Computer Modeling in Engineering & Sciences CMES, vol.121, no.2, pp.593-608, 2019

IDSH: An Improved Deep Supervised Hashing Method for Image Retrieval

Chaowen Lu[1, a], Feifei Lee[1, a, *], Lei Chen[1], Sheng Huang[1] and Qiu Chen[2, *]

Abstract: Image retrieval has become more and more important because of the explosive growth of images on the Internet. Traditional image retrieval methods have limited image retrieval performance due to the poor image expression abhility of visual feature and high dimension of feature. Hashing is a widely-used method for Approximate Nearest Neighbor (ANN) search due to its rapidity and timeliness. Meanwhile, Convolutional Neural Networks (CNNs) have strong discriminative characteristics which are used for image classification. In this paper, we propose a CNN architecture based on improved deep supervised hashing (IDSH) method, by which the binary compact codes can be generated directly. The main contributions of this paper are as follows: first, we add a Batch Normalization (BN) layer before each activation layer to prevent the gradient from vanishing and improve the training speed; secondly, we use Divide-and-Encode Module to map image features to approximate hash codes; finally, we adopt center loss to optimize training. Extensive experimental results on four large-scale datasets: MNIST, CIFAR-10, NUS-WIDE and SVHN demonstrate the effectiveness of the proposed method compared with other state-of-the-art hashing methods.

Keywords: Image retrieval, convolutional neural network, hash functions, center loss.

1 Introduction

Due to the popularity of social media in the Internet and mobile terminals, the number of digital images is growing rapidly. More and more visual tasks have been widely studied in artificial intelligence and computer vision. Content Based Image Retrieval (CBIR) refers to the process of obtaining images that are relevant to a query image from a large collection based on their visual content [Datta, Li and Wang (2005)]. The key issue of the CBIR is to extract valuable semantic information from raw data in order to eliminate the semantic gap. So image representations and similarity measure become critical to such a task. Suppose

[1]School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai, 200093, China.

[2]Major of Electrical Engineering and Electronics, Graduate School of Engineering, Kogakuin University, 1-24-2, Nishi-shinjuku, Sinjuku-ku, Tokyo, 163-8677, Japan.

[a]Both authors contributed equally to this work.

[*]Corresponding Authors: Feifei Lee. Email: feifeilee@ieee.org;
Qiu Chen. Email: q.chen@ieee.org.

CMES. doi:10.32604/cmes.2019.07796 www.techscience.com/cmes

that both the images in the database and the query image are presented by real-valued features, the simplest way to find the same or similar images is to sort the database images according to the distances between the database images and the query image in the feature space, and return the closest ones. However, for a database with millions of images, a linear search of the database will take a lot of time and memory.

Although several hand-crafted features, such as have been proposed to reflect the images representation [Lowe (2004); Bay, Tuytelaars and Gool (2006); Qiu (2002)], the performance of these visual descriptors is still limited until the recent breakthrough in deep learning. Some studies have shown that the performance on various vision tasks can be improved, such as image features representations [Girshick, Donahue, Darrell et al. (2014); Oquab, Bottou, Laptev et al. (2014); Sharif Razavian, Azizpour, Sullivan et al. (2014)], image classification [Krizhevsky, Sutskever and Hinton (2012); Szegedy, Liu, Jia et al. (2015)], face recognition [Taigman, Yang, Ranzato et al. (2014); Lin, Li and Tang (2017); Tang, Lin, Li et al. (2018)] and so on. These achievements are attributed to the ability of deep CNN to learn rich image representations. Benefiting from the produced binary codes in hashing method, fast image search can be carried out via Hamming distance measurement, which reduces the computational cost and further optimizes the efficiency of the search. The problem of hashing image retrieval is how to effectively encode massive images into available feature representations so as to improve retrieval performance. Semantic Hashing [Salakhutdinov and Hinton (2009)] uses a multi-layer auto-encoder to construct binary feature, with the raw pixels of images being used as input. Recent studies [Liu, Wang, Ji et al. (2012); Norouzi and Fleet (2011); Kulis and Darrell (2009)] have shown that combining supervised information can raise the performance of hash learning.

However, there are some drawbacks in these methods mentioned above. First, the traditional hand-crafted features contain incomplete semantic information. Secondly, not end-to-end methods consume a lot of memory. Thirdly, some methods take much time during the data preparation. To address this problem, we propose in this paper a CNN architecture based on improved deep supervised hashing (IDSH) method, by which the binary compact codes can be generated directly. Different from traditional hashing methods, our method is with the following characteristics:

1. The proposed method can quickly learn hash functions to generate binary codes of images because of the existence of end-to-end module. The hash features become more prominent by Divide-and-Encode Module, which can represent accurate semantic information of the corresponding images.

2. With some modifications to the network model, our method achieves the best retrieval performance on some public datasets compared with the state-of-the-art works.

3. The addition of center loss can improve the retrieval effectiveness on simple datasets, which can be applied to certain datasets.

The rest of this paper is organized as follows: Section 2 overviews the related work. The proposed method is elaborated in detail in Section 3. The details of the experiments and the results are described in Section 4. Finally, we draw the conclusion in Section 5.

2 Related work

Because of the advantages of fast retrieval speed and low storage cost, more and more scholars have studied for hashing-based image retrieval. The current learning-based hashing methods can be roughly divided into unsupervised and supervised hashing methods.

Unsupervised hashing methods learn hash functions with unlabeled training data, which encode input images to binary codes. The most representative of the methods is Locality Sensitive Hashing [Gionis, Indyk and Motwani (1999)], and there are also many other unsupervised hashing algorithms in subsequent studies, such as Semantic Hashing [Salakhutdinov and Hinton (2009)], Iterative Quantization [Gong, Lazebnik, Gordo et al. (2012)].

Supervised hashing methods use supervised information from the labeled data to learn hash function to generate compact bit-wise representations. Binary Reconstruction Embedding (BRE) [Kulis and Darrell (2009)] is proposed to minimize the error between distances of data points and those of the corresponding hash codes. Minimal Loss Hashing (MLH) [Norouzi and Fleet (2011)] constructs an objective function based on structured SVM for hash function learning. Different from previous hashing methods, Supervised Hashing with Kernels (KSH) [Liu, Wang, Ji et al. (2012)] is a kernel-based method that does not train hash functions directly by minimizing the Hamming distance between hash codes instead of minimizing the inner product of hash codes, where it is proved that minimizing the inner product of hash codes is equivalent to implicitly minimizing the Hamming distance.

These methods mentioned above are based on hand-crafted visual features (e.g., GIST [Oliva and Torralba (2001)]), which limit the retrieval performance. In recent years, deep hashing methods have been used in large-scale image retrieval. The success of image representation method based on deep network is mainly due to their ability of automatically learning effective image representation.

DHLE [Lu, Song, Xie et al. (2017)] has adopted point-wise training for simultaneous feature extracting and hash function learning. DFH [Zhou, Zeng and Chen (2019)] proposes a deep forest-based method for hashing learning that aims to learn shorter binary codes to achieve effective and efficient image retrieval. DSDH [Li, Sun, He et al. (2017)] combines pairwise label information and the classification information to learn the hash codes within one stream framework. DSH [Liu, Wang, Shan et al. (2016)] is proposed to reduce the heterogeneity between image pairs and constructs a loss function to obtain effective hash codes to ensure the richness of image information. CNNH [Xia, Pan, Lai et al. (2014)] is proposed to divide hash learning process into two stages. First similarity information of the data is used to construct similarity matrix and the corresponding hash codes are obtained, then the obtained hash codes and image labels are input to learn image features and hash function based on deep convolution network. But it cannot deal with large-scale images because the matrix factorization costs much storage memory. Pan et al. [Lai, Pan, Liu et al. (2015)] (DNNH) proposed an end-to-end supervised hashing method to learn hash function, where using a triplet loss to preserve the relative similarities of images. And other methods [Zhao, Huang, Wang et al. (2015); Zhang, Lin, Zhang et al.

(2015); Wang, Shi and Kitani (2016); Zhou, Huang, Zhang et al. (2017); Deng, Chen, Liu et al. (2018); Zhou, Po, Liu et al. (2019)] also enforce the network to learn binary-like outputs that preserve the semantic relations of image-triplets. However, it costs much time on screening lots of triplet pairs of images in early stage. In recent years, with the development of convolutional neural networks, some methods [Yang, Xie, Yin et al. (2017); Cao, Long, Wang et al. (2017); Gui, Liu, Sun et al. (2018); Li, Sun, He et al. (2017); Li and Li (2015); Zhang and Peng (2017); Li, Miao, Wang et al. (2018); Wang, Lee and Chen (2019); Yang, Lin and Chen (2018); Wu, Dai, Liu et al. (2019); Ge, Zhang, Xia et al. (2019); Shi, Sapkota, Xing et al. (2018)] have greatly improved their performance. In order to avoid previous problems, we propose a one-stage supervised deep hashing (IDSH) method via a deep convolution network that maps input images to binary codes directly.

The purpose of the hashing image retrieval is to learn the suitable features by using the available image information to increase the accuracy of image retrieval. Due to the significant progress of deep features, we propose an improved end-to-end hash learning approach for the best compatibility of representation learning and hash coding.

The purpose of the hashing image retrieval is to learn the suitable features by using the available image information to increase the accuracy of image retrieval. Due to the significant progress of deep features, we propose an improved end-to-end hash learning approach for the best compatibility of representation learning and hash coding.

Table 1: Configurations of shared CNN

Type	Filter size/stride
Convolution1	11*11/ 4
Max Pool1	3*3 / 2
Convolution2	5*5 / 2
Max Pool2	3*3 / 2
Convolution3	3*3 / 1
Convolution4	3*3 / 1
Convolution5	3*3 / 1
Max Pool3	3*3 / 2

3 The proposed IDSH approach

In this section, we will introduce the proposed fast hashing method, which incorporates hashing learning and feature learning. The proposed improved deep supervised hashing (IDSH) method includes two components. First, in order to generate robust hashing codes of corresponding images, we use Batch Normalization (BN) layer and Divide-and-Encode Module together based on Alexnet network. Secondly, center loss is applied to the network for better intra-class distance. The whole training process is end-to-end and stable. The details of the proposed retrieval process framework are illustrated in Fig. 1. The configurations of shared CNN are shown in Tab. 1.

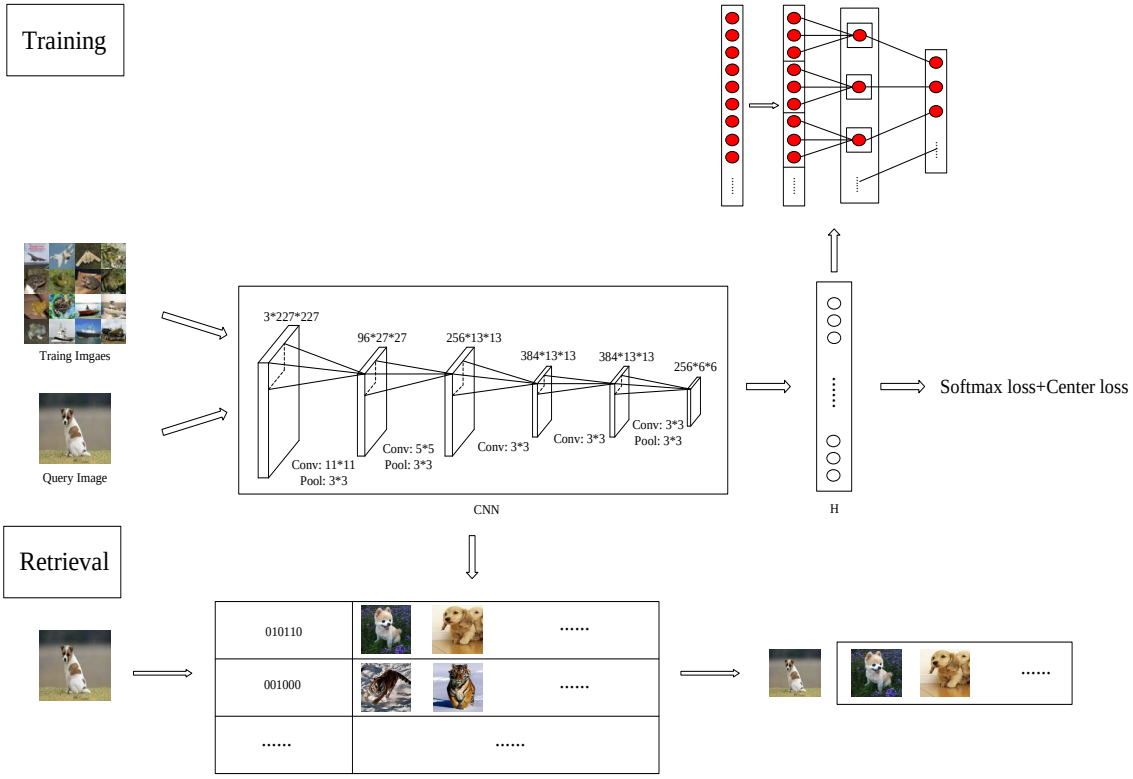

Figure 1: Overview of the proposed deep architecture for hashing

3.1 Batch normalization and divide-and-encode module

Batch Normalization (BN) [Ioffe and Szegedy (2015)] helps to avoid the vanishing gradient problem and boost the learning speed, so we add a BN layer before each activation layer based on the original Alexnet Network. The ideal of BN is that increasing gradient to avoid the vanishing gradient problem. The specific operation is normalizing each scalar feature independently by making it have the mean of zero and the variance of 1. Therefore the convergence becomes faster, and the speed of training is accelerated. Inspired by Oliva et al. [Oliva and Torralba (2001)], we take the Divide-and-Encode Module as the hash layer.

The function of sigmoid and tanh is basically the same that be able to output the approximate hash. For convenience, we use the sigmoid activation funtion to generate hash features. As can be seen in Fig. 2, the input deep features are divided into q slice with self-defined equal length. Each output of the mapped slice by fully-connected layer is restricted in the range [0,1] by a sigmoid layer. Then the q output hash bits are concatenated to be a q-bit code, which defined as s. The discrete hash features are obtained by threshold

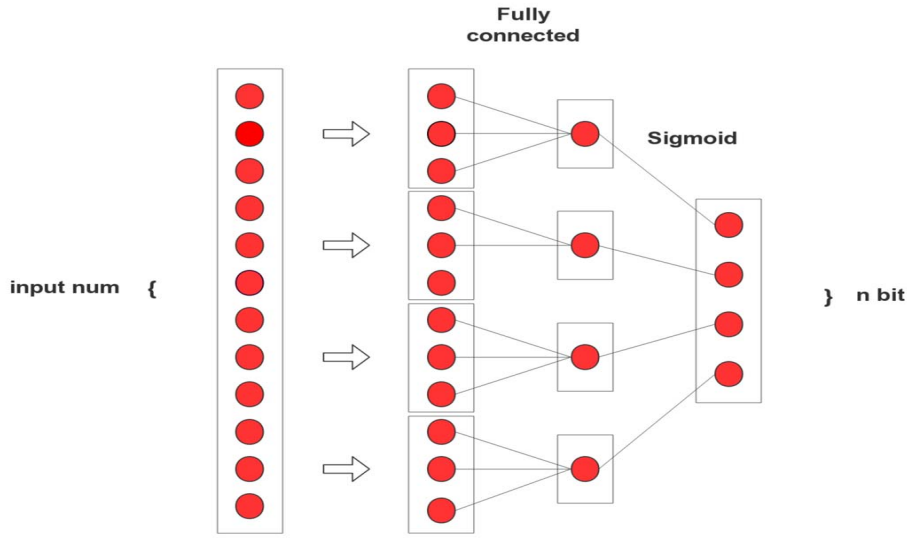

Figure 2: A divide-and-encode module

function. The threshold is formulated in Eq. (1).

$$f(s) = \begin{cases} 0 & \text{s<0.5} \\ 1 & \text{s>=0.5} \end{cases} \tag{1}$$

In previous methods, each hash code could be associated with the whole input image feature vector, which leads to redundancy among hash codes. Compared with the direct use of fully-connected layer followed by a sigmoid layer, the key of the Divide-and-Encode Module is to reduce the redundancy among the hash bits.

3.2 Center loss

Instead of using triplet loss function in previous methods, we input a single image directly for training. The softmax loss function is used for training the network, given by Eq. (2).

$$L_s = -\sum_{i=1}^{m} \log \frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^{n} e^{W_{y_i}^T x_i + b_{y_i}}} \tag{2}$$

M is the training batch size, $x_i \in R^d$ denotes the deep image feature, belonging to the $y_i th$ class. The W and b are the weights and bias for the last layer of the network. However when using the softmax loss alone in the network, the effect is often less than expectation due to the large intra-class variations. How to get more useful compact binary codes to effectively present the corresponding image features is such a significant problem.

Intuitively, minimizing the intra-class variations while keeping the features of different classes separable are the key. For this goal, we try to use center loss to improve the discriminative ability of the deeply learned features and make the features more close on the basis of softmax loss, which makes the expressive ability of features more powerful. The center loss function is formulated in Eq. (3).

$$L_c = -\frac{1}{2}\sum_{i=1}^{m} \|x_i - c_{y_i}\|_2^2 \tag{3}$$

The $c_{y_i} \in R_d$ represents the $y_i th$ class center of image features. Meanwhile, c_{y_i} should be updated with the change of deep features. In other words that it should take the entire training set into account and average the features of every class in each iteration, which is inefficient. It seems difficult to be implemented. Therefore, the center loss cannot be used directly. Then, two necessary modifications are made to address this problem. First, the mini-batch replaces the entire training set to achieve the function of updating the centers. The centers are calculated by averaging the features of the corresponding classes in each iteration. Second, a scalar α is used to control the learning rate of the centers to avoid large perturbations caused by few mislabeled samples.

The gradients of L_c with respect to x_i and update equation of c_{y_i} are computed as follows:

$$\frac{\partial L_c}{\partial x_i} = x_i - c_{y_i} \tag{4}$$

$$\Delta c_j = \frac{\sum_{i=1}^{m} \delta(y_i = j) \cdot (c_j - x_i)}{1 + \sum_{i=1}^{m} (y_i = j)} \tag{5}$$

$$c_j^{t+1} = c_j^t - \alpha \cdot \Delta c_j^t \tag{6}$$

where δ(condition) =1 if condition is satisfied, and δ(condition)=0 if not. And α is set in [0, 1], t refers to the number of iteration. Eq. (6) shows that centers would be updated in each iteration if necessary.

The joint of the softmax loss and center loss function is given in Eq. (8).

$$\begin{aligned} L &= L_s + L_c \\ &= -\sum_{i=1}^{m} \log \frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^{n} e^{W_{y_i}^T x_i + b_{y_i}}} + \frac{\lambda}{2}\sum_{i=1}^{m} \|x_i - c_{y_i}\|_2^2 \end{aligned} \tag{7}$$

The scalar λ is a weight coefficient used to balance the two loss functions. If we only train CNN with center loss, the centers and deeply learned features will degraded to zeros. Simply using either of them cannot achieve discriminative feature learning. So it is necessary for us to combine the two loss functions.

We use SGD to optimize the loss function during the training process.

4 Experimental results

In order to verify the performance of the proposed hashing method, in this section, we construct experiments and demonstrate the performance of our proposed image representation based on deep hash codes on three publicly available datasets. To better show the superiority of our method clearly, some state-of-the-art hashing methods are compared, including unsupervised methods LSH [Gionis, Indyk and Motwani (1999)], SH [Salakhutdinov and Hinton (2009)], ITQ [Gong, Lazebnik, Gordo et al. (2012)], and supervised methods DFH [Zhou, Zeng and Chen (2019)], DHLE [Lu, Song, Xie et al. (2017)], DSDH[Li, Sun, He et al. (2017)], DSH [Liu, Wang, Shan et al. (2016)], DNNH [Lai, Pan, Liu et al. (2015)], CNNH [Xia, Pan, Lai et al. (2014)], KSH [Liu, Wang, Ji et al. (2012)], MLH [Norouzi and Fleet (2011)], BRE [Kulis and Darrell (2009)], ITQ-CCA [Gong, Lazebnik, Gordo et al. (2012)].

Based on our preliminary experimental results, we finally set up two empirical parameters. The scalar α is set to 0.5, the scalar λ is set to 0.008. The experimental environment for the evaluation is a computer with an E5-2630 v3 CPU, 32GB of RAM, and an NVIDIA K4200.

Table 2: Divisions of datasets

Datasets	Training set	Test set
CIFAR-10	50000	10000
NUS-WIDE	59300	10000

4.1 Datasets and evaluation metrics

We conduct the experiments on two image datasets, CIFAR-10 dataset contains color tinny images and NUS-WIDE dataset is a multi-label dataset with complicated objects. Both datasets are full of rich information, which are introduced in detail as shown in Tab. 2.

To evaluate the quality of hashing, we use the following evaluation metrics: the Mean Average Precision (mAP) for different code lengths, Precision curves within Hamming distance 2 and Precision curves w.r.t. different number of top returned samples, which are most commonly used by the scientific community for benchmarking. We evaluate the retrieval results based on whether the query image and the returned images have the same labels. The definition of evaluation criteria is as follows:

Mean Average Precision(mAP): mAP is the overall assessment measure of retrieval performance.It is the mean value of avearge percision(AP) of all queries, where AP is calculated by Eq. (8).

$$AP = \frac{1}{R}\sum_{k=1}^{n}\frac{k}{R_k} \times rel_k \tag{8}$$

where n is the size of the dataset, R is the total number of related images in the dataset, and

R_k is the number of the related images in the top k returns. rel_k is an indicator function with $rel_k = 1$ if the image at position k is relevant, and $rel_k = 0$ otherwise.

Precision@k: It is the percentage of true neighbors on the top k retrieved samples and can be calculated in Eq. (9).

$$Precision = \frac{\sum_{i=1}^{k} rel_i}{k} \tag{9}$$

where i denotes the ith image in the top k returned images, and rel_i is also an indicator function with the same meaning as mentioned above.

Table 3: Mean Average Precision (mAP) for different hashing code numbers on CIFAR-10 and NUS-WIDE datasets

Methods	CIFAR-10				NUS-WIDE			
	12-bit	24-bit	36-bit	48-bit	12-bit	24-bit	36-bit	48-bit
IDSH (with C-L)	**0.8360**	**0.8523**	**0.8474**	**0.8581**	**0.6062**	**0.6082**	**0.6127**	**0.6124**
IDSH (without C-L)	0.8388	0.8490	0.8642	0.8526	0.6140	0.6162	0.6150	0.6070
DHLE	0.8220	0.8210	0.8330	0.8620	–	–	–	–
DFH	0.4570	0.5130	0.5240	0.5590	0.6220	0.6590	0.6740	0.6950
DSDH	0.7400	0.7860	0.8010	0.8200	–	–	–	–
DSH	0.6157	0.6512	0.6607	0.6755	0.5483	0.5513	0.5582	0.5621
DNNH	0.5708	0.5875	0.5899	0.5904	0.5471	0.5367	0.5258	0.5248
CNNH	0.5425	0.5604	0.5664	0.5574	0.4315	0.4358	0.4451	0.4332
KSH	0.2948	0.3723	0.4019	0.4167	0.4331	0.4592	0.4659	0.4692
BRE	0.1589	0.1632	0.1697	0.1717	0.3556	0.3581	0.3549	0.3592
MLH	0.1844	0.1994	0.2053	0.2094	0.3829	0.3930	0.3959	0.3990
CCA-ITQ	0.1653	0.1960	0.2085	0.2176	0.3874	0.3977	0.4146	0.4188
ITQ	0.1080	0.1088	0.1117	0.1184	0.3425	0.3464	0.3522	0.3576
SH	0.1319	0.1278	0.1364	0.1320	0.3401	0.3374	0.3343	0.3332
LSH	0.1277	0.1367	0.1407	0.1492	0.3329	0.3392	0.3450	0.3474

4.2 Retrieval results

Tab. 3, Fig. 3 and Fig. 4 show the comparison results of search accuracy on all of the datasets. The results of comparison method are obtained from the experimental results provided by their authors, respectively.

As we can see from Tab. 3, the mAP of the proposed hashing method performs better than other state-of-the-art supervised hashing methods. In general, compared with other methods, our method obtains the best mAP on each of the two datasets. For example,

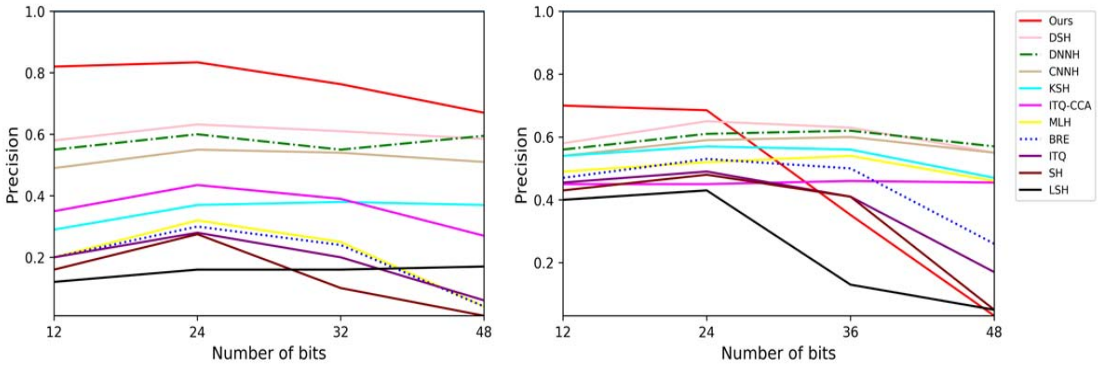

Figure 3: The precision of different hash bits on CIFAR-10 and NUS-WIDE

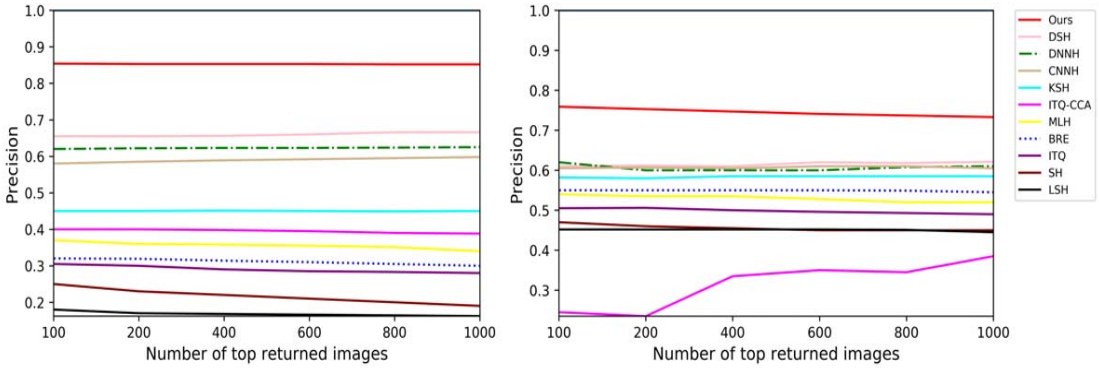

Figure 4: The precision curves with 48 bits w.r.t. different number of top returned samples on CIFAR-10 and NUS-WIDE

compared with DSH, the mAP of method increases by 18.26%~2.03% / 5.03%~5.79% on CIFAR-10 / NUS-WIDE. On the other hand, in constrast to tradition methods, the deep learning methods have been greatly improved. The methods used make image retrieval easy to apply in new domains.

These results show that our hashing method can learn useful representation of images which preserve similarities. But in the case of the results on center loss, we notice that the improvement is not stable as expected. Further relevant experiments will described in Section 4.3.

The precision of different hash bits and the precision curves with 48 bits w.r.t. different number of top returned samples on each dataset are depicted in Fig. 3 and Fig. 4. From the figures, some of the results show that our method is reliable.

4.2.1 Comparison results of BN against without BN

We use CIFAR-10 as the comparison dataset, we implement and compare the search results of the proposed framework with batch normalization to its alternative without batch

normalization. The results of comparison are shown in the Tab. 4. As we can see in Tab. 4, the search results of the network with batch normalization perform better than the alternative without batch normalization. With the addition of batch normalization, the results on each hash bit have been improved. For example, the mAP increases by 2.9%~9.11% and the precision increases 0.1%~10.5%. The fundamental reason why search results have been improved is that the distribution of training data is more balanced due to the addition of batch normalization. In addition, by comparing the mAP in Tab. 3 and Tab. 4, the addition of Divide-and-Encode Module also improves the retrieval precision. As mentioned in previous section, the division of hash features by Divided-and-Encode Module reduces redundancy between features.

Table 4: Comparison results of the proposed framework with BN against without BN on CIFAR-10

Methods	12-bit	24-bit	32-bit	48-bit
		mAP		
with BN	0.8020	0.8240	0.8300	0.8380
without BN	0.7100	0.7950	0.7920	0.8050
	Precision within Hamming radius 2			
with BN	0.7700	0.7920	0.7720	0.6890
without BN	0.6650	0.7790	0.7540	0.6880

4.2.2 Comparison results of the Softmax loss against with Center loss

We can see clearly from the experimental data above that there are some improvements on the data compared with the traditional works. But we find that there are almost no improvements on two data sets after adopting center loss, it is even a downward trend in some cases. This is inconsistent with the results we envisaged using center loss. We hold the view that it is because of the complexity of the data sets that causes center loss not work. In order to verify the effectiveness of the center loss, we try to select two simple data sets to perform a comparison experiment.

MNIST dataset consists of 70K 28×28 grayscale images of handwritten digits from 0 to 9. There are 60000 training images and 10000 test images.

The Google street-view house number dataset (SVHN) consists of 32×32 images of house number digits captured from Google Streetview. It has about 600 k images for training and 26 k images for testing.

We randomly select 10,000 images as test data, and the remaining are training data. The training environment is still consistent with the above.

We choose these two datasets for two reasons. First, the two datasets are simpler than former datasets, each class of data is more similar, which conform to the concept of the proposed 'center'. Secondly, we choose the two instead of one to avoid the contingency of experimental data, which verify the effectiveness of center loss. We implement and

compare the mAP of the proposed method with center loss to the architecture only with softmax loss on the two datasets. Several results are shown in the Tab. 5 and Fig. 5.

Table 5: Mean Average Precision (mAP) for different hashing code numbers on the MNIST and SVHN datasets

Methods	MNIST				SVHN			
	12-bit	24-bit	36-bit	48-bit	12-bit	24-bit	36-bit	48-bit
IDSH (with C-L)	0.9682	0.9773	0.9756	0.9783	0.9305	0.9378	0.9443	0.9462
IDSH (without C-L)	0.9530	0.9616	0.9787	0.9720	0.9303	0.9422	0.9284	0.9422

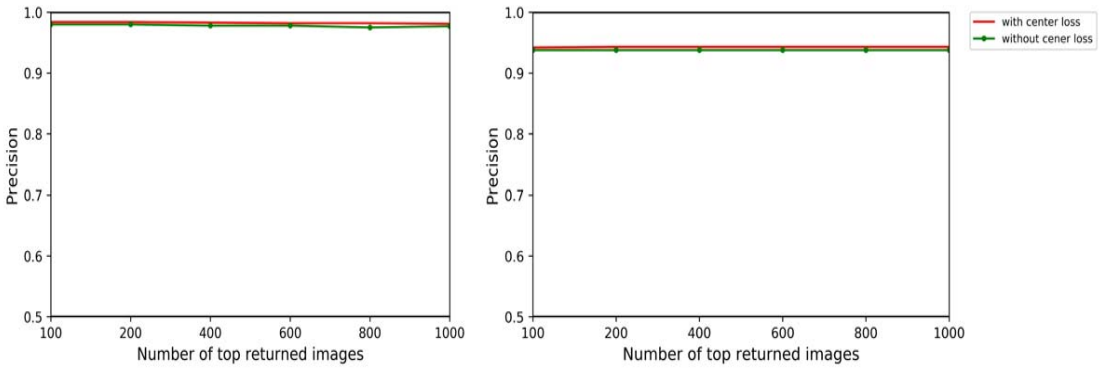

Figure 5: The comparison precision of 48-bit on MNIST and SVHN

As can be seen from Tab. 5 and Fig. 5, the results of the proposed method on simple datasets show some good performance.

The mAP for different hashing code numbers on the two datasets shows the effectiveness of center loss. Compared with the method based on softmax loss, the proposed method with center loss shows a relative increase of 0.63%~1.52% / 0.02%~1.59% on MNIST / SVHN, respectively. In addition, there are still slight declines on some hash bits, which attribute from insufficient or differences from data. Fig. 5 also visually shows the good performance of center loss. We conclude that the main reason for this is probably that the difference between the same classes of data is different. Conversely, under the circumstance of data with larger variability, the center cannot be updated due to large intra-class distance leading to unbalanced center so that it cannot be optimized.

This auxiliary experiment demonstrates that center loss has the promotion ability on highly similar dataset. On the other hand, from these experimental results, we can realize the center more clearly. If only using softmax loss freas supervision standard, deeply learned features would contain large intra-class variations, which can still be further optimized. However, simply using the center loss also could not achieve discriminative feature

learning. So the center loss is jointly used to supervise the CNNs to solve the problem of intra-class, as confirmed by our experiments.

4.2.3 Illustration of retrieval

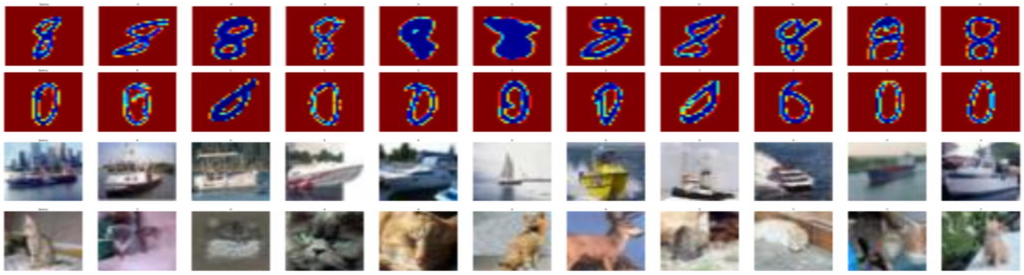

Figure 6: Top 10 retrieved images on CIFAR-10 and MNIST datasets

The top 10 retrieved images on MNIST and CIFAR-10 are shown in Fig. 6 as an illustration. The hash length is 48 bits. The first is query image, followed by ten retrieval images. It performs well in top 10 retrieval images.

4.2.4 Efficiency of hashing learning

The experimental environment for the evaluation is a computer with an E5-2630 v3 CPU, 32GB of RAM, and an NVIDIA K4200. The method we proposed is implemented based on the open source Caffe framework. Similar to general CNNs, the time complexity of shared CNN is O($\sum_{l=1}^{D} M_l^2 * k_l^2 * C_{l-1} * C_l$). D denotes the depth of shared CNN, l is the lth convolution kernel, C_i is the number of convolution kernels. The total time complexity is accumulation of all the time complexity convolution kernels. The time of computing the hamming distance between two 48 bits binary codes is less than that of traditional exhaustive search with high dimensional features. On the other hand, due to the supplement of Divide-and-Encode Module, the parameters have been reduced in our experiments. It saves some time in training phase.

5 Conclusion

In this paper, we present an effective supervised hashing method for fast image retrieval. Our proposed IDSH is simple but efficient to learn hash function that generates the same or similar binary codes directly. We use Batch Normalization (BN) layers before activation layers and Divide-and-Encode Module to generate compact binary codes. Furthermore, we use center loss and softmax loss to optimize on training stage. The experimental results on the some datasets demonstrate that the effectiveness of the proposed hashing method compared with other state-of-the-art ones. In addition, the comparisons of mAP on MNIST and SVHN also show center loss can improve retrieval performance to a certain extent.

References

Bay, H.; Tuytelaars, T.; Gool, L. V. (2006): SURF: speeded up robust features. *9th European Conference on Computer Vision*, pp. 404-417.

Cao, Z. J.; Long, M. S.; Wang, J. M.; Yu, P. S. (2017): HashNet: deep learning to hash by continuation. *IEEE International Conference on Computer Vision*, pp. 5608-5617.

Datta, R.; Li, J.; Wang, J. Z. (2005): Content-based image retrieval: approaches and trends of the new age. *7th ACM SIGMM International Workshop on Multimedia Information Retrieval*, pp. 253-262.

Deng, C.; Chen, Z. J.; Liu, X. L.; Gao, X. B.; Tao, D. C. (2018): Triplet-based deep hashing network for cross-modal retrieval. *IEEE Transactions on Image Processing*, vol. 27, no. 8, pp. 3893-3903.

Ge, L. W.; Zhang, J.; Xia, Y.; Chen, P.; Wang, B. et al. (2019): Deep spatial attention hashing network for image retrieval. *Journal of Visual Communication and Image Representation*, vol. 63.

Gionis, A.; Indyk, P.; Motwani, R. (1999): Similarity search in high dimensions via hashing. *25th International Conference on Very Large Data Bases*, pp. 518-529.

Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. (2014): Rich feature hierarchies for accurate object detection and semantic segmentation. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 580–587.

Gong, Y. C.; Lazebnik, S.; Gordo, A.; Perronnin, F. (2012): Iterative quantization: a procrustean approach to learning binary codes for large-scale image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 12, pp. 2916-2929.

Gui, J.; Liu, T. L.; Sun, Z. A.; Tao, D. C.; Tan, T. N. (2018): Fast supervised discrete hashing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 2, pp. 490-496.

Ioffe, S.; Szegedy, C. (2015): Batch normalization: accelerating deep network training by reducing internal covariate shift. *32nd International Conference on International Conference on Machine Learning*, pp. 448-456.

Krizhevsky, A.; Sutskever, I.; Hinton, G. E. (2012): Imagenet classification with deep convolutional neural networks. *25th International Conference on Neural Information Processing Systems*, pp. 1097-1105.

Kulis, B.; Darrell, T. (2009): Learning to hash with binary reconstructive embeddings. *22nd International Conference on Neural Information Processing Systems*, pp. 1042-1050.

Lai, H. J.; Pan, Y.; Liu, Y.; Yan, S. C. (2015): Simultaneous feature learning and hash coding with deep neural networks. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3270-3278.

Li, J. Y.; Li, J. H. (2015): Fast image search with deep convolutional neural networks and efficient hashing codes. *12th International Conference on Fuzzy Systems and Knowledge Discovery*, pp. 1285-1290.

Li, Q.; Sun, Z. A.; He, R.; Tan, T. (2017): Deep supervised discrete hashing. *31st Annual Conference on Neural Information Processing Systems*, pp. 2482-2491.

Li, Y.; Miao, Z.; Wang, J. B.; Zhang, Y. F. (2018): Deep binary constraint hashing for fast image retrieval. *Electronics Letters*, vol. 54, no. 1, pp. 25-27.

Lin, J.; Li, Z. C.; Tang, J. H. (2017): Discriminative deep hashing for scalable face image retrieval. *26th International Joint Conference on Artificial Intelligence*, pp. 2266-2272.

Liu, H. M.; Wang, R. P.; Shan, S. G.; Chen, X. L. (2016): Deep supervised hashing for fast image retrieval. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2064-2072.

Liu, W.; Wang, J.; Ji, R. R.; Jiang, Y. G.; Chang, S. F. (2012): Supervised hashing with kernels. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2074-2081.

Lowe, D. G. (2004): Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, vol. 60, no. 2, pp. 91-110.

Lu, X. C.; Song, L.; Xie, R.; Yang, X. K.; Zhang, W. J. (2017): Deep hash learning for efficient image retrieval. *IEEE International Conference on Multimedia and Expo Workshops*, pp. 579–584.

Norouzi, M.; Fleet, D. J. (2011): Minimal loss hashing for compact binary codes. *28th International Conference on Machine Learning*, pp. 353-360.

Oliva, A.; Torralba, A. (2001): Modeling the shape of the scene: A holistic representation of the spatial envelope. *International journal of Computer Vision*, vol. 42, no. 3, pp. 145-175.

Oquab, M.; Bottou, L.; Laptev, I.; Sivic, J. (2014): Learning and transferring mid-level image representations using convolutional neural networks. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1717-1724.

Qiu, G. P. (2002): Indexing chromatic and achromatic patterns for content-based colour image retrieval. *Pattern Recognition*, vol. 35, no. 8, pp. 1675-1686.

Salakhutdinov, R.; Hinton, G. E. (2009): Semantic hashing. *International Journal of Approximate Reasoning*, vol. 50, no. 7, pp. 969-978.

Sharif Razavian, A.; Azizpour, H.; Sullivan, J.; Carlsson, S. (2014): Cnn features off-the-shelf: an astounding baseline for recognition. *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 806-813.

Shi, X. S.; Sapkota, M.; Xing, F. Y.; Liu, F. J.; Cui, L. et al. (2018): Pairwise based deep ranking hashing for histopathology image classification and retrieval. *Journal of Visual Communication and Image Representation*, vol. 81.

Szegedy, C.; Liu, W.; Jia, Y. Q.; Sermanet, P.; Reed, S. et al. (2015): Going deeper with convolutions. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1-9.

Taigman, Y.; Yang, M.; Ranzato, M.; Wolf, L. (2014): Deepface: closing the gap to human-level performance in face verification. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1701-1708.

Tang, J. H.; Lin, J.; Li, Z. C.; Yang, J. (2018): Discriminative deep quantization hashing for face image retrieval. *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 12, pp. 1-9.

Wang, X. F.; Lee, F. F.; Chen, Q. (2019): Similarity-preserving hashing based on deep neural networks for large-scale image retrieval. *Journal of Visual Communication and Image Representation*, vol. 61, pp. 260-271.

Wang, X. F.; Shi, Y.; Kitani, K. M. (2016): Deep supervised hashing with triplet labels. *13th Asian Conference on Computer Vision*, pp. 70-84.

Wu, D. Y.; Dai, Q.; Liu, J.; Li, B.; Wang, W. P. (2019): Deep incremental hashing network for efficient image retrieval. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 9069-9077.

Xia, R. K.; Pan, Y.; Lai, H. J.; Liu, C.; Yan, S. C. (2014): Supervised hashing for image retrieval via image representation learning. *28th AAAI Conference on Artificial Intelligence*, pp. 2156-2162.

Yang, D. B.; Xie, H. T.; Yin, J.; Liu, Y. Z.; Yan, C. G. (2017): Supervised deep quantization for efficient image search. *IEEE International Conference on Multimedia and Expo Workshops*, pp. 525-530.

Yang, H. F.; Lin, K.; Chen, C. S. (2018): Supervised learning of semantics-preserving hash via deep convolutional neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 2, pp. 437-451.

Zhang, J.; Peng, Y. X. (2017): SSDH: semi-supervised deep hashing for large scale image retrieval. *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 1, pp. 212-225.

Zhang, R. M.; Lin, L.; Zhang, R.; Zuo, W. M.; Zhang, L. (2015): Bit-scalable deep hashing with regularized similarity learning for image retrieval and person re-identification. *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 4766-4779.

Zhao, F.; Huang, Y. Z.; Wang, L.; Tan, T. N. (2015): Deep semantic ranking based hashing for multi-label image retrieval. *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1556-1564.

Zhou, C.; Po, L. M.; Liu, M.; Yuen, W. Y.; Wong, P. H. et al. (2019): Deep hashing with triplet labels and unification binary code selection for fast image retrieval. *International Conference on Multimedia Modeling*, pp. 277-288.

Zhou, M.; Zeng, X. H.; Chen, A. Z. (2019): Deep forest hashing for image retrieval. *Pattern Recognition*, vol. 95, pp. 114-127.

Zhou, Y. F.; Huang, S. S.; Zhang, Y.; Wang, Y. F. (2017): Deep hashing with triplet quantization loss. *IEEE Visual Communications and Image Processing*, pp. 1-4.