selection for fast image retrieval," in *MultiMedia Modeling. MMM 2019. Lecture Notes in Computer Science, Vol. 11295*, eds I. Kompatsiaris, B. Huet, V. Mezaris, C. Gurrin, W. H. Cheng, and S. Vrochidis (Cham: Springer), 277–288.

Zhu, H., Long, M. S., Wang, J. M., and Cao, Y. (2016). "Deep hashing network for efficient similarity retrieval," in Thirtieth *Aaai Conference on Artificial Intelligence* (Phoenix, AZ), 2415–2421.

Zhu, J., Shu, Y., Zhang, J., Wang, X., Wu, S. (2021). Triplet-object loss for large scale deep image retrieval. *Int. J. Mach. Learn. Cyber.* 1–9. doi: 10.1007/s13042-021-01330-8

Zhu, L., Lu, X., Cheng, Z. Y., Li, J. J., and Zhang, H. X. (2020). Deep collaborative multi-view hashing for large-scale image search. *IEEE Trans. Image Process.* 29, 4643–4655. doi: 10.1109/TIP.2020.2974065

Conflict of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.