Neurocomputing xxx (xxxx) xxx

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Unsupervised semantic deep hashing☆

Sheng Jin, Hongxun Yao, Xiaoshuai Sun*, Shangchen Zhou

Harbin Institute of Technology, China

ARTICLE INFO

Article history:
Received 14 May 2018
Revised 9 November 2018
Accepted 8 January 2019
Available online xxx

Communicated by Dr. Min Xu

Keywords:
Deep learning
Unsupervised hashing
Semantic loss

ABSTRACT

In recent years, deep hashing methods have been proved to be effective since it employs convolutional neural network to learn features and hashing codes simultaneously. However, these methods are mostly supervised. In real-world applications, it is a time-consuming and overloaded task for annotating a large number of images. In this paper, we propose a novel unsupervised deep hashing method for large-scale image retrieval. Our method, namely unsupervised semantic deep hashing (**USDH**), uses semantic information preserved in the CNN feature layer to guide the training of network. We enforce four criteria on hashing codes learning based on VGG-19 model: 1) preserving relevant information of feature space in hashing space; 2) minimizing quantization loss between binary-like codes and hashing codes; 3) improving the usage of each bit in hashing codes by using maximum information entropy, and 4) invariant to image rotation. Extensive experiments on CIFAR-10, NUSWIDE have demonstrated that **USDH** outperforms several state-of-the-art unsupervised hashing methods for image retrieval. We also conduct experiments on Oxford 17 datasets for fine-grained classification to verify its efficiency for other computer vision tasks.

1. Introduction

With the explosive increase of data, searching for the content relevant image, or other media data remains a challenge because of a large amount of computational cost and the accuracy requirement. In the early stage, researchers focus on data-independent methods. Locality-Sensitive Hashing [1] and its variants are proposed. But it has a lower accuracy since the semantic information of data is not considered during the coding process. In recent years, the data-dependent hashing methods [2] attract more attention since its compact representation and superior accuracy performance. Compared with data-independent hashing method, data-dependent hashing methods improve retrieval performance via training on the dataset.

Data-dependent methods mainly include supervised hashing methods [3], unsupervised hashing methods [4] and semi-supervised hashing methods [5]. These supervised methods make use of the class information provided in the manual labels, where the supervised information is used in three forms: point-wise labels, pair-wise labels, and ranking labels. Some representative works have been proposed, e.g. Semantic Hashing [6], Binary Reconstruction Embedding [7], Minimal Loss Hashing [8], Kernel-based Supervised Hashing [3], Hamming Distance Metric Learning [9], and Column Generation Hashing [10]. Although the supervised hashing methods and semi-supervised hashing methods have been proved to gain better accuracy with compacter hashing codes, it is a time-consuming and heavy workload task in the practical application. In the past years, some classical unsupervised hashing methods also have been developed, e.g. Isotropic Hashing [11], Spherical Hashing [12], Discrete Graph Hashing [13], Locally Linear Hashing [14], Asymmetric Inner-product Binary Coding [15] and Scalable Graph Hashing [16].

In these traditional hashing methods, each image is initially represented by a hand-crafted feature. However, these features may not preserve accurate semantic information. And they also may not be suitable for generating binary codes. Due to these facts, the accuracy of image retrieval could not meet our requirement. Over the last five years, deep learning has been proved to be effective in computer vision because it could automatically extract high-level semantic feature to represent images that is robust to the variances of the object. Salakhutdinov and Hinton [6] firstly proposed hashing method based on deep neural networks. However, in CNNH [6], the input of the network is still hand-crafted features, which is the most crucial limitation.

Very recently, Convolutional Neural Network Hashing [17] introduces an end-to-end network into hashing learning. However, this method has limitations since it cannot perform feature learning and hashing codes learning simultaneously. Followed [17], new variants of deep hashing have been proposed, e.g, Deep Neural Network Hashing [18], Deep Semantic Ranking Hashing [19], deep supervised hashing [20] and DeepBit [21], which extract features

☆ Fully documented templates are available in the elsarticle package on CTAN.

* Corresponding author.
E-mail addresses: 16B903055@stu.hit.edu.cn (S. Jin), h.yao@hit.edu.cn (H. Yao), xiaoshuaisun@hit.edu.cn (X. Sun), sczhou@hit.edu.cn (S. Zhou).

https://doi.org/10.1016/j.neucom.2019.01.020
0925-2312/

and learn hashing codes simultaneously. These methods are more effective and perform more efficiently in image retrieval task. However, most of these deep hashing methods, except DeepBit [21] and DBD-MQ [22], are pure supervised. DBD-MQ [22] propose a quantization method for hashing learning. This method does not utilize the rigid sign function for binarization and considers the binarization as a multi-quantization task. The goal of DeepBit [21] is to learn rotation-invariant hashing codes, which is similar to RICNN [23]. The Deepbit method tries to make hashing codes invariant to rotation by minimizing the difference between the hashing codes that describe the reference image and that of rotated one. However, this method only considers rotation invariance of images. The invariance among different images with the same class label cannot be guaranteed.

In view of the limitation of the DeepBit [21] method, we propose a novel unsupervised hashing method, called unsupervised semantic hashing (**USDH**). Our goal is able to learn hashing codes invariant to other attribute factors besides rotation. Without label information, we attempt to mine the semantic information preserved in the feature space for learning better hashing codes, which is proved to be effective in some traditional hashing methods.

However, general deep hashing methods use the end-to-end network to learn hashing codes directly from the input image. So two problems need to be solved. The first problem is that we should extract semantic information-preserved features by the deep network in an unsupervised manner. Extensive related experimental results proved the pre-trained models can achieve high performance on a new dataset for classification tasks, whether the parameters of the convolution layers is updated or not. Motivated by the success of these pre-trained models, the feature maps extracted from these convolution layers preserve accurate semantic information. In this paper, these feature maps are used as features. The second problem is how to use the semantic information of these feature maps. We propose a novel semantic loss. The key idea of the novel loss is that the hashing codes is required to preserve the neighbor structure in feature space.

The main contributions of **USDH** are outlined as follows:

USDH is an unsupervised end-to-end deep hashing framework. Compared with the DeepBit method, **USDH** not only considers rotation invariance in a single image but also preserves the semantic information of image pairs.

USDH proposes a novel deep unsupervised hashing method to preserve the semantic information in the feature space. It regards the output of fully-connected layer as representation descriptor of the image. We propose a novel semantic loss. The novel loss component requires hashing codes approximating the similarity computed by representation descriptors of the image.

Experiments on general datasets show that **USDH** can outperform other unsupervised methods to achieve the state-of-the-art performance in image retrieval applications. And it is also quite effective for fine-grained classification.

2. Related work

Unsupervised hashing: Data-based hashing method can be divided into two mainstreams which depend on whether label information is used or not: supervised method and unsupervised method. For the first mainstreams, compared with supervised hashing, unsupervised methods do not need the label in training stage. There exists some representative method, for example, Spectral Hashing [4] proposes a novel spherical hashing scheme pulls hashing codes which are neighbors in feature space together and also required hashing codes to be balanced and uncorrelated. Chang et al. propose Binary Reconstructive Embedding (BRE) [7]. The method is designed by minimizing a cost function measuring the difference between the metric and reconstructed distance in hamming space. Some quantization methods have been proposed. Jegou et al. [24] introduce product quantization and decompose the space into the Cartesian product of low dimensional subspaces. These subspaces are quantized separately. The famous Iterative Quantization method [25] is proposed by Gong via minimizing the quantization error of mapping this data to the vertices of a zero-centered binary hypercube and it is worth to note this method can be used without supervised data by embedding PCA. The OPQ [26] optimizes product quantization by minimizing quantization distortions. The optimal problem is solved by searching for optimal codebooks and space decomposition. Since OPQ performs significantly better when the underlying distribution is unimodal, Kalantidis and Avrithis [27] propose Locally Optimized Product Quantization. This method partitions data in cells and locally optimizes one product quantizer per cell on the residual distribution. As described in this section, these unsupervised method use the hand-crafted feature which limits their performance.

Deep learning: In current years, deep learning, especially deep convolutional network, has been proved effective for many computer vision tasks, including classification, detection [28], segmentation, captioning and image matching. Krizhevsky et al. [29] propose the famous deep neural network architecture named Alex-Net. Alex-Net contains five convolutional and three fully-connected layers, and the paper employs some effective techniques for preventing the overfitting problem, such as data augmentation and drop-out. Very recently, VGG model [30] is proposed by Simonyan and Zisserman. The main contribution of the model is a thorough evaluation of networks of increasing depth using an architecture with very small (3×3) convolution filters. Extensive experiments have proved that depth of network is key factors affecting the efficiency of the deep model. So many deep frameworks are introduced, such as google-net [31], res-net [32] and same variants of res-net [33]. In our work, the architecture of our deep network is the same as that of VGG. However, we adopt different loss function to obtain hashing codes in an unsupervised fashion.

Deep hashing: The success of the deep convolutional network is attributed to their ability to learn effective feature. These features are robust to within-class variance and have high-level semantic information. In the recent years, the deep convolutional network is introduced into image retrieval. Salakhutdinov and Hinton [6] firstly employ deep neural networks to learn efficient hashing code. But this method still uses the hand-crafted feature as the input of network and just improves performance with non-linear representation ability of network. Xia et al. [17] propose a deep hashing method named CNNH by learning image representation automatically. CNNH includes two training stages. In the first stage, hashing codes are approximated by decomposing a pair-wise similarity matrix. In the second stage, the deep network is trained to fit the approximated hashing codes. However, this method has its limitation that the image representation could not give feedback to binary codes and the computation of matrix decomposition is large. Existing deep hashing methods, including Deep Neural Network Hashing [18], Deep Hashing Network [34] and deep supervised hashing [20] improve retrieval performance via simultaneously learning image representation and hashing codes. Deep Neural Network Hashing [18] uses a triplet ranking hashing loss to preserve semantic information. Deep Hashing Network [34] has further improved DNNH by controlling quantization loss and preserving semantic information by pair-wise cross-entropy loss. These methods have been proved superior over the state-of-the-art traditional hashing methods. However, these deep hashing methods need label information.

Most recently, compared with the DeepBit method, DBD-MQ [22] do not utilize the rigid sign function for binarization and considers the binarization as a multi-quantization task. A

Please cite this article as: S. Jin, H. Yao and X. Sun et al., Unsupervised semantic deep hashing, Neurocomputing, https://doi.org/10.1016/j.neucom.2019.01.020

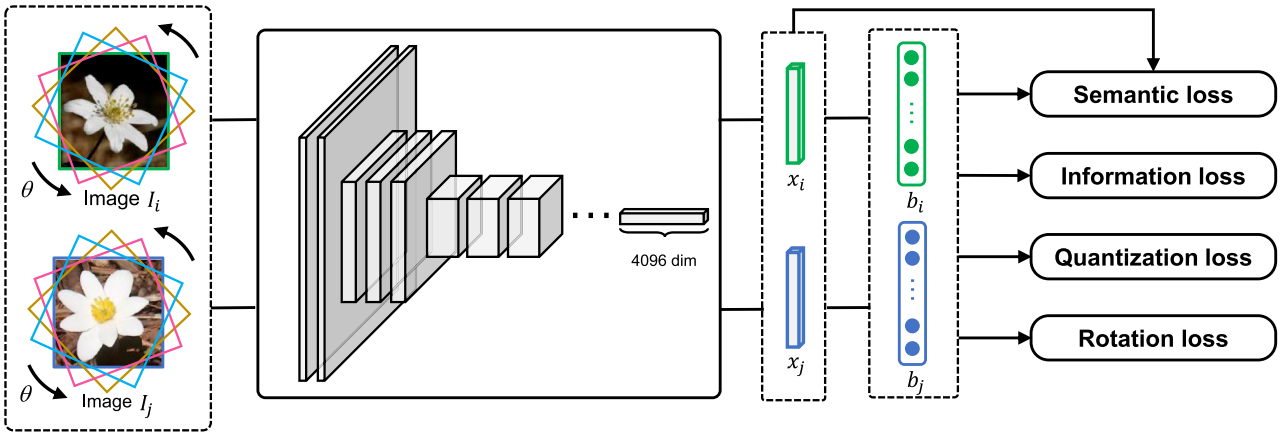


Fig. 1. We enforce four criterions on the loss function to learn efficient hashing codes based on VGG-19 architecture. In the training stage, the hashing codes are learned by the form of image pairs. On the first stage, we train the deep model by minimizing quantization loss, information loss and use the mid-level feature to guide the process of learning hashing codes. On the second stage, we augment dataset with rotation, hashing codes are learned to be invariant to rotation by minimization the distance between that represents reference image and that of rotated one.

K-AutoEncoders (KAEs) network was applied to jointly learn the parameters and the binarization functions under a deep learning framework. In our paper, we propose a deep unsupervised hashing method, but compared to existing deep unsupervised hashing method, we adopt a pair of images as the input of network and require the hashing code to preserve the similarity information in the feature space. The similar idea is proved to be effective in image classification [23].

3. Unsupervised semantic deep hashing

3.1. Overview

Fig. 1 shows the framework of the proposed USDH method. Given an unlabeled dataset, our goal is to use the semantic information preserved in the feature space to learn more accurate hashing codes. With this intuition in mind, we propose a novel unsupervised deep hashing network, which consists of two steps, learning features from the input images and learning hashing codes from the learned features. Specially, VGG-19 is used as the base model. We regard the output of the second full-connected layer as image features. Then we introduce the novel semantic loss to make use of these features, which is designed to guide the training of the whole network.

The loss function consists of other three components, including the information loss, the quantization loss, and the rotation loss. We adopt the information loss to improve the usage of each bit. The rotation loss, which is proved to be effective in deepbit [21], is used to keep the learned hashing codes invariant to rotation. And the binary constraint of hashing codes makes it intractable to train an end-to-end deep model with the backpropagation algorithm. So we relax the discrete constraint in the training phase. The quantization loss is devised to measure the loss between binary-like codes and hashing codes.

The whole cost function is written as below:

$$J = J_1 + J_2 + J_3 + J_4, \tag{1}$$

J_1 represents semantic loss, J_2 represents quantization loss, J_3 represents information loss, J_4 represents rotation loss.

3.2. Semantic loss

The semantic loss is designed to preserve the neighbor structure of the feature space. To preserve semantic information in the feature space, firstly, we should adopt an optimal feature to represent images and use a proper formula to measure the similarity of images in the feature space, then we let similarity computed by the hashing codes of image pairs approximate the similarity measured in the original feature space.

Firstly, we adopt the VGG-19 model to process the images and use the output of the second full-connected layer as our image feature. Currently, many research works have proved that high-level feature of the convolutional neural network has sufficient semantic information and these mid-features are robust to inner-class including rotation, shape and color variance. There also exist different metrics to measure similarity in the feature space. We adopt a widely-used metric that is defined as:

$$S_{i,j} = e^{\frac{-\|x_i - x_j\|_2}{\rho d}}, \tag{2}$$

Where d denotes the dimension of the second full-connected layer and ρ is a positive constant parameter. $S_{i,j} \in (0, 1]$ can represent a similarity degree of the images i and j. The hashing codes of image i is denoted as b_i.

We require hashing codes preserving relevant semantic information. More specifically, if $S_{i,j}$ is near to 1, we assume hashing codes b_i and b_j has smaller distance. But if $S_{i,j}$ is near to 0, then b_i and b_j has larger distance. For each training batch, we can obtain a similarity matrix. We try to use the similarity degree in the feature space to guide the learning of hashing codes. To do so, in hamming space, we also define a similarity measure, and then the similarity measure defined in hashing space is required to be as similar as possible to the similarity matrix defined in the original feature space.

According to this constraint, the neighbor points in the feature space are still neighbors in the hashing space. Specifically, $b_i \in \{0, 1\}$ is relaxed to (0,1), then the hashing codes is linearly transformed to $(-1, 1)$:

$$\widetilde{b}_i = 2b_i - 1, \tag{3}$$

where $\widetilde{b}_i \in (-1, 1)$. The inner product of $\widetilde{b}_i$and $\widetilde{b}_j$ is in the range of $(-k, k)$, where k is the length of hashing codes. Then the inner product is linear transformed to (0,1) via $\frac{\widetilde{b}_i^T * \widetilde{b}_j + k}{2k}$. The result of the linear transformation is also regarded as a similarity degree. And it fits in with the assumption on information loss that each bit of hashing codes plays the same role. The function of semantic loss is

Please cite this article as: S. Jin, H. Yao and X. Sun et al., Unsupervised semantic deep hashing, Neurocomputing, https://doi.org/10.1016/j.neucom.2019.01.020

written as:

$$J_1 = \sum_{i,j} \left| S_{i,j} - \frac{\widetilde{b}_i^T * \widetilde{b}_j + k}{2k} \right|_1. \quad (4)$$

With this loss function, the deep model is trained by the back-propagation algorithm with batch gradient descent method. To solve this, the gradient of semantic loss function need to be computed. Since l_1 norm is non-differentiable at some certain point, we employ sub-gradient to overcome the problem and we define the sub-gradient at this point to be equal to the right-hand derivative. The gradient of semantic loss is defined as:

$$\frac{\partial J_1}{\partial b_i} = \sum_j sgn\left(S_{i,j} - \frac{\widetilde{b}_i^T * \widetilde{b}_j + k}{2k} \right) * \frac{\widetilde{b}_j}{2k} \quad (5)$$

where

$$sgn(x) = \begin{cases} 1 & x \geq 0, \\ -1 & x < 0. \end{cases}$$

3.3. Quantization loss

Since it is difficult to directly optimize discrete loss function, we should relax the objective function to transform the discrete problem into a continuous optimization problem. As discussed in [20], some widely-used relaxation scheme working with non-linear functions, such as sigmoid and tanh function, would inevitably slow down or even restrain the convergence of the network [29]. To overcome such limitation, we still use the relu function as activation function of the second fully-connected layer. Then the output of the network is quantized to the binary codes. The quantization function is written as:

$$f(b_i) = \begin{cases} 1 & b_i \geq 0.5, \\ 0 & b_i < 0.5. \end{cases}$$

where $f(x)$ denotes the binarization function.

To decrease this loss, we let the value of network's output near to 1 or 0. First, the hashing codes b_i is linearly transformed to $(-1, 1)$ in the same way. Then the result of linear transformation is changed into an absolute value. The absolute value of the hashing codes $|\widetilde{b}_i|$ should be near to 1. Finally, the quantization loss is defined as:

$$J_2 = \alpha \sum_i \left| |\widetilde{b}_i| - 1 \right|_1, \quad (6)$$

where$|.|$ denotes element-wise absolute value, and $\|.\|_1$ denotes l_1 norm. α is a weighting parameter.

To train the model, the gradient of J_2 need to be computed. The sub-gradient is taken to replace the gradient of J_2 because of the non-differentiate point in the absolute operation and l_1 norm. The gradient is written as:

$$\frac{\partial J_2}{\partial b_i} = \begin{cases} 2\alpha & b_i \geq 1 \quad or \quad 0 < b_i < 0.5, \\ -2\alpha & otherwise. \end{cases}$$

3.4. Information loss

As the main assumption of semantic loss, each bit of hashing codes should play an equivalent impact, which means each bit should have the same mean value. Inspired by the efficiency of DeepBit [21] method, we also maximize the capability of each bit in hashing codes to express information. So we further enhance the hashing codes by assuming that each bit has half-chance to be one. Based on this constraint, the balanced distribution criterion can be written:

$$\mu_i = \frac{1}{m} \sum_{i=1}^{m} b_i(m), \quad (7)$$

where μ_i denotes the mean value of ith bite of hashing codes, $\|.\|_2$ denotes l_2 norm and m denotes the size of training batch.

3.5. Rotation loss

Existing widely-used hand-crafted features should be invariant to rotation and scale. Inspired by this motivation, we also rotate the images and pull hashing codes that represent the reference image and that of the rotated one together. The proposed rotation-invariance criterion can be written as:

$$J_4 = \sum_{i=1}^{m} \sum_{\theta=0}^{2R} \|b_{\theta,i} - b_i\|, \quad (8)$$

Where $b_{\theta,i}$ denotes hashing codes of image i with rotation θ.

4. Experiment

In order to test the performance of our proposed method, we conduct experiments on four datasets, including three widely used image retrieval datasets: CIFAR-10 and NUSWIDE dataset, as well as one recognition dataset: Oxford flower17. Similar to other image retrieval task, our method is also evaluated based on mean accuracy precision at top 1000. Compared with some representative unsupervised hashing methods, such as KMH [35], SphH [12], SpeH [4], PCAH [5], LSH [1], PCA-ITQ [25], DH [36], DeepBit [21] and DBD-MQ [22], experimental results verify that our proposed method outperforms these existing unsupervised hashing method. In order to prove our method is flexible for other computer vision applications, we also conduct experiments for fined-grained recognition on Oxford flower17 dataset.

4.1. Dataset and evaluation metric

CIFAR-10 dataset consists of 60000 32×32 images in 10 classes. Each image in dataset belongs to one class (6000 images per class). The dataset is divided into two parts: the training set (5000 images per class) and testing set (1000 images per class).

NUSWIDE dataset is a multi-label dataset. NUSWIDE contains nearly 270k images associated with 81 semantic concepts. Followed [17], We select the 21 most frequent concept. Each of concepts is associated with at least 5000 images. The dataset is split into the training set and testing set. We sample 100 images from each concept to form a testing set and the remaining images are treated as a training set.

Oxford 17 flower dataset consists of 1360 images belonging to 17 mutually classes. Each class contains 80 images. The dataset is divided into three parts, including training set, testing set, and validation set, with 40 images, 20 images, and 20 images respectively. In our experiment, we ignore validation set.

We mainly use Mean Average Precision (MAP)and Precision-Recall curves for quantitative evaluation.

4.2. Implementation details

The **USDH** method is implemented based on Caffe and the deep model is trained by batch gradient descend. As shown in Fig. 1, We use VGG-19 as the base model, and the model is firstly trained on Imagenet dataset. Then the output layer of VGG-19 is replaced by hashing layer.

In the training stage, the image is regarded as input in the form of batch and every two images in the same batch construct an image pair. The parameters of the deep model are updated by minimizing the objective function, including semantic loss, quantization loss, information loss and rotation loss. We conduct experiments for learning 16-bit, 32-bit, 48-bit hashing codes, respectively on

Please cite this article as: S. Jin, H. Yao and X. Sun et al., Unsupervised semantic deep hashing, Neurocomputing, https://doi.org/10.1016/j.neucom.2019.01.020

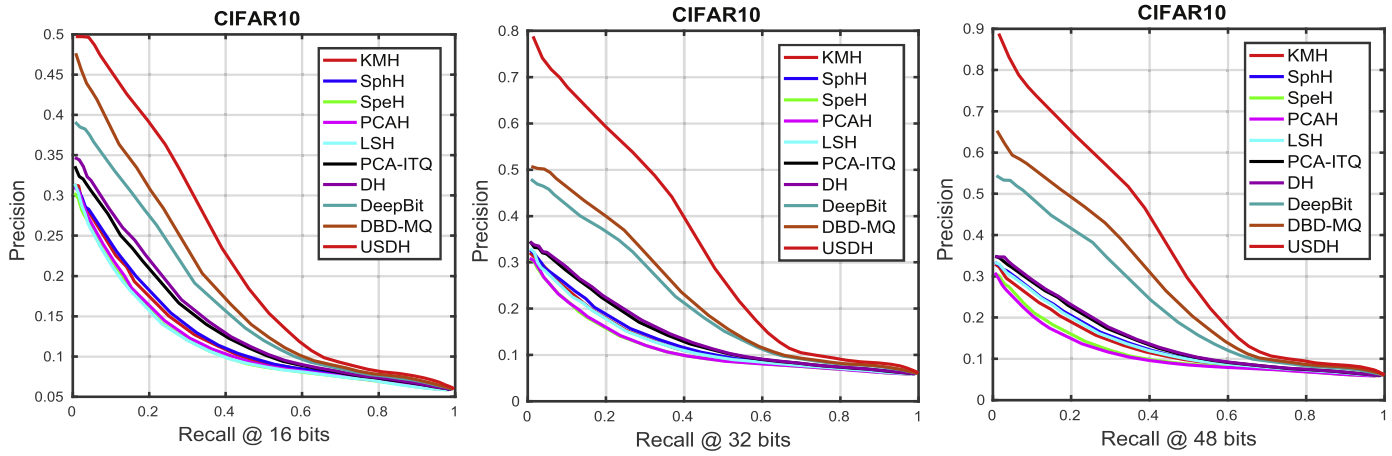


Fig. 2. Recall vs. precision curve on the CIFAR-10 dataset showed the results of different unsupervised hashing methods corresponding to 16-bits, 32-bits, 64-bits.

Table 1
Mean Average Precision (MAP) results for different number of bits CIFAR-10.

Method	16-bit	32-bit	64-bit
KMH	13.59	13.93	14.46
SphH	13.98	14.58	15.38
SpeH	12.55	12.42	12.56
PCAH	12.91	12.60	12.10
LSH	12.55	13.76	15.07
PCA-ITQ	15.67	16.20	16.64
DH	16.17	16.62	16.96
DeepBit	19.43	24.86	27.73
DBD-MQ	21.53	26.50	31.85
USDH	**26.13**	**36.56**	**39.27**

Table 2
MAP results for different number of bits NUSWIDE.

Method	16-bit	32-bit	48-bit
SphH	41.30	42.40	43.10
SpeH	43.30	42.62	42.44
PCAH	42.90	43.70	41.40
LSH	40.30	42.60	42.30
PCA-ITQ	45.28	46.82	47.70
DH	42.20	44.80	48.00
DeepBit	38.30	40.10	41.20
USDH	**64.07**	**65.68**	**65.87**

Table 3
Effectiveness of different loss function.

Dataset	Method	12 bits	24 bits	48 bits
CIFAR10	deepbit	19.43	24.86	27.73
	deepbit+semanticloss	23.31	32.48	36.18
	USDH	**26.13**	**36.55**	**39.27**
NUSWIDE	deepbit	38.30	40.10	41.20
	deepbit+semanticloss	60.28	61.22	61.73
	USDH	**64.07**	**65.68**	**65.87**

the the CIFAR-10 dataset and NUSWIDE dataset. In this paper, we propose the multiple loss functions. So we further evaluate these loss functions. The semantic loss is proved more important and our quantization loss also improves performance. Since the efficiency of the semantic loss, robustness analysis is discussed. We conduct experiments by different parameters ρ in the semantic loss. The constant parameters ρ are respectively set as d, $\frac{d}{2}$, $\frac{d}{4}$. Where d denotes the dimension of the output of the second fully-connected layer. To prove the efficiency of hashing codes learned by **USDH**, we also conduct experiments for other computer vision field, such as fine-grained classification.

Network parameters In our method, the value of hyper-parameter α is 0.01. We use the mini-batch stochastic gradient descent with 0.9 momentum. We set the value of the margin parameters m as $k/4$, where k is the bits of hashing codes. The mini-batch size of images is fixed as 32 and the weight decay parameter as 0.0005.

4.3. Results on image retrieval

Similar to DeepBit [21] method, the dataset is split into two parts. More specially, 10000 images are selected randomly as query image and then we conduct retrieval task on the remaining images for both CIFAR-10. We define similarity label based on semantic-level labels and images from the same class are considered similar. The Mean Average Precision (MAP,%) at top 1000 of different unsupervised hashing methods on CIFAR-10 dataset was shown in table1. The experimental results on Table 1 show that **USDH** outperforms existing best retrieval performance by 4.6%, 10.1%, 7.3% and improves DeepBit method by 6.7%, 11.7%, 11.5%, correspond to different hash bits, respectively 16 bits, 32 bits and 64 bits. we also conduct experiments for large-scale image retrieval. The precision curves are shown in Fig. 2.

For NUSWIDE dataset, we follow the setting in [17], and if two images share at least one same label, they are considered same. As shown in Table 2 and Fig. 3, our method absolute increases of 25.77%, 25.58%, 24.67% in the average MAP for different bits on NUSWIDE dataset.

Based on results of the experiment, **USDH** is proved to be effective for image retrieval and the semantic information among different images in feature space improves significantly performance.

4.4. Exploration experiment

Component analysis of loss function: Our loss function consists of four components. In this section, we evaluate the effectiveness of two major components: semantic loss and quantization loss. The results on CIFAR-10 and NUSWIDE are shown in Table 3. It is worth mentioning that the semantic loss has improved the performance by 7.62% and 15% compared to the DeepBit method. And the quantization loss proposed in our paper has further improved the performance by 4.07% on both datasets. The experimental results of deepbit method combined with the semantic loss for fine-grained classification is shown on Table 5. The semantic loss has improved the performance by 4.6% compared to the DeepBit method.

Please cite this article as: S. Jin, H. Yao and X. Sun et al., Unsupervised semantic deep hashing, Neurocomputing, https://doi.org/10.1016/j.neucom.2019.01.020

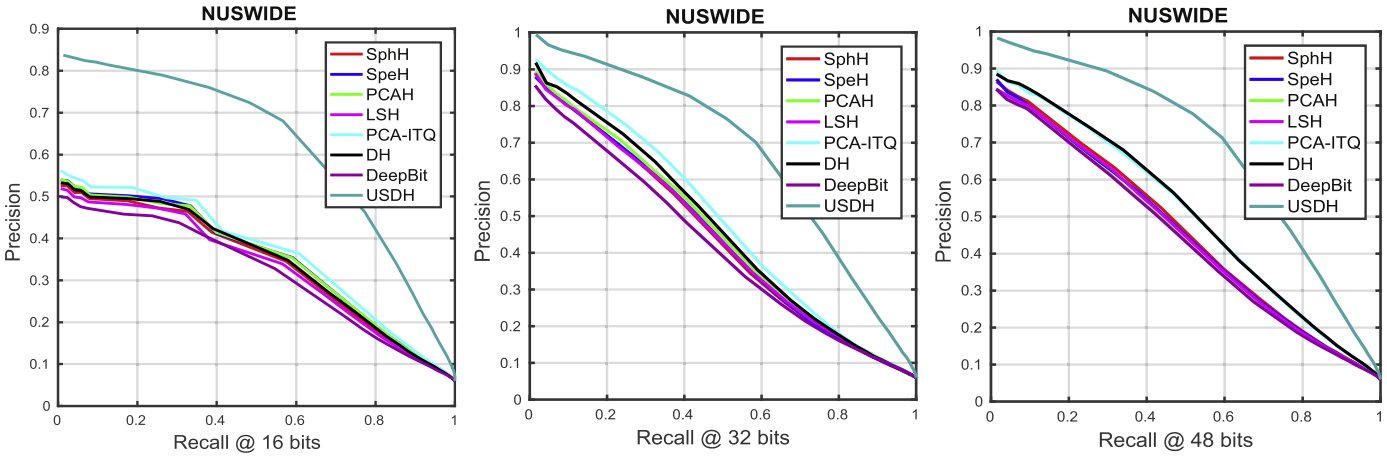


Fig. 3. Recall vs. precision curve on the NUSWIDE dataset showed the results of different unsupervised hashing methods corresponding to 16-bits, 32-bits, 64-bits.

Table 4
Comparison of image retrieval MAP of our **USDH** with respect to different values of parameters ρ.

ρ	1	1/2	1/4	1/8
MAP	39.27	39.02	39.20	39.11

Table 5
The recognition accuracy for fine grained classification on Oxford17 dataset compared with different features.

Feature	Accuracy	Training time(s)
Colour	60.9 ± 2.1%	3
Texture	70.2 ± 1.3%	4
HOG	63.7 ± 2.7%	3
HSV	58.5 ± 4.5%	4
SIFT-Boundary	59.4 ± 3.3%	4
SIFT-Internal	70.6 ± 1.6%	4
DeepBit	75.1 ± 2.5%	0.07
DeepBit+semanticloss	79.7 ± 2.2%	0.07
USDH	**81.3 ± 2.1%**	**0.07**

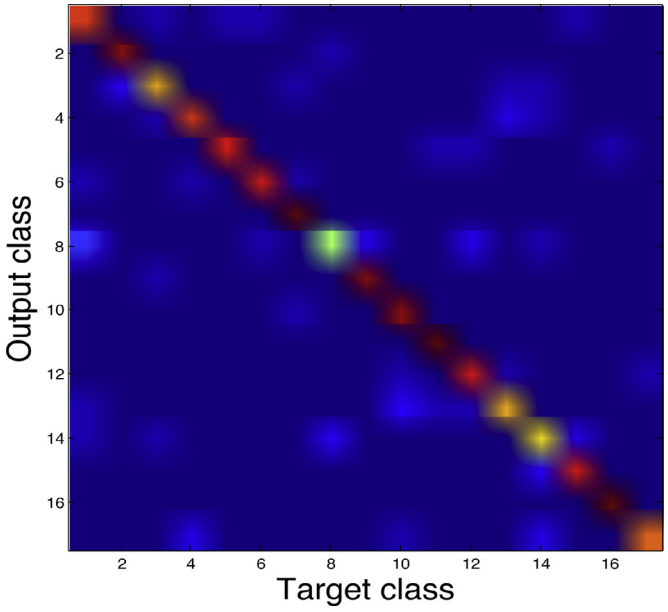


Fig. 4. Confusion matrix of Oxford 17 flower classification using the proposed **USDH**.

Robustness analysis of semantic loss: Since all these experimental results have shown the effectiveness of semantic loss, the next experiment would focus on the influences of different parameter settings. We set the parameter ρ in different value, including 1, $\frac{1}{2}$, $\frac{1}{4}$, $\frac{1}{8}$, to conduct experiments on CIFAR10 to learn the 64-bits hashing codes, where d denotes the dimension of second full-connected layer. Table 4 reveals that semantic loss is robust to the value of parameter ρ. The experimental results suggest that the hashing codes learned by **USDH** focus on the relative relationship of image features, instead of their exact similarity value.

4.5. Results on fined grained classification

Different from supervised hashing method, **USDH** learns hashing codes without label information. Thus, it has more practical potential which benefits not only image retrieval but also other computer vision tasks such as fine-grained classification. To verify it, we conduct experiments on fine-grained classification on Oxford17 dataset. Our method is compared with some low-level features, such as Color and HSV, and some mid-level features, like SIFT, HOG and so on. It is worth mentioning that DeepBit is also a deep unsupervised hashing method. However, DeepBit method [21] only requires hashing codes invariant to rotation and not considers the within-class variance among different images.

Fine-grained classification is a classic computer vision task and refers to discriminating categories of the same sub-class belong to the different superclass. This task requires image descriptors invariant to within-class variance. More specially, for flower classification, within-class variances include color difference, shape deformation and pose. We select multi-SVM as classifier and conduct experiments with different features (Fig. 4).

Table 5 shows classification accuracy of the 17 categories by using different features. Since within-class variance limits the effectiveness of the traditional color descriptor and hand-crafted shape descriptor, hashing codes learned by the deep network has a superior performance and improves 10.7% than the SIFT-Internal feature. Compared with DeepBit, our method still improves 6.2%. Additionally, our method is same fast as the DeepBit method and more faster than traditional descriptor since it has low dimension. From the above experiment, the proposed **USDH** method has been proved the effectiveness and efficiency of the fine-grained classification task.

Please cite this article as: S. Jin, H. Yao and X. Sun et al., Unsupervised semantic deep hashing, Neurocomputing, https://doi.org/10.1016/j.neucom.2019.01.020

5. Conclusions

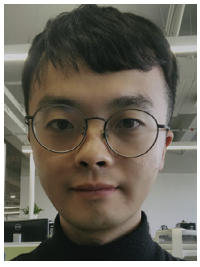
In this paper, we propose a novel unsupervised deep hashing method, named unsupervised semantic deep hashing method. The parameters of the deep neural network are fine-tuned according to four loss function: 1) semantic loss; 2) quantization loss; 3) information loss; and 4) rotation loss. Compare with previous unsupervised deep hashing methods, **USDH** requires hashing codes to preserve the relevant semantic information in the feature space. Extensive experiments on the CIFAR-10 dataset and NUSWIDE dataset demonstrate that our proposed method outperforms existing unsupervised hashing method for image retrieval task. And the experimental results on Oxford17 dataset also prove that the hashing code learned by **USDH** is also effective on other computer vision tasks, such as fine-grained classification.

Acknowledgment

This work was supported by the National Natural Science Foundation of China under Project nos. 61772158, 61702136 and U1711265.

References

[1] A. Andoni, P. Indyk, Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions, in: Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS), IEEE, 2006, pp. 459–468.
[2] G. Lin, C. Shen, Q. Shi, A. van den Hengel, D. Suter, Fast supervised hashing with decision trees for high-dimensional data, in: Proceedings of the CVPR, 2014, pp. 1963–1970.
[3] W. Liu, J. Wang, R. Ji, Y.-G. Jiang, S.-F. Chang, Supervised hashing with kernels, in: Proceedings of the CVPR, IEEE, 2012, pp. 2074–2081.
[4] Y. Weiss, A. Torralba, R. Fergus, Spectral hashing, in: Advances in Neural Information Processing Systems, 2009, pp. 1753–1760.
[5] J. Wang, S. Kumar, S.-F. Chang, Semi-supervised hashing for scalable image retrieval, in: Proceedings of the CVPR, IEEE, 2010, pp. 3424–3431.
[6] R. Salakhutdinov, G. Hinton, Semantic hashing, Int. J. Approx. Reason. 50 (7) (2009) 969–978.
[7] B. Kulis, T. Darrell, Learning to hash with binary reconstructive embeddings, in: Advances in Neural Information Processing Systems, 2009, pp. 1042–1050.
[8] M. Norouzi, D.M. Blei, Minimal loss hashing for compact binary codes, in: Proceedings of the ICML, 2011, pp. 353–360.
[9] M. Norouzi, D.J. Fleet, R.R. Salakhutdinov, Hamming distance metric learning, in: Advances in Neural Information Processing Systems, 2012, pp. 1061–1069.
[10] X. Li, G. Lin, C. Shen, A. Hengel, A. Dick, Learning hash functions using column generation, in: Proceedings of the ICML, 2013, pp. 142–150.
[11] W. Kong, W.-J. Li, Isotropic hashing, in: Advances in Neural Information Processing Systems, 2012, pp. 1646–1654.
[12] J.-P. Heo, Y. Lee, J. He, S.-F. Chang, S.-E. Yoon, Spherical hashing, in: Proceedings of the CVPR, IEEE, 2012, pp. 2957–2964.
[13] W. Liu, C. Mu, S. Kumar, S.-F. Chang, Discrete graph hashing, in: Advances in Neural Information Processing Systems, 2014, pp. 3419–3427.
[14] G. Irie, Z. Li, X.-M. Wu, S.-F. Chang, Locally linear hashing for extracting non–linear manifolds, in: Proceedings of the CVPR, 2014, pp. 2115–2122.
[15] F. Shen, W. Liu, S. Zhang, Y. Yang, H. Tao Shen, Learning binary codes for maximum inner product search, in: Proceedings of the CVPR, 2015, pp. 4148–4156.
[16] Q.-Y. Jiang, W.-J. Li, Scalable graph hashing with feature transformation., in: Proceedings of the IJCAI, 2015, pp. 2248–2254.
[17] R. Xia, Y. Pan, H. Lai, C. Liu, S. Yan, Supervised hashing for image retrieval via image representation learning, in: Proceedings of the AAAI, 2014, pp. 2156–2162.
[18] H. Lai, Y. Pan, Y. Liu, S. Yan, Simultaneous feature learning and hash coding with deep neural networks, in: Proceedings of the CVPR, 2015, pp. 3270–3278.
[19] F. Zhao, Y. Huang, L. Wang, T. Tan, Deep semantic ranking based hashing for multi-label image retrieval, in: Proceedings of the CVPR, 2015, pp. 1556–1564.
[20] H. Liu, R. Wang, S. Shan, X. Chen, Deep supervised hashing for fast image retrieval, in: Proceedings of the CVPR, 2016, pp. 2064–2072.
[21] K. Lin, J. Lu, C.-S. Chen, J. Zhou, Learning compact binary descriptors with unsupervised deep neural networks, in: Proceedings of the CVPR, 2016, pp. 1183–1192.
[22] Y. Duan, J. Lu, Z. Wang, J. Feng, J. Zhou, Learning deep binary descriptor with multi-quantization, in: Proceedings of the CVPR, 2017.
[23] G. Cheng, P. Zhou, J. Han, Learning rotation-invariant convolutional neural networks for object detection in vhr optical remote sensing images, IEEE Trans. Geosci. Remote Sens. 54 (12) (2016) 7405–7415.
[24] H. Jegou, M. Douze, C. Schmid, Product quantization for nearest neighbor search, IEEE Trans. PAMI 33 (1) (2010) 117–128.
[25] Y. Gong, S. Lazebnik, A. Gordo, F. Perronnin, Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval, IEEE Trans. PAMI 35 (12) (2013) 2916–2929.
[26] T. Ge, K. He, Q. Ke, J. Sun, Optimized product quantization for approximate nearest neighbor search, in: Proceedings of the CVPR, 2013, pp. 2946–2953.
[27] Y. Kalantidis, Y. Avrithis, Locally optimized product quantization for approximate nearest neighbor search, in: Proceedings of the CVPR, 2014, pp. 2329–2336.
[28] J. Han, D. Zhang, G. Cheng, N. Liu, D. Xu, Advanced deep-learning techniques for salient and category-specific object detection: a survey, IEEE Signal Process. Mag. 35 (1) (2018) 84–100.
[29] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.
[30] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, International Conference on Learning Representations, 2015.
[31] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the CVPR, 2015, pp. 1–9.
[32] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the CVPR, 2016, pp. 770–778.
[33] S. Xie, R. Girshick, P. Dollár, Z. Tu, K. He, Aggregated residual transformations for deep neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1492–1500.
[34] H. Zhu, M. Long, J. Wang, Y. Cao, Deep hashing network for efficient similarity retrieval., in: Proceedings of the AAAI, 2016, pp. 2415–2421.
[35] K. He, F. Wen, J. Sun, K-means hashing: An affinity-preserving quantization method for learning binary compact codes, in: Proceedings of the CVPR, 2013, pp. 2938–2945.
[36] K. Lin, H.-F. Yang, J.-H. Hsiao, C.-S. Chen, Deep learning of binary hash codes for fast image retrieval, in: Proceedings of the CVPR Workshops, 2015, pp. 27–35.



Sheng Jin is currently a Ph.D. candidate in Computer Science and Technology om Harbin Institute of Technology. He also is a research intern in Alibaba. Before that, he received his B.S. degrees in Applied Mathematic from Harbin Institute of Technology. His research interests include computer vision and machine learning.

Hongxun Yao received the B.S. and M.S. degrees in computer science from the Harbin Shipbuilding Engineering Institute, Harbin, China, in 1987 and in 1990, respectively, and received Ph.D. degree in computer science from Harbin Institute of Technology in 2003. Currently, she is a professor with School of Computer Science and Technology, Harbin Institute of Technology. Her research interests include computer vision, pattern recognition, multimedia computing, human-computer interaction technology. She has 6 books and over 200 scientific papers published, and won both the honor title of the New Century Excellent Talent in China and enjoy special government allowances expert in Heilongjiang Province, China.

Xiaoshuai Sun received the B.S. degree in Computer Science from Harbin Engineering University 07. received the M.S. and Ph.D. degree in Computer Science and Technology om Harbin Institute of Technology in 2009 and 2015 respectively. He is currently a lecturer at School of Computer Science and Technology, Harbin Institute of Technology, China. He was a post-doc research fellow (2015-2016) at School of Information Technology and Electrical Engineering, the University of Queensland, Australia. He was a Research Intern with Microsoft Research Asia (2012-2013) and also a winner of Microsoft Research Asia Fellowship (2011). He owns 3 authorized patents and has authored over 60 papers in referred journals and conferences, including IEEE Transactions on Image Processing, Pattern Recognition, ACM Multimedia, IEEE CVPR and AAAI.

Shangchen Zhou is currently a research intern in Sensetime Research. Before that, he received his Bachelor's and Master's degrees from the University of Electronic Science and Technology of China and Harbin Institute of Technology in 2015 and 2018, respectively. His research interests include computer vision and machine learning.