Image Retrieval Based on ResNet and KSH

Jinyun Lu

Chongqing Educational Evaluation Institute, Chongqing 400000, China.

jimmloo@qq.com

Abstract. Hashing have had been successfully applied to different methods regarding a wide range of problems. It is fairly very effective for large-scale image retrieval in reducing the processing time. Although a number of hashing methods have been developed in recent years. Most of them the methods are based on hand-crafted features, which might not be optimally compatible with the hashing procedure for dealing with large datasets. Furthermore, recently, deep hash learning has been proposed to generate hash code, simultaneously, extracting to better extract the image features, which has shown better performance than the traditional methods of hand-crafted features. In this paper, we propose a new supervised hashing framework based on deep Residual Networks and kernel-based supervised hashing (KSH). Firstly, we exploit the learning abilities of deep residual network to mine the inherent hidden relationship of image content, extract deep feature descriptors, and increase the visual expression of images Secondly, kernel-based supervised hashing is applied to learn from the high-dimensional image feature and map into low-dimensional hamming space and achieve compact Hash codes. Finally, image retrieval is accomplished in low-dimensional hamming space. Experimental results of MNIST, CIFAR-10, CIFAR-100 and Caltech 256 show that the expression ability of visual feature is effectively improved and the image retrieval performance is substantially boosted compared with other related methods.

Keywords: ResNet, KSH, hash learning, deep learning.

INTRODUCTION

With explosive growing of image data available on the Web, quick content-based image searching in large-scale image datasets becomes on demand, approximate nearest neighbor(ANN)[1] search in large-scale images has become a rather hot search topic in recent years. However, it is difficult to find an image that is the nearest to the item searching for. The distance computation between the searching item and database item is costly[2]. It's very difficult to find the nearest neighbor.

In content-based image retrieval, images representation and computational costs play a vital role. Many researchers are committed to retrieve relevant images from the large-scale image datasets, efficiently. Because of the huge cost of calculation, traditional linear search is no longer suitable for retrieval of large corpus. Thus, many researchers have been devoted to ANN, for its fast responding speed and low memory consumption. ANN and hashing, instead of linear search, are put forward to accelerate. These methods aim to learn low-dimensional similarity-preserving codes representation that encode similar input images into nearby binary hash codes. Thanks to the binary codes generated by these methods, fast search can be achieved by Hamming distance calculation, which reduces the computational cost[2].

Traditional image retrieval systems based on supervised learning usually contain two main steps. Firstly, images are encoded with a vector of traditional hand-crafted visual descriptor constraining redundant information, such as image semantics against noise. Secondly, the hashing learning is presented as pairwise optimization problem preserving the pairwise label information in the learned Hamming space. Since the two steps above are two separate problems, it easy leading to dissatisfied results, such as the representation of the feature may not be optimally compatible with the coding process. Furthermore, the manual feature projecting processing usually requires additional knowledge.

Hashing method has become one of the most popular and effective solution for approximate nearest neighbor techniques[2] Hashing method projects high-dimensional features into low-dimensional space, and then generate a compact binary codes[2].

Recent studies have shown that deep convolutional neural networks (CNNs) significantly improve the performance of various visual tasks[3], such as object detection, image classification and segmentation. These achievements are reaffirmed by the ability of deep CNN to learn the rich intermediate image descriptors.

In this paper, we investigate the recently widely-applied technique, deep learning, to learn hash. For this investigation, we apply deep Residual Networks (ResNet)[4] to extract deep feature descriptor, next KSH[5] is applied to learn hash codes. Experimental results on the MNIST, CIFAR-10, CIFAR-100 and Caltech 256, ResNet hash achieves better performance than other related methods.

RELATED WORK

In order to obtain the intrinsic hidden relation of large image data and generate more representative features, Hinton et al use deep learning to extracting more effective image features. Tang et al[6] used the sparsity connection on the first level of Deep Belief Network(DBN), reduce probability noise was used to improve the robustness of DBN output characteristics Lee et al[7] constructed a convolution depth confidence network (CDBN), which can be used to learn effective higher-order features from image. Huang et al[8] proposed a Local Convolutional Restricted Boltzmann Machines(LCRBM) method on the basis of CDBN. The model uses the overall structure of the object class to learn the characteristics and achieves better performance in the face recognition task. He et al[9] through the convolution layer and the full connection layer of convolution Neural Network, add SPP (Spatial Pyramid Pooling) layer to learn the different sizes of images directly and generate multiscale features. Nearest neighbor retrieval methods (such as R-tree, KD-tree), it is difficult to apply to large scale data, but, hash learning can solve large-scale image retrieval problems. Xia et al[10] propose a supervised deep hash learning method CNNH and CNN+. By decomposing the semantic similarity matrix of the trained image data into a binary hash code keeps the similarity, the depth convolution network is trained by the label data of the image and the approximate hash code, and a better retrieval precision is obtained. Lin et al[11] propose a hash learning framework based on deep convolution neural network, which adds a hash layer on the network structure to learn hash coding, and can also learn the representation of the image and obtain accurate retrieval performance.

METHOD

Based Deep Residual Network Extract Deep Feature Description.

The section headings are in boldface capital and lowercase letters. Second level headings are typed as part of the succeeding paragraph (like the subsection heading of this paragraph). All manuscripts must be in English, also the table and figure texts, otherwise we cannot publish your paper. Please keep a second copy of your manuscript in your office. When receiving the paper, we assume that the corresponding authors grant us the copyright to use the paper for the book or journal in question. When receiving the paper, we assume that the corresponding authors grant us the copyright to use the paper for the book or journal in question. When receiving the paper, we assume that the corresponding authors grant us the copyright to use.

CNN is widely used to extract the deep feature of the learning image data. CNN takes the local perception area of the image as the input of the network, and then transmits the information to different layers. Each layer has an invariant feature of translation, rotation and scaling through a digital filter.

Deep residual network performs well in the field of computer vision as: target detection, image classification and image segmentation[4, 12, 13]. Compared with other networks, Resnet can quickly reduce neural network training time. When transferring, ResNet ensure the output characteristics of the expression, and use normalization and global avg_pool to generalize the network more widely.

SIFT, Gist and other methods to extract the image features will lose the original information of the image, and the extracted feature dimension is easy to generate dimension disaster, the features also are local features, and the semantic information can't be embedded with this kind of method, which belongs to unsupervised learning. As ResNet can extract high-level semantic information and low dimension, it is also a supervised learning.

We use the ResNet network to extract the depth eigenvalue of the image frame, as shown in the figure. The size of the input image of the neural network is 224×224 the size of the output is 256 depth characteristics.

FIGURE 1. Extracting Features Description through Resnet Training

TABLE 1. Architectures for Resnet Framework Building Blocks are shown in Brackets with the Numbers of Blocks Stacked Down-Sampling is Performed by Conv2_1, Conv3_1, and Conv4_1 with a Strode of 2.

Layer name	Con1	Con2_x	Con3_x	Con4_x	Avg_pool	Feature layer	Classification layer
Configuration	[3×3,16]	$\begin{bmatrix} 3 \times 3, 16 \\ 3 \times 3, 16 \end{bmatrix} \times 5$	$\begin{bmatrix} 3 \times 3, 32 \\ 3 \times 3, 32 \end{bmatrix} \times 5$	$\begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 5$	256	256	

Image Retrieval Based on Supervised Kernel Hash

Given image feature dataset $X = \{x_1, \dots, x_n\} \subset \mathbb{R}^d$, the aim of hashing is find hash function $h : \mathbb{R}^d \rightarrow \{1, -1\}$, KSH using kernel function $\kappa : \mathbb{R}^d \cdot \mathbb{R}^d \rightarrow \mathbb{R}$ to construct hash function. Hashing function define as follows:

In order to enhance the resolution of hash function for linear and nonseparable high dimensional data, hash codes are constructed by using kernel function. Where using Resnet extract feature is $X = \{x_1, \dots, x_n\} \subset \mathbb{R}^d$, kernel function is $\kappa : \mathbb{R}^d \cdot \mathbb{R}^d \rightarrow \mathbb{R}$, and hash function is $h : \mathbb{R}^d \rightarrow \{1, -1\}$.

$$f(x) = \sum_{j=1}^m \kappa(x_{(j)}, x) a_j - b \quad (1)$$

$$h(f(x)) = \text{sgn}(f(x)) = \begin{cases} 1, & f(x) > 0 \\ -1, & f(x) \leq 0 \end{cases} \quad (2)$$

Where $x_1, \dots, x_n$ are m samples uniformly selected at random, $a_j \in \mathbb{R}$ and $b \in \mathbb{R}$ is bias.

In order to achieve fast hash map, m is always less than n . The hash function, in addition to satisfying the similar consistency between the low dimension space and the original high dimensional space, should also ensure that the hash encoding of the production is balanced, that is, the hash function should satisfy the $\sum_{i=1}^n h(x_i) = 0$, and

then the bias value $b = \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^m \kappa(x_j, x) a_j$ is replaced by the eq. 1 we obtain

$$\begin{aligned} f(x) &= \sum_{j=1}^m (\kappa(x_j, x) - \frac{1}{n} \sum_{i=1}^n \kappa(x_j, x)) a_j \\ &= a^T \bar{k}(x) \end{aligned} \quad (3)$$

Where $a = [a_1, \dots, a_m]^T$ and $\mathbb{R}^d \rightarrow \mathbb{R}^m$ is a vectorial map defined by:

$$k = [\kappa(x_1, x) - \mu_1, \dots, \kappa(x_m, x) - \mu_m]^T \quad (4)$$

$\mu = \frac{1}{n} \sum_{i=1}^n \kappa(x_j, x_i)$ Can be precomputed. The coefficient vector a came as a random direction drawn. Since completely determines a hash function, we seek to learn by leveraging supervised information so that the resulted hash function is discriminative

Suppose that r hash bits are needed. So, we have to find r coefficient vectors $a_1, \dots, a_r$ to construct hash functions $h_k(x) = \text{sgn}(a^T \bar{k}(x)) \mid k \in [1, r]$. The label information for the training images can be obtained by the semantic correlation and space distance of the image, and $\text{lable}(x_i, x_j) = 1$ indicates that the image x_i and x_j are similar; $\text{lable}(x_i, x_j) = -1$, on the other hand, represents a large difference between the image x_i and the x_j . To explicitly record the pairwise relationships among $X_l = \{x_1, \dots, x_l\}$, we define a label matrix $S \in \mathbb{R}^{l \times l}$ as

$$s_{ij} = \begin{cases} 1, & \text{lable}(x_i, x_j) = 1 \\ -1, & \text{lable}(x_i, x_j) = -1 \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

In this, $\text{lable}(x_i, x_j) = 1$, $s_{ii} = 1$, $s_{ij} = 0$ indicates that the similarity between images is uncertain. In order to enhance the discrimination ability of hash codes, we can efficiently judge the similarity between images x_i and x_j in Hamming space. Directly optimizing the Hamming distances is nontrivial because of the complex mathematical formula:

$$D_h(x_i, x_j) = \begin{cases} 0, & s_{ij} = 1 \\ r, & s_{ij} = -1 \end{cases} \quad (6)$$

In order to minimize the distance between the similar matrix $s' = \frac{1}{r} H_l H_l^T$ and the supervised matrix s , the objective function is

$$\min \Gamma = \left\| \frac{1}{r} H_l H_l^T - s \right\|_F^2 \quad (7)$$

Where $H = \begin{bmatrix} \text{code}(x) \\ \dots \\ \text{code}(x_l) \end{bmatrix} \in \{1, -1\}^{l \times r}$ denotes the code matrix of X , and $\|\bullet\|_F^2$ presents the Frobenius norm.

We can generalize $\text{sgn}(\bullet)$ to take the elementwise sign operation for any vector or matrix input, and then express the code matrix H_l as

$$H_l = \begin{bmatrix} h_1(x_1) & \dots & h_r(x) \\ \dots & \dots & \dots \\ h_1(x_1) & \dots & h_r(x) \end{bmatrix} = \text{sgn}(\bar{K}_l A) \quad (8)$$

Where $\bar{K}_l = [\bar{k}(x_1), \dots, \bar{k}(x_l)]^T \in \mathbb{R}^{l \times m}$, after substituting H_l in eq. (7) with eq. (8), we obtain an analytical form of the objective function $\Gamma(A)$:

$$\min_{A \in \mathbb{R}^{m \times r}} \Gamma(A) = \left\| \frac{1}{r} \text{sgn}(\bar{K}_l A) (\text{sgn}(\bar{K}_l A))^T - S \right\|_F^2 \quad (9)$$

We rewrite $\Gamma(A)$ in eq. (9) as

$$\min_{A \in \mathbb{R}^{m \times r}} \left\| \sum_{k=1}^r \text{sgn}(\bar{K}_l a_k) (\text{sgn}(\bar{K}_l a_k))^T - rS \right\|_F^2 \quad (10)$$

$a_1^*, \dots, a_{k-1}^*$ We need to estimate a_k and define matrix $R_{k-1} = rS - \sum_{i=1}^{k-1} \text{sgn}(\bar{K}_l a_i^*) (\text{sgn}(\bar{K}_l a_i^*))^T$, where $R_0 = rS$ is estimated by greedy algorithm, and a_k is gradually estimated.

$$\begin{aligned} & \left\| \text{sgn}(\bar{K}_l a_k) (\text{sgn}(\bar{K}_l a_k))^T - R_{k-1} \right\|_F^2 \\ &= -2(\text{sgn}(\bar{K}_l a_k))^T R_{k-1} \text{sgn}(\bar{K}_l a_k) + l^2 + \text{tr}(R_{k-1}^2) \\ &= -2(\text{sgn}(\bar{K}_l a_k))^T R_{k-1} \text{sgn}(\bar{K}_l a_k) + \text{const} \end{aligned} \quad (11)$$

Discarding the constant term, we arrive at a cleaner cost

$$\mathcal{G}(a_k) = -(\text{sgn}(\bar{K}_l a_k))^T R_{k-1} \text{sgn}(\bar{K}_l a_k) \quad (12)$$

The $\text{sgn}(x)$ function in the objective function makes $\mathcal{G}(a_k)$ discontinuous, and $\mathcal{G}(a_k)$ is not a convex function, it is hard for $\mathcal{G}(a_k)$ to directly minimize it. We use $\varphi(x)$ instead of $\text{sgn}(x)$ to get the approximate objective function $\tilde{\mathcal{G}}(a_k)$:

$$\tilde{\mathcal{G}}(a_k) = -(\mathcal{G}(\bar{K}_l a_k))^T R_{k-1} \varphi(\bar{K}_l a_k) \quad (13)$$

We can minimize $\tilde{\mathcal{G}}(a_k)$ by gradient descent method and get $\tilde{\mathcal{G}}(a_k)$ gradient about a_k :

$$\nabla \tilde{\mathcal{G}}(a_k) = -\bar{K}_l^T ((R_{k-1} b) \odot (1 - b \odot b)) \quad (14)$$

Where $b = \varphi(\bar{K}_l a_k) \in \mathbb{R}^l$, $1 = [1, \dots, 1] \in \mathbb{R}^l$, $\odot$ representation Hadamard product.

Using supervising the learning of the vector coefficients a , the hash function $H(x)$ and hash table H are generated to map the deep features of the query image by mapping to get $code_r(x_q)$, and the distance between the hash codes in $code_r(x_q)$ and hash table H is calculated, and the nearest image is returned as the retrieval result.

EXPERIMENT RESULTS

In this section, we validate our RSKH hash learning on several public datasets of image retrieval, including MNIST, CIFAR-10, CIFAR-100 and Caltech 256. For each dataset, the images are split into a training set and a query set. We compare our methods with eight state-of-the-art approaches.

Data Sets

We evaluate our method on two standard large image datasets with semantic labels: MNIST, CIFAR-10, CIFAR-100 and Caltech 256.

MNIST Dataset contains 10 categories of the handwritten digits from 0 to 9. There are 60k training images, and 10k test images. All the digits are normalized to gray-scale images with size 28×28 . We use 10K images as the query set and the other 60K as the training samples.

CIFAR-10 dataset consists of 60, k 32×32 color images which are categorized into 10 classes (6k images per class). It is a single-label dataset in which each image belongs to one of the ten classes. We use 10K images as the query set and the other 50K as the training samples.

CIFAR-100 Dataset contains 100 object categories and each class consists of 6, 00 images, resulting in a total of 60,000 images. The dataset is split into training and query sets, with 50,000 and 10,000 images respectively.

Caltech 256 Dataset contains a total of 30,607 images, split between 256 distinct object categories and a background category. We randomly selected 70% as training sets, the rest as the query sets.

For hashing methods which use hand-crafted features, we represent each image in datasets by a 512-dimensional GIST vector. For deep hashing methods, we first resize all images to be 224×224 pixels and then directly use the raw image pixels as input. For data pre-processing, we follow the standard way of feature normalization by making each dimension of the feature vectors to have zero mean and equal variance.

Experimental Baselines

We compare our method with several state-of-the-art hashing methods. These methods can be categorized into five classes:

Data-dependent hashing methods with hand-crafted features, including Locality-Sensitive Hash(LSH)[14], Locality-Sensitive Binary Codes from Shift-Invariant Kernels(SKLSH)[15].

Unsupervised hashing methods with hand-crafted features, including Spectral Hashing(SH)[16], Spherical Hashing (SpH)[17], Iterative Quantization(ITQ)[18], Density Sensitive Hashing(DSH)[19].

To evaluate the quality of hashing, we use evaluation metrics: Mean Average Precision (MAP).

Results of Search Accuracies

In table 2 and table 3, we report the Map results with different code lengths on MNIST, CIFAR10, CIFAR100, Caltech 256. In these tables, the best results are in boldface. From these tables, we can see that RSKH hashing achieves better results than data-dependent hashing method, unsupervised hashing methods. The longer the hash codes length is, the higher the retrieval precision is. RSKH hashing shows slightly but consistently better search accuracies than other methods, which verifies that incorporating both the approximate hash codes and image tags in training helps to learn a better shared image representation and enhance the hashing performance. These results verify that using Residual Networks simultaneously learning useful representation of images and KSH get hash codes of preserving similarities can benefit each other.

TABLE 2. Image Retrieval Results (MAP) With Various Number of Bits on the MNIST Dataset and the CIFAR 256.

Method	MNIST					CIFAR 10				
Bits	8	16	48	64	128	8	16	48	64	128
RKSH	0.972	0.974	0.989	0.985	0.990	0.763	0.778	0.852	0.869	0.871
ITQ	0.649	0.779	0.874	0.890	0.908	0.209	0.240	0.280	0.288	0.302
LSH	0.345	0.504	0.751	0.768	0.827	0.154	0.182	0.220	0.238	0.262
SH	0.105	0.105	0.105	0.105	0.105	0.106	0.106	0.106	0.106	0.106
SKLSH	0.237	0.370	0.512	0.539	0.636	0.139	0.147	0.158	0.165	0.192
DSH	0.365	0.565	0.603	0.598	0.660	0.184	0.214	0.243	0.250	0.268
SpH	0.381	0.523	0.545	0.595	0.690	0.166	0.181	0.191	0.195	0.216

TABLE 3. Image Retrieval Results (MAP) with Various Number of Bits on the CIFAR 100 Dataset and the Caltech 256.

Method	CIFAR 100					Caltech 256				
Bits	8	16	48	64	128	8	16	48	64	128
RKSH	0.244	0.293	0.472	0.532	0.573	0.463	0.625	0.644	0.698	0.706
ITQ	0.064	0.112	0.196	0.208	0.245	0.037	0.066	0.104	0.113	0.128
LSH	0.044	0.074	0.151	0.176	0.221	0.020	0.035	0.080	0.089	0.116
SH	0.029	0.029	0.029	0.029	0.029	0.009	0.009	0.009	0.009	0.009
SKLSH	0.040	0.052	0.072	0.087	0.120	0.006	0.006	0.006	0.007	0.007
DSH	0.054	0.086	0.129	0.151	0.188	0.022	0.041	0.070	0.076	0.087
SpH	0.057	0.094	0.131	0.148	0.182	0.006	0.006	0.006	0.005	0.005

SUMMARY

In this paper, we have proposed a novel supervised hashing method for image retrieval based on ResNet and SKH, called RSKH. Hashing generates bitwise hash codes for images via a carefully designed deep architecture. RSKH hashing can learn better codes than other methods without end-to-end architecture. Experiments on real datasets show that RSKH hashing can outperform other methods to achieve the state-of-the-art performance in image retrieval applications.

REFERENCES

1. Andoni, A., and Indyk, P.: 'Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions', in Editor (Ed.) ^ (Eds.): 'Book Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions' (IEEE, 2006, edn.), pp. 459-468.
2. Gionis, A., Indyk, P., and Motwani, R.: 'Similarity search in high dimensions via hashing', in Editor (Ed.) ^ (Eds.): 'Book Similarity search in high dimensions via hashing' (1999, edn.), pp. 518-529.
3. Sharif Razavian, A., Azizpour, H., Sullivan, J., and Carlsson, S.: 'CNN features off-the-shelf: an astounding baseline for recognition', in Editor (Ed.) ^ (Eds.): 'Book CNN features off-the-shelf: an astounding baseline for recognition' (2014, edn.), pp. 806-813.
4. He, K., Zhang, X., Ren, S., and Sun, J.: 'Deep residual learning for image recognition', in Editor (Ed.) ^ (Eds.): 'Book Deep residual learning for image recognition' (2016, edn.), pp. 770-778.

5. Liu, W., Wang, J., Ji, R., Jiang, Y.-G., and Chang, S.-F.: 'Supervised hashing with kernels', in Editor (Ed.) ^ (Eds.): 'Book Supervised hashing with kernels' (IEEE, 2012, edn.), pp. 2074-2081.
6. Tang, Y., and Eliasmith, C.: 'Deep networks for robust visual recognition', in Editor (Ed.) ^ (Eds.): 'Book Deep networks for robust visual recognition' (2010, edn.), pp. 1055-1062.
7. Lee, H., Grosse, R., Ranganath, R., and Ng, A.Y.: 'Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations', in Editor (Ed.) ^ (Eds.): 'Book Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations' (ACM, 2009, edn.), pp. 609-616.
8. Huang, G.B., Lee, H., and Learned-Miller, E.: 'Learning hierarchical representations for face verification with convolutional deep belief networks', in Editor (Ed.) ^ (Eds.): 'Book Learning hierarchical representations for face verification with convolutional deep belief networks' (2012, edn.), pp. 2518-2525.
9. He, K., Zhang, X., Ren, S., and Sun, J.: 'Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition', *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 2014, 37, (9), pp. 1904-1916.
10. Xia, R., Pan, Y., Lai, H., Liu, C., and Yan, S.: 'Supervised Hashing for Image Retrieval via Image Representation Learning', in Editor (Ed.) ^ (Eds.): 'Book Supervised Hashing for Image Retrieval via Image Representation Learning' (2014, edn.), pp. 2.
11. Lin, K., Yang, H.-F., Hsiao, J.-H., and Chen, C.-S.: 'Deep learning of binary hash codes for fast image retrieval', in Editor (Ed.) ^ (Eds.): 'Book Deep learning of binary hash codes for fast image retrieval' (2015, edn.), pp. 27-35.
12. He, K., Zhang, X., Ren, S., and Sun, J.: 'Identity mappings in deep residual networks', in Editor (Ed.) ^ (Eds.): 'Book Identity mappings in deep residual networks' (Springer, 2016, edn.), pp. 630-645.
13. Zagoruyko, S., and Komodakis, N.: 'Wide residual networks', *arXiv preprint arXiv: 1605.07146*, 2016.
14. Kulis, B., and Grauman, K.: 'Kernelized locality-sensitive hashing for scalable image search', in Editor (Ed.) ^ (Eds.): 'Book Kernelized locality-sensitive hashing for scalable image search' (2010, edn.), pp. 2130-2137.
15. Raginsky, M., and Lazebnik, S.: 'Locality-sensitive binary codes from shift-invariant kernels', in Editor (Ed.) ^ (Eds.): 'Book Locality-sensitive binary codes from shift-invariant kernels' (2009, edn.), pp. 1509-1517.
16. Weiss, Y., Torralba, A., and Fergus, R.: 'Spectral hashing', in Editor (Ed.) ^ (Eds.): 'Book Spectral hashing' (2009, edn.), pp. 1753-1760.
17. Heo, J.-P., Lee, Y., He, J., Chang, S.-F., and Yoon, S.-E.: 'Spherical hashing', in Editor (Ed.) ^ (Eds.): 'Book Spherical hashing' (IEEE, 2012, edn.), pp. 2957-2964.
18. Gong, Y., Lazebnik, S., Gordo, A., and Perronnin, F.: 'Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013, 35, (12), pp. 2916-2929.
19. Jin, Z., Li, C., Lin, Y., and CAI, D.: 'Density sensitive hashing', *IEEE transactions on cybernetics*, 2014, 44, (8), pp. 1362-1371.