



Article

DMCH: A Deep Metric and Category-Level Semantic Hashing Network for Retrieval in Remote Sensing

Haiyan Huang ¹, Qimin Cheng ^{2,*}, Zhenfeng Shao ¹, Xiao Huang ³ and Liyuan Shao ²¹ State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, Wuhan 430079, China; huanghaiyan@whu.edu.cn (H.H.); shaozhenfeng@whu.edu.cn (Z.S.)² School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan 430074, China; m201977098@hust.edu.cn³ Department of Environmental Sciences, Emory University, Atlanta, GA 30322, USA; xiao.huang2@emory.edu

* Correspondence: chengqm@hust.edu.cn; Tel.: +86-027-87792516

Abstract: The effectiveness of hashing methods in big data retrieval has been proved due to their merit in computational and storage efficiency. Recently, encouraged by the strong discriminant capability of deep learning in image representation, various deep hashing methodologies have emerged to enhance retrieval performance. However, maintaining the semantic richness inherent in remote sensing images (RSIs), characterized by their scene intricacy and category diversity, remains a significant challenge. In response to this challenge, we propose a novel two-stage deep metric and category-level semantic hashing network termed DMCH. First, it introduces a novel triple-selection strategy during the semantic metric learning process to optimize the utilization of triple-label information. Moreover, it inserts a hidden layer to enhance the latent correlation between similar hash codes via a designed category-level classification loss. In addition, it employs additional constraints to keep bit-uncorrelation and bit-balance of generated hash codes. Furthermore, a progressive coarse-to-fine hash code sorting scheme is used for superior fine-grained retrieval and more effective hash function learning. Experiment results on three datasets illustrate the effectiveness and superiority of the proposed method.



Citation: Huang, H.; Cheng, Q.; Shao, Z.; Huang, X.; Shao, L. DMCH: A Deep Metric and Category-Level Semantic Hashing Network for Retrieval in Remote Sensing. *Remote Sens.* **2024**, *16*, 90. <https://doi.org/10.3390/rs16010090>

Academic Editors: Pedram Ghamisi, Benoit Vozel, Ce Zhang, Danfeng Hong and Qiqi Zhu

Received: 28 September 2023

Revised: 15 December 2023

Accepted: 21 December 2023

Published: 25 December 2023



Copyright: © 2023 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 (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: deep hash learning; category-level semantics; remote sensing image retrieval

1. Introduction

The extraordinary advancements in satellite technology mark the advent of the era of big data in remote sensing, posing unprecedented challenges to traditional data mining technologies [1]. The analysis and utilization of big data play an important role in hazard alerts, emergency response, and so on [2,3]. A notable increase in remote sensing archives has amplified the demand for content-based image retrieval (CBIR) within the field of remote sensing [4–6]. A typical CBIR system comprises two key components: visual feature extraction and similarity measurement. However, the rapid and convenient availability of high-resolution remote sensing images (RSIs) has substantially increased the dimensionality of visual feature vectors. Consequently, there is a pressing necessity to develop strategies for reducing storage space and computational complexity associated with high-dimensional representations.

An approximate nearest neighbor (ANN) search [7] has been proposed to decrease computational complexity resulting from the exhaustive similarity of calculations of high-dimensional feature vectors in traditional nearest neighbor searches. Hashing methods, as a type of classical ANN technique, have received wide attention for their effectiveness in CBIR due to their efficiency in feature compacting. Hashing-based ANN transforms high-dimensional data into concise binary hash codes, where the similarity between image features is approximated by calculating the Hamming distance between binary codes. Traditional hashing methods [8–10] employ hand-crafted feature descriptors as input for

hash learning to generate low-dimensional binary hash codes. These methods can be categorized into unsupervised [11–13], semi-supervised [14,15], and supervised [16–18].

The significant strides made in deep learning technology [19] have spurred the emergence of deep hashing methods. Over the past decades, a variety of deep hashing methods [20–24] have been introduced, demonstrating satisfactory performance in large-scale natural image retrieval. These methods commonly employ label information, in either pair or triplet forms, as supervision information to enhance the performance of hashing networks. For more comprehensive insights into methods utilizing pairwise and triplet label information, readers are referred to [25–28] and [29–31], respectively. In addition to effective hash function learning, hash code ranking is an essential but often overlooked aspect that significantly impacts the retrieval performance of hashing methods. It determines how to rank similar images that have the same Hamming distance from the query image. Current ranking schemes primarily include those based on weighted Hamming distance [32,33] and asymmetric distances [34,35]. Despite notable progress in hashing-based retrieval for remote sensing images (RSIs), the preservation of semantic similarity during hash learning remains challenging.

To address this problem, we introduce a novel two-stage deep metric and category-level semantic hashing network, named DMCH, for the retrieval of remote sensing images. In the first stage of feature extraction, a pre-trained deep neural network (DNN) is utilized to extract image features as an intermediate representation. In hash learning, a triplet selection strategy is integrated to facilitate the learning of a Hamming metric space, ensuring superior preservation of semantics. Then, a hidden layer is introduced to enhance the category-level semantics of hash codes. Additional constraints are proposed to the overall objective function to enhance the conciseness and discriminative power of hash codes, thereby optimizing retrieval performance. Furthermore, a hash code ranking scheme is employed to discriminate between the similarities of images possessing identical hash codes by evaluating the more detailed real-valued feature vectors prior to relaxation. Lastly, the efficacy and superior performance of the DMCH method are evaluated through extensive experiments on three publicly available benchmark datasets.

The main contributions of the proposed DMCH are as follows:

- (1) We propose a deep metric and category-level semantic hashing network for remote sensing image retrieval. Our quantitative and qualitative experimental results conclusively demonstrate that the DMCH surpasses contemporary deep hashing retrieval methodologies in performance.
- (2) We propose a semantic-preserving loss function designed to enhance the conciseness and discriminatory capability of the generated hash codes and facilitate the extraction of distinct features from RSIs.

Comparative experiments have been conducted on three public datasets with classic and recent hashing methods in remote sensing image retrieval, including PRH [12], KLSH [11], DPSH [23], MiLaN [29], FAH [36], AHCL [37], and DSSH [38] to demonstrate the effectiveness of our DMCH model.

The structure of this paper is organized as follows: Section 2 reviews traditional and deep hashing methods for retrieval tasks, respectively. Section 3 presents the proposed DMCH in detail. Experimental results and their implications are discussed in Section 4. Finally, Section 5 concludes the paper.

2. Related Work

2.1. Traditional Hash Learning Methods

Traditional hash learning methods typically rely on hand-crafted image descriptor features for image representation. Demir et al. [11] proposed two kernel-based nonlinear hashing methods: the unsupervised kernel-based method (KULSH) and the supervised kernel-based method (KLSH). Compared with the earlier KNN-based and classifier-based methods, both KULSH and KLSH significantly reduced retrieval time and storage space while the retrieval accuracy was slightly reduced. Li et al. [12] proposed the partial random

hash (PRH), a method consisting of two steps. Firstly, a random projection was applied to map the original RSI into Hamming space, and then a transformation weight matrix was learned to obtain binary codes. Reato et al. [13] proposed using the original cluster sensitive multi-hash codes for RS image representation and designed a descriptor of the original sensitive cluster to generate multi-hash codes and subsequently map these codes into a common space for measuring distance measurement between RSIs. In addition to the unsupervised hash methods mentioned above, there are a small number of semi-supervised hash learning methods. For example, Wang et al. [14] proposed a semi-supervised hash technology leveraging labeled data to explore the semantic similarity between RSIs while using unlabeled data during training to prevent overfitting and enhance framework robustness. Kim et al. [15] adopted the concept of linear discriminant analysis and proposed a semi-supervised discriminant hashing method. Their approach used labeled data for model training to maximize the correlation of binary codes and regularization to prevent overfitting.

Several supervised hash learning methods have been proposed to effectively leverage label information, thereby enhancing retrieval performance. Liu et al. [16] proposed a kernel-based supervised hashing (KSH) method to map the original image features into a compact binary hashing code via existing annotation information. Through training, KSH narrowed the Hamming distance between similar images and increased the Hamming distance between dissimilar images. Norouzi et al. [17] proposed a minimal loss hashing (MLH) method, which utilizes label information to establish similar relationships between input RSIs and constructed a hinge loss function for model training. Furthermore, Shen et al. [18] proposed a novel supervised discrete hashing method that directly optimizes bit-by-bit hash codes, departing from traditional relaxation methods, to enhance the discrimination of hash codes.

2.2. Deep Hash Learning Methods

The emergence of deep hash learning methods is inspired by the strong feature learning capabilities of DNNs. It has a wide range of applications in various fields, such as medical image retrieval [39–41]. In the RSI field, Liu et al. [20] proposed a supervised deep hash retrieval model leveraging generative adversarial networks (GAN), termed GAN-assisted hashing. This approach includes defining a new loss function to obtain more discriminative binary hash codes. Moreover, Li et al. [21] proposed a lightweight quantization depth hashing framework, incorporating a class-intensive paired loss function within the hashing layer to mitigate the challenges posed by data imbalance.

Supervised methods leverage semantic label information, whether in pair or triplet forms, during the network model training to further improve performance. In recent years, various deep-supervised hashing methods have emerged, yielding commendable retrieval performance. In 2014, Xia et al. [22] proposed a staged convolutional neural network (CNN) hashing method. This approach first decomposed the pairwise similarity matrix to obtain approximate hash codes and then used the learned hash codes as supervision while learning image features and hash functions simultaneously. Based on this work, Li et al. [23] subsequently proposed a deep pairwise supervised hashing (DPSH) method. In DPSH, a deep neural network is employed to automatically learn deep features and hash functions under the supervision of the semantics of the images. The end-to-end architecture enabled the various parts of the network to give feedback to each other to better guide the learning process of the hashing code. Lin et al. [42] proposed a new deep hashing method, named deep learning of binary hash codes, which built a hash function in the hidden layer, enhancing potential correlation between similar hashing codes without additional artificial similarity matrix. Moreover, Zhao et al. [43] proposed a deep semantic sorting hashing method that used multi-level semantic similarity information among multi-label images to learn the hash function.

In the RS field, Chen et al. [25] proposed a deep hashing method leveraging pairwise label information. This method efficiently mapped high-dimensional image features into binary hash codes, which had high time efficiency and accurate retrieval capabilities.

Similarly, Li et al. [26] proposed an end-to-end deep hashing network that used paired annotation information to learn hash codes and hash functions. Their method achieved impressive performance on large-scale RSI datasets. Han et al. [27] proposed a Heaviside function-like deep residual hashing network to binarize the input RSIs to obtain the corresponding hash codes, employing the powerful feature expression capabilities of deep residual networks to establish associations between RSIs. Song et al. [28] introduced a deep hashing method for RSIR, and they utilized a dual-stream network, allowing simultaneous optimization for image retrieval and classification, thereby enhancing retrieval performance significantly. On this basis, they also designed a new loss function to measure the similarity loss between pairs of RSIs.

In addition to the deep hash learning methods based on paired label information, hash learning methods based on triple have been proposed. Roy et al. [29] proposed a metric learning-based deep hashing network to address the scarcity of labeled RSIs. This method first used the Inception network pre-trained on ImageNet for deep features extraction and then combined three different loss functions for hashing network training to generate corresponding binary hash codes. Chen et al. [30] used triple supervision information to learn deep hash codes effectively and minimize the loss of accuracy caused by the binarization process. Meanwhile, Shan et al. [31] proposed a new proxy metric learning network that integrated hash learning and proxy-based metric learning. Their method included a new hash loss function to reduce the quantization loss.

In recent studies, researchers have applied novel unsupervised learning, meta-learning, asymmetric methods, and cross-modal to remote sensing image hash retrieval. For example, Fernandez-Beltran et al. [44] introduced a novel unsupervised hashing method that takes advantage of the generative nature of probabilistic topic models to encapsulate the hidden semantic patterns of the data into the final binary representation. Tang et al. [45] proposed a new supervised hash learning method based on meta-learning, which could achieve well-retrieval performance with a few labeled training samples. Song et al. [37] proposed an asymmetric way to solve the time-consuming problem of generating the hash codes of large-scale database images. Chen et al. [46] proposed a novel quadruple-based hashing network to learn relative semantic similarity relationships of hash codes in cross-modal remote sensing image–sound retrieval.

Compared with traditional methods, deep hash learning methods have become the mainstream trend due to the multi-layer nonlinear ability of the convolutional neural network in image representation.

3. A Deep Metric and Category-Level Semantic Hashing Network

In this section, we introduce the basic idea of deep hashing-based remote sensing image retrieval in Section 3.1. Then, we describe our retrieval model's system architecture and retrieval process in detail in Section 3.2, while we introduce the constructions of triplets in Section 3.3. We focus on the design of the loss function in Section 3.4. In Section 3.5, we will introduce a coarse-to-fine hash code ranking scheme in remote sensing image retrieval.

3.1. The Basic Idea of Deep Hashing-Based Remote Sensing Image Retrieval

Suppose that $I = \{I_i\}_{i=1}^n$ represents RSIs, where each image I_i corresponds to a C-dimensional category label vector $y_i \in \{0, 1\}^C$, and C represents the amount of categories of the datasets. Hash learning aims to learn a series of nonlinear hash functions $f: I \rightarrow \{0, 1\}^K$, which can satisfy the requirements for semantics-preserving in the mapping process, and K means the length of mapped hash codes.

3.2. System Architecture of Deep Hashing Model

As shown in Figure 1, the proposed deep metric and category-level hash network DMCH consists of two main components: a deep feature extraction network and a hashing network. The pipeline can be described as follows: the input images are first fed into a pre-trained Inception-v3 network, and the feature vectors extracted in the previous layer of

the classification output layer are used as intermediate representations. Then, information triples are strategically selected as inputs to the hashing network. The hashing network consists of two fully connected layers ($f_1 - f_2$), the hidden layer (H) and the classification output layer (output), and the triples, category-level classification loss and ideal hash code constraints are designed in order to maintain semantic similarity.

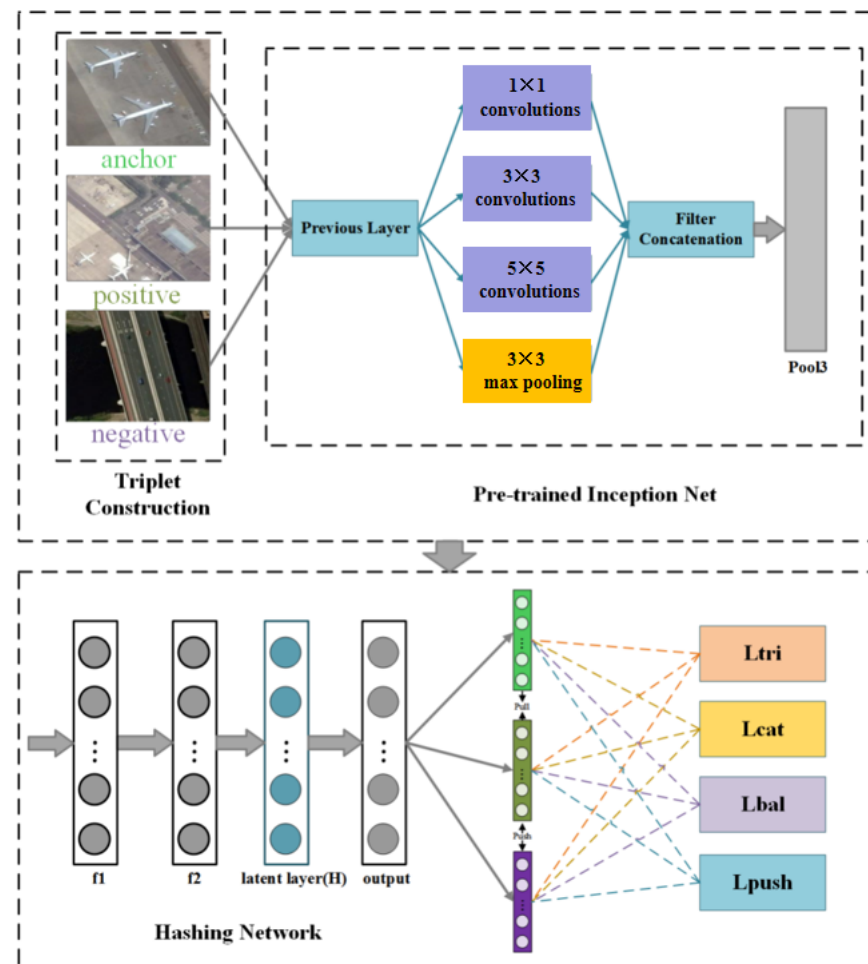


Figure 1. The overall framework of DMCH. It is divided into two parts: a deep feature extraction process and a hash learning process. The deep feature extraction process consists of extracting image features using pre-trained Inception Net as the backbone network; then, triples are selected as the input to the hash network. The hash learning process combines the triplet loss L_{tri} , category-level classification loss L_{cat} , hash code binarization constraint L_{bal} , and bit balance constraint L_{push} to learn the hash codes.

In the first stage, Inception-v3 pre-trained on ImageNet is adopted as the backbone network, hoping to overcome over-fitting resulting from limited labeled samples in feature extraction. Compared with previous architecture, Inception-v3 consists of 42 layers, has fewer parameters, and has more computational efficiency. For each image, represented as $I = \{I_i\}_{i=1}^n$, the extracted 2048-dimensional feature vector of the previous layer of the classification output layer, represented as $G = \{G_1, G_2, \dots, G_n\}$, $G_i \in 2048$, is adopted as the intermediate representation of the input RSF.

Then, effective triplet $\{G_j^z\}_{j=1}^M$ from the $G = \{G_1, G_2, \dots, G_n\}$, $G_i \in 2048$, obtained in the first stage are fed into the hashing network. Suppose $z = \{a, p, n\}$ represents the selected triplets, and M illustrates how many triplets participate in training. The weights of the hash layers are initialized randomly to train the network from scratch. The learned hash functions of DMCH, $f: R^{2048} \rightarrow R^K$, encode each high-dimensional feature vector

into a short K-bit binary sequence while keeping the similarity relationship between the original space and the hash space consistent.

In the hash learning process, the hashing network consists of two fully connected layers $f_1 - f_2$, a latent layer (H), and an output layer (output), which contains 1024, 512, K, and C neurons, respectively, where K represents the desired length of the hash codes. In the first two layers, the Leaky ReLU [43] nonlinear activation function is employed to speed up the network convergence, and the sigmoid activation function is employed in the hidden layer to limit the output value to the range from 0 to 1. The purpose of inserting a hidden layer is to formulate a correlation between the image category label and the hash codes while enhancing the potential correlation between the hash codes of images from the same category. The classification output layer is used to obtain the network softmax output corresponding to each input image and then to calculate the cross-entropy classification loss.

After hashing network training, the final hash functions can be obtained by quantizing the real-valued vector output by the network's hidden layer. Specifically, for query image I_q and its feature vector G_q , the real-valued vector output is thresholded by the hidden layer through the *sign* function. Therefore, the K-bits binary-like hash codes are binarizing into corresponding K-bits binary hash codes, which are shown as follows:

$$b_q = (\text{sign}(\sigma(f(G_q)) - 0.5) + 1)/2, \quad (1)$$

where $\sigma(\cdot)$ represents the output value of the hidden layer neuron, defined by $\sigma(z) = 1/(1 + e^{-z})$ with a real value of z . $\text{sign}(\cdot)$ denotes the sign function, where $\text{sign}(v) = 1$ when $v > 0$ and -1 otherwise. Finally, the Hamming distance between b_j and b_q is calculated, and the top-k semantically similar images I_q are returned.

3.3. The Construction of Triplets

Existing hashing methods, e.g., MiLaN [29], do not use an effective strategy to select triplets but only select samples by simple random sampling, which does not make full use of the triplets information, resulting in slow convergence of the model and insufficient discriminative features learned by the network, thus affecting the retrieval performance. Inspired by the idea of [47] in the field of person re-identification, we intend to mine only semi-hard positive pairs and negative pairs in a batch. In other words, we adopt a small-scale data and large-batch online training scheme named BatchAll. Specifically, a batch of $P \times K$ images is formed by randomly selecting P classes and randomly selecting K images from each class. Next, an image G_j^q is chosen from the $P \times K$ images to act as the anchor sample. Then, according to the category of the anchor, a positive image G_j^p of the same category and a negative image G_j^n of the different category are randomly chosen from images G. Thus, a total of $P \times K \times (K - 1) \times (P \times K - K)$ triplets can be got for each batch. Finally, the triplets with a triplet loss greater than 0 are picked out as the input of the final hash network, making the hash network easier to converge and the learned hash codes more discriminative.

3.4. Objective Function

The goal of hash learning is to learn a set of nonlinear hash functions. Through these functions, the semantic similarity between the data points in the original feature space can be preserved as much as possible. That is to say, the Hamming distance of samples from the same class should be smaller than that of samples from different classes. Aiming to this goal, DMCH adopts the triplet loss to learn the metric space, in which image pairs with the same semantic label should be closer than image pairs with different labels. Unlike pointwise and pairwise losses, the triplet loss pulls similar pairs closer and pushes dissimilar pairs

apart. The training process makes the distance between the anchor-positive pair closer than that between the anchor-negative pair. The L_{Tri} is designed as follows:

$$L_{tri} = \sum_{j=1}^M \max \left(0, \|f(G_j^a) - f(G_j^p)\|_2^2 - \|f(G_j^a) - f(G_j^n)\|_2^2 + m \right), \quad (2)$$

where (G_j^a, G_j^p, G_j^n) represents the selected triplet samples, $\|\cdot\|_2^2$ represents the L_2 norm of the vector, which $f(\cdot)$ represents the hash mapping function, and m represents the minimum boundary threshold, which is used to control the relative distance between positive and negative samples, ensuring that the hash code of the same type of image should be closer than the hash code of different types of images.

The above triplet loss L_{tri} is able to learn the similarity-preserved metric space well. Still, the category-level semantics of the RSIs have not taken advantage of the learning of the hash functions. To learn the category-level hash code, a hidden layer with K units is inserted between the fully connected layer and the classification output layer for the purpose of modeling the relationship between the image label and the hash codes. We assume that the category labels are implicitly determined by a series of potential latent attributes, and the status of each attribute value is on or off (corresponding to 1 or 0 of binary code). When an input image receives its binary-like hash codes in the hidden layer, the classification result depends on the value of these latent attributes. This suggests that classification loss optimization can strengthen the latent correlation between similar hash codes so that images with similar semantics are guaranteed to have similar binary hash codes. The L_{Cat} is designed as follows:

$$L_{cat} = \sum_{j=1}^M \left(\varphi(y_j^a, Y_j^a) + \varphi(y_j^p, Y_j^p) + \varphi(y_j^n, Y_j^n) \right), \quad (3)$$

where φ denotes the cross-entropy loss function, Y_j^a, Y_j^p, Y_j^n , respectively, represent the softmax outputs of anchor samples, positive samples and negative samples through the classification layer, and y_j^a, y_j^p, y_j^n , respectively, correspond to their category label information.

Apart from preserving the semantic similarity of the hash codes, we also add two additional hash code constraints to ensure the bit-balance and bit-uncorrelation of the generated binary codes. On one hand, the activation value of each unit in the hidden layer is encouraged to be approximately $\{0,1\}$. Since each latent node is activated by the sigmoid function, the range of its value should be from 0 to 1. To make the generated real-value hash codes closer to 0 or 1, DMCH maximizes the sum of squared errors between the activation value and 0.5. K denotes the length of hash codes. The specific definition of L_{Push} can be rewritten as follows:

$$L_{push} = -\frac{1}{K} \sum_{j=1}^M \left(\|f(G_j^a) - 0.5\|^2 + \|f(G_j^p) - 0.5\|^2 + \|f(G_j^n) - 0.5\|^2 \right). \quad (4)$$

On the other hand, the bit-balance is also considered in the hash learning process. It means that each hash bit has a 50% probability of being 0 or 1 so that the hash codes are more separated, and the discrete distribution of entropy is maximized. In this case, the hash codes perform best. To this end, we encourage each bit of the hash codes to be 0 or 1 on average, which is achieved by minimizing the squared errors between the average value of the latent layer and 0.5. The definition of L_{bal} can be reformulated as follows:

$$L_{bal} = \sum_{j=1}^M \left((\text{mean}(f(G_j^a)) - 0.5)^2 + (\text{mean}(f(G_j^p)) - 0.5)^2 + (\text{mean}(f(G_j^n)) - 0.5)^2 \right). \quad (5)$$

In summary, the overall objective function of DMCH can be rewritten as follows:

$$\min_{\{W\}} L_{Overall} = \min_{\{W\}} (L_{tri} + \lambda L_{cat} + \gamma L_{push} + \alpha L_{bal}), \quad (6)$$

where λ , γ , and α are the weights of L_{cat} , L_{push} , and L_{bal} , respectively, and W denotes the parameters of DMCH. The overall objective function includes four parts (L_{tri} , L_{cat} , L_{push} , and L_{bal}). The L_{tri} ensures that the hash codes of the anchor are more similar to that of the positive sample than that of the negative sample. The L_{cat} preserves category-level semantics of *hash* codes and enhances the latent correlation between similar hash codes. The L_{bal} and L_{push} constraints guarantee the bit-balance and bit-uncorrelation of the generated hash codes. Through this design, DMCH integrates classification and retrieval in a unified deep hashing network and generates effective hash codes while preserving semantics. The detailed learning process of DMCH is summarized as Algorithm 1.

Algorithm 1 DMCH Algorithm.

Input:

First stage: Remote sensing image set $I = \{I_i\}_{i=1}^n$.

Second stage: M triplet set $\{G_j^a, G_j^p, G_j^n\}_{j=1}^M, G_i \in R^{2048}$ with the corresponding class label

$\{y_j^a, y_j^p, y_j^n\}_{j=1}^M$.

Output: The weight parameters W of DMCH.

(1) Initialize the weight of the hash layers randomly i ;

(2) Pick triplet construction image feature $\{G_j^a, G_j^p, G_j^n\}_{j=1}^M, G_i \in R^{2048}$.

(3) Calculate the binary hash codes $f(G_j^a), f(G_j^p), f(G_j^n)$ obtained by the output of the latent layer H by forward propagation.

(4) Compute the overall objective function $L_{Overall}$ of the DMCH.

(5) Optimize weight parameter W by Adam optimizer.

Until:

Convergence or a preset number of training iterations is satisfied.

Return: W .

3.5. Coarse-to-Fine Hash Code Ranking

Inspired by [42], DMCH introduces the coarse-to-fine hierarchical search strategy learned to better distinguish feature similarities. This approach considers that the quantization process of the real-valued vector will cause the loss of the detailed features while utilizing detailed features before quantization to perform similarity measures will increase retrieval time and memory consumption. In [42], their hierarchical search strategy takes the Hamming distance as a coarse-level similarity measurement to retain the simplicity and efficiency of hash code-based retrieval while taking the Euclidean distance between their corresponding real-valued feature vectors before quantization as a fine-level similarity measurement to improve retrieval performance.

The coarse-to-fine hash code ranking further distinguishes the similarity between the retrieved image with the same hash codes and the query image by using feature vectors with more detailed features. In DMCH, we assume that the hash codes generated by candidate image set are I_i is $\{h_{i,1}, h_{i,2}, \dots, h_{i,K}\}$, the corresponding real-valued hash codes before quantization are $\{u_{q,1}, u_{q,2}, \dots, u_{q,K}\}$, and the hash codes of the query image are $\{h_{q,1}, h_{q,2}, \dots, h_{q,K}\}$, the coarse-to-fine hash code ranking scheme can be expressed by

$$D(h_i, h_q) = \sum_{j=1}^K \max(h_{i,j} \oplus h_{q,j} (1 - (h_{i,j} \oplus h_{q,j}))) \times \frac{||u_{i,j}| - |h_{q,j}||}{[K - \sum_{j=1}^K h_{i,j} \oplus h_{q,j}]_+}, \quad (7)$$

where the max function ensures that the Hamming distance plays a leading role in the similarity measurement. If the hash bits are different, the corresponding distance is 1. In

this case, the retrieval image with a smaller Hamming distance is more similar to the query image. If the hash bits are the same, the detailed feature vector before quantization is used for further fine-grained similarity measurement.

The time and space complexity of DMCH are analyzed. With regard to time complexity, we focus on the Hamming distance-based similarity calculation and the coarse-to-fine hash code ranking process. Assuming that the number of candidate images is M , the time complexity is about the sum of linear order $O(N)$ in stage one and $O(M \log M)$ in stage two. Considering practical applications, the number of returned images M is much smaller than that of all candidate images N in the dataset, so the time complexity can be approximated as $O(N)$. With regard to space complexity, the additional space complexity that stores additional real-valued feature vectors is $O(NK)$, where the length of hash codes is K , and the real-valued feature vectors before relaxation are K -dimensional real numbers. That means DMCH can improve retrieval accuracy with limited additional space complexity.

4. Experimental Results and Analysis Results

In this section, we initially provide a succinct overview of the three publicly available benchmark datasets, the conventional metrics, and the configurations employed in our experiments. Subsequently, we highlight the contributions of the triplet loss, category-level classification loss, and the ideal hash code constraint within the realm of RSIR. Following this, we benchmark DMCH against various representative baseline models to validate its effectiveness. Lastly, we present the results through visual illustrations.

4.1. Datasets

In this paper, we utilized three major benchmark datasets for the classification and retrieval of RSIs: UCMD [48], AID [49], and NWPU-RESISC45 [50].

UCMD [48]: This dataset, released by the University of California in 2010, comprises 2100 images spanning 21 categories, each measuring 256×256 pixels. The images, sourced from aerial orthophotos captured by the National Geological Survey of the United States, have established UCMD as a widely used benchmark dataset in the field of RSI classification and retrieval.

AID [49]: Launched by Wuhan University in 2017, the AID dataset contains 10,000 images across 30 different scene categories. The images were collected from Google Earth, representing diverse global regions, each with dimensions of 600×600 pixels and a spatial resolution ranging from 8 to 0.5 m. Compared to UCMD, the AID dataset features higher intra-class differences due to varied imaging and temporal conditions.

NWPU-RESISC45 [50]: One of the largest benchmark datasets for remote sensing scene classification, NWPU-RESISC45 was released by Northwestern Polytechnical University in 2017. It comprises 31,500 images from 45 scene categories, each containing 700 images with dimensions of 256×256 pixels and a spatial resolution between 30 and 0.2 m. Like AID, the images were gathered from Google Earth. The NWPU-RESISC45 dataset presents unique challenges, including a large number of images, a wide variety of scene classes, significant within-class diversity, and high between-class similarity characteristics.

4.2. Metrics and Experimental Settings

To assess retrieval precision and efficiency, we have employed the established metrics of mean average precision (mAP) and retrieval time (ms), respectively. The experiments were conducted on a workstation equipped with an Intel Core i7 Processor, 32 GB of RAM, and an NVIDIA 12 GB TITAN Xp GPU. The system was operated using Ubuntu 16.04, with a deep learning platform composed of TensorFlow 1.2.0, SciPy 1.1.0, and Pillow 5.1.0. Python 2.7 was the chosen programming language for this research. The configuration settings are as follows. In the stage of feature extraction, Inception-V3, pre-trained on the ImageNet dataset, is selected as the backbone network. The extracted 2048-dimensional feature vector output by Inception-V3 is used to describe the high-level visual features of the input RSIs. All input images are resized to 256×256 pixels. The train-test split

strategy follows 60% and 40%. In the stage of hash learning, the first three classes are selected randomly from the training dataset, and then 30 images are selected randomly from each class to form a batch of 90 images. Next, the BatchAll strategy is employed to ensure effective triplets are selected for hash learning network training. The value of the threshold margin in triplet loss is set to 0.2, and the hyperparameter $\gamma = 0.001$. The weight of the hash layer is initialized randomly, and the Adam optimizer is adopted to optimize the overall objective function with a learning rate of 10^{-4} . The Adam hyper-parameters β_1 and β_2 are set to 0.5 and 0.9, respectively. Finally, the stopping condition is when the epoch reaches 800, or the network converges.

4.3. Results and Analysis

4.3.1. Results of Ablation Experiments

The effectiveness of triplet loss, BatchAll strategy, category-level semantics, hash-code binarization, and hash-code balance constraints is evaluated through ablation experiments. In our experiments, we mark DMCH without using the triplet loss as DMCH-nTri and mark DMCH without using the BatchAll strategy to select triplets as DMCH-nBatchAll-Tri. Similarly, DMCHnCat, DMCH-nPush, and DMCH-nBal refer to the DMCH without classification loss, the DMCH without hash-code binarization constraints, and the DMCH without hash-code balance constraints, respectively. The results of ablation experiments on three benchmark datasets are shown in Tables 1–3. Bit-K from 32 to 64 improves the performance of the model more than bit-K from 64 to 96, which may be due to the discrimination of the selected features from high to low. It can also be seen that the model has better performance with the increase of bit bits, indicating that higher bit bits can better maintain the semantic information of the image.

Table 1. Results of ablation experiments on UCMD.

Constraint	K = 32	K = 64	K = 96
DMCH-nTri	0.8495	0.8532	0.8578
DMCH-nBatchAll-Tri	0.9017	0.9110	0.9113
DMCH-nCat	0.9065	0.9125	0.9099
DMCH-nPush	0.9114	0.9206	0.9298
DMCH-nBal	0.8703	0.8783	0.8865
DMCH	0.9185	0.9266	0.9291

Table 2. Results of ablation experiments on AID.

Constraint	K = 32	K = 64	K = 96
DMCH-nTri	0.9083	0.9167	0.9230
DMCH-nBatchAll-Tri	0.9385	0.9484	0.9479
DMCH-nCat	0.9363	0.9480	0.9490
DMCH-nPush	0.9480	0.9525	0.9561
DMCH-nBal	0.9449	0.9458	0.9477
DMCH	0.9544	0.9570	0.9577

Table 3. Results of ablation experiments on NWPU-RESISC45.

Constraint	K = 32	K = 64	K = 96
DMCH-nTri	0.7408	0.7691	0.7964
DMCH-nBatchAll-Tri	0.7939	0.8123	0.8310
DMCH-nCat	0.7411	0.7914	0.8010
DMCH-nPush	0.7904	0.8216	0.8429
DMCH-nBal	0.7468	0.7698	0.7917
DMCH	0.7974	0.8287	0.8437

The results clearly highlight the pivotal role of the triplet loss, category-level semantics, and hash code bit balance constraint in the effective learning of hash codes to maintain semantics. The DMCH leverages triplet loss to enhance the intra-class similarity and inter-class disparity between samples, thereby pulling positive samples and anchor samples closer while pushing negative samples and anchor samples further apart. The DMCH model also utilizes BatchAll to select valid ternary groups, which furnish more useful information for learning the hash function. Moreover, the DMCH model has shown an improvement in the mAP value by up to 5.63% on the NWPU-RESISC45 dataset when compared to the DMCH-nCat model. This underlines the necessity for the DMCH model to employ category-level semantic information to guide hash function learning. Lastly, comparing the retrieval performance of the DMCH model with the DMCH-nPush and DMCH-nBalancing models, it becomes evident that the hash code bit equalization constraint is crucial for the DMCH model to learn compact and efficient hash codes.

The DMCH model is displayed in the t-SNE 2D visual scatterplot of the hash code generated on the UC Merced test set and the hash code scatter plot of the DMCH-nCat model, in which different colored point classes represent different land cover categories. As shown in Figure 2, the hash codes generated by the DMCH model proposed in this paper present better intra-class aggregation. The hash code learned by the DMCH model has a stronger discriminant ability, which makes the hash code boundaries between different types of samples clearer because class-level semantics can enhance the potential correlation between similar hash codes.

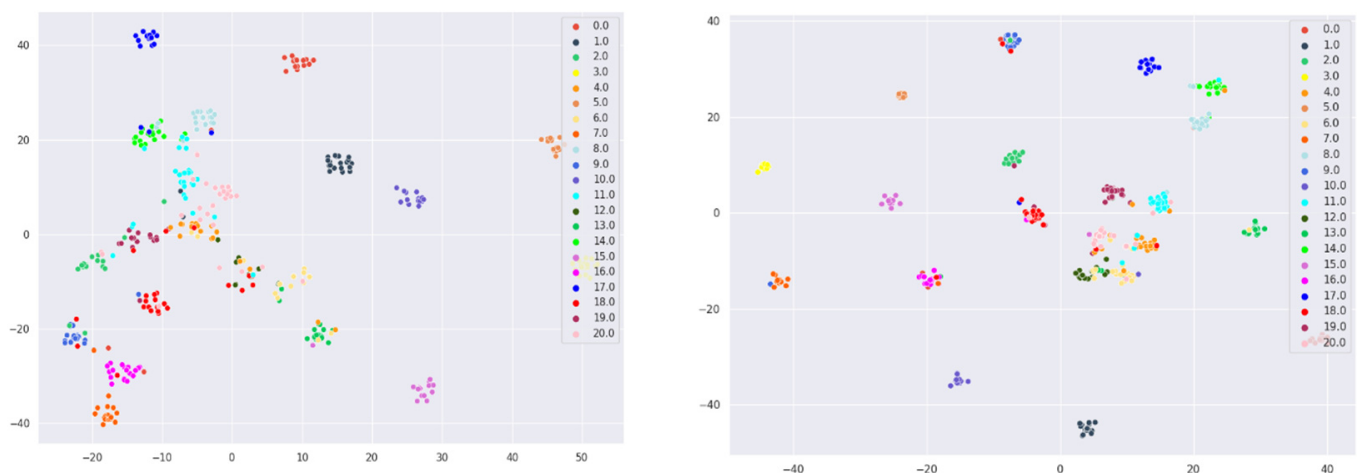


Figure 2. t-SNE 2D visualization scatter plots of DMCH-nCat and DMCH on UCMD ($k = 32$), the proposed model DMCH shows better intra-class aggregation.

4.3.2. Results on Benchmark Datasets

We also compare DMCH and several baselines including PRH [12], KSLSH [11], DPSH [23], MiLaN [29], AHCL [37], FAH [36], and DSSH [38]. Among them, PRH and KSLSH belong to traditional unsupervised and supervised hashing methods, respectively, which are developed for retrieval tasks of RSIs. DPSH and MiLaN are representative deep learning-based hashing methods. The FAH introduces an adversarial regularization to hashing learning. AHCL generates the hash codes of query and database images in an asymmetric way. The difference between DPSH and MiLaN is that the former adopts pairwise loss, and the latter adopts triplet loss. Tables 4–6 report the comparison results of different methods, which demonstrate the superiority of DMCH. The effectiveness of the hash codes ranking scheme is also verified through comparative experiments. Concretely, in the coarse stage, the first M similar images are returned based on the similarity calculation in Hamming space. In the fine stage, the M similar images are reordered based on the

similarity calculation in Euclidean space. We mark the retrieval results by coarse-to-fine hash code ranking scheme as DMCH+. The best experimental results are labeled in bold.

Table 4. Comparison with baselines on UCMD.

Methods	Description	K = 32	K = 64	K = 96
PRH	Unsupervised/shallow	0.1557	0.1744	0.1858
KSLSH	Supervised/shallow	0.6307	0.6536	0.6680
DPSH	Supervised/deep	0.7478	0.8174	0.8640
MiLaN	Supervised/deep	0.9171	0.9176	0.9178
FAH	Supervised/deep	0.9114	0.9122	0.9233
AHCL	Supervised/deep	0.9121	0.9231	0.9237
DSSH	Supervised/deep	0.9156	0.9248	0.9274
DMCH	Supervised/deep	0.9185	0.9266	0.9291
DMCH+	Supervised/deep	0.9283	0.9302	0.9361

Table 5. Comparison with baselines on AID.

Methods	Description	K = 32	K = 64	K = 96
PRH	Unsupervised/shallow	0.1425	0.1624	0.1669
KSLSH	Supervised/shallow	0.4953	0.5330	0.5589
DPSH	Supervised/deep	0.3008	0.3394	0.3546
MiLaN	Supervised/deep	0.9255	0.9378	0.9410
FAH	Supervised/deep	0.9109	0.9168	0.9172
AHCL	Supervised/deep	0.9436	0.9457	0.9501
DSSH	Supervised/deep	0.9518	0.9533	0.9556
DMCH	Supervised/deep	0.9544	0.9570	0.9577
DMCH+	Supervised/deep	0.9588	0.9610	0.9637

Table 6. Comparison with baselines on NWPU-RESISC45.

Methods	Description	K = 32	K = 64	K = 96
PRH	Unsupervised/shallow	0.1524	0.1726	0.1820
KSLSH	Supervised/shallow	0.3815	0.3920	0.4012
DPSH	Supervised/deep	0.6812	0.7524	0.7835
MiLaN	Supervised/deep	0.7638	0.8039	0.8272
FAH	Supervised/deep	0.6724	0.7149	0.7351
AHCL	Supervised/deep	0.7740	0.7921	0.8323
DSSH	Supervised/deep	0.7892	0.8256	0.8415
DMCH	Supervised/deep	0.7974	0.8287	0.8437
DMCH+	Supervised/deep	0.8209	0.8427	0.8560

The DMCH method proposed in this study outperforms several deep hashing methods, including DSH, DHN, DPSH, and MiLaN, consistently achieving the highest mAP values across all tests. The enhanced version, DMCH+, further boosts retrieval performance. Interestingly, when compared with the latest AHCL method, the DMCH method shows an improvement of about 2% on the NWPU-RESISC45 dataset. Of particular note is the fact that traditional hashing methods demonstrate inferior performance on the AID dataset. This could be attributed to the significant variance in the number of images across different categories within this dataset. Traditional hashing methods do not account for the imbalance in image category numbers, consequently impacting retrieval performance negatively.

In contrast, the proposed DMCH model addresses this issue effectively. By learning the ternary metric space for semantic similarity preservation and preserving category-level semantic information of hash codes through a hidden layer in the hash network, the DMCH model rectifies the problem of uneven semantic space distribution. This, in turn, amplifies the potential correlation between similar hash codes. Furthermore, the DMCH model

accounts for the hash codes' binarization and bit-balancing constraints, thereby consistently demonstrating high retrieval accuracy across diverse datasets. This multi-faceted approach to enhancing retrieval performance underscores the efficacy and versatility of the DMCH model in diverse conditions.

When considering the type of dataset and the number of images it contains, our DMCH model demonstrates pronounced advantages when applied to the AID dataset, which features a diverse array of data and a substantial volume of images. This suggests that the model's performance relies on the dataset utilized, with large-scale datasets proving particularly conducive to model training. The NWPU-RESISC45 dataset currently presents the most significant challenge due to its higher intra-class diversity and inter-class similarity. Despite these complexities, our model delivers a highly competitive performance, affirming the effectiveness of our strategy that utilizes a fine-to-coarse selection of triples and hashing schemes. In future research, we plan to accommodate the multi-scale and multi-angle characteristics inherent in remote-sensing images. By designing a more sophisticated feature extraction algorithm, we aim to enhance the precision of image retrieval, thus advancing the robustness and applicability of our DMCH model.

4.3.3. Visualization Results

In this section, we present the category retrieval performance of DMCH on three datasets. Figure 3 illustrates the visual retrieval results of DMCH for three categories on three datasets. The left column shows the query image (marked in blue), and the right part shows the returned top 20 images, in which the images marked in red indicate the wrong categories (i.e., in Figure 3c, the query image is an airport, yet the image marked in red is an overpass).

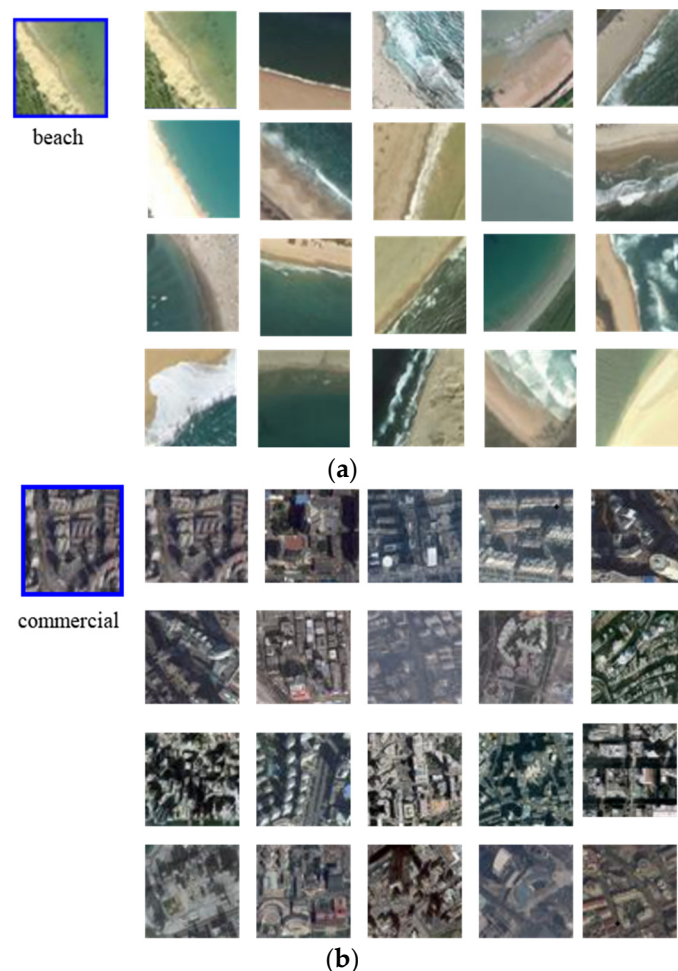


Figure 3. Cont.

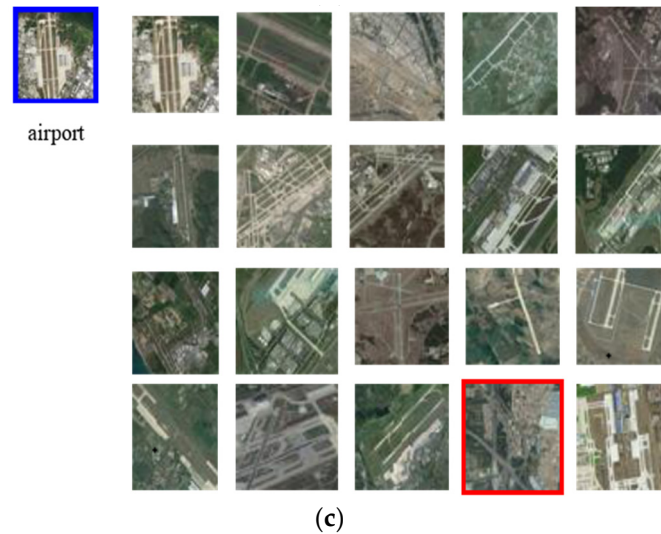


Figure 3. Top 20 retrieval results returned by DMCH. (a) UCMD. (b) AID. (c) NWPU-RESISC45.

4.3.4. Efficiency Analysis

Table 7 illustrates the retrieval time of DMCH on three datasets. Although the retrieval time increases slightly with the increase of hash bits, it proves that DMCH can meet the requirements for real-time retrieval tasks.

Table 7. Results time (milliseconds) of DMCH on the three datasets.

Datasets	Retrieval Time (ms)		
	K = 32	K = 64	K = 96
UCMD	85.55	96.01	104.25
AID	112.83	119.99	125.07
NWPU-RESISC45	161.34	172.46	181.61

5. Conclusions

In this study, we propose a novel two-stage deep metric and category-level semantic hashing network, DMCH, for remote sensing image retrieval (RSIR). The core objective of DMCH is to maintain semantic similarity when mapping high-dimensional remote sensing image features to low-dimensional binary codes. It not only elevates retrieval accuracy but also guarantees heightened retrieval efficiency and reduces storage space simultaneously. DMCH employs a triplet selection strategy for semantic metric space learning, enabling maximum exploitation of the triplet label information. Moreover, a hidden layer is integrated into the hashing network to amplify the potential relevance between analogous hash codes. Additionally, DMCH employs a hash code binarization constraint and a bit-balancing constraint to generate more effective and distinctive binary hash codes. A progressive coarse-to-fine hash code ranking scheme is also utilized within DMCH to enhance its retrieval performance further. The proficiency and superiority of DMCH are confirmed via rigorous experimentation on three benchmark datasets. In the future, we will consider combining deep hashing with cross-modal retrieval to achieve mutual retrieval of remote-sensing images, text, and sound.

Author Contributions: Methods, L.S. and Q.C.; software, L.S. and H.H.; validation, H.H. and L.S.; formal analysis, L.S.; investigation, H.H.; resources, Z.S.; data curation, L.S.; writing—original draft preparation, L.S. and Q.C.; writing—review and editing, reorganizing and rewriting the manuscript, H.H. and X.H.; visualization, L.S.; supervision, Z.S.; project management, Z.S.; funding acquisition, Q.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China (42090012, 42271352), the Guangxi Science and Technology Programme (Gui Ke 2021AB30019),

the Hubei Key R&D Programme (2022BAA048), the Sichuan Key R&D Programme (2022YFN0031, 2023YFN0022, 2023YFS0381), and the Shanxi Provincial Science and Technology Major Special Project (202201150401020).

Data Availability Statement: The data that support the findings of this study are available from the corresponding author, upon reasonable request.

Acknowledgments: The authors would like to thank the reviewers for reviewing this paper and providing important feedback throughout its development.

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

References

- Chi, M.; Plaza, A.; Benediktsson, J.A.; Sun, Z.; Shen, J.; Zhu, Y. Big data for remote sensing: Challenges and opportunities. *Proc. IEEE* **2016**, *104*, 2207–2219. [\[CrossRef\]](#)
- Zhang, L.; Zhang, L. Artificial intelligence for remote sensing data analysis: A review of challenges and opportunities. *IEEE Geosci. Remote Sens. Mag.* **2022**, *10*, 270–294. [\[CrossRef\]](#)
- Yadav, S.K.; Borana, S.L.; Parihar, S.K. Application of Geospatial Technology for Disaster Management Preparedness in Jodhpur City. *Int. J. Curr. Res.* **2017**, *9*, 60397–60404.
- Ouyang, X.; Xu, Y.; Mao, Y.; Liu, Y.; Wang, Z.; Yan, Y. Blockchain-Assisted Verifiable and Secure Remote Sensing Image Retrieval in Cloud Environment. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2022**, *16*, 1378–1389. [\[CrossRef\]](#)
- Shao, Z.; Zhou, W.; Deng, X.; Zhang, M.; Cheng, Q. Multilabel remote sensing image retrieval based on fully convolutional network. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2020**, *13*, 318–328. [\[CrossRef\]](#)
- Zhang, X.; Li, W.; Wang, X.; Wang, L.; Zheng, F.; Wang, L.; Zhang, H. A Fusion Encoder with Multi-Task Guidance for Cross-Modal Text–Image Retrieval in Remote Sensing. *Remote Sens.* **2023**, *15*, 4637. [\[CrossRef\]](#)
- Har-Peled, S.; Kumar, N. Approximate nearest neighbor search for low-dimensional queries. *SIAM J. Comput.* **2013**, *42*, 138–159. [\[CrossRef\]](#)
- Weiss, Y.; Torralba, A.; Fergus, R. Spectral hashing. In Proceedings of the 21st International Conference on Neural Information Processing Systems, Vancouver, BC, Canada, 8–10 December 2008.
- Gong, Y.; Lazebnik, S.; Gordo, A.; Perronnin, F. Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.* **2012**, *35*, 2916–2929. [\[CrossRef\]](#)
- Liu, W.; Mu, C.; Kumar, S.; Chang, S.F. Discrete graph hashing. In Proceedings of the 27th International Conference on Neural Information Processing Systems, Cambridge, MA, USA, 8–13 December 2014.
- Demir, B.; Bruzzone, L. Hashing-based scalable remote sensing image search and retrieval in large archives. *IEEE Trans. Geosci. Remote Sens.* **2015**, *54*, 892–904. [\[CrossRef\]](#)
- Li, P.; Ren, P. Partial randomness hashing for large-scale remote sensing image retrieval. *IEEE Geosci. Remote Sens. Lett.* **2017**, *14*, 464–468. [\[CrossRef\]](#)
- Reato, T.; Demir, B.; Bruzzone, L. An unsupervised multicode hashing method for accurate and scalable remote sensing image retrieval. *IEEE Geosci. Remote Sens. Lett.* **2018**, *16*, 276–280. [\[CrossRef\]](#)
- Wang, J.; Kumar, S.; Chang, S.F. Semi-supervised hashing for scalable image retrieval. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; IEEE: New York, NY, USA; pp. 3424–3431.
- Kim, S.; Choi, S. Semi-supervised discriminant hashing. In Proceedings of the 2011 IEEE 11th International Conference on Data Mining, Washington, DC, USA, 11–14 December 2011; pp. 1122–1127.
- Liu, W.; Wang, J.; Ji, R.; Jiang, Y.G.; Chang, S.F. Supervised hashing with kernels. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 2074–2081.
- Norouzi, M.; Fleet, D. Minimal Loss Hashing for Compact Binary Codes. In Proceedings of the 28th International Conference on Machine Learning, Madison, WI, USA, 28 June–2 July 2011.
- Shen, F.; Shen, C.; Liu, W.; Tao, H. Supervised discrete hashing. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 37–45.
- LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [\[CrossRef\]](#) [\[PubMed\]](#)
- Liu, C.; Ma, J.; Tang, X.; Zhang, X.; Jiao, L. Adversarial hash-code learning for remote sensing image retrieval. In Proceedings of the IGARSS 2019—2019 IEEE International Geoscience and Remote Sensing Symposium, Yokohama, Japan, 28 July–2 August 2019; pp. 4324–4327.
- Li, P.; Han, L.; Tao, X.; Zhang, X.; Grecos, C.; Plaza, A.; Ren, P. Hashing nets for hashing: A quantized deep learning to hash framework for remote sensing image retrieval. *IEEE Trans. Geosci. Remote Sens.* **2020**, *58*, 7331–7345. [\[CrossRef\]](#)
- Xia, R.; Pan, Y.; Lai, H.; Liu, C.; Yan, S. Supervised hashing for image retrieval via image representation learning. In Proceedings of the AAAI Conference on Artificial Intelligence, Québec City, QC, Canada, 27–31 July 2014; Volume 28, No. 1.
- Zhu, H.; Long, M.; Wang, J.; Cao, Y. Deep hashing network for efficient similarity retrieval. In Proceedings of the AAAI Conference on Artificial Intelligence, Phoenix, AZ, USA, 12–17 February 2016; Volume 30, No. 1.
- Liu, P.; Liu, Z.; Shan, X.; Zhou, Q. Deep Hash Remote-Sensing Image Retrieval Assisted by Semantic Cues. *Remote Sens.* **2022**, *14*, 6358. [\[CrossRef\]](#)

25. Chen, C.; Zou, H.; Shao, N.; Sun, J.; Qin, X. Deep semantic hashing retrieval of remote sensing images. In Proceedings of the IGARSS 2018—2018 IEEE International Geoscience and Remote Sensing Symposium, Valencia, Spain, 22–27 July 2018; pp. 1124–1127.
26. 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.* **2017**, *56*, 950–965. [\[CrossRef\]](#)
27. Han, L.; Li, P.; Bai, X.; Grecos, C.; Zhang, X.; Ren, P. Cohesion intensive deep hashing for remote sensing image retrieval. *Remote Sens.* **2019**, *12*, 101. [\[CrossRef\]](#)
28. Song, W.; Li, S.; Benediktsson, J.A. Deep hashing learning for visual and semantic retrieval of remote sensing images. *IEEE Trans. Geosci. Remote Sens.* **2020**, *59*, 9661–9672. [\[CrossRef\]](#)
29. Roy, S.; Sangineto, E.; Demir, B.; Sebe, N. Metric-learning-based deep hashing network for content-based retrieval of remote sensing images. *IEEE Geosci. Remote Sens. Lett.* **2020**, *18*, 226–230. [\[CrossRef\]](#)
30. Chen, Y.; Lu, X. Deep category-level and regularized hashing with global semantic similarity learning. *IEEE Trans. Cybern.* **2020**, *51*, 6240–6252. [\[CrossRef\]](#)
31. Liu, P.; Wang, Y.; Zhou, Q.; Wang, Z. Deep hashing using proxy loss on remote sensing image retrieval. *Remote Sens.* **2021**, *13*, 2924.
32. Zhang, X.; Zhang, L.; Shum, H.Y. QsRank: Query-sensitive hash code ranking for efficient ϵ -neighbor search. In Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 2058–2065.
33. Jiang, Y.G.; Wang, J.; Chang, S.F. Lost in binarization: Query-adaptive ranking for similar image search with compact codes. In Proceedings of the 1st ACM International Conference on Multimedia Retrieval, Trento, Italy, 18–20 April 2011; pp. 1–8.
34. Gordo, A.; Perronnin, F.; Gong, Y.; Lazebnik, S. Asymmetric distances for binary embeddings. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *36*, 33–47. [\[CrossRef\]](#) [\[PubMed\]](#)
35. Lv, Y.; Ng, W.W.; Zeng, Z.; Yeung, D.S.; Chan, P.P. Asymmetric cyclical hashing for large scale image retrieval. *IEEE Trans. Multimed.* **2015**, *17*, 1225–1235. [\[CrossRef\]](#)
36. Liu, C.; Ma, J.; Tang, X.; Liu, F.; Zhang, X.; Jiao, L. Deep hash learning for remote sensing image retrieval. *IEEE Trans. Geosci. Remote Sens.* **2020**, *59*, 3420–3443. [\[CrossRef\]](#)
37. Song, W.; Gao, Z.; Dian, R.; Ghamisi, P.; Zhang, Y.; Benediktsson, J.A. Asymmetric hash code learning for remote sensing image retrieval. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5617514. [\[CrossRef\]](#)
38. Chen, Y.; Huang, J.; Mou, L.; Jin, P.; Xiong, S.; Zhu, X.X. Deep Saliency Smoothing Hashing for Drone Image Retrieval. *IEEE Trans. Geosci. Remote Sens.* **2023**, *61*, 4700913. [\[CrossRef\]](#)
39. Öztürk, Ş.; Alhudhaif, A.; Polat, K. Attention-based end-to-end CNN framework for content-based x-ray image retrieval. *Turk. J. Electr. Eng. Comput. Sci.* **2021**, *29*, 2680–2693. [\[CrossRef\]](#)
40. Öztürk, Ş. Hash code generation using deep feature selection guided siamese network for content-based medical image retrieval. *Gazi Univ. J. Sci.* **2021**, *34*, 733–746. [\[CrossRef\]](#)
41. Öztürk, Ş. Class-driven content-based medical image retrieval using hash codes of deep features. *Biomed. Signal Process. Control.* **2021**, *68*, 102601. [\[CrossRef\]](#)
42. Lin, K.; Yang, H.F.; Hsiao, J.H.; Chen, C.S. Deep learning of binary hash codes for fast image retrieval. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Boston, MA, USA, 7–12 June 2015; pp. 27–35.
43. 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, Boston, MA, USA, 7–12 June 2015; pp. 1556–1564.
44. Fernandez-Beltran, R.; Demir, B.; Pla, F.; Plaza, A. Unsupervised remote sensing image retrieval using probabilistic latent semantic hashing. *IEEE Geosci. Remote Sens. Lett.* **2020**, *18*, 256–260. [\[CrossRef\]](#)
45. Tang, X.; Yang, Y.; Ma, J.; Cheung, Y.M.; Liu, C.; Liu, F.; Zhang, X.; Jiao, L. Meta-hashing for remote sensing image retrieval. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 5615419. [\[CrossRef\]](#)
46. Chen, Y.; Xiong, S.; Mou, L.; Zhu, X.X. Deep quadruple-based hashing for remote sensing image-sound retrieval. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 4705814. [\[CrossRef\]](#)
47. Hermans, A.; Beyer, L.; Leibe, B. In defense of the triplet loss for person re-identification. *arXiv* **2017**, arXiv:1703.07737.
48. Yang, Y.; Newsam, S. Bag-of-visual-words and spatial extensions for land-use classification. In Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, San Jose, CA, USA, 2–5 November 2010; pp. 270–279.
49. Xia, G.S.; Hu, J.; Hu, F.; Shi, B.; Bai, X.; Zhong, Y.; Zhang, L.; Lu, X. AID: A benchmark data set for performance evaluation of aerial scene classification. *IEEE Trans. Geosci. Remote Sens.* **2017**, *55*, 3965–3981. [\[CrossRef\]](#)
50. Cheng, G.; Han, J.; Lu, X. Remote sensing image scene classification: Benchmark and state of the art. *Proc. IEEE* **2017**, *105*, 1865–1883. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.