A NOVEL DEEP HASHING METHOD WITH TOP SIMILARITY FOR IMAGE RETRIEVAL

Qibing Qin[1], Zhiqiang Wei[1,2,], Lei Huang[1,2], Jie Nie[1], Xiaopeng Ji[1]*
[1]Ocean University of China, Qingdao 266000, China
[2] Qingdao National Laboratory for Marine Science and Technology, Qingdao 266000, China
15114585538@163.com , {weizhiqiang, huangl, niejie, jxiaopeng}@ouc.edu.cn

ABSTRACT

Due to the advantages of retrieval speed and storage space, deep hashing methods have become a research hotspot in the field of large-scale image retrieval. Most of existing deep hashing methods pay close attention to similarity between images without images at the top of the ranking list similar to query targets. In the paper, a novel deep hashing model is proposed to preserve top images similar to the query images and optimize the quality of hash codes for image retrieval. Specifically, the optimized AlexNet is utilized to extract discriminative image representations and learn hashing functions simultaneously. The loss function based on acceleration strategy is designed to ensure similarity between returned images at the top of the ranking list and query images. In addition, we implement the model training in a batch-process fashion to low the image storage. Moreover, our extensive experiments on standard benchmarks demonstrate that our method outperforms several state-of-the-art deep hashing methods.

Index Terms— Deep Hashing, Image Retrieval, Similarity, Ranking List, AlexNet.

1. INTRODUCTION

The Content-Based Image Retrieval (CBIR) has been a very active research domain in computer vision for large image database [1,2]. Due to efficient retrieval speed and low storage cost, the hashing that belongs to one of the nearest neighbor search methods has been widely used in the field of large-scale image retrieval [3]. Existing hashing methods can be divided into data-independent and data-dependent. Unlike data-independent approaches, data-dependent methods try to learn hash function from training data, which is called learning-based hashing approaches [4].

According to whether supervision information of training samples is used, learning-based hashing methods can be divided into two categories in detail: unsupervised learning and supervised learning. Based on supervised information, supervised hash learning methods are further divided into three categories. (1)Point-wise methods, which use instance semantic labels to learn hash functions, including Canonical Correlation Analysis Iterative Quantization (CCA-ITQ) [5], isotropic hashing (IsoHash) [6], Supervised Semantics Preserving Deep Hashing (SSDH) [7],etc. (2)Pair-wise methods, which utilize the pair-wise label information between images, include Supervised Deep Hashing (SDH) [8], Convolutional Neural Networks Hashing (CNNH) [9] Deep Pairwise-Supervised Hashing (DPSH) [10], Deep Hashing Network (DHN) [11], etc. (3) Triplet-wise methods, which use the forms of triples for training model, contain deep regularized similarity comparison Hashing (DRSCH) [12], deep semantic ranking hashing (DSRH) [13], etc.

Although supervised hashing learning have achieved good performance in large-scale image retrieval tasks, most of the existing hashing methods only consider the similarity between images and the location information of in the retrieval ranking list is rarely discussed. Moreover, users always pay too much attention to the top results in query list, and do not get used to caring about those at the bottom of the ranking list in the content-based retrieval. So, it is critical to preserve similarity between images ranked at the top of the ranking list and query images in the content-based retrieval [14,15]. Meanwhile, the models mentioned above (e.g., classical deep convolutional network AlexNet) have semantic gaps mapping the high-dimensional feature vectors to low-dimensional hash codes when generating hash codes based on target images.

In this paper, a novel deep hashing with preserving top images similarity based on acceleration strategy is proposed in this paper so as to generate compact binary codes and solve these problems we have discussed above. The overall framework of our model is shown in Fig.1. Firstly, we are inspired by [16] and optimize the internal structure of the classical deep convolutional network AlexNet to improve the feature representation ability of the network and produce high quality hash codes compared with other networks. Optimized AlexNet is used to extract discriminative image representations and learn hash functions simultaneously. Secondly, the loss function based on acceleration strategy is designed to preserve similarity between images ranked at the top of the ranking list and query images in large-scale image retrieval. Thirdly, we train our deep hashing model in a batch-process fashion to cope with the large amount of stored images. Furthermore, experimental results show our framework exceeds several hashing methods.

2. PROPOSED APPROACHE

* Corresponding author: Zhiqiang Wei

2.1. Deep Architecture

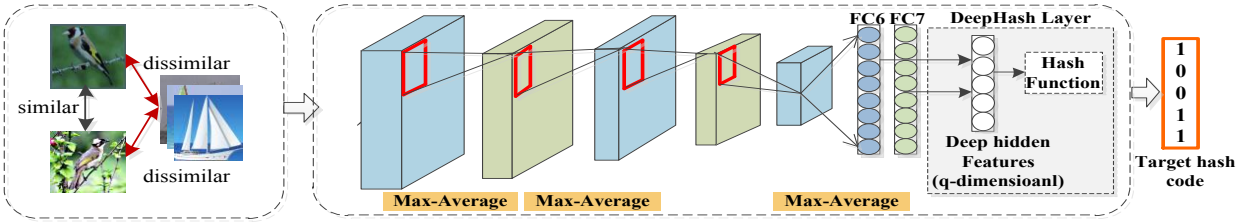

Fig. 1. The overall framework of our model

The classical deep convolutional neural network Alexnet consists of five convolutional layers, three pooling layers, and three fully connected layers [17]. Although we can increase the depth of network to reduce semantic gaps, it will also add complexity with training model. Therefore, we optimize pooling layers and fully connected layers of Alexnet to get discriminative middle-level feature descriptors. (1)The max-average pooling strategy is adopted to strengthen the local feature presentation ability. (2) Maxout activation [18] is used instead of Sigmoid or ReLu to fit the distribution of global features in fully connected layers. (3)The last fully connected layer is replaced by a novel hidden hashing layer to convert extracted image features into compact hash codes.

2.1.1. Max-Average Pooling

It is conductive to represent image feature that the key role for pooling is to get core features and discard the irrelevant features. The features after convolution often contain some location information (e.g., relative position, absolute position, etc.). When the distributions of image features are smooth and flat, max-pooling often discards relevant local spatial information to reduce image feature extraction and representation. In this case, the average pooling is often used without losing local relevant information [16]. The pooling operation in the paper is defined as follows.

$$f(v) = \partial_1 \max v_i + \partial_2 \frac{1}{p}\sum_{i=1}^{p} v_i \qquad (1)$$

Where $\partial_1 + \partial_2 = 1$, $\max v_i$ represents the maximum pooling operation, $\frac{1}{p}\sum_{i=1}^{p} v_i$ represents the average pooling operation.

In phase of training model, we hope our model can be very expressive for any input image. Therefore, we set $\partial_1 = \partial_2 = 0.5$ to enhance the robustness of model. In phase of testing model, we set $\partial_1 = \partial_2 = 0.5$ to measure the saliency feature and the average feature as same importance with high-quality images as input; we set $\partial_1 = 0.3$ and $\partial_2 = 0.7$ to strengthen the importance of the average feature to reduce the impact of image noise on feature extraction and representation with low-quality images as input. In our experiments, CIFAR-10 is with low quality pixels while NUS-WIDE is with high quality pixels.

2.1.2. Maxout Activation for Global Features

Recent researches have shown that Maxout could fit any dimensional function compared with traditional activation functions [18]. For the given input data, Maxout is defined as follows.

$$h_i(x) = \max_{j\in[1,k]} z_{ij} \qquad (2)$$

Where $z_{ij} = x^T W \ldots_{ij} + b_{ij}, W \in \mathbb{R}^{d\times m\times k}$ and $b \in \mathbb{R}^{m\times k}$ are parameters which needs to be learned.

In our proposed model, Maxout with dropout operation, which could achieve excellent performance, is used in fully connected layer to fit the distribution of global features. Meanwhile, Maxout activation will increase network parameters. During our experiments in this paper, we reduce the number of nodes of FC6 and FC7 to 2048, as well reduce the dropout rate from 0.5 to 0.3 to achieve the retrieval efficiency.

2.1.3. Deep Hash Functions

Inspired by [19], we add a bypassing connection between the FC6 layer and the last hash layer to reduce the loss of information. Recent researches have shown that the features from FC7 are dependent on labels too much and have strong feature expression invariance, which is not conducive to capture subtle semantic features [13]. Therefore, hash layer in our model is connected with the FC6 and FC7 layers simultaneously to improve image feature presentation. The hash function is defined as follows.

$$h(x;\mathbf{w}) = sign(\mathbf{w}^T[f_{fc6}(x); f_{fc7}(x)]) \qquad (3)$$

Where $\mathbf{w}$ represents the weights of hash layer, $f_{fc6}(\bullet)$ and $f_{fc7}(\bullet)$ respectively denote feature vectors from the outputs of FC6 layer and FC7 layer. For the sake of concision, bias terms and parameters of $f_{fc6}(\bullet)$ and $f_{fc7}(\bullet)$ are omitted. We could compute $h(x,\mathbf{W}) = [h_1(x;\mathbf{w_1}), h_2(x;\mathbf{w}_2), \ldots, h_K(x;\mathbf{w}_K)]$ to get K-bit binary codes.

2.2. Loss Function

Most of the deep hashing methods pay close attention to similarity between images without top similarity preserving. We introduce the loss function which makes images ranked in the top similar to the query images in the section.

During training stage, we feed a group of images to our model each iteration, including the query image I, similar image I^+ and dissimilar images $\{I_k^-\}_{k=1}^n$, where n is the number of dissimilar images. Then we can obtain the

corresponding hash codes respectively, denoted as $b(I)$, $b(I^+)$ and $\{b(I_k^-)\}_{k=1}^n$ when original images as the inputs. Intuitively, the Hamming distance between $b(I)$ and $b(I^+)$ can get closer than that between $b(I)$ and $b(I_k^-)$ for any $k \in \{1,n\}$ [15]. Therefore, the "rank" of the similar image I^+ with respect to query image I is defined as follows.

$$R(I,I^+) = \sum_{k=1}^{n} f\{(\|b(I)-b(I^+)\|_{\mathrm{H}} - \|b(I)-b(I_k^-)\|_{\mathrm{H}}) > 0\} \quad (4)$$

Where $\|\bullet\|_{\mathbf{H}}$ denotes the Hamming space distance, f is the Boolean function. $R(I,I^+)$ represents the number of dissimilar images. Due to the images at the top of the ranking list similar to the query images, we defined the loss function as follows.

$$Loss(R(I,I^+)) = -\frac{\psi^2}{\psi + R(I,I^+)} \quad (5)$$

Where parameter ψ controls the reduction rate of first derivative. Eq.(5) means the similarity in top of ranking list is more important than that in the bottom. According to Eq.(5), the object function is defined as follows.

$$L = \frac{1}{Z}\sum_{I}\sum_{I^+} -\frac{\psi^2}{\psi + R(I,I^+)} \quad (6)$$

Where Z is the number of input image groups. We need to relax it as follows due to the non-differentiability of Eq.(3).

$$f_h(I) = sigmoid(w_h^T g(I)) \quad (7)$$

Where $g(I)$ denotes feature vectors from the outputs of FC6 layer and FC7 layer. Then we can obtain a q-bit binary code by simple quantization $b(I) = sign(f_h(I) - 0.5)$. Meanwhile, it is relaxed as follows by *Sigmoid* and L_1-*norm* due to the non-differentiability of Eq.(4).

$$\hat{R}(I,I^+) = \sum_{k=1}^{n} sigmoid(\|f_h(I)-f_h(I^+)\|_1 - \|f_h(I)-f_h(I_k^-)\|_1) \quad (8)$$

According to Eq.(6) and Eq.(8), the overall objective function can be written as follows. Where **W** is the weights of our network. The third term of Eq.(9) is used to make each bit averaged over the training data to maximize the entropy of binary codes [20].

$$\begin{aligned} L = & \frac{1}{Z}\sum_{I}\sum_{I^+} -\frac{\psi^2}{\psi + \hat{R}(I,I^+)} \\ & + \frac{\partial}{2}\sum_i \|b(I_i) - f_h(I_i)\|_2 \\ & + \frac{\beta}{2}\sum_i \left\| mean_i(f_h(I_i) - 0.5) \right\|_2 + \frac{\lambda}{2}\|\mathbf{W}\|_2 \end{aligned} \quad (9)$$

3. LEARNING ALGORITHM

3.1. Joint Optimization based on Acceleration

We use Stochastic Gradient Descent to optimize network parameters in training process. Each mini-batch contains query image I, similar image I^+ and dissimilar images $\{I_k^-\}_{k=1}^n$, and the gradients of the objective function L with respect to $f_h(I)$, $f_h(I^+)$ and $\{f_h(\{I_k^-\}_{k=1}^n)\}$ are as following.

$$\begin{aligned} \frac{\partial L}{\partial f_h(I)} = & \frac{1}{Z}\frac{\psi^2}{(\psi + \hat{R}(I,I^+))^2}\frac{\partial \hat{R}(I,I^+)}{\partial f_h(I)} \\ & + \alpha(f_h(I) - b(I)) + \frac{\beta}{n+2}(mean_i(f_h(I_i) - 0.5)) \end{aligned} \quad (10)$$

$$\begin{aligned} \frac{\partial L}{\partial f_h(I^+)} = & \frac{1}{Z}\frac{\psi^2}{(\psi + \hat{R}(I,I^+))^2}\frac{\partial \hat{R}(I,I^+)}{\partial f_h(I^+)} \\ & + \alpha(f_h(I^+) - b(I^+)) + \frac{\beta}{n+2}(mean_i(f_h(I_i) - 0.5)) \end{aligned} \quad (11)$$

$$\begin{aligned} \frac{\partial L}{\partial f_h(I_k^-)} = & \frac{1}{Z}\frac{\psi^2}{(\psi + \hat{R}(I,I^+))^2}\frac{\partial \hat{R}(I,I^+)}{\partial f_h(I_k^-)} \\ & + \alpha(f_h(I_k^-) - b(I_k^-)) + \frac{\beta}{n+2}(mean_i(f_h(I_i) - 0.5)) \end{aligned} \quad (12)$$

Obviously, $\frac{\partial \hat{R}(I,I^+)}{\partial f_h(I)}$ and $\frac{\partial \hat{R}(I,I^+)}{\partial f_h(I^+)}$ is easy to be calculated. Since different samples could contain same image, we find that the overall gradient of the dissimilar images can be generated by calculating the gradient of each dissimilar image. For computing $\frac{\partial \hat{R}(I,I^+)}{\partial f_h(I_d^-)}$, we adopt the calculation methods for gradient to greatly reduce computation cost as shown in Eq.(13).

$$\frac{\partial \hat{R}(I,I^+)}{\partial f_h(I_k^-)} = \sum_{i=1}^{n}\frac{\partial \hat{R}(I,I^+)}{\partial f_h(I_i^-)} \quad (13)$$

Eq.(13) is very similar to traditional calculation method of loss function based on partial derivative. The only difference is that partial differential is calculated with respect to the outputs of all dissimilar images in the groups as follows Eq.(13).

3.2. Batch-Process Fashion

We implement the model training in a batch-process fashion to avoid loading all the data at once. The following steps are mainly included each iteration. Suppose that the training data contains *K* categories and each category includes *O* images. Firstly, we randomly select *K* categories randomly as query images. For each query image, we construct a fixed number of groups, in which the image with different label from query image is randomly selected from the remaining categories. In this way, the images distributed over the generated training samples are relatively centralized. Considering that the images and categories are randomly selected each iteration, our method can generate all possible groups with enough iterations.

4. EXPERMENTS

4.1. Datasets and Settings

We validate our algorithm on CIFAR10 and NUS-WIDE datasets. The CIFAR10 dataset contains 60,000 32×32 color images of 10 classes [21]; The NUS-WIDE dataset consists of 269,648 images [22]. We select the subset of images

annotated with the 21 most frequently happened classes in NUS-WIDE dataset [23]. Training data is used to train model and the query image is searched within the query set by applying the leave-one-out [12]. In our experiments, network is initialized by the weights of AlexNet which has been trained on the ImageNet dataset. The parameters ψ , ∂ and β are empirically set as 20, 0.01 and 0.1 respectively.

Table 1. mAP w.r.t. different number of bits on two datasets.

Method	CIFAR10(mAP)				NUS-WIDE(mAP)			
	16bits	24bits	32bits	48bits	16bits	24bits	32bits	48bits
Ours	**0.819**	**0.832**	**0.838**	**0.846**	**0.782**	**0.789**	**0.792**	**0.801**
DHTSP-F [15]	0.812	0.826	0.833	0.835	0.766	0.779	0.783	0.786
DTSPH [14][15]	0.782	0.800	0.803	0.805	0.770	0.786	0.788	0.789
DPSH [10]	0.742	0.764	0.765	0.770	0.696	0.711	0.726	0.730
DRSCH [12]	0.613	0.622	0.631	0.631	0.618	0.622	0.623	0.628
DSRH [13]	0.608	0.617	0.617	0.618	0.609	0.618.	0.621	0.631
SDH [8]	0.384	0.412	0.405	0.432	0.530	0.535	0.540	0.536

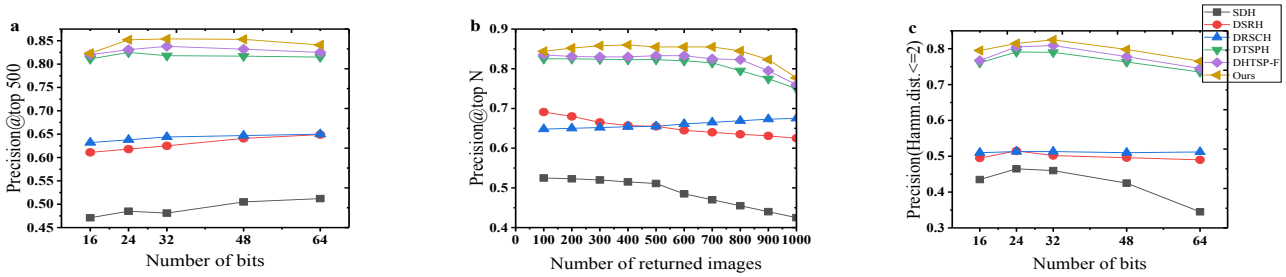

Fig 2. The results of comparison methods on the CIFAR10 dataset

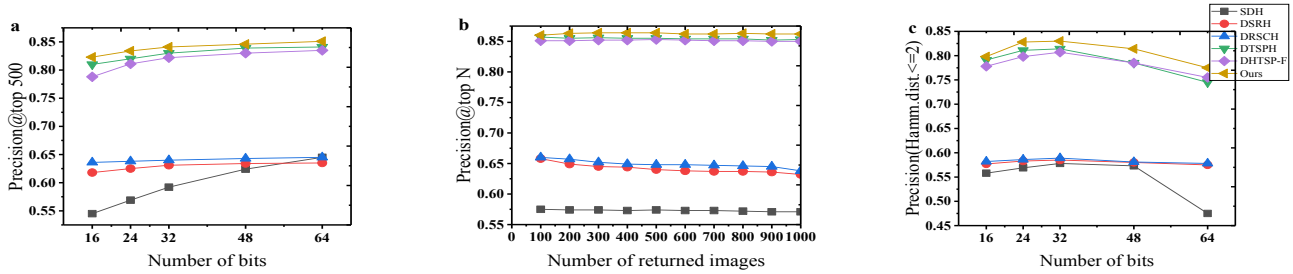

Fig 3. The results of comparison methods on the NUS-WIDE dataset

4.2. Experimental Results

We compare our method with five state-of-the-art hashing methods, including SDH, DRSH, DRSCH, DTSPH, DHTSP-F by four evaluation metrics, which are mean average precision (Table 1), precision at top 500 samples w.r.t. different code lengths (Section a in Fig.2. or Fig.3), precision curves with 64 bits wr.t. different numbers of top returned samples(Section b in Fig.2. or Fig.3), and precision within Hamming distance 2(Section c in Fig.2. or Fig.3).

Performance on CIFAR10 dataset. The experimental results as presented in Table 1 show that mAP of our model exceed that of other deep hash models. The mPA from calculation by using the proposed model is 0.819, 0.832, 0.838, 0.846 with hashing code lengths from 16 to 48 bits as shown in Table 1 respectively. The experimental results for other three evaluation metrics show that our model achieves promising performance as shown in Fig.2.

Performance on NUS-WIDE dataset. For NUS-WIDE dataset is very large, we calculate mAP values within the top 50,000 returned samples. In particular, mAP of our model increase from 0.770 to 0.782 with 16-bits hash code. Experimental results on other three evaluation metrics shown in Fig.3 introduce that our model achieve more performance than other hash model which has been discussed in large scale image retrieval tasks.

5. CONCLUSION

In the paper, we propose a novel end-to-end deep hashing model with preserving the images on the top of the result list similar to query images to improve performance of image retrieval effectively. The optimized AlexNet is used to extract better feature descriptors and generate the high-quality hash codes. Experimental results show that our method outperform several state-of-the-art deep hashing methods.

Acknowledgments. This work is supported by the National Natural Science Foundation of China (No. 61872326, No.61672475, No. 61702471).

6. REFERENCES

[1] Li L, Yu M, Shao L. Learning Short Binary Codes for Large-scale Image Retrieval [J]. *IEEE Transactions on Image Processing*, 2017, 26(3):1289-1299.

[2] Wang, Guan'an, et al. "Semi-Supervised Generative Adversarial Hashing for Image Retrieval." *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018. pp. 469-485

[3] Liu, L., Shen, F., Shen, Y., Liu, X., & Shao, L. Deep sketch hashing: Fast free-hand sketch-based image retrieval. *In Proc. CVPR.2017*.pp. 2862-2871.

[4] Yan C, Xie H, Yang D, et al. Supervised hash coding with deep neural network for environment perception of intelligent vehicles[J].*IEEE transactions on intelligent transportation systems*, 2018, 19(1): 284-295.

[5] Gong Y, Lazebnik S. Iterative quantization: A procrustean approach to learning binary codes[C]// *2011 IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, 2011: 817-824.

[6] Kong W, Li W J. Isotropic hashing[C]// *International Conference on Neural Information Processing Systems*. Curran Associates Inc. 2012:1646-1654.

[7] Yang H F, Lin K, Chen C S. Supervised Learning of Semantics-Preserving Hashing via Deep Neural Networks for Large-Scale Image Search. Computer Science*, arXiv:150700101*, 2015.

[8] Liong V E, Lu J, Wang G, et al. Deep hashing for compact binary codes learning[C]//*Computer Vision and Pattern Recognition*. IEEE, 2015:2475-2483.

[9] Xia R, Pan Y, Lai H, et al. Supervised hashing for image retrieval via image representation learning[C]//*AAAI*.2014, 1(2014): 2.

[10] Li W J, Wang S, Kang W C. Feature learning based deep supervised hashing with pairwise labels[J].*arXiv preprint arXiv:1511.03855*, 2015.

[11] Zhu H, Long M, Wang J, et al. Deep Hashing Network for efficient similarity retrieval[C]// *Thirtieth AAAI Conference on Artificial Intelligence*. AAAI Press, 2016:2415-2421.

[12] Zhang R, Lin L, Zhang R, et al. Bit-Scalable Deep Hashing With Regularized Similarity Learning for Image Retrieval and Person Re-Identification.[J]. *IEEE Transactions on Image Processing*, 2015, 24(12):4766-4779.

[13] Zhao F, Huang Y, Wang L, et al. Deep semantic ranking based hashing for multi-label image retrieval[C]// *Computer Vision and Pattern Recognition*. IEEE, 2015:1556-1564.

[14] Li Q, Fu H, Kong X. Deep Top Similarity Preserving Hashing for Image Retrieval[C]//*International Conference on Image and Graphics*. Springer, Cham, 2017: 206-215.

[15] Li Q, Fu H, Kong X, et al. Deep hashing with top similarity preserving for image retrieval[J].*Multimedia Tools and Applications*, 2018: 1-21.

[16] Bai C, Huang L, Pan X, et al. Optimization of deep convolutional neural network for large scale image retrieval[J]. *Neurocomputing*, 2018, 303: 60-67.

[17] Krizhevsky, Alex, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. *International Conference on Neural Information Processing Systems Curran Associates Inc*. 2012:1097-1105.

[18] I. Goodfellow , D. Warde-Farley , M. Mirza , A. Courville , Y. Bengio , Maxout net-works, in: Proceedings of the International Conference on Machine Learning, 2013, pp. 1319–1327.

[19] Sun Y, Wang X, Tang X. Deep Learning Face Representation from Predicting 10,000 Classes[C]// *Computer Vision and Pattern Recognition. IEEE*, 2014:1891-1898.

[20] Norouzi M, Fleet D J, Salakhutdinov R R. Hamming distance metric learning[C]//*Advances in neural information processing systems*. 2012: 1061-1069.

[21] Krizhevsky, Alex, and Geoffrey Hinton. Learning multiple layers of features from tiny images. Vol. 1. No. 4. *Technical report, University of Toronto*, 2009.

[22] Chua T S, Tang J, Hong R, et al. NUS-WIDE: a real-world web image database from National University of Singapore[C]//*Proceedings of the ACM international conference on image and video retrieval. ACM*, 2009: 48.

[23] Zhu X, Zhang L, Huang Z. A sparse embedding and least variance encoding approach to hashing[J]. *IEEE Trans Image Process*, 2014, 23(9):3737-3750.