

Article

Online Hashing for Scalable Remote Sensing Image Retrieval

Peng Li ^{1,2}, Xiaoyu Zhang ³, Xiaobin Zhu ⁴ and Peng Ren ^{1,2*}

¹ College of Information and Control Engineering, China University of Petroleum (East China), Qingdao 266580, China; lipeng@upc.edu.cn

² State Key Laboratory of Mathematical Engineering and Advanced Computing, Wuxi 214125, China

³ Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China; zhangxiaoyu@iie.ac.cn

⁴ College of Computer and Information Engineering, Beijing Technology and Business University, Beijing 100048, China; brucezhucas@gmail.com

* Correspondence: pengren@upc.edu.cn; Tel.: +86-532-8698-3478

Received: 9 April 2018; Accepted: 3 May 2018; Published: 4 May 2018



Abstract: Recently, hashing-based large-scale remote sensing (RS) image retrieval has attracted much attention. Many new hashing algorithms have been developed and successfully applied to fast RS image retrieval tasks. However, there exists an important problem rarely addressed in the research literature of RS image hashing. The RS images are practically produced in a streaming manner in many real-world applications, which means the data distribution keeps changing over time. Most existing RS image hashing methods are batch-based models whose hash functions are learned once for all and kept fixed all the time. Therefore, the pre-trained hash functions might not fit the ever-growing new RS images. Moreover, the batch-based models have to load all the training images into memory for model learning, which consumes many computing and memory resources. To address the above deficiencies, we propose a new online hashing method, which learns and adapts its hashing functions with respect to the newly incoming RS images in terms of a novel online partial random learning scheme. Our hash model is updated in a sequential mode such that the representative power of the learned binary codes for RS images are improved accordingly. Moreover, benefiting from the online learning strategy, our proposed hashing approach is quite suitable for scalable real-world remote sensing image retrieval. Extensive experiments on two large-scale RS image databases under online setting demonstrated the efficacy and effectiveness of the proposed method.

Keywords: hashing; remote sensing image retrieval; online learning

1. Introduction

With the rapid development of satellite and aerial vehicle technologies, we have entered an era of remote sensing (RS) big data. Automatic knowledge discovery from massive RS data has become increasingly urgent. Among emerging RS big data mining efforts, large-scale RS image retrieval has attracted an increasing amount of research interest due to its broad applications in the RS research community. For example, a fast and accurate retrieval of similar satellite cloud images can provide valuable judging information for short-term weather forecasting. Besides, in the disaster rescue scenario, a fast rescue and optimal resources allocating also depend on the real-time and precise retrieval strategies for the photographs of disaster area.

In earlier RS image retrieval systems, RS image retrieval mainly relied on manual tags in terms of sensor types, waveband information, and geographical locations of remote sensing images. However, the manual generation of tags is quite time consuming and becomes especially prohibitive when the

volume of remote sensing images is oversized. As an effective method to manage a large number of images, content-based image retrieval (CBIR) can retrieve the interesting images according to their visual content. In recent years, content-based RS image retrieval has been comprehensively studied [1–4], in which the similarity of RS images is measured by different kinds of visual descriptors. More specifically, local invariant [5], morphological [6], textural [7–9], and data-driven features [10–13] have been evaluated in terms of content-based RS image retrieval tasks. To further improve image retrieval performance levels, Li et al. [14] proposed a multiple feature-based remote sensing image retrieval approach by combining handcrafted features and data-driven features via unsupervised feature learning. Wang et al. [15] proposed a multilayered graph model for hierarchically refining retrieval results from coarse to fine. Although some encouraging progress has been made, there remains a great challenge for the content-based RS image retrieval tasks. For the aforementioned visual descriptors, their dimensions can be in the hundreds or even thousands. Exhaustively comparing the high dimensional feature descriptor of an inquiry remote sensing image with each image in the retrieval set is computationally expensive and impossible to achieve on an oversized database. Besides, the storage of the image descriptors is also a bottleneck for large-scale RS image retrieval problems.

Hashing technique is a potential solution to cope with big data retrieval due to its excellent ability in compact feature representation. The hashing methods map the input images from the high dimensional feature space to a low dimensional code space, i.e., hamming space, where each image is represented by a short binary code. It is extremely fast to perform image retrieval over such binary codes, because the hamming distance between binary codes can be efficiently calculated with XOR operation even in a modern CPU. Moreover, binary code representation significantly reduces the amount of memory required for storing the large-scale content information of images. Existing hashing approaches can be broadly categorized as data-independent and data-dependent schemes. Data-independent methods usually adopt random projections as hash functions without using any training data. One representative data-independent method is Locality Sensitive Hashing (LSH) [16–18], which projects data points to a random hyperplane and then conducts random thresholding. Although this data-independent random scheme is quite computationally efficient, it usually cannot achieve satisfactory retrieved results because it totally disregards the image data structure. Moreover, to achieve a reasonable recall rate, the LSH based methods typically require long codes and multiple tables, which degrade the search efficiency in practice. On the contrary, data-dependent hashing methods attempt to learn good data-aware hash codes by utilizing various machine learning techniques, which are usually demonstrated to be more effective than data-independent LSH. Data-dependent hashing can further be divided into unsupervised hashing [19–23] and supervised hashing methods [24–30]. For example, spectral hashing [19] and Principal Component Analysis (PCA) based hashing methods [20] belong to the unsupervised category, which does not utilize the label information of training images when learning the binary codes. Supervised hashing approaches, such as kernel-based supervised hashing [25], supervised discrete hashing [27] and deep hashing methods [29], incorporate the label information to learn semantic hashing functions.

Due to the great success of hashing in the field of natural image retrieval, many efforts have been devoted to develop efficient hashing methods for large-scale RS images retrieval tasks recently. More specifically, kernel-based nonlinear hashing was first introduced into the remote sensing community by Demir and Bruzzone [31]. Then, Li and Ren [32] proposed a novel unsupervised hashing method named partial randomness hashing (PRH) for efficient hash function construction. In [33], a novel large-scale RS image retrieval approach was proposed based on deep hashing neural networks under the supervision of labeled images. Ye et al. [34] proposed a multiple-feature learning framework for large-scale RS image hashing problem, which takes multiple complementary features as the input and learns the hybrid hash functions.

Although the hashing-based RS image retrieval methods have achieved some improvements for large-scale applications, there exist two important problems that are rarely exploited in the existing RS

image hashing approaches. (1) The existing RS image hashing methods are based on a batch learning fashion, which assume all training images are available in advance for training and the hash functions keep unchanged once the learning procedure finished. However, in many real-world RS applications, the RS images become available continuously in streaming fashion. For example, the satellite transmits remote sensing images back to the data center every day. In such environments, the RS image database is enriched by time and the new incoming images may have different distribution with the existing images or even belong to a totally new category that has never been seen before. Thus, for the batch-based hashing methods, the pre-learned hash functions may be inappropriate for the new RS images over time. One solution is to accumulate all the available data and repeatedly do batch learning to re-train new hash functions, which is a quite inefficient learning manner, especially for time-consuming hashing methods. (2) The batch-based hashing methods usually have to load all the training RS images into the memory for hash function learning. Thus, these methods make very high demands on the computing hardware such as CPU and memory, which limits their practical application on many mobile remote sensing devices. In addition, for many real large-scale RS image databases, it is even impossible to load the whole training dataset into memory, let alone training hash model. Therefore, batch-based hashing on large-scale data often results in a great deal of computational time and memory cost, which does not satisfy the requirement of the real-world applications.

To overcome the above problems, we propose a novel online hashing method for fast and scalable RS image retrieval in this paper. Online learning approaches are quite efficient for streaming data modeling [35–37]. Specifically, we first formulate our hash model based on a partial random auto-encoder and then develop a novel online hash function learning scheme to continuously update the hash model such that it fits the sequentially arriving new images over time. Our online hashing method only employs the new RS images to optimize the hash functions at each learning round and do not need to revisit all the available data, which has greatly reduced the demands on computing and memory costs. Even for the oversized RS image database that is difficult to handle using batch hashing methods, one can divide the whole big dataset into many small chunks and then implement binary code learning through our proposed online hashing method. As a result, our proposed method is very suitable and efficient for scalable RS image retrieval tasks. The main contributions of this paper are summarized as follows:

- (1) A novel online hashing method is developed for scalable RS image retrieval problem. To the best of our knowledge, our work is the first attempt to exploit online hash function learning in the large-scale remote sensing image retrieval literature.
- (2) By learning the hash functions in an online manner, the parameters of our hash model can be updated continuously according to the new obtained RS images by time, which in contrast is one main drawback of the existing batch hashing methods.
- (3) The proposed online hashing approach reduces the computing complexity and memory cost in the learning process compared with batch hashing methods. Experimental results show the superiority of our online hashing for scalable RS image retrieval tasks.

The rest of the paper is organized as follows. In Section 2, the proposed online RS image hashing method is described in detail. Extensive experiments are conducted in Section 3 to evaluate the performance of our proposed method as well as other compared approaches. Finally, conclusions are given in Section 4.

2. The Proposed Approach

Our proposed hashing approach contains two main steps: (1) hash model formulation, which defines the form of hash model used in the paper; and (2) online hash function learning, which describes how to update the hash functions dynamically based on the sequentially arriving data. The illustration of the proposed online hashing approach for scalable RS image retrieval is shown in Figure 1.

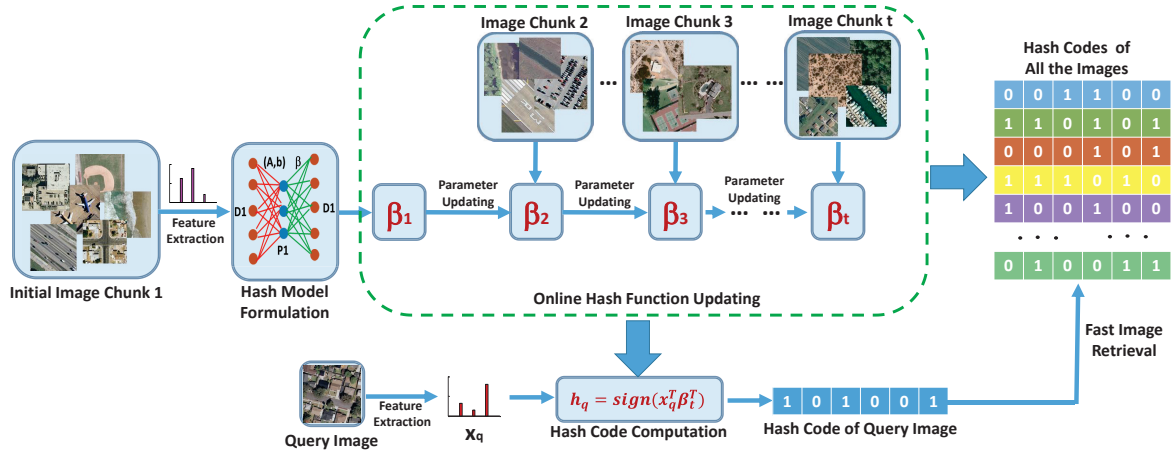


Figure 1. The illustration of the proposed online hashing approach for scalable remote sensing image retrieval.

2.1. Hash Model Formulation

Suppose that the RS image dataset used for training contains n images. Specifically, $x_i \in \mathbb{R}^d$ is a d -dimensional feature vector for the i -th image and the feature vectors for all n images are $\{x_1, x_2, \dots, x_n\}$. Denote $X = [x_1, x_2, \dots, x_n] \in \mathbb{R}^{d \times n}$ as the whole data matrix. The corresponding binary code matrix for the dataset is $H = [h_1, h_2, \dots, h_n] \in \{-1, 1\}^{r \times n}$, where r is the code length. The hash code vector for the i -th image is a column of H and is denoted as h_i . The goal of hashing is to learn hash functions that encode the original RS images from the d -dimensional feature space into an r -dimensional hamming space.

Our hash model is formulated by a partial random auto-encoder which includes both forward and backward parameters. First, the whole data matrix is randomly projected from the d -dimensional feature space to an r -dimensional relaxed hamming space with sigmoid activation function as follows:

$$P = g(X^T \cdot A + \mathbf{1}_n b) \quad (1)$$

where $A \in \mathbb{R}^{d \times r}$ is a randomly generated projection matrix and $b \in \mathbb{R}^r$ is a randomly generated bias row vector. $g(x) = 1/(1 + e^{-x})$ is the sigmoid activation function and $\mathbf{1}_n$ denotes the $n \times 1$ column vector in which all the elements are equal to 1. $P \in \mathbb{R}^{n \times r}$ is the projected data matrix. This is the forward procedure, whose parameters are randomly generated.

Then, a linear model parameter β is employed to fit randomly projected data P back to the original data X and β is learned by minimizing the following problem:

$$\hat{\beta} = \arg \min_{\beta} \|P \cdot \beta - X^T\|^2 \quad (2)$$

The optimal linear model parameter can be simply computed as follows:

$$\hat{\beta} = P^+ X^T \quad (3)$$

where the superscript $+$ denotes the Moore–Penrose generalized inverse of a matrix. P^+ can be given by $P^+ = (P^T P)^{-1} P^T$. This is the backward procedure, whose parameters are optimized based on the training images. Our hash model is inspired by extreme learning machine (ELM) approach [38], however supervised ELM computes forward to a target label matrix while our model computes backward to the original feature data X^T . Therefore, our method is in fact an unsupervised data-dependent hashing scheme.

Finally, the hash codes H for all the images in the training database can be simply obtained by $H = \text{sign}(X^T \hat{\beta}^T)$.

2.2. Online Hash Function Learning

It is easy to observe from Section 2.1 that the defined hash model is a batch-learning based hashing approach, in which all the training images are assumed to be available in advance and the hash model parameters keep fixed once the learning procedure is finished. However, as we have explained in Section 1, such hashing methods are not well adapted to the scalable streaming RS images, which is a common scenario in the real-world applications. For example, as more and more new RS images are available, the pre-learned hash functions may become unsuitable or even fail for hash code generation. Moreover, it is even impossible to load all the images into memory for learning when the training dataset is oversized. In this part, we introduce a novel online hashing scheme which can update the hash functions continuously so that it can fit the sequentially available RS images.

We assume that the new RS images are available in a stream form. Let D_i denote the data chunk received at round i , $i = \{1, 2, \dots\}$. One highlight of online learning is that when learning new information at round t , the algorithm should not access the previously seen image data D_1, \dots, D_{t-1} . Given a chunk of initial training set D_1 , we can compute its hash code as $H_1 = \text{sign}(D_1^T \hat{\beta}_1^T)$, where $\hat{\beta}_1$ is obtained by the partial random hash model based on Equation (3) as $\hat{\beta}_1 = Q_1^{-1} P_1^T D_1^T$ where $Q_1 = P_1^T P_1$, and P_1 is obtained based on Equation (1) as $P_1 = g(D_1^T \cdot A + \mathbf{1}_n b)$.

Suppose that we are given another chunk of data D_2 , the proposed method becomes minimizing the following problem if considering both image datasets D_1 and D_2 :

$$\hat{\beta}_2 = \arg \min_{\beta_2} \left\| \begin{bmatrix} P_1 \\ P_2 \end{bmatrix} \beta_2 - \begin{bmatrix} D_1^T \\ D_2^T \end{bmatrix} \right\|^2 \quad (4)$$

where the optimized $\hat{\beta}_2$ can be given by

$$\hat{\beta}_2 = \left(\begin{bmatrix} P_1 \\ P_2 \end{bmatrix}^T \begin{bmatrix} P_1 \\ P_2 \end{bmatrix} \right)^{-1} \begin{bmatrix} P_1 \\ P_2 \end{bmatrix}^T \begin{bmatrix} D_1^T \\ D_2^T \end{bmatrix} \quad (5)$$

If we let $\left(\begin{bmatrix} P_1 \\ P_2 \end{bmatrix}^T \begin{bmatrix} P_1 \\ P_2 \end{bmatrix} \right)$ be denoted by Q_2 , then

$$Q_2 = P_1^T P_1 + P_2^T P_2 = Q_1 + P_2^T P_2 \quad (6)$$

and $\hat{\beta}_2$ can be rewritten as

$$\begin{aligned} \hat{\beta}_2 &= Q_2^{-1} (P_1^T D_1^T + P_2^T D_2^T) \\ &= Q_2^{-1} (Q_1 Q_1^{-1} P_1^T D_1^T + P_2^T D_2^T) \\ &= Q_2^{-1} (Q_1 \hat{\beta}_1 + P_2^T D_2^T) \\ &= Q_2^{-1} [(Q_2 - P_2^T P_2) \hat{\beta}_1 + P_2^T D_2^T] \\ &= \hat{\beta}_1 + Q_2^{-1} P_2^T (D_2^T - P_2 \hat{\beta}_1) \end{aligned} \quad (7)$$

From Equations (6) and (7), we can see that $\hat{\beta}_2$ can be expressed as a function of $\hat{\beta}_1$ based on the new data chunk D_2 .

Without loss of generality, we can easily get a recursive form for the streaming data chunk D_i as new images arrive. When the k -th chunk of image set is received, we have

$$Q_k = Q_{k-1} + P_k^T P_k \quad (8)$$

$$\hat{\beta}_k = \hat{\beta}_{k-1} + Q_k^{-1} P_k^T (D_k^T - P_k \hat{\beta}_{k-1}) \quad (9)$$

By recursively applying (8) and (9), the hash model parameter $\hat{\beta}$ is updated with respect to the new available RS images and the learned hash codes for all the images are also improved continuously with the streaming data. More importantly, we only have to handle the current data chunk without needing to access the whole image set at each round. Therefore, our method is less constrained by the computational and space cost limitation compared with the batch hashing approaches.

The learning process of the proposed online partial randomness hashing (OPRH) method is summarized in Algorithm 1.

Algorithm 1 Online Binary Code Learning with OPRH

- 1: **Input:** Streaming image data chunk D_1, D_2, \dots, D_k , code length r
 - 2: **Output:** Hash codes H for all the images
 - 3: Randomly generate a projection matrix $A \in \mathbb{R}^{d \times r}$ and a bias row vector $b \in \mathbb{R}^r$
 - 4: Compute P_1 by $P_1 = g(D_1^T \cdot A + \mathbf{1}_n b)$
 - 5: Compute $Q_1 = P_1^T P_1$ and $\hat{\beta}_1 = Q_1^{-1} P_1^T D_1^T$
 - 6: **for** $i = 2 : k$ **do**
 - 7: Compute P_i by $P_i = g(D_i^T \cdot A + \mathbf{1}_n b)$
 - 8: Update Q_i with Equation (8)
 - 9: Update $\hat{\beta}_i$ with Equation (9)
 - 10: **end for**
 - 11: Compute the hash codes H for the whole database $X = [D_1, D_2, \dots, D_k]$ by $H = \text{sign}(X^T \hat{\beta}_k^T)$
-

2.3. Complexity Analysis

We analyze the complexity of our proposed online partial randomness hashing method. Specifically, for a stream of data chunk D_1, D_2, \dots, D_t , we update the hash model parameters at every round $i = 1, 2, \dots, t$. We analyze both the time and space complexity for hash function learning at each round.

Time Complexity: The time complexity of computing P_k at each round is $O(n_k dr)$, where n_k is the number of images in the k -th chunk, d is the dimensionality of the original feature vectors and r is the length for hash codes. The complexity of updating Q_k and $\hat{\beta}_k$ can be $O(n_k r^2)$ and $O(n_k r^3 + n_k dr)$, respectively. Thus, the overall time complexity for each round is $O(n_k r^3 + n_k dr)$.

Space Complexity: In our method, all the operations at each round are conducted on a data chunk D_k without accessing the whole dataset, space overhead of which is $O(n_k d)$. The Q_k and $\hat{\beta}_k$ updating steps occupy a space of $O(n_k r + r^2)$ and $O(dr)$ space is needed to store the final learned hash model parameter $\hat{\beta}_t$. Thus, the overall space complexity is $O(n_k d + r^2 + dr)$.

In the light of the above observations, our OPRH approach is quite suitable for scalable RS image hashing and fast retrieval because the operated data chunk at each round is much smaller than the whole large dataset. Especially when the RS image set is oversized and impossible to be loaded into the memory, we can divide the whole image set into many small chunks and employ our OPRH method for binary code learning, which can be easily finished even on a ordinary computer.

3. Experiments

3.1. Datasets and Settings

In this section, we conduct extensive experiments to evaluate the performance of our proposed OPRH. Two issues are verified in the following experiments: (1) large-scale RS image retrieval performance of our method compared to state-of-the-art batch-based hashing algorithms; and (2) the effectiveness and efficiency of the proposed OPRH method under online setting.

Two public large-scale satellite datasets are used in the experiments, i.e., SAT-4 and SAT-6 airborne datasets [39], which contain 500,000 and 405,000 images, respectively. SAT-4 dataset contains four classes and SAT-6 contains six classes. All the images in these two datasets are normalized to 28×28 pixels in size. Some example images from the two datasets are shown in Figure 2. One thousand images are randomly selected from each dataset as testing queries and the remaining images are used for training and retrieval database. We extract a 512-dimensional GIST descriptor [40] for each image as visual feature representation. Given an input image, a GIST descriptor is computed as follows: (a) convolve the image with 32 Gabor filters at 4 scales and 8 orientations, producing 32 feature maps of the same size of the input image; (b) divide each feature map into 16 regions (by a 4×4 grid), and then average the feature values within each region; and (c) concatenate the 16 averaged values of all 32 feature maps, resulting in a $16 \times 32 = 512$ -dimensional GIST descriptor. GIST summarizes the gradient information (scales and orientations) for different parts of an image, which provides a rough description of the scene.

We compare our approach with both batch-based hashing methods and online hashing methods. The batch-based hashing methods include two recent RS image hashing methods, Partial Randomness Hashing (PRH) [32] and Kernel Unsupervised Locality Sensitive Hashing (KULSH) [31], and four hashing approaches, Inductive Hashing on Manifolds (IMH) [23], Isotropic Hashing (IsoHash) [22], Iterative Quantization (ITQ) [20], and Spherical Hashing (SpH) [21], used in computer vision. The compared two online hashing methods are Online Kernel-based Hashing (OKH) [35] and Online Sketch Hashing (OSH) [36], which are used in the natural image processing literature, because our proposed approach is the first online hashing method for large-scale RS image retrieval. For the batch-based hashing methods, all the training images are used to learn the hash functions. For the online hashing methods, we randomly divide the whole training set into 1000 different chunks to simulate the online condition and the hash functions are updated in a streaming way.

To perform fair evaluations, we adopt the hamming ranking search commonly used in the literature. All the images in the database are ranked according to their hamming distance to the query and the desired neighbors are returned from the top of the ranked list. The retrieval performance is measured with average precision of the top K returned examples and the overall precision–recall curves. More specifically, precision and recall are defined as follows:

$$precision = \frac{true\ positive}{true\ positive + false\ positive} \quad (10)$$

$$recall = \frac{true\ positive}{true\ positive + false\ negative} \quad (11)$$

According to Equation (10), we get precision of the top K returned examples for a query image if the correctly predicted samples divided by K . The average precision is obtained by averaging the precision scores over all the test queries.

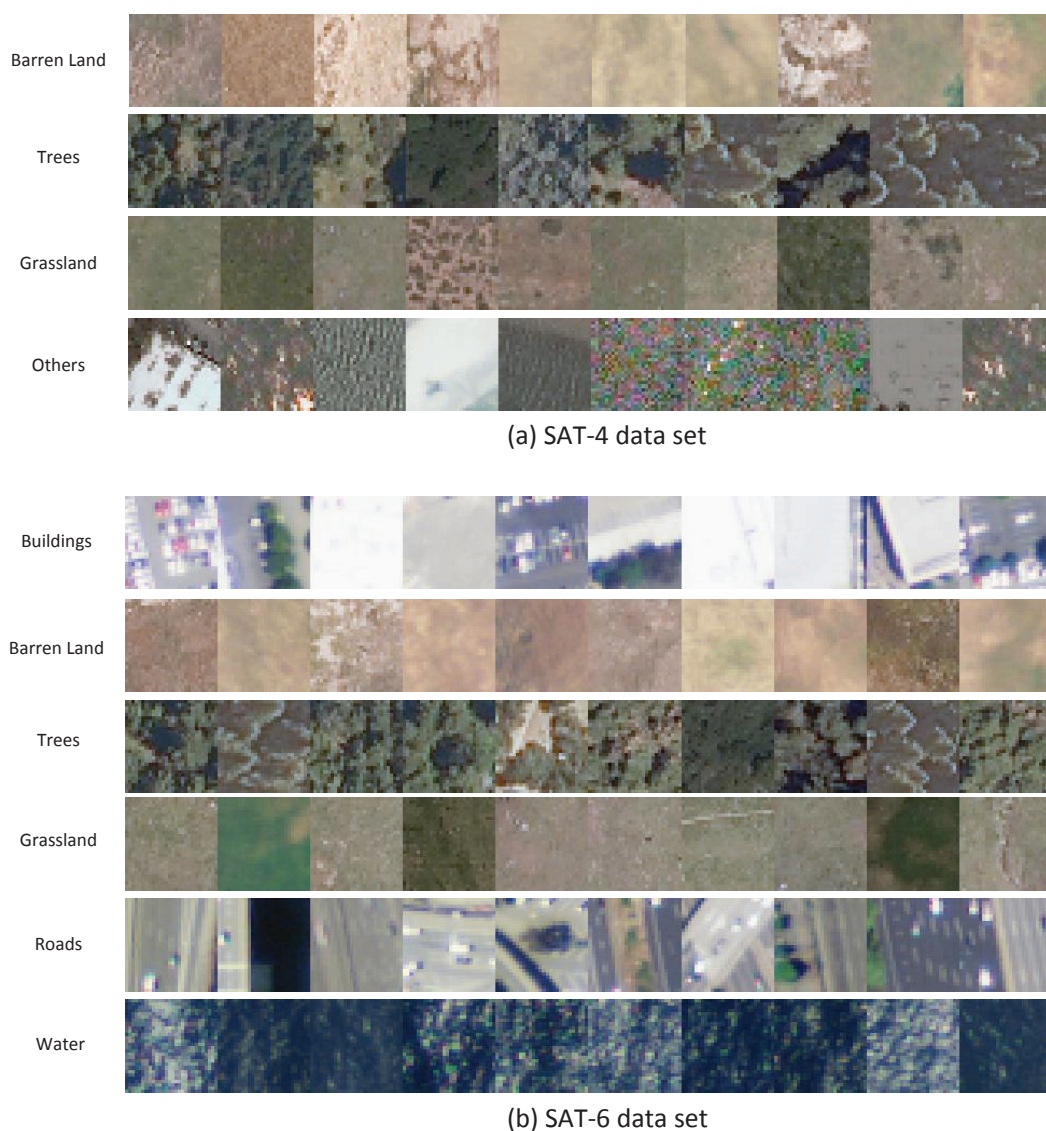


Figure 2. Some sample images from: (a) SAT-4 dataset; and (b) SAT-6 dataset.

3.2. Results and Analysis

Tables 1 and 2 show the average precision of the Top-10 and Top-100 retrieved image samples by different hashing methods on the two datasets. We can observe that the ITQ and PRH methods achieve relative better results among the batch-based hashing methods under varied hash bits. For the online hashing methods, the proposed OPRH achieves better results compared with the competitors in most cases. By comparing our OPRH method with the batch-based hashing methods, we can find that our OPRH obtains comparable performance to the batch methods on SAT-4 dataset while sometimes achieves even better results than all of the other compared approaches on SAT-6 dataset, which has indicated the effectiveness of the proposed online hashing method. The performance gain our OPRH approach may be attributed to the backward learning procedure, which helps learn more accurate projection parameter to enhance the representational ability of hash codes.

The average precision with respect to different retrieved samples and the precision-recall curves of compared hashing methods on the two datasets are shown in Figure 3. Since too many curves will be overlapped and hard to distinguish, we only keep the online hashing methods and the batch-based RS hashing methods in the figure. In Figure 3a–c,g–i, we can observe that our OPRH method consistently outperforms OSH and OKH methods when the retrieved images increase and the improvements are

more notable for long code length. The reason may be that OSH and OKH have large quantization error when generating binary codes while our OPRH can reduce the error in code binarization through the backward decoder learning procedure. Precision–recall curve reflects the overall image retrieval performance of different hashing approaches. In Figure 3d–f,j–l, we also find that our OPRH method achieves the best results among the compared online hashing methods. The proposed OPRH method has comparable overall performance to batch-based PRH method and much better than KULSH approach on the two datasets.

Table 1. The comparison of mean precision of the top K returned examples for different methods on the SAT-4 dataset with varied hash bits.

Methods	Top-10			Top-100		
	32-bits	48-bits	64-bits	32-bits	48-bits	64-bits
IMH	0.560	0.538	0.548	0.550	0.524	0.541
IsoHash	0.606	0.640	0.655	0.576	0.594	0.597
ITQ	0.636	0.653	0.662	0.609	0.607	0.610
SpH	0.596	0.623	0.658	0.563	0.588	0.607
KULSH	0.492	0.507	0.553	0.476	0.479	0.526
PRH	0.607	0.621	0.665	0.592	0.595	0.622
OKH	0.439	0.516	0.600	0.418	0.480	0.561
OSH	0.603	0.637	0.647	0.568	0.596	0.596
OPRH	0.608	0.630	0.656	0.598	0.594	0.616

Table 2. The comparison of mean precision of the top K returned examples for different methods on the SAT-6 dataset with varied hash bits.

Methods	Top-10			Top-100		
	32-bits	48-bits	64-bits	32-bits	48-bits	64-bits
IMH	0.583	0.626	0.604	0.575	0.614	0.582
IsoHash	0.667	0.680	0.673	0.635	0.645	0.642
ITQ	0.672	0.691	0.681	0.649	0.660	0.653
SpH	0.642	0.664	0.694	0.616	0.631	0.657
KULSH	0.413	0.459	0.452	0.418	0.496	0.520
PRH	0.651	0.682	0.683	0.629	0.658	0.652
OKH	0.541	0.619	0.638	0.521	0.592	0.617
OSH	0.669	0.684	0.680	0.639	0.650	0.647
OPRH	0.645	0.699	0.705	0.631	0.672	0.677

To explicitly compare the online hash function updating process at each round for the online hashing methods, we compute the the average retrieval precision of different methods after each round and show it in Figure 4. It is obvious that the proposed OPRH method outperforms both OKH and OSH approaches on the two datasets. Moreover, OKH has big fluctuations during the online learning process and its performance even deteriorates as the number of received chunks increases on SAT-4 dataset, while our proposed OPRH achieves quite stable improvement when more and more new image chunks are available for training. To show the online updating process of our approach more intuitively, we give an visual example for image retrieval in Figure 5, which shows the first returned 16 samples to the query image by our method after different learning rounds. We can see that the retrieval results become more and more accurate as the learning round increases. The reason is that the learned hash functions improve continuously as new training images are obtained and thus the generated hash codes also become more accurate. This also indicates that our proposed online hashing method can fit the new available streaming data very well, which is the shortcoming of batch-based hashing methods in contrary.

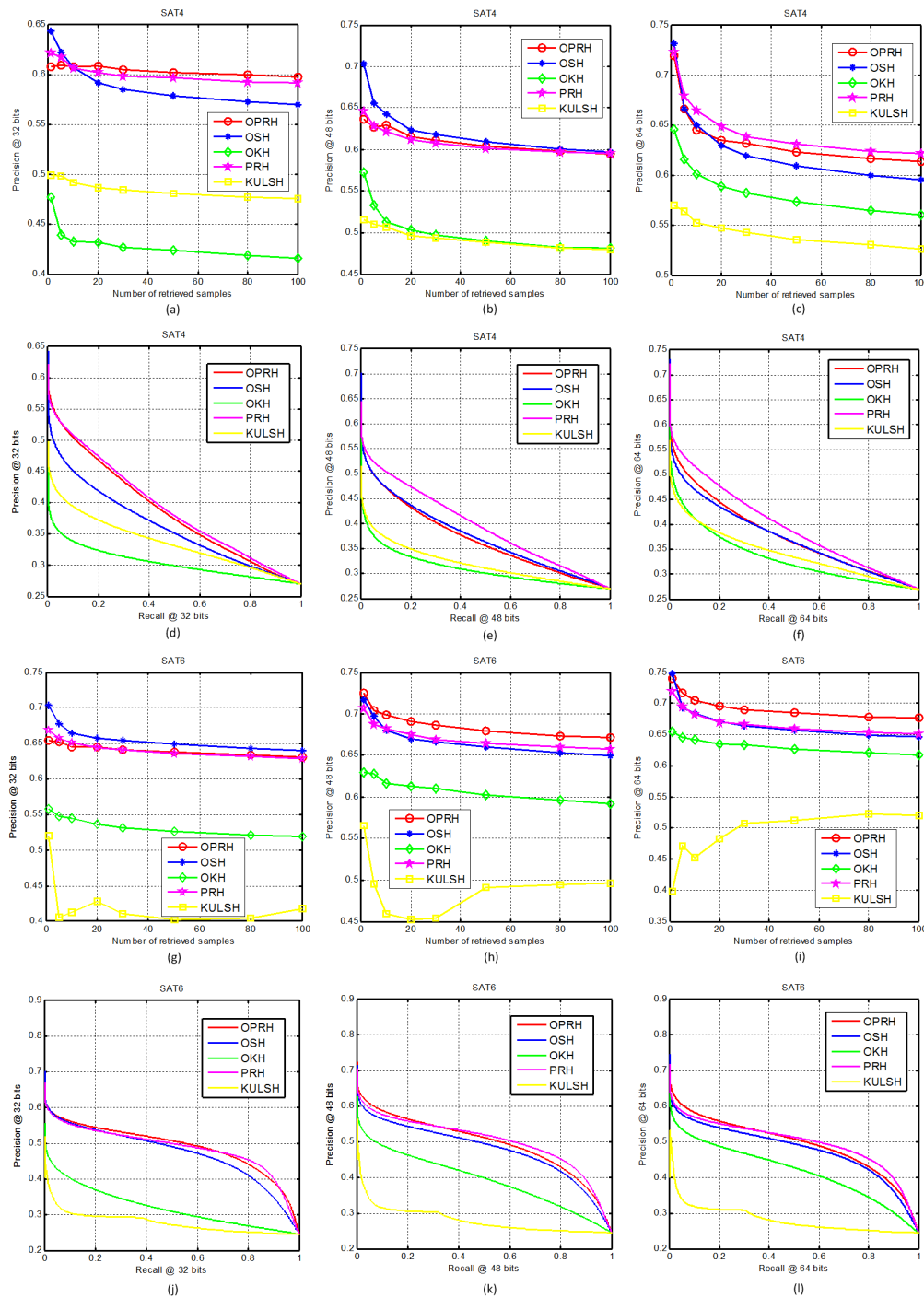


Figure 3. The average precision with respect to different retrieved samples and precision-recall curves for the compared methods on the two datasets: (a–f) SAT-4; and (g–i) SAT-6.

We also compare the learning efficiency of different hashing methods, which is shown in Table 3. All experiments are implemented with MATLAB code and run on a PC with Intel Core-i5 2.3 GHz CPU, 8 GB RAM. For the batch-based hashing methods, we report their total time on the whole training image set and for the online hashing methods, we show both their average updating time at each round and the accumulated time of total rounds. Among the batch-based methods, PRH and IsoHash are much more efficient than other methods. Among the online hashing methods, our OPRH approach has the fastest updating time at each round and more than 10 times faster than the compared OSH method. The accumulated time of total 1000 rounds of our OPRH is still comparable to the PRH method. For memory cost, the online hashing approaches are much lower than the batch-based

hashing methods. This is easy to explain because the online hashing methods only have to handle a small image chunk at each learning round while the batch-based methods have to load all the images into the memory for training. More specifically, the PRH algorithm occupies about 1.2 GB RAM to store the data and parameters in the learning process on SAT-6 dataset with 64-bits in our experiments while only 1.8 MB RAM is needed for our OPRH method. SAT-6 dataset only has 405,000 images. Imagine that, if we are given a RS image dataset consisting of a million or billion images, which is impossible to be loaded into the memory for training, the batch-based hashing methods would not work. However, our proposed online hashing method is still able to do hash function learning by segmenting the whole database into many small chunks. Therefore, the proposed OPRH method is quite suitable for hash code learning and fast image retrieval on oversized RS image sets, which is expected in real-world applications.

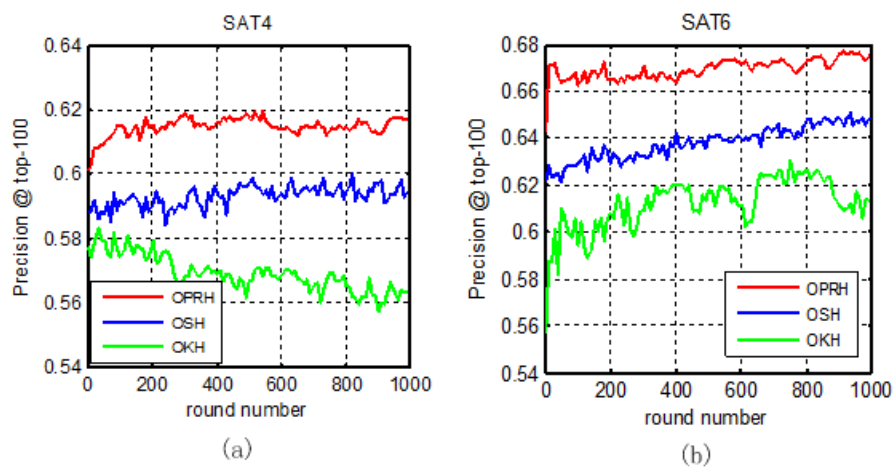


Figure 4. Comparison of average precision at each round of the online hashing methods on: (a) SAT-4 dataset; and (b) SAT-6 dataset (64-bits).

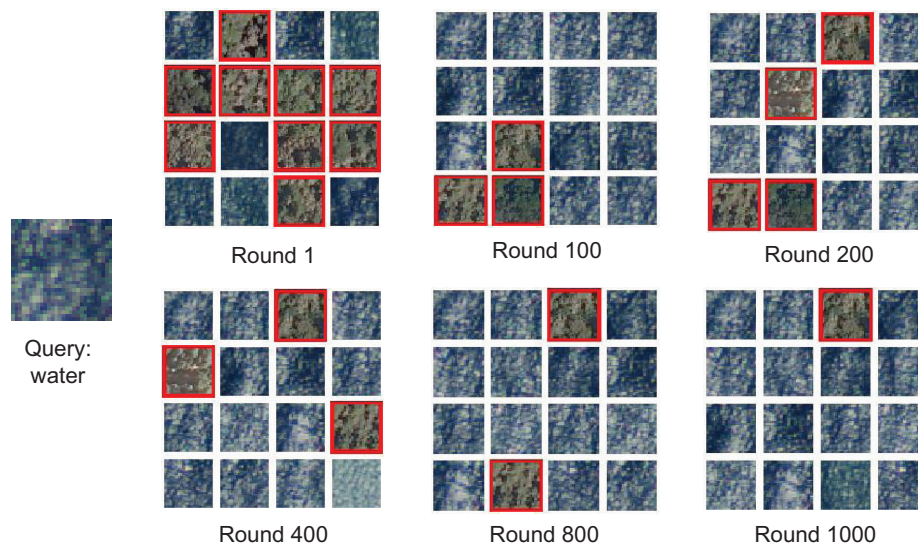


Figure 5. Visualized retrieval example after different rounds by our OPRH method on SAT-6 dataset with 64-bits. Top-16 returned image patches for the query are shown for each round and the false positives are annotated with a red rectangle.

Table 3. The comparison of training time (in seconds) and memory cost (MB) for different kinds of hashing methods.

Methods	SAT-4 Dataset			SAT-6 Dataset		
	Round Time	Total Time	Memory Cost	Round Time	Total Time	Memory Cost
IMH	-	67.6	3696	-	67.7	2990
IsoHash	-	5.5	4915	-	5.8	3942
ITQ	-	47.9	5857	-	61.1	5529
SpH	-	196.3	5109	-	200	4177
KULSH	-	10.3	3901	-	8.2	3143
PRH	-	4.6	1556	-	5.0	1198
OKH	0.32	315.8	10.4	0.27	267	8
OSH	0.11	113.5	4.4	0.11	105.4	3.5
OPRH	0.01	12	2.3	0.009	8.7	1.8

To evaluate the large-scale RS image retrieval performance of our proposed hashing approach and direct linear search strategy, we conduct image retrieval experiments with our OPRH method and ℓ_2 linear scan. For our OPRH method, image retrieval is carried out with learned binary codes in the hamming space. ℓ_2 linear scan directly does image retrieval in the original feature space based on the Euclidean distance of feature vectors. Besides the GIST descriptor used in the previous experiments, CNN feature is also adopted to evaluate the generalizing ability of our OPRH method. We choose AlexNet as the CNN feature extraction model and the output 4096-dimensional feature of the fully connected layer fc7 is extracted for each image. PCA is applied to reduce the dimensionality to 1024 and form the final feature vector for the images. The comparison of average search time per image and mean precision of Top-100 retrieved samples is shown in Table 4. From the results, we can find that, when using CNN feature instead of GIST descriptor, the average retrieval precision can be improved by 30–40% on the two datasets. This is attributed to the powerful representation ability of CNN feature, which is able to learn more high-level semantic information. For different image search strategies, the direct search in the original image feature space can obtain higher accuracy than hashing-based search methods in most cases. However, by sacrificing a little accuracy, the hashing approaches can obtain much faster search speed than the traditional direct search method. For example, compared with direct search in the CNN feature space, OPRH + CNN achieves more than 60 times speed acceleration with only 1% drop in the retrieval accuracy on the SAT-6 dataset. The reason is that our OPRH approach conducts image retrieval based on binary codes and the hamming distance between different codes can be efficiently calculated with XOR operation, which is much more faster than the computation of Euclidean distance in the feature space.

Table 4. The comparison of average search time (in seconds) and accuracy (mean precision of Top-100 retrieved samples) between our proposed hashing method in the hamming space (with 64-bits) and ℓ_2 linear scan in the original feature space based on different feature representations.

	GIST ℓ_2 Scan		CNN ℓ_2 Scan		OPRH+GIST		OPRH+CNN	
	Time	Precision@100	Time	Precision@100	Time	Precision@100	Time	Precision@100
SAT-4	1.93	0.60	4.01	1	0.06	0.61	0.06	0.98
SAT-6	1.67	0.69	3.15	0.98	0.05	0.67	0.05	0.97

Finally, to demonstrate the superiority of the proposed hashing approach for real-world large-scale remote sensing image retrieval, we generate a synthetic dataset consisting of 100 million samples of 1000 dimension. Due to the size of the synthetic dataset, it exceeds the processing ability of traditional batch-based hashing approaches and brute force linear search schemes. However, by dividing the dataset into one million small chunks, our OPRH hashing approach only has to handle 100 samples

at each round and can finish hash model training in 33 min on our ordinary PC. With learned binary codes of 64-bits, fast image retrieval from 100 million samples can be carried out at the speed of 5 s per image. These results demonstrate that the proposed OPRH is scalable to massive streaming remote sensing data even on a common computer.

4. Conclusions

In this paper, we have proposed a novel online hashing method, named online partial randomness hashing (OPRH), for retrieving scalable remote sensing image databases. Benefiting from the online learning scheme, the hash model parameters can be updated continuously according to the streaming image data, which is a common scenario in the real-world applications. Therefore, the hash codes learned by our approach have better generalization ability compared with the batch-based hashing approaches. More importantly, the batch-based hashing methods will face difficulties when handling very large database due to the high complexity and space limitation while the proposed method can be easily applied to oversized dataset by dividing it into several small chunks. Thus, our OPRH method is very suitable for large-scale remote sensing image retrieval. Extensive experiments on two public large-scale satellite datasets have demonstrated the effectiveness and efficiency of our approach.

Our proposed online hashing method can be used in many real-time remote sensing applications due to its adapting ability to variations in datasets as they grow and diversify. For example, on-orbit processing of satellite remote sensing images can be conducted through our online hashing method to improve the efficiency of information processing. Real-time retrieval from huge historical satellite cloud pictures with our proposed approach can provide the forecaster more effective information for short-term weather forecasting. At the same time, there are also some challenging issues that need to be addressed for our proposed online hashing approach. The hash functions of our approach are updated gradually according to the changing database, but the updating frequency needs to be well decided in real-world applications. Too high frequency is time-consuming while low updating frequency may lead to unsatisfactory retrieval results. In addition, the hash code indexing must also be frequently updated when hash functions change. This may cause inefficiencies in the real-world systems. Therefore, solutions must be simultaneously developed to alleviate this particular problem in future work.

Author Contributions: P.L. and P.R. conceived and designed the experiments; X.Zhang performed the experiments; X.Zhu analyzed the data; and P.L. wrote the paper. All authors read and approved the final manuscript.

Acknowledgments: This work was supported in part by the National Natural Science Foundation of China (Grant 61602517 and 61501475), the National Key R&D Program of China (Grant SQ2017YFB140187), Qingdao Applied Fundamental Research (Grant 16-5-1-11-jch), the Open Project Program of the State Key Laboratory of Mathematical Engineering and Advanced Computing (Grant 2017A05), the Open Project Program of National Laboratory of Pattern Recognition (Grant 201800018), and the Fundamental Research Funds for Central Universities (Grant 18CX02110A).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Du, R.; Chen, Y.; Tang, H.; Fang, T. Study on content-based remote sensing image retrieval. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Seoul, South Korea, 25–29 July 2005; pp. 707–710.
2. Wang, Q.; Zhu, G.; Yuan, Y. Statistical quantization for similarity search. *Comput. Vis. Image Underst.* **2014**, *124*, 22–30. [[CrossRef](#)]
3. Yang, J.; Liu, J.; Dai, Q. An improved Bag-of-Words framework for remote sensing image retrieval in large-scale image databases. *Int. J. Digit. Earth* **2015**, *8*, 273–292. [[CrossRef](#)]
4. Sevilla, J.; Bernabe, S.; Plaza, A. Unmixing-based content retrieval system for remotely sensed hyperspectral imagery on GPUs. *J. Supercomput.* **2014**, *70*, 588–599. [[CrossRef](#)]

5. Yang, Y.; Newsam, S. Geographic image retrieval using local invariant features. *IEEE Trans. Geosci. Remote Sens.* **2013**, *51*, 818–832. [[CrossRef](#)]
6. Aptoula, E. Remote sensing image retrieval with global morphological texture descriptors. *IEEE Trans. Geosci. Remote Sens.* **2014**, *52*, 3023–3034. [[CrossRef](#)]
7. Newsam, S.; Wang, L.; Bhagavathy, S.; Manjunath, B.S. Using texture to analyze and manage large collections of remote sensed image and video data. *Appl. Opt.* **2004**, *43*, 210–217. [[CrossRef](#)] [[PubMed](#)]
8. Luo, B.; Aujol, J.F.; Gousseau, Y.; Ladjal, S. Indexing of satellite images with different resolutions by wavelet features. *IEEE Trans. Image Process.* **2008**, *17*, 1465–1472. [[PubMed](#)]
9. Rosu, R.; Donias, M.; Bombrun, L.; Said, S.; Regniers, O.; Da Costa, J.-P. Structure tensor Riemannian statistical models for CBIR and classification of remote sensing images. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 248–260. [[CrossRef](#)]
10. Zhou, W.; Shao, Z.; Diao, C.; Cheng, Q. High-resolution remotesensing imagery retrieval using sparse features by auto-encoder. *Remote Sens. Lett.* **2015**, *6*, 775–783. [[CrossRef](#)]
11. Zhou, W.; Newsam, S.; Li, C.; Shao, Z. Learning low dimensional convolutional neural networks for high-resolution remote sensing image retrieval. *Remote Sens.* **2017**, *9*, 489. [[CrossRef](#)]
12. Du, Z.; Li, X.; Lu, X. Local structure learning in high resolution remote sensing image retrieval. *Neurocomputing* **2016**, *207*, 813–822. [[CrossRef](#)]
13. Wang, Q.; Wan, J.; Yuan, Y. Deep metric learning for crowdedness regression. *IEEE Trans. Trans. Circuits Syst.* **2017**. [[CrossRef](#)]
14. Li, Y.; Zhang, Y.; Tao, C.; Zhu, H. Content-based high-resolution remote sensing image retrieval via unsupervised feature learning and collaborative affinity metric fusion. *Remote Sens.* **2016**, *8*, 709. [[CrossRef](#)]
15. Wang, Y.; Zhang, L.; Tong, X.; Zhang, L.; Zhang, Z.; Liu, H.; Xing, X.; Takis Mathiopoulos, P. A three-layered graph-based learning approach for remote sensing image retrieval. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 6020–6034. [[CrossRef](#)]
16. Andoni, A.; Indyk, P. 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), Berkeley, CA, USA, 21–24 October 2006; pp. 459–468.
17. Datar, M.; Immorlica, N.; Indyk, P.; Mirrokni, V. Locality-sensitive hashing scheme based on p-stable distributions. In Proceedings of the 20th Annual Symposium on Computational Geometry (SCG), Brooklyn, NY, USA, 8–11 June 2004; pp. 253–262.
18. Kulis, B.; Grauman, K. Kernelized locality-sensitive hashing for scalable image search. In Proceedings of the IEEE 12th International Conference on Computer Vision (ICCV), Kyoto, Japan, 27 September–4 October 2009; pp. 2130–2137.
19. Weiss, Y.; Torralba, A.B.; Fergus, R. Spectral hashing. In Proceedings of the Advances in Neural Information Processing Systems (NIPS), Vancouver, BC, Canada, 8–11 December 2008; pp. 1753–1760.
20. Gong, Y.; Lazebnik, S. Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 2916–2929. [[CrossRef](#)] [[PubMed](#)]
21. Heo, J.; Lee, Y.; He, J.; Chang, S.; Yoon, S. Spherical hashing: binary code embedding with hyperspheres. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *37*, 2304–2316. [[CrossRef](#)] [[PubMed](#)]
22. Kong, W.; Li, W. Isotropic hashing. In Proceedings of the Advances in Neural Information Processing Systems (NIPS), Lake Tahoe, NV, USA, 3–6 December 2012; pp. 1655–1663.
23. Shen, F.; Shen, C.; Shi, Q.; Hengel, A.; Tang, Z. Inductive hashing on manifolds. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Portland, OR, USA, 23–28 June 2013; pp. 1562–1569.
24. Norouzi, M.; Fleet, D.J. Minimal loss hashing for compact binary codes. In Proceedings of the 28th International Conference on Machine Learning (ICML), Bellevue, WA, USA, 28 June–2 July 2011; pp. 353–360.
25. Liu, W.; Wang, J.; Ji, R.; Jiang, Y.; Chang, S. Supervised hashing with kernels. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Providence, RI, USA, 16–21 June 2012; pp. 2074–2081.
26. Lin, G.; Shen, C.; Shi, Q.; Hengel, A.; Suter, D. Fast supervised hashing with decision trees for high-dimensional data. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Columbus, OH, USA, 23–28 June 2014; pp. 1971–1978.

27. Shen, F.; Shen, C.; Liu, W.; Shen, H. Supervised discrete hashing. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 37–45.
28. Xia, R.; Pan, Y.; Lai, H.; Liu, C.; Yan, S. Supervised hashing via image representation learning. In Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence (AAAI), Quebec City, QC, Canada, 27–31 July 2014; pp. 2156–2162.
29. Zhao, F.; Huang, Y.; Wang, L.; Tan, T. Deep semantic ranking based hashing for multi-label image retrieval. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 1556–1564.
30. Li, W.; Wang, S.; Kang, W. Feature learning based deep supervised hashing with pairwise labels. In Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence (IJCAI), New York, NY, USA, 7–15 July 2016; pp. 1711–1717.
31. Demir, B.; Bruzzone, L. Hashing-based scalable remote sensing image search and retrieval in large archives. *IEEE Trans. Geosci. Remote Sens.* **2016**, *54*, 892–904. [[CrossRef](#)]
32. Li, P.; Ren, P. Partial randomness hashing for large-scale remote sensing image retrieval. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 464–468. [[CrossRef](#)]
33. Li, Y.; Zhang, Y.; Huang, X.; Zhu, H.; Ma, J. Large-scale remote sensing image retrieval by deep hashing neural networks. *IEEE Trans. Geosci. Remote Sens.* **2018**, *56*, 950–965. [[CrossRef](#)]
34. Ye, D.; Li, Y.; Tao, C.; Xie, X.; Wang, X. Multiple feature hashing learning for large-scale remote sensing image retrieval. *ISPRS Int. J. Geo-Inform.* **2017**, *6*, 364. [[CrossRef](#)]
35. Huang, L.; Yang, Q.; Zheng, W. Online hashing. In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), Beijing, China, 3–9 August 2013; pp. 1422–1428.
36. Leng, C.; Wu, J.; Cheng, J.; Bai, X.; Lu, H. Online sketching hashing. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 7–12 June 2015; pp. 2503–2511.
37. Liang, N.; Huang, G.; Saratchandran, P.; Sundararajan, N. A fast and accurate online sequential learning algorithm for feedforward networks. *IEEE Trans. Neural Netw.* **2006**, *17*, 1411–1423. [[CrossRef](#)] [[PubMed](#)]
38. Huang, G.; Zhu, Q.; Siew, C. Extreme learning machine: Theory and applications. *Neurocomputing* **2006**, *70*, 489–501. [[CrossRef](#)]
39. Basu, S.; Ganguly, S.; Mukhopadhyay, S.; Dibiano, R.; Karki, M.; Nemani, R. DeepSat: A learning framework for satellite imagery. In Proceedings of the SIGSPATIAL International Conference on Advances in Geographic Information Systems (SIGSPATIAL/GIS), Bellevue, WA, USA, 3–6 November 2015; pp. 37:1–37:10.
40. Oliva, A.; Torralba, A. Modeling the shape of the sence: A holistic representation of spatial envelope. *Int. J. Comput. Vis.* **2001**, *42*, 145–175. [[CrossRef](#)]



© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).