Pseudo Label based Unsupervised Deep Discriminative Hashing for Image Retrieval

Qinghao Hu[1,2], Jiaxiang Wu[1,2], Jian Cheng[1,2,3*], Lifang Wu[4], Hanqing Lu[1,2]
[1]Institute of Automation, Chinese Academy of Sciences , Beijing, China
[2]University of Chinese Academy of Sciences, Beijing, China
[3]Center for Excellence in Brain Science and Intelligence Technology, CAS, Beijing, China
[4]School of ICE, Beijing University of Technology, Beijing, China
{ qinghao.hu, jiaxiang.wu, jcheng, luhq }@nlpr.ia.ac.cn, lfwu@bjut.edu.cn

ABSTRACT

Hashing methods play an important role in large scale image retrieval. Traditional hashing methods use hand-crafted features to learn hash functions, which can not capture the high level semantic information. Deep hashing algorithms use deep neural networks to learn feature representation and hash functions simultaneously. Most of these algorithms exploit supervised information to train the deep network. However, supervised information is expensive to obtain. In this paper, we propose a pseudo label based unsupervised deep discriminative hashing algorithm. First, we cluster images via K-means and the cluster labels are treated as pseudo labels. Then we train a deep hashing network with pseudo labels by minimizing the classification loss and quantization loss. Experiments on two datasets demonstrate that our unsupervised deep discriminative hashing method outperforms the state-of-art unsupervised hashing methods.

CCS CONCEPTS

• **Computing methodologies → Visual content-based indexing and retrieval**; **Image representations**; **Neural networks**;

KEYWORDS

Deep Hashing; Unsupervised hashing; Pseudo Labels

1 INTRODUCTION

With the rapid development of mobile Internet, people share thousands of images every day using popular photo sharing applications. As a result, the amount of images on the web is experiencing an explosive growth. How to retrieve interesting images from a large scale image collection efficiently has attracted lots of attention.

In general, nearest neighbour search is a straightforward way to accomplish this task. However, exact nearest neighbour search usually consumes much time when retrieving from a large scale image database. To improve the search speed, various approximate nearest neighbour (ANN) search methods have been proposed. Among these algorithms, tree-based methods and hashing-based methods are two main streams. The tree-based algorithms usually partition the search space and build an efficient tree data structure which enables fast ANN search. One of the typical tree-based methods is the KD-tree algorithm [2] and it provides logarithmic search time $O\left(log\, n\right)$. Nevertheless, the key problem of tree-based methods is the curse of dimensionality i.e. the search performance degrades quickly for high dimensions. Hashing-based algorithms, on the other hand, try to map the high dimensional data to a low-dimensional Hamming space while still preserving the relative similarities with high probabilities. Since the Hamming distance between binary hash codes can be calculated extremely fast, ANN search can be conducted simply by calculating Hamming distance between binary hash codes. Roughly speaking, hashing algorithms can be categorized into two classes: data-independent and data-dependent hashing algorithms. Locality sensitive hashing (LSH) [3] is one representative algorithm of data-independent hashing methods, which doesn't require the training data. For data-dependent hashing methods, on the other hand, hashing functions are learned based on a set of training samples. These algorithms can be further divided into supervised hashing methods and unsupervised hashing methods. Supervised hashing methods utilize semantic labels or other supervised information to guide the training procedure of hashing functions. Various supervised hashing algorithms have been proposed, such as Binary Reconstruction Embedding (BRE) [10], Minimal Loss Hashing (MLH) [19] , Supervised Discrete Hashing (SDH)[21] and so on. On the other hand, unsupervised hashing methods, such as Iterative Quantization(ITQ) [4], Spectral Hashing [23], and Inductive Manifold Hashing(IMH) [22], learn the hashing functions only by utilizing the unlabeled training data. In general, supervised hashing methods usually achieve better performance than unsupervised methods, but it's quite expensive to obtain semantic labels, especially for large-scale datasets. Thus in this paper, we focus on the unsupervised hashing methods.

Recent years have witnessed the success of deep learning in various computer vision tasks such as image classification [9], object detection [20], image segmentation [17] and so on. Attracted by the powerful feature representation of deep learning, a lot of deep learning based hashing algorithms have been proposed [1, 11, 14, 24, 25]. These algorithms showed that feature learning and hash function learning can be accomplished simultanously via

[*] The corresponding author.

MM '17, October 23–27, 2017, Mountain View, CA, USA
© 2017 Association for Computing Machinery.
ACM ISBN 978-1-4503-4906-2/17/10…$15.00
https://doi.org/10.1145/3123266.3123403

deep neural networks. However, most of the deep learning based hashing methods utilized the supervised information such as semantic labels [25], pair-wise similarity labels [15, 24, 26], triplet labels [11]. Considering the unsupervised learning scenarios, Liong *et al.* [1] proposed an unsupervised deep hashing algorithm via a multi-layer neural network. Lin *et al.* [14] proposed to utilize the transfer learning ability of deep learning via using the pretrained VGG16 network. Nevertheless, the performance of these two unsupervised algorithms is not good enough.

In this paper, we propose a novel pseudo label based unsupervised discriminative hashing method. First, we extract deep features for each image using a convolutional neural network (CNN). Since images with similar features are more likely to be closely related, we generate pseudo labels via K-means clustering. Based on the pseudo labels, we train a deep hashing network with classification loss and quantization loss. The pseudo label based classification loss enables learning more discriminative hash codes. Extensive experiments are conducted on two publicly available benchmarks. The experimental results demonstrate that our proposed algorithm outperforms the state-of-art unsupervised algorithms.

2 RELATED WORK

In the past decade, a lot of unsupervised hashing methods have been proposed. Weiss *et al.* built a connection between hashing and balanced graph partitioning problem and proposed the Spectral Hashing [23] algorithm which obtained an eigenvetor solution by using a spectral relaxation. Since the cost of building a graph is $O\left(n^2\right)$, it's unaffordable for large n. To overcome this problem, Liu *et al.* [16] used a set of anchor points to approximate the data neighborhood structure. The similarity between database points and anchor points are calculated, and the adjacency matrix then can be approxiamted based on these similarities. The cost of building the anchor graph is nearly linear in n, which is far more efficient than the Spectral Hashing algorithm. Following this line, Jiang *et al.* proposed Scalable Graph Hashing (SGH) [7], which can effectively approximate the whole graph without explicitly computing the similarity graph matrix. Gong *et al.* proposed the ITQ [4] algorithm. They first projected data into a low-dimentional space via principal component analysis (PCA), then the quantization loss of mapping this low-dimentional data to the vertices of binary hypercube is minimized. An alternating method was adopted to solve the problem, which tended to find a rotation to balance the variance of different dimensions of data. Following this line, Kong *et al.* proposed the Isotropic Hashing (IsoH) [8]. Isotropic Hashing aimed to learn projection functions which can produce projected dimensions with equal variances. While many hashing algorithms use hyperplane based hashing functions, a hypersphere based hashing functions called Spherical Hashing(SpH) was proposed [5]. Compared to hyperplane-based hashing functions, spherical hashing maps more spatially coherent data points into a binary code. Shen *et al.* proposed the Inductive Manifold Hashing (IMH) algorithm [22], which built a connection between manifold learning methods and hashing algorithms.

In recent years, deep learning has attracted lots of attentions due to its powerful feature representation [12]. Xia *et al.* [24] incorporated deep convolutional neural networks with hashing algorithms. They first learned the hash codes using the pairwise similarities, followed by a stage which learned the image representation and hash function together. The learned hash codes can guide the image representation and feature hashing, but the learned image representation can not give any feedback to the learning of hash codes [11] . To solve this problem, Lai *et al.* proposed a triplet ranking loss based deep hashing algorithm. Their proposed algorithm enabled simultaneous feature learning and hash function learning. Later works such as [15], [26] used simlar framework but with different loss functions. Liu *et al.* proposed a supervised deep hashing algorithm by minimizing Hamming distance between similar images and enlarging hamming distance between dissimilar images. Besides, they imposed regularization on the network's outputs to approximate the binary hash codes. Zhu *et al.* linked the pairwise Hamming distances with pair-wise similarities by using maximum a posterior (MAP) [26]. They proposed the bimodal Laplacian prior for the network's output to regularize the output's value. Most of deep hashing algorithms exploited supervised information, few unsupervised deep hashing have been proposed. One of the difficulties for unsupervised deep hashing is designing proper loss function. Liong *et al.* proposed the DH [1] algorithm which used a multi-layer neural network to learn the hashing functions. Independent loss, quantization loss and balanced loss were introduced to guide the training procedure of deep network. Lin *et al.* [14] used the pretrained VGG16 network to utilize the transfer learning power of deep networks. In addition to the quantization loss, they also incorporated rotation invariant loss to minimize the Hamming distance between image and their rotated images. However, the performance of unsupervised deep hashing is not satisfying and more effort are required to be devoted to unsupervised deep hashing.

3 THE PROPOSED APPROACH

Given a set of images $\mathcal{X} = \{\mathbf{x_i}\}_{i=1}^{N}$, we denote the corresponding set of hash codes as $\mathcal{B} = \{\mathbf{b_i}\}_{i=1}^{N}$. Our goal is to learn a deep hashing function which maps an image $\mathbf{x_i}$ to M-bit hash codes $\mathbf{b_i}$.

In this section, we propose a novel pseudo label based unsupervised deep hashing algorithm. First, we extract the deep features of each image using pretrained CNN models. Here we utilize the VGG16 network due to its powerful feature representation ability. Afterwards, a clustering procedure is conducted on these features and the resulting cluster labels of images are regarded as pseudo labels for the later training stage. Given pseudo labels, a convolutional neural network is trained for simultaneous feature representation and hash encoding. The overall framework of our proposed algorithm is depicted in Fig.1. The following subsections shall present technical details of our proposed algorithm .

3.1 Pseudo Label Generation

Using supervised information in deep hashing has two-fold benefits. One is that supervised information can improve the discriminative power of hash codes. The other one is that the loss function can be easily designed with supervised information. However, in

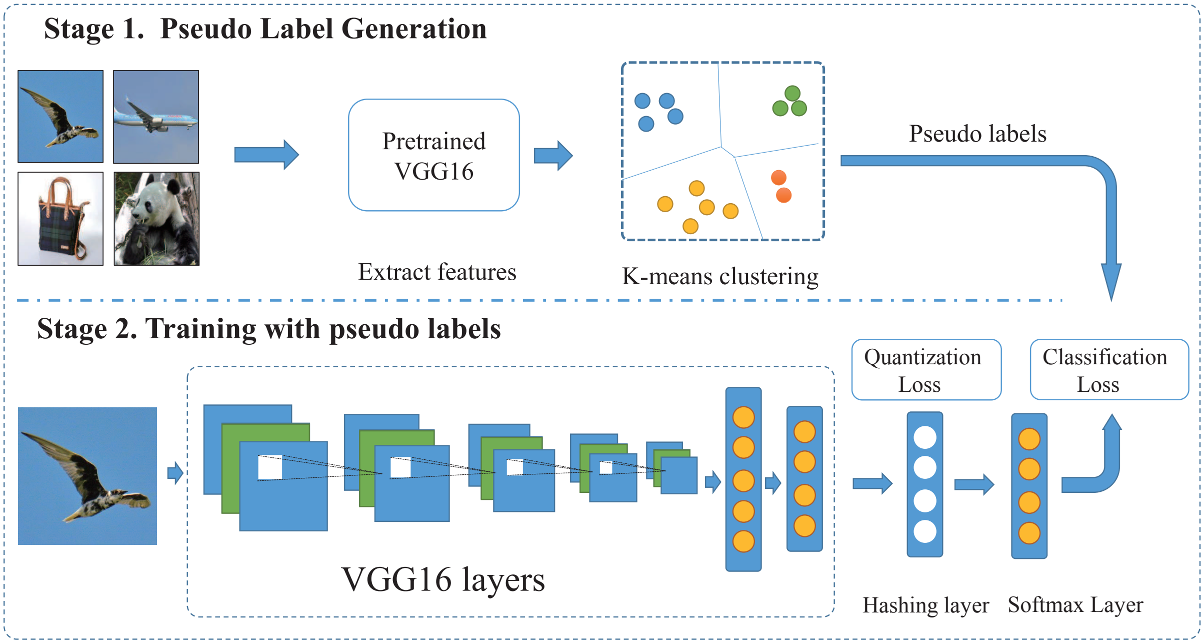

Figure 1: The framework of our proposed unsupervised deep discriminative hashing algorithm. In stage 1, a pretrained VGG16 network is used to extract features from each image. Then we use the K-means algorithm to cluster the images. The cluster label of an image is treated as its pseudo label. In stage 2, we use a deep hashing neural network to learn features and hash functions simultaneously. A hash encoding layer is added on the top of VGG16 network to generate hash codes. Quantization loss and classification loss are used to guide the training procedure of network. Best viewed in color.

the scenario of unsupervised hashing algorithms, the labels of images or other supervised information are not available. To solve this, our goal is to generate pseudo labels which have similar properties with semantic category labels. The generated pseudo labels are supposed to help improve the discriminative power of hash codes. Since *deep* features extracted by CNN can capture high level semantic information, we assume that, if two images are semantically similar, their deep features are closely related. Based on this assumption, we propose to generate pseudo labels via K-means clustering. First, a pretrained VGG16 network is adopted to extract features from each image. Specifically, the activations of fc7 layer of VGG16 network are treated as the 4096-dimensional feature for each image. The pretrained model enables us to utilize the transfer learning power of deep networks. Then we cluster the images via K-means algorithm. By assuming that images in the same cluster are likely to have the same label, we treat the cluster label (cluster id) as the pseudo label of an image. For the image set $\mathcal{X}$, the corresponding pseudo labels are denoted as $\mathcal{Y} = \{\mathrm{y_i}\}_{i=1}^{N}$. Fig. 2 shows some images from clusters after K-means clustering. For each row of Fig. 2, we randomly pick 5 images from one cluster. The left column of each row shows the top three most frequent concepts of images from the cluster.

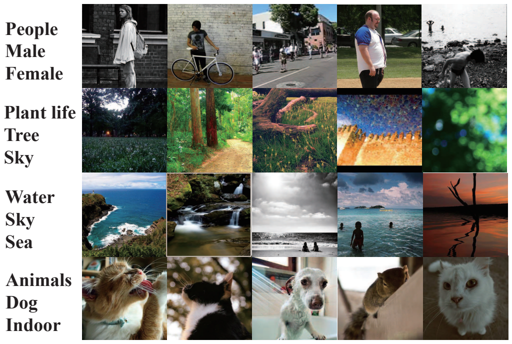

Figure 2: The clustering result of K-means. For each cluster, the top three most concepts of images are shown at the left column in each row. Best viewed in color.

3.2 Training Deep hashing networks with Pseudo Labels

Network architecture Following [14], we adopt the VGG16 network as our base network architecture. The $fc8$ layer is replaced by a hash encoding layer which is designed to map deep features to M-bit binary codes. The hash encoding layer is a fully-connected layer which consists of M output nodes. Here we denote the hash encoding layer's activations as $H(\mathbf{x};\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is the parameters of all the precedent layers. Thus the hash codes is given by

$$b = sgn(H(\mathbf{x};\boldsymbol{\theta})) \tag{1}$$

where sgn denotes the sign function, and $sgn(x)$=1 for $x > 0$ and -1 otherwise.

Loss function To improve the discriminative power of hash codes, we use a softmax classification loss to guide the training of deep hashing model. Suppose that we cluster images into K clusters in the pseudo label generation stage, a softmax layer with K output nodes is added on the top of hash encoding layer. Given images $\mathcal{X}$ and pseudo labels $\mathcal{Y}$, the problem can be formulated as :

$$\begin{aligned} \min\ L(\mathcal{X},\mathcal{Y}) &= L_c(\mathcal{B},\mathcal{Y}) \\ &= L_c(sgn(H(\mathcal{X};\boldsymbol{\theta})),\mathcal{Y}) \end{aligned} \tag{2}$$

where $L_c(\cdot,\cdot)$ denotes the softmax classification loss. And $L_c(\cdot,\cdot)$ is formulated as:

$$L_c(\mathcal{B},\mathcal{Y}) = -\sum_{i=1}^{N}\sum_{j=1}^{K} \mathbb{1}(\mathbf{y}_i = j)\, log\left(\frac{e^{\mathbf{w}_j^T\mathbf{b}_i+v_j}}{\sum_{j=1}^{K} e^{\mathbf{w}_j^T\mathbf{b}_i+v_j}}\right) \tag{3}$$

where $\mathbf{w}_j \in \mathcal{R}^M$ and $v_j \in \mathcal{R}$ are the weights and bias of softmax layer respectively. The motivation is that the learned hash codes should be capable of predicting the pseudo label of each image.

Since the sgn function is not differentiable, the problem defined in (2) cannot be optimized via back-propagation (BP) directly. Following [13], we reformulate the problem (2) as:

$$\begin{aligned} &\min\ L_c(H(\mathcal{X};\boldsymbol{\theta}),\mathcal{Y}) \\ &s.t.\ \mathbf{b}_i = H(\mathbf{x}_i;\boldsymbol{\theta}) \quad \forall i = 1,2,\cdots N \\ &\qquad \mathbf{b}_i \in \{-1,1\}^M \quad \forall i = 1,2,\cdots N \end{aligned} \tag{4}$$

Due to the binary constraint on $\mathbf{b_i}$ and equality constraint, the problem (4) is equivalent to the problem (2). By adding a penalty term to the loss, we remove the equality constraint:

$$\begin{aligned} &\min\ L_c(H(\mathcal{X};\boldsymbol{\theta}),\mathcal{Y}) + \lambda\sum_{i=1}^{N} \|\mathbf{b}_i - H(\mathbf{x}_i;\boldsymbol{\theta})\|_2^2 \\ &s.t.\ \mathbf{b}_i \in \{-1,1\}^M \quad \forall i = 1,2,\cdots N \end{aligned} \tag{5}$$

The first term in Eq.(5) is the classification loss defined by pseudo labels, which improves the discriminative power of hash codes. The second term in Eq. (5) can be regarded as the quantization loss of hash encoding layer. It helps to reduce the quantization error during binarization.

Eq.(5) can be easily solved by an alternating optimization method, i.e. update binary codes with network parameters fixed, and vice versa.

Update $\mathcal{B}$ Given parameters $\boldsymbol{\theta}$ is fixed, $H(\mathcal{X};\boldsymbol{\theta})$ can be updated using forward propagation. Then the following equation offer the optimal solution of $\mathcal{B}$:

$$\mathbf{b}_i = sgn(H(\mathbf{x}_i;\boldsymbol{\theta})) \quad \forall i = 1,2,\cdots N \tag{6}$$

Update network parameters Given $\mathcal{B}$ is fixed, the derivatives of loss $L(\mathcal{X},\mathcal{Y})$ wrt. $H(\mathbf{x}_i;\boldsymbol{\theta})$ can be calculated as follows:

$$\frac{\partial L(\mathcal{X},\mathcal{Y})}{\partial H(\mathbf{x}_i;\boldsymbol{\theta})} = \frac{\partial L_c(H(\mathbf{x}_i;\boldsymbol{\theta}),\mathbf{y}_i)}{\partial H(\mathbf{x}_i;\boldsymbol{\theta})} + 2\lambda(H(\mathbf{x}_i;\boldsymbol{\theta}) - \mathbf{b}_i) \tag{7}$$

The first term in (7) can be obtained by back-propagation. Thus the whole network can be trained directly by standard back-propagation. We summarize the overall training algorithm in Algorithm 1.

Algorithm 1: Pseudo label based unsupervised deep discriminative hashing

```
Input: Training set $\mathcal{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots \mathbf{x}_n]$ and pseudo labels
       $\mathcal{Y} = [\mathbf{y}_1, \mathbf{y}_2, \cdots \mathbf{y}_n]$
Output: $\boldsymbol{\theta}$
Initialize deep hashing network with pretrained VGG16
  model
while $iter \leq max\_iters$ do
    Sampling a mini-batch $x$ from data $X$
    Forward propagation
    Update $H(\mathbf{x};\boldsymbol{\theta})$
    Update $\mathbf{b}$ using Eq.(6);
    Calculate the loss $L$ using the Eq. (5)
    Calculate gradients using Eq.(7)
    Update $\boldsymbol{\theta}$ via backward propagation
end
return $\boldsymbol{\theta}$;
```

4 EXPERIMENTS

4.1 Datasets and Setting

To evaluate our proposed method, we conduct extensive experiments on two public benchmark datasets, **CIFAR10** and **Flickr**

- **CIFAR10**[1] dataset consists of 60,000 colour images in 10 classes. Each class contains 6000 images in size 32× 32.
- **Flickr**[2] dataset contains 25,000 images collected from Flickr. Each image is labelled with one of the 38 concepts.

For CIFAR10 dataset, we randomly select 100 images per class as the test set and the rest of images are treated as the training set. For Flickr dataset, we randomly select 1000 images as test set and the rest of dataset images are taken as training set, following [26]. For CIFAR10 and Flickr, we resize all images to size 256×256 in order to use the VGG16 network.

We compare our proposed method with eight unsupervised hashing methods including LSH [3], Spectral Hashing(SH) [23], ITQ [4], AGH [16], Spherical Hashing(SpH) [5], Isotropic Hashing(IsoH) [8], IMH [22] and Scalable Graph Hashing (SGH) [7]. Besides, we also compare our method with two unsupervised deep hashing method i.e. DH [1] and DeepBit [14]. For deep learning based methods, raw image pixels are directly used as input. For traditional unsupervised hashing methods, 512-dimensional GIST features are extracted from each image, following [1]. We implement our proposed method using the open source Caffe [6] framework. As mentioned above, we use the VGG16 network and initialize the $conv1-$

[1] https://www.cs.toronto.edu/ kriz/cifar.html

[2] http://press.liacs.nl/mirflickr/

Table 1: mAP of different hashing algorithms on CIFAR10 and Flickr benchmark.

Method	CIFAR10 (mAP)			Flickr (mAP)		
	16-bit	32-bit	64-bit	16-bit	32-bit	64-bit
LSH [3]	0.1252	0.1317	0.1451	0.5532	0.5593	0.5637
SH [23]	0.1265	0.1252	0.1254	0.5524	0.5541	0.5531
ITQ [4]	0.1599	0.1680	0.1750	0.5623	0.5636	0.5642
AGH [16]	0.1495	0.1517	0.1431	0.5619	0.5580	0.5537
SpH [5]	0.1424	0.1538	0.1624	0.5619	0.5604	0.5635
IsoH [8]	0.1550	0.1584	0.1639	0.5638	0.5603	0.5626
IMH [22]	0.1752	0.1930	0.1974	0.5684	0.5722	0.5741
SGH [7]	0.1579	0.1660	0.1743	0.5647	0.5649	0.5662
DH [1]	0.1617	0.1662	0.1669	–	–	–
DeepBit [14]	0.1943	0.2486	0.2773	0.5789	0.5810	0.5892
Our Method	**0.5165**	**0.5720**	**0.5956**	**0.8266**	**0.8411**	**0.853**

*conv*5 and $fc6 - fc7$ layer with pretrained model. We use the mini-batch stochastic gradient descent (SGD) with 0.9 momentum . And the weight decay parameters are fixed as 0.0005.

4.2 Results on CIFAR10 and Flickr Benchmark

We compare our proposed method with traditional unsupervised hashing methods on CIFAR10 and Flickr benchmark. Following [14], we evaluated our methods based on the mean Average Precision(mAP) of top 1,000 retrieved images. Table. 1 shows the mAP result of different hashing algorithms with respect to different hash bits. From Table. 1, it's clear that our method outperforms other state-of-art algorithms by a large margin consistently on both CIFAR10 and Flickr benchmark. In terms of deep-learning based hashing algorithm, DeepBit [14] and our proposed algorithm demonstrate superior performance over non-deep unsupervised hashing methods, showing the powerful feature representation of VGG16 network. However, DH [1] algorithm surprisingly fails to outperform ITQ sometimes. It may be caused by the shallow neural network used by DH [1]. DeepBit [14] and our proposed algorithm both use VGG16 network, but we use different loss functions with DeepBit. Our proposed algorithm benefits from pseudo label based classification loss which can help generate more discriminative hash codes.

As mentioned above, our proposed algorithm enables simultaneous feature learning and hash encoding. To show the effectiveness of our algorithm, we also compare our method to non-deep unsupervised hashing algorithm using pre-trained deep features. In detail, for these non-deep hashing algorithms, we represent each image as a 4096-dimensional feature vector which is extracted from $fc7$ layer by using a pre-trained VGG16 network. The mAP result is reported in Table 2. The performance of traditional unsupervised hashing methods have improved by a large margin on both CIFAR10 and Flickr benchmark by using deep features, which shows the powerful feature representation of deep neural networks. We also noticed that, by using deep features, ITQ or IMH is superior than DeepBit and DH on CIFAR10 benchmark. On Flikcr benchmark, most of traditional hashing algorithms using deep features are superior than DeepBit and DH. However, even using the deep features, these hashing methods are still inferior to our proposed method, which demonstrates that our proposed method generates more discriminative hash codes.

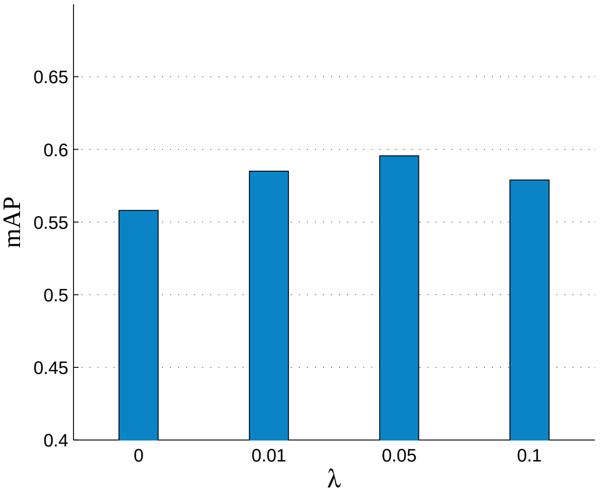

Figure 3: mAP with respect to different hyper-parameter λ on CIFAR10 benchmark.

4.3 Sensitivity to the hyper-parameters

In this subsection, we explored the influence of different hyper-parameters to the proposed algorithm. In the procedure of pseudo labels generation, we used the K-means algorithm to cluster the images. We conducted a series of experiments on CIFAR10 and Flickr benchmark by varying the value of K and leaving other setting unchanged. Figure 4 reports the mAP of varying K value. While the value of K increases from 10 to 100, the mAP first gradually increases and then degrades. One reason for this phenomenon is that the purity of a cluster will increase with the value of K increasing. The higher the purity is, the higher the probability of that images in the same cluster belong to the same class is. Thus the pseudo labels will be more convincing. However, images in the same category may be divided into different clusters when the value of K gets vary large, which makes the mAP degrade. Anyway, the mAP is not sensitive to K in a wide range of K. And the performance is

Table 2: mAP of different hashing algorithms using Deep features on CIFAR10 and Flickr benchmark.

Method	CIFAR10 (mAP)			Flickr (mAP)		
	16	32	64	16	32	64
LSH+CNN [3]	0.1640	0.1903	0.2390	0.5834	0.5729	0.6115
SH+CNN [23]	0.2210	0.1975	0.1805	0.6013	0.5899	0.5807
ITQ+CNN [4]	0.3308	0.3416	0.3589	0.6816	0.6784	0.6785
AGH+CNN [16]	0.3057	0.2621	0.2272	0.6745	0.6500	0.6459
SpH+CNN [5]	0.2148	0.2517	0.2816	0.6816	0.6784	0.6785
IsoH+CNN [8]	0.2647	0.3019	0.3183	0.6195	0.6364	0.6490
IMH+CNN [22]	0.4272	0.4126	0.4463	0.7066	0.7161	0.7241
SGH+CNN [7]	0.2642	0.2866	0.3228	0.6242	0.6319	0.6391
DH [1]	0.1617	0.1662	0.1669	–	–	–
DeepBit [14]	0.1943	0.2486	0.2773	0.5789	0.5810	0.5892
Our Method	**0.5165**	**0.5720**	**0.5956**	**0.8266**	**0.8411**	**0.853**

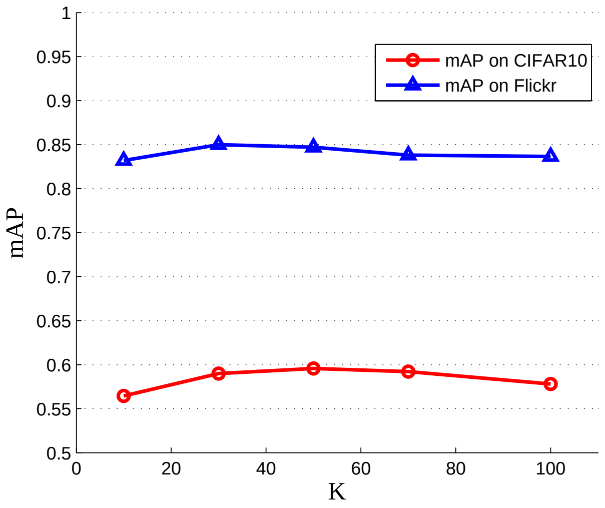

Figure 4: mAP with different hyper-parameter K on CIFAR10 and Flickr benchmark.

still superior than other state-of-art algorithms even the optimal K is not chosen.

We also explored the effect of hyper-parameter λ. Fig. 3 reports the mAP of our proposed algorithm on CIFAR10 with varying λ. λ equals to zero means that we drop the quantization loss. The mAp drops about 4 percentage when λ equals to zero, showing the necessity of quantization loss.

To show that our algorithm enables features learning and hash coding together, we conduct another series of experiments with fixed VGG16 network. In detail, we fixed the parameters of $conv1-fc7$ layers unchanged during training procedure, which means the feature learning is disabled. Fig.5 shows the mAP with or without fixed VGG16 on CIFAR10 benchmark. It can be figured out that the mAP of algorithm with fixed VGG16 is about 5 percentage lower.

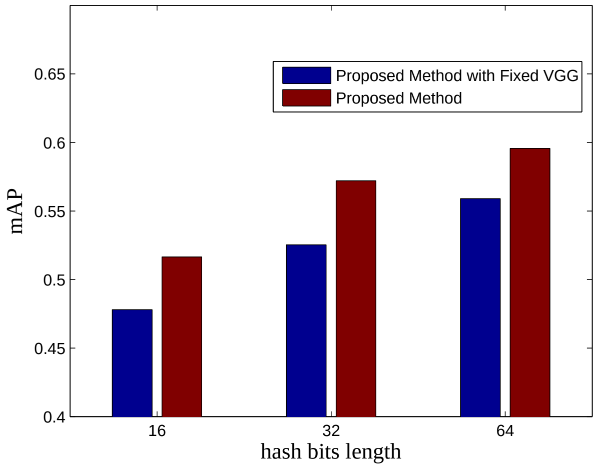

Figure 5: mAP with or without fixed VGG16 on CIFAR10 benchmark.

4.4 Experiments on Classification

Following the setting in [14], we trained a multi-class SVM classifier on Oxford 17 Category Flower Dataset [18] using the hash codes generated by our method. The categorization accuracy of our method is 88.2% which outperforms 75.2% in DeepBit [14] by a large margin.

5 CONCLUSION

Hashing algorithms have been widely used in large scale image retrieval task. Traditional hashing methods encode hand-crafted features into hash codes, resulting in unsatisfying performance. Recent years has witnessed the success of deep learning. Deep learning based hashing algorithms have attracted a lot of attentions. However, most of them used supervised information, which is usually hard to obtain. In this paper, we proposed a pseudo label based unsupervised deep discriminative hashing algorithm. We first generate pseudo labels via K-means clustering, then train a deep convolution neural network to generate hash codes. Classification loss and quantization loss is introduced to optimize the network. Extensive experiments on CIFAR10 and Flickr demonstrate that our algorithm outperforms the state-of-art unsupervised hashing algorithms.

ACKNOWLEDGEMENTS

This work was supported in part by National Natural Science Foundation of China (No.61332016), the Scientific Research Key Program of Beijing Municipal Commission of Education (KZ201610005012) and Jiangsu Key Laboratory of Big Data Analysis Technology.

REFERENCES

[1] Venice Erin Liong, Jiwen Lu, Gang Wang, Pierre Moulin, and Jie Zhou. 2015. Deep hashing for compact binary codes learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2475–2483.

[2] Jerome H Friedman, Jon Louis Bentley, and Raphael Ari Finkel. 1977. An algorithm for finding best matches in logarithmic expected time. *ACM Transactions on Mathematical Software (TOMS)* 3, 3 (1977), 209–226.

[3] Aristides Gionis, Piotr Indyk, Rajeev Motwani, et al. 1999. Similarity search in high dimensions via hashing. In *VLDB*, Vol. 99. 518–529.

[4] Yunchao Gong, Svetlana Lazebnik, Albert Gordo, and Florent Perronnin. 2013. Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35, 12 (2013), 2916–2929.

[5] Jae-Pil Heo, Youngwoon Lee, Junfeng He, Shih-Fu Chang, and Sung-Eui Yoon. 2012. Spherical hashing. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2957–2964.

[6] Yangqing Jia, Evan Shelhamer, Jeff Donahue, Sergey Karayev, Jonathan Long, Ross Girshick, Sergio Guadarrama, and Trevor Darrell. 2014. Caffe: Convolutional Architecture for Fast Feature Embedding. *arXiv preprint arXiv:1408.5093* (2014).

[7] Qing-Yuan Jiang and Wu-Jun Li. 2015. Scalable Graph Hashing with Feature Transformation.. In *IJCAI*. 2248–2254.

[8] Weihao Kong and Wu-Jun Li. 2012. Isotropic hashing. In *Advances in Neural Information Processing Systems*. 1646–1654.

[9] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*. 1097–1105.

[10] Brian Kulis and Trevor Darrell. 2009. Learning to hash with binary reconstructive embeddings. In *Advances in neural information processing systems*. 1042–1050.

[11] Hanjiang Lai, Yan Pan, Ye Liu, and Shuicheng Yan. 2015. Simultaneous feature learning and hash coding with deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3270–3278.

[12] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature* 521, 7553 (2015), 436–444.

[13] Wu-Jun Li, Sheng Wang, and Wang-Cheng Kang. 2015. Feature learning based deep supervised hashing with pairwise labels. *arXiv preprint arXiv:1511.03855* (2015).

[14] Kevin Lin, Jiwen Lu, Chu-Song Chen, and Jie Zhou. 2016. Learning compact binary descriptors with unsupervised deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1183–1192.

[15] Haomiao Liu, Ruiping Wang, Shiguang Shan, and Xilin Chen. 2016. Deep supervised hashing for fast image retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2064–2072.

[16] Wei Liu, Jun Wang, Sanjiv Kumar, and Shih-Fu Chang. 2011. Hashing with graphs. In *Proceedings of the 28th international conference on machine learning (ICML-11)*. 1–8.

[17] Jonathan Long, Evan Shelhamer, and Trevor Darrell. 2015. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3431–3440.

[18] M-E Nilsback and Andrew Zisserman. 2006. A visual vocabulary for flower classification. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, Vol. 2. IEEE, 1447–1454.

[19] Mohammad Norouzi and David M Blei. 2011. Minimal loss hashing for compact binary codes. In *Proceedings of the 28th international conference on machine learning (ICML-11)*. 353–360.

[20] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*. 91–99.

[21] Fumin Shen, Chunhua Shen, Wei Liu, and Heng Tao Shen. 2015. Supervised discrete hashing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 37–45.

[22] Fumin Shen, Chunhua Shen, Qinfeng Shi, Anton Van Den Hengel, and Zhenmin Tang. 2013. Inductive hashing on manifolds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1562–1569.

[23] Yair Weiss, Antonio Torralba, and Rob Fergus. 2009. Spectral hashing. In *Advances in neural information processing systems*. 1753–1760.

[24] Rongkai Xia, Yan Pan, Hanjiang Lai, Cong Liu, and Shuicheng Yan. 2014. Supervised Hashing for Image Retrieval via Image Representation Learning.. In *AAAI*, Vol. 1. 2.

[25] Ziming Zhang, Yuting Chen, and Venkatesh Saligrama. 2016. Efficient training of very deep neural networks for supervised hashing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1487–1495.

[26] Han Zhu, Mingsheng Long, Jianmin Wang, and Yue Cao. 2016. Deep Hashing Network for Efficient Similarity Retrieval.. In *AAAI*. 2415–2421.