Hashing Nets for Hashing: A Quantized Deep Learning to Hash Framework for Remote Sensing Image Retrieval

Peng Li, Lirong Han, Xuanwen Tao, *Student Member, IEEE*, Xiaoyu Zhang, Christos Grecos, *Senior Member, IEEE*, Antonio Plaza, *Fellow, IEEE*, and Peng Ren, *Senior Member, IEEE*

***Abstract*— Fast and accurate remote sensing image retrieval from large data archives has been an important research topic in the remote sensing research literature. Recently, hashing-based remote sensing image retrieval has attracted extreme attention because of its efficient search capabilities. Especially, deep remote sensing image hashing algorithms have been developed based on convolutional neural networks (CNNs) and have shown effective retrieval performance. However, implementing a deep hashing network tends to be highly expensive in terms of storage space and computing resources to be suitable for on-orbit remote sensing image retrieval, which usually operates on resource-limited devices such as satellites and unmanned aerial vehicles (UAVs). To address this limitation, we propose to hash a deep network that in turn hashes remote sensing images. Specifically, we develop a quantized deep learning to hash (QDLH) framework for large-scale remote sensing image retrieval. The weights and activation functions in the QDLH framework are binarized to low-bit representations, which require comparatively much less storage space and computing resources. The QDLH results in a lightweight deep neural network for effective remote sensing image hashing. We conduct extensive experiments on two public remote sensing image data sets by incorporating several state-of-the-art network architectures into our QDLH methodology for remote sensing image hashing. The experimental results demonstrate that the proposed QDLH is effective in saving hardware resources in terms of both storage and computation. Moreover, superior remote sensing image retrieval performance is also achieved by our QDLH, compared with state-of-the-art deep remote sensing image hashing methods.**

Manuscript received December 2, 2019; revised February 23, 2020 and March 2, 2020; accepted March 9, 2020. Date of publication April 6, 2020; date of current version September 25, 2020. This work was supported by the National Natural Science Foundation of China under Project U1906217, Project 61971444, and Project 61871378. *(Corresponding author: Peng Ren.)*

Peng Li is with the College of New Energy, China University of Petroleum (East China), Qingdao 266580, China (e-mail: lipeng@upc.edu.cn).

Lirong Han and Peng Ren are with the College of Oceanography and Space Informatics, China University of Petroleum (East China), Qingdao 266580, China (e-mail: lironghan_upc@163.com; pengren@upc.edu.cn).

Xuanwen Tao and Antonio Plaza are with Hyperspectral Computing Laboratory, Department of Technology of Computers and Communications, University of Extremadura, E-10071 Caceres, Spain (e-mail: aplaza@unex.es).

Xiaoyu Zhang is with the Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China (e-mail: zhangxiaoyu@iie.ac.cn).

Christos Grecos is with the School of Computing, National College of Ireland, Dublin 1, D01 K6W2 Ireland (e-mail: christos.grecos@ncirl.ie).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TGRS.2020.2981997

***Index Terms*— Class intensive, deep hashing, quantized deep network, remote sensing images retrieval.**

I. Introduction

With the development of remote sensing technologies in recent years, a large number of remote sensing images have been collected by optical, synthetic aperture radar (SAR), light detection and ranging (LiDAR) and other instruments. The extensive databases of collected remote sensing images comprise abundant information, which plays an important role in many fields [1]–[3] such as environmental monitoring and disaster rescue. However, the explosive growth of remote sensing images renders a great challenge on how to retrieve the desired images from tremendously large data sets effectively and efficiently. In this scenario, remote sensing image retrieval has attracted significant attention in the community.

According to image description strategies, traditional remote sensing image retrieval algorithms are divided into two categories: text-based image retrieval (TBIR) [4] methods and content-based image retrieval (CBIR) [5] methods. TBIR methods rely mainly on manually annotated text information to describe the content of the images, and it is inefficient and unsuitable for large-scale remote sensing image retrieval tasks due to the required extensive labor. In turn, CBIR methods extract different types of visual features automatically for image representation, and image retrieval is carried out based on the features extracted by various similarity metrics. There have been many works on context-based remote sensing image retrieval [6]. However, the dimension of visual features can be in the order of thousands, which requires enormous storage space and consumes considerable time for linear similarity scan on a big image data set.

In order to cope with the curse of dimensionality, many approximate nearest neighbor (ANN) search methods [7] have been developed for CBIR tasks. Among these various ANN search methods, hashing [8] is a powerful technique for big data retrieval because of its excellent ability in the task of compacting features efficiently. Hashing algorithms generally learn a set of hash functions to project the original images from high-dimensional feature space to a low-dimensional hamming

space, in which the images are represented by binary hash codes and the intrinsic similarity structure is preserved as well. Thus, the image similarity can be efficiently evaluated based on the hamming distance of the binary hash codes instead of the Euclidean distance, which is of great importance for large-scale image processing tasks. Compared with traditional CBIR algorithms, the hashing algorithms significantly reduce the amount of memory storage occupation and extremely improve the retrieval speed because the image similarity is efficiently measured through the hamming distance between two hash codes.

Classical hashing methods, such as locality sensitive hashing (LSH) [9], spectral hashing (SH) [10], and iterative quantization (ITQ) [11], [12]–[15] have been successfully applied to various big data tasks including large-scale image retrieval. In recent years, hashing algorithms have been introduced to the field of remote sensing image retrieval. Lukac *et al.* [16] proposed a parallelization of kernelized locality-sensitive hashing (KLSH) using graphical processing units (GPUs), in order to perform fast parallel satellite image retrieval. Demir and Bruzzone [17] and Reato *et al.* [18] introduced kernelized unsupervised and supervised hashing into scalable remote sensing image retrieval and further developed a multicode hashing scheme based on the segmented regions of remote sensing images. Li and Ren [19] and Li *et al.* [20] proposed a fast unsupervised hashing algorithm for large-scale remote sensing image retrieval and extended it to an online scheme to handle constantly updated remote sensing images. Ye *et al.* [21] developed a multifeature learning method on the remote sensing image hashing problem.

Most of the existing remote sensing image hashing methods rely on hand-crafted features for hash code learning. It is commonly observed that such low-level hand-crafted features cannot accurately reflect high-level semantic information contained in the images. Therefore, the retrieval accuracy of previous remote sensing image hashing methods fails to meet the demands of practical applications in many cases. In order to benefit from the representational power of deep learning techniques, many deep convolutional neural networks (CNNs) have been adopted for image understanding tasks [22]–[30] (e.g., classification [31], detection [32], and semantic segmentation [33]) in the computer vision and machine learning fields. In this scenario, deep hashing networks (DHNs) have been developed for natural image retrieval tasks. Tang and Li [34] present a weakly supervised multimodal hashing model, which is trained based on the weakly supervised tag information and visual information for scalable social image retrieval. Considering the special characteristics of remote sensing images, Li *et al.* [35], [36] introduced deep hashing neural networks (DHNNs) into large-scale remote sensing image retrieval tasks and obtained significant improvements in retrieval accuracy over previous remote sensing image hashing methods. Han *et al.* [37] presented a cohesion intensive deep hashing to overcome the data imbalance problem of remote sensing image data set.

DHNNs have achieved state-of-the-art performance for remote sensing image retrieval purposes [35]. They are supposed to operate on a comparatively computational powerful work station and retrieve historical remote sensing images. On the other hand, on-orbit retrieval of remote sensing images that are captured in real-time is also an important issue. A specific scenario is that a satellite or a UAV, which is just equipped with limited computational resources, operates on-orbit and captures a large number of images in real-time. The wireless bandwidth for real-time communication between the remote sensing device and the Earth is limited, and it would be beneficial that only images with important features rather than all captured images are transited from the remote sensing device to the Earth. This requires image retrieval on the (resource limited) remote sensing device. However, there remain important problems that hinder deep hashing from such practical remote sensing uses.

TABLE I
MEMORY AND COMPUTATION CONSUMPTION OF DIFFERENT DEEP MODELS

Model	Model size(MB)	Million parameters	FLOPs (G)
AlexNet	> 200	61	0.72
VGGNet	> 500	138	15.5
GoogLeNet	~ 50	6.8	1.6

First, the CNNs have extensive requirements for hardware resources (i.e., memory and computational power). One major reason for the computational overloads is that the deep networks are usually characterized by a large number of parameters with float values. For example, as shown in Table I, the widely used VGGNet model [38] requires more than 500 MB of storage and over 15 B floating-point operations (FLOPs) to classify a single 224×224 image. When the network becomes deeper, these numbers are even larger. Therefore, the existing DHNNs are difficult to be deployed on resource-limited remote sensing devices such as satellites and UAVs for on-orbit image retrieval applications. To the best of our knowledge, despite the availability of some deep model compression schemes introduced to natural image classification problems, the topic of how to develop compressed DHNNs for remote sensing image retrieval has not been fully studied yet. Existing neural network compression methods are mainly divided into two categories: one is to change the structure of the model, such as network pruning [39], and the other is the low-bit expression of parameters [40], [41]. Though a pruned network reduces the number of parameters and operations, it requires FLOP for processing the remote sensing images. The FLOPs are not a desirable option for on-orbit remote sensing image retrieval with embedded devices such as field programmable gate array (FPGA), which essentially requires fixed-point operations for parallel processing and fast computation. Second, remote sensing image data sets usually suffer from the data imbalance problem, which occurs when the number of retrieved images is far less than the number of irrelevant images in a data set. The data imbalance problem requires the hash codes within one class to exhibit strong intraclass similarity and discriminate against a large number of images from other classes. However, existing remote sensing retrieval algorithms usually use indiscriminate hash losses for characterizing image pairs [35], and tend to neglect the data imbalance problem.

Taking the above issues into consideration, this article presents a novel quantized deep learning to hash (QDLH) framework for remote sensing image retrieval. Specifically, the hash codes of remote sensing images are learned through a quantized DHN, in which the filter weights are quantized as 1-bit representations and the activations are quantized as 2-bit representations. After compression, our QDLH model significantly saves memory for convolutional computation, which provides an efficient solution for practical applications on resource-limited remote sensing devices such as satellites and UAVs. In addition, our QDLH employs a class-intensive pairwise loss function in the hash layer learning to address the data imbalance problem and generates more accurate binary hash codes. The main contributions of this article can be summarized as follows.

1) We present a novel strategy of hashing nets for hashing. Our new strategy exploits hashing techniques from two perspectives. First, a deep network is hashed with all weights and activation quantized. This effectively compresses the deep net. Second, the hashing net is used for hashing remote sensing images for efficient retrieval. To the best of our knowledge, we are among the first ones to develop a deep neural network quantization framework for remote sensing image hashing and retrieval tasks. This validates the possibility of using quantized DHNs in remote sensing observation devices with limited resources, such as satellites and UAVs.
2) We introduce the weighted pairwise entropy loss function [37], which have been used in penalizing full precision nets for remote sensing image retrieval, into the training procedure of our QDLH framework. It intensifies the intraclass cohesion of hash codes and improves the remote sensing image retrieval accuracy.
3) We evaluate the proposed QDLH extensively on two public remote sensing image data sets with four popular CNNs (i.e., AlexNet, VGGNet, GoogLeNet, and ResNet-18) for remote sensing image retrieval tasks. Our experimental results demonstrate the efficiency and effectiveness of the proposed framework, which achieves state-of-the-art performance for remote sensing image retrieval.

The remainder of this article is organized as follows. Section II gives a brief review of traditional full precision deep neural networks. Section III presents the proposed QDLH framework in detail. Section IV presents the experiments and analyzes the results of two public remote sensing image data sets. The final section concludes this article with some remarks and hints at plausible future research lines.

II. Review of Full Precision Deep Networks

As a prerequisite for our quantized deep hashing framework, we review the full precision deep networks in terms of CNNs with float weights and activations. Despite the great representational power, a full precision deep network has two disadvantages, that is, the large storage requirement and the high computational complexity, rendering a challenging task for deploying it on a source-limited remote sensing device.

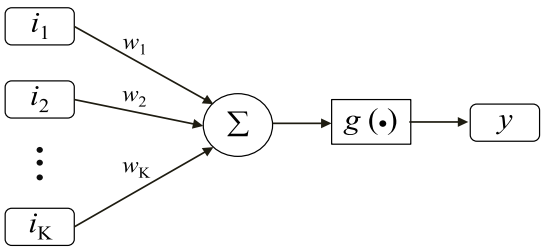


Fig. 1. Structure of a neuron. The neuron receives signals from other neurons as input through a weighted connection, and then an activation function is used to produce the output y of the neuron.

We take the basic CNN as an example to briefly analyze the computational overloads required by the deep networks. A CNN is composed of layers of processing units (as shown in Fig. 1) that roughly model the computations simulating neurons. Each activation function behaves in terms of a neuron unit as follows:

$$y = g\left(\sum_{k=1}^{K} w_k i_k\right) \tag{1}$$

where w_k is a float weight, i_k is the input of the neuron, and y is the output of the neuron. K is the number of neurons connected by the neuron and $g(\cdot)$ is the nonlinear activation function. The large storage requirement of CNNs arises from the fact that the number of parameters is large and the parameter values are of the float format which is represented with long word width. In the CNN with one neuron unit formulated as (1), there are K (e.g., 10 000) parameters (e.g., w_k) for each layer and the CNN may have tens of thousands of layers. Each parameter w_k is formatted by a float value that occupies 128 bits or more. The high computational complexity of full precision CNNs is caused by a large number of float value-based operations including additions, subtractions, and multiplications. The operation in terms of (1) for each processing layer requires at least K float multiplication in addition to nonlinear procedures and additions. The deep network requires considerably multiple times of such operations.

The widely used AlexNet [42] has eight layers in total, involving 61 M weight parameters and more than 729 M FLOPs for one round of inference execution. The execution overloads of other deeper networks such as GoogLeNet and ResNet are even larger. Although these CNNs can be trained on high-performance GPUs or central processing unit (CPU) clouds efficiently, operating these deep networks on resource-limited remote sensing devices remains a challenging problem.

In order to make quick responses by using a deep network deployed on a remote sensing device, both the deep network and the remote sensing information should be compressed. This is achieved by three major procedures. First, the float values of parameter [e.g., w_k in (1)] are converted to binary representations, which save storage space. Second, based on the binarized parameters, original operations such as multiplications [e.g., K multiplications in (1)] based on float values are instead computed with binary manipulations, which

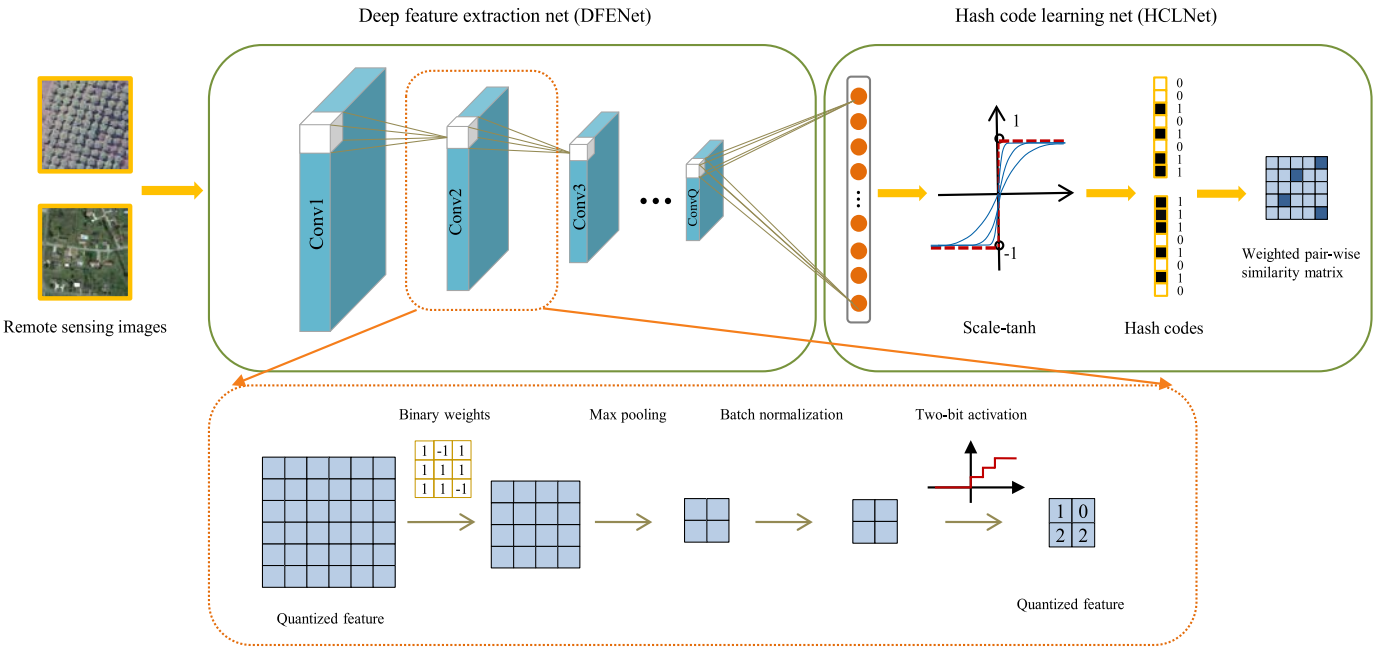


Fig. 2. Illustration of the proposed QDLH framework. The QDLH is composed of a DFENet and HCLNet. The DFENet consists of several quantized convolution layers, and it extracts deep features from the input remote sensing images. The HCLNet contains a fully connected layer and a scale-tanh layer, and it produces the binary hash codes for the input images through a weighted pairwise loss function.

saves computational power. Third, the remote sensing data are compressed in terms of hash codes whose Hamming distance characterizes data categorization. It is clear that the underlying manipulation of the three procedures transforms hash continuous representations into binary. To this end, the overall novel hashing framework will be presented in Section III.

III. Quantized Deep Learning to Hash

In this section, we introduce our QDLH framework for remote sensing image retrieval. In addition, the sections are organized as follows. Section III-A describes the overall architecture of our QDLH, which is composed of a quantized deep convolutional feature extraction net and a hash code learning net (HCLNet). Sections III-B and III-C describe these two networks in detail, separately.

A. Overall Architecture

Our QDLH framework consists of a deep feature extraction net (DFENet) and an HCLNet. As shown in Fig. 2, the DFENet can take the form of various off-the-shelf deep CNN architectures such as AlexNet, VGGNet, GoogLeNet, and ResNet. It generates high-level semantic feature representations. Different from existing DHNNs [35], which adopt full precision deep nets, we employ a quantized DFENet (Q-DFENet) for efficient image feature extraction. As illustrated in Fig. 2, we quantize the kernel weights of the convolutional layers into 1-bit forms and the activations of each layer output into 2-bit forms, separately. The HCLNet contains a fully connected layer and a scale-tanh layer. The fully connected layer maps high-dimensional features of the DFENet to a lower L-dimensional space. The scale-tanh layer further converts the L-dimensional real value feature representations into binary hash codes. In order to obtain more accurate hash codes for the remote sensing images, we adopt a class intensive pairwise loss function in the HCLNet to address the data imbalance problem. Different from the DHNNs in [35] which treat each image pair equally, our loss function makes the hash codes within a scene to have strong intraclass similarity and discriminate against a large number of images from other classes.

In one training iteration, full precision weights are computed through backpropagation. Then, the weights are quantized for efficient forward computation. The network output for the next training iteration is computed based on the quantized network. Once the whole training procedure has finished, the deep network is hashed such that a discrete representation is developed for network weight characterization and inference.

B. Q-DFENet

In this section, we first introduce our Q-DFENet in detail. The Q-DFENet is composed of many quantized convolutional layers. Different from the traditional feature extraction network used in deep remote sensing image hashing, the quantized convolutional layers we adopted in the feature extraction network have binary convolutional kernel weights and 2-bit activation functions, which lead to a lightweight network for remote sensing image inference.

1) Weight Binarization: As we have introduced in Section I, one main problem with the deep CNNs is its large memory consumption. Recently, it has been shown that weight quantization achieves a very large reduction in memory [43]. As introduced in this article, the weight quantization methods for deep nets include 2-bit quantization, ternary quantization, and binary quantization. The representation accuracy would be

higher if more bits are employed in weight quantization. However, more bits lead to larger resource consumption (i.e., storage space and computation complexity) of the compressed network. Therefore, one should seek a trade-off between accuracy and efficiency in real applications. We aim to develop a quantized DHN for remote sensing image retrieval applications on resource-limited embedded devices. Specifically, we focus on the binary weight quantization scheme because the binary convolutional operation can be realized efficiently based on the logical units such as FPGA chips. In addition, according to the previous works, the small accuracy loss of binary quantization is usually acceptable compared with the improvements for resource consumption. The experimental results validate that our binary scheme achieves efficient performance with acceptable accuracy. This strategy utilizes a binary filter $\mathbf{B} \in \{+1, -1\}^{c\times w\times h}$ where (c, w, h) represents channels, width and height respectively, and a scale parameter $\alpha > 0$ to approximate the full precision kernel weight $\mathbf{W} \in \mathrm{R}^{c\times w\times h}$, as $\mathbf{W} \approx \alpha\mathbf{B}$. We assume kernels have no bias terms. The convolutional operation is approximated as follows:

$$\mathbf{I} * \mathbf{W} \approx \alpha\,(\mathbf{I} \pm \mathbf{B}) \tag{2}$$

where $\mathbf{I}$ represents the layer input, $*$ represents convolution operations, and $\pm$ represents a convolution operation with addition and subtraction operations. Without loss of generality, let $\mathbf{B} \in \{+1, -1\}^n$ and $\mathbf{W} \in \mathrm{R}^n$ are denoted in vectors where $n = c \times w \times h$. The goal of weight quantization is to make the quantized value $\alpha\mathbf{B}$ approximate the original value $\mathbf{W}$ as much as possible. To this end, we define

$$G\,(\mathbf{B}, \alpha) = \|\mathbf{W} - \alpha\mathbf{B}\|_2^2. \tag{3}$$

By expanding (3), we obtain the following form:

$$G\,(\mathbf{B}, \alpha) = \mathbf{W}^T\mathbf{W} - 2\alpha\mathbf{W}^T\mathbf{B} + \alpha^2\mathbf{B}^T\mathbf{B}. \tag{4}$$

Thus, the optimal approximation is formulated as

$$\langle\hat{\alpha}, \hat{\mathbf{B}}\rangle = \arg\min_{\alpha,\mathbf{B}} G\,(\mathbf{B}, \alpha). \tag{5}$$

The weight $\mathbf{W}$ can be obtained when we find the optimal binary weight $\hat{\mathbf{B}}$ and scale parameter $\hat{\alpha}$ and $\mathbf{W}^T\mathbf{W}$ can be considered as a constant value in (4) and (5). In addition, $\mathbf{B}^T\mathbf{B} = n$ is also a constant value due to $\mathbf{B} \in \{+1, -1\}^n$. Furthermore, α is a positive and constant value when computing $\hat{\mathbf{B}}$, and the optimal solution of $\hat{\mathbf{B}}$ is simply obtained by

$$\hat{\mathbf{B}} = \arg\max_{\mathbf{B}} \left\{\mathbf{W}^T\mathbf{B}\right\} = \mathrm{sign}\,(\mathbf{W}) \quad \text{s.t. } \mathbf{B} \in \{+1, -1\}^n. \tag{6}$$

Then we compute for the optimal scale parameter value α. According to the above analysis of (4), the optimization function for finding α is converted to

$$G\left(\hat{\mathbf{B}}, \alpha\right) = \alpha^2 n - 2\alpha\mathbf{W}^T\hat{\mathbf{B}}. \tag{7}$$

The derivative of the above equation is

$$\frac{\partial G\left(\hat{\mathbf{B}}, \alpha\right)}{\partial\alpha} = 2\alpha n - 2\mathbf{W}^T\mathbf{B}. \tag{8}$$

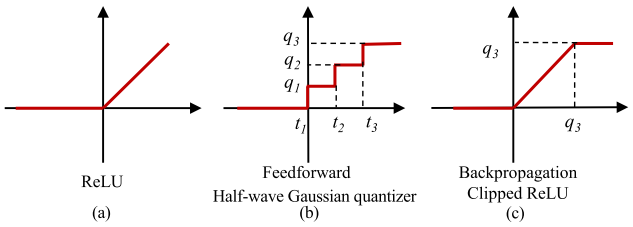


Fig. 3. Illustration of (a) ReLU function, (b) 2-bit half-wave Gaussian quantizer of ReLU, and (c) clipped ReLU function.

By setting the derivative in (8) to be zero, the optimal scale parameter $\hat{\alpha}$ is computed as follows:

$$\hat{\alpha} = \frac{\mathbf{W}^T\hat{\mathbf{B}}}{n}. \tag{9}$$

By substituting the optimal $\hat{\mathbf{B}} = \mathrm{sign}(\mathbf{W})$ into (9), the solution of optimal $\hat{\alpha}$ is given as

$$\hat{\alpha} = \frac{\mathbf{W}^T\mathrm{sign}\,(\mathbf{W})}{n} = \frac{\sum|\mathbf{W}_i|}{n} = \frac{1}{n}\|\mathbf{W}\|_{\ell 1}. \tag{10}$$

We observe that the optimal scale parameter $\hat{\alpha}$ is the mean value of ℓ_1-norm of $\mathbf{W}$. $\mathbf{B}$ is the binary representation of the full precision weights in $\mathbf{W}$. It plays a major role in compressing the deep network in terms of hashing nets. In this way, the full precision weight term $\mathbf{W}$ is quantized to the binary weight term $\mathbf{B}$, resulting in tremendous memory reduction for image inference. Furthermore, complex convolution operations can be replaced by simple addition and subtraction operations.

2) 2-Bit Activation Quantization: Although the filter weight quantization reduces the required memory and does not involve multiplication during the convolution operation, addition and subtraction operations require FLOPs because the input of the current layer and the output of the previous layer are float values. In order to further compress the convolutional network and accelerate the convolution operation, we conduct activation quantization to map the output of each layer to several discrete values. However, the activation quantization is more difficult than the weight quantization because the quantized activation functions often suffer from the problem of derivative vanishing, which gives rise to great difficulty for backpropagation learning of the whole network. A rectified linear unit (ReLU) activation function is often adopted in the deep networks and a half-wave Gaussian quantization method [44] is employed for ReLU quantization.

A ReLU retains the biological inspiration of a neuron, which is activated only when the input exceeds a threshold. The ReLU has the following form [as seen in Fig. 3(a)]:

$$g\,(x) = \begin{cases} x, & x \geq 0; \\ 0, & x < 0. \end{cases} \tag{11}$$

A quantizer $Q(x)$ is defined as a piecewise constant function that maps all values of x into a quantization level. Considering that the activation function ReLU is a half-wave rectifier and the dot-product in (1) tends to be close to a Gaussian distribution, we adopt a half-wave Gaussian quantizer for $Q(x)$

with the following form:

$$Q(x) = \begin{cases} q_i, & \text{if } x \in (t_i, t_{i+1}] \\ 0, & x \leq 0 \end{cases} \tag{12}$$

where $q_i \in \mathbb{R}^+ (i = 1, 2, \ldots, m)$ represents different quantization values and $t_i \in \mathbb{R}^+ (t_1 = 0$ and $t_{m+1} = +\infty)$ is the optimal quantization parameter for the Gaussian distribution. In addition, a constant quantization interval $\Delta = t_{i+1} - t_i$ is adopted to guarantee a uniform quantizer. The optimal quantization parameters of the uniform half-wave Gaussian quantizer q_i^* and t_i^* only depend on the mean and variance of the dot product distribution. Furthermore, q_i^* and t_i^* are obtained by the Lloyds algorithm [45], which is similar to the k-means algorithm.

We utilize a batch normalization layer before activation and it forces the responses of each convolutional layer to have zero mean and unit variance. The Lloyds algorithm is applied to the data from the entire nets to generate a single quantizer for forward inference in all layers. We adopt a 2-bit half-wave Gaussian quantizer ($m = 3$) for activation quantization, as shown in Fig. 3(b).

The half-wave Gaussian quantizer is a stepwise function whose derivative is almost zero. It may cause the vanishing gradient problem during the backpropagation learning of the deep networks. In order to address this problem, we adopt a piecewise function, that is, clipped ReLU [see Fig. 3(c)], to approximate ReLU and its half-wave Gaussian quantizer in backpropagation. The clipped ReLU is formulated as follows:

$$\widetilde{Q}_c(x) = \begin{cases} q_m, & x > q_m; \\ x, & x \in (0, q_m]; \\ 0, & \text{otherwise.} \end{cases} \tag{13}$$

The reason for using the clipped ReLU is that it not only matches with the half-wave Gaussian quantizer on the tail but also is endowed with nonzero gradients in the interval $(0, q_m]$. The clipped gradients enable a stable optimization, especially for CNNs [46]. Finally, the original activation function ReLU is quantized to a 2-bit half-wave Gaussian quantizer in the forward step and approximated with a clipped ReLU in the backpropagation.

3) Efficiency Analysis: The Q-DFENet is efficient in terms of both memory and computation. A comparison of efficiency analysis for different deep nets is shown in Fig. 4. We observe that the traditional full precision nets require addition, subtraction, and multiplication for convolutional operations. In contrast, the Q-DFENet merely requires bit count operations. In addition, about $32\times$ memory is reduced by storing the binary model parameters instead of the real values with float precision. Consequently, the Q-DFENet is very suitable to be applied to the embedded remote sensing devices with limited resources for online remote sensing image processing. The detailed experimental performance analysis for remote sensing image hashing with the Q-DFENet will be given in Section IV.

C. HCLNets

1) Hashing Layers: In this section, we describe our HCLNet. As shown in Fig. 2, the HCLNet consists of a hash layer and a scale-tanh layer that converts the extracted deep image features to binary hash codes. The fully connected hash layer maps the remote sensing images from high-dimensional deep feature space to a lower L-dimensional real value code space and the scale-tanh layer further converts the L-dimensional real value codes to binary hash codes.

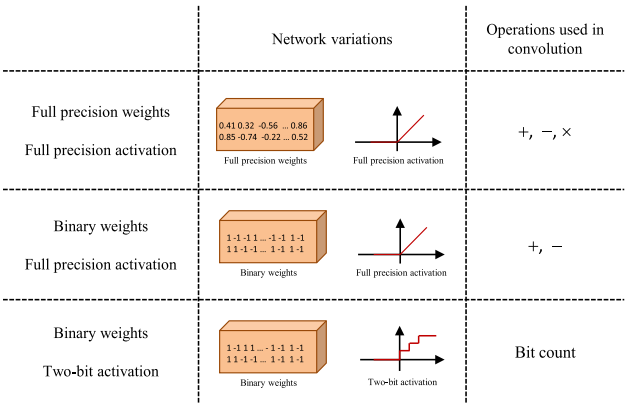


Fig. 4. Comparison of deep CNNs with different kinds of filter weights and activations.

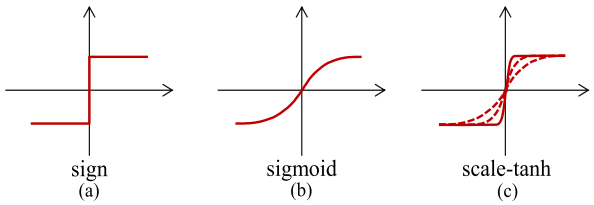


Fig. 5. Illustration of (a) sign, (b) sigmoid, and (c) scale-tanh functions.

Intuitively, the Heaviside function (i.e., sign function) may be the most effective activation function in the fully connected hash layer to produce discrete binary codes for hashing directly. However, the derivative of the sign function is zero almost everywhere, severely increasing the difficulty of the backpropagation learning for the whole network. The sigmoid function is often used for approximation in the existing deep remote sensing image hashing methods [35]. However, the sigmoid function is a continuous function and its output is not discrete. An additional threshold is usually conducted to obtain the final hash codes, and it may lead to a big quantization error and decreased retrieval performance. In order to generate binary hash codes from the network directly, inspired by hashnet [47], we place a scale-tanh layer after the fully connected hash layer to gradually approximate the sign function. The illustrations of the sign, sigmoid, and scale-tanh function are shown in Fig. 5. The scale-tanh has the following form:

$$\lim_{\tau \to \infty} \tanh(\tau y) = \text{sign}(y) = \begin{cases} +1, & y \geq 0 \\ -1, & y < 0 \end{cases} \tag{14}$$

where τ is a scale parameter. By gradually increasing τ in each iteration of the backpropagation learning, the scale-tanh comes closer to the sign function, and finally, the layer produces exactly binary hash codes without the need for an additional threshold.

2) Class Intensive Objective Function: After introducing our deep hash network structure, we describe the objective

function which is used for the whole network learning. One widely used objective function is the pairwise loss function [35] that learns the parameters and hash codes for remote sensing images. However, the DHNNs method [35] treats each image pair equally in learning. In practical remote sensing operations, different classes tend to have considerably different numbers of images. In order to address this data imbalance problem, we employ a class-intensive pairwise loss function for model training.

Given a remote sensing image data set with N images and their semantic class labels u_i, $i = 1, 2, \ldots, N$. We define $\boldsymbol{C} = \{c_{ij}\}^{N\times N}$ as a common class indicator matrix. If the image pair u_i and u_j have the same label, $c_{ij} = 1$, and otherwise $c_{ij} = 0$. In addition, $\mathbf{H} \in \{-1, +1\}^{L\times N}$ denotes the corresponding hash code matrix output from the final layer of our network for the whole data set, where each column h_i represents one image and L is the hash code length. The inner product ${h_i}^T h_j$ is used to measure the hamming distance between two hash codes. Specifically, a big ${h_i}^T h_j$ value reflects a small hamming distance. Our goal is to make the hamming distance between hash codes of similar image pairs as small as possible, and vice versa. In this scenario, for each image pair, the conditional probability $P(c_{ij}|h_i, h_j)$ is defined as

$$P(c_{ij}|h_i, h_j) = \left(\frac{1}{1+e^{-\lambda {h_i}^T h_j}}\right)^{c_{ij}} \left(1-\frac{1}{1+e^{-\lambda {h_i}^T h_j}}\right)^{1-c_{ij}} \quad (15)$$

where λ is a hyperparameter whose value is less than 1. Specifically, if the image pair u_i and u_j are similar, a large value of inner product ${h_i}^T h_j$ is obtained, leading to a large value of $P(c_{ij} = 1|h_i, h_j)$, and vice versa. Then the log-likelihood of $\mathbf{H}$ can be derived as follows:

$$\log P\left(\mathbf{C}|\mathbf{H}\right) = \sum_{i,j=1}^{N} \left[\log\left(1 + e^{\lambda {h_i}^T h_j}\right) - \lambda c_{ij}\left({h_i}^T h_j\right)\right]. \quad (16)$$

The above log-likelihood function in (16) has been widely adopted in deep hashing included [35]. However, data imbalance normally exists in remote sensing image hashing and retrieval. Specifically, the number of relevant images to the target image is usually far less than the number of irrelevant images in the data set. Therefore, we employ a weight parameter γ_{ij} to address this data imbalance problem [37], which leads to a weighted log-likelihood function as follows:

$$\begin{aligned} L &= \sum_{i,j=1}^{N} \gamma_{ij} \log P\left(c_{ij}|h_i, h_j\right) \\ &= \sum_{i,j=1}^{N} \gamma_{ij} \left[\log\left(1 + e^{\lambda {h_i}^T h_j}\right) - \lambda c_{ij}\left({h_i}^T h_j\right)\right] \end{aligned} \quad (17)$$

where

$$\gamma_{ij} = \begin{cases} N/N_i, & c_{ij} = 1 \\ N/\left(N - N_i\right), & c_{ij} = 0 \end{cases} \quad (18)$$

and N_i represents the number of images relevant to the ith image. γ_{ij} is the weight for each training pair, which is used to tackle the data imbalance problem by weighting the training pairs according to the importance of misclassifying that pair. Finally, the class-intensive log-likelihood loss function is formulated by minimizing the following problem for the whole network:

$$\min_{\Omega} L = \min_{\Omega} \sum_{i,j=1}^{N} \gamma_{ij} \left[\log\left(1 + e^{\lambda {h_i}^T h_j}\right) - \lambda c_{ij}\left({h_i}^T h_j\right)\right] \quad (19)$$

where Ω represents the whole set of parameters for the proposed quantized DHN. The backpropagation algorithm is applied to computing the optimal values for the parameters. When $N_i < N - N_i$, the weight parameter $\gamma_i j$ enables the image similarities within the same remote sensing scene to play a more dominant role than those from different scenes in the learning procedure. Thus, we obtain more accurate hash codes for the remote sensing images and the retrieval performance is improved over subtle quantization errors.

D. Observations

Section III-B has described how to quantize a full precision deep network into that with binary weights. Section III-C has described how to binarize the network outputs, that is, enabling the network to output a piece of hashing code for representing one remote sensing image. Studies on hashing models [43], [48] into quantized representations have been recently conducted in the literature. Most of these quantized models are applied to tasks such as regression and recognition, which tend to transform images into continuous outputs, not binary codes. In addition, the research on hashing images into binary codes has been going on for decades [49], [50] but the processing models for hashing images are not hashed themselves but represented with full precision. This renders a seemingly contradictory situation that hashed models is not used for hashing data, and on the contrary, hash codes are generated by complicated models that are not hashed. To close this gap, we propose to hash nets for hashing data. We refer to the strategy as hashing nets for hashing. Hashing nets are realized by the quantized deep nets presented in Section III-B. Hashing data are generated by the quantized deep network as well as the network output characterization presented in Section III-C. To the best of our knowledge, we are among the first to develop the strategy of hashing nets for hashing. Experimental evaluations in Section IV will validate the effectiveness and efficiency of our proposed framework.

IV. Experimental Results

In this section, we conduct extensive experiments on two public remote sensing image data sets to evaluate the efficiency and effectiveness of the proposed QDLH framework for remote sensing image retrieval. Section IV-A introduces the experimental environment, parameter settings, information of the data sets and neural network structure used in the experiments. Sections IV-B and IV-C analyze the evaluation results of the proposed framework.

A. Experimental Settings and Evaluation Metrics

We conduct our experiments on two benchmark remote sensing image data sets.

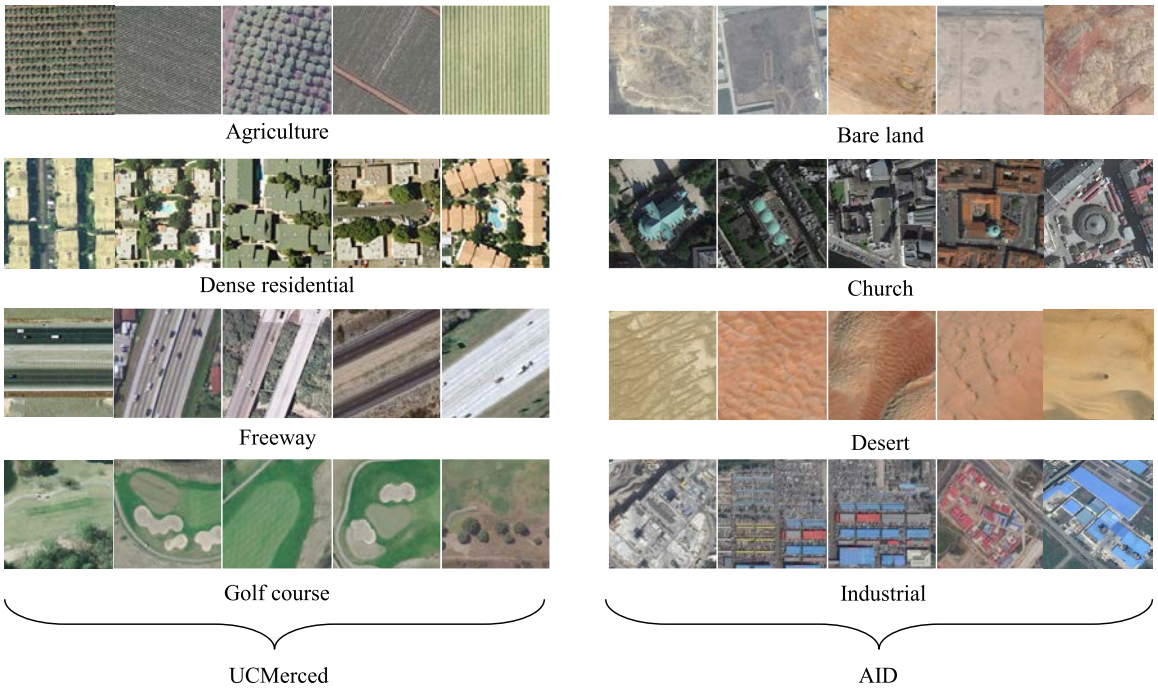


Fig. 6. Sample images from the UCMerced and AID data sets.

The UCMerced data set [51] was produced by the University of California. These images are manually extracted from the U.S. national city map produced by the U.S. Geological Survey. It contains 21 land cover classes, each of which includes 100 images. Each image has 256×256 pixels and the pixel resolution is one foot. Similar to DHNNs [35], in order to augment the UCMerced data set for deep neural network training, we rotate the original images by 90°, 180°, and 270°, separately. In this way, the volume of the UCMerced data set is increased to 8400. In our experiments, 7400 images are used as the retrieval database and to train our quantized DHN and the remaining 1000 images are used as query data for testing.

The AID data set [52] is a large-scale remote sensing image data set produced by Wuhan University in 2017. The data set was taken from Google Earth images and tagged by experts in the field of remote sensing image interpretation. The AID data set contains 10 000 images from 30 different scenes in total. Each scene has 200–400 images and each image has 256×256 pixels. The AID data set contains 10 000 remote sensing images which are sufficient for training deep nets. We do not conduct image rotation for the AID data set. We randomly select 8000 images as training data, and the remaining 2000 images are used as testing data. We give some sample images about the above two data sets in Fig. 6.

The proposed QDLH framework is composed of two parts: the Q-DFENet and the HCLnet. In order to verify the effectiveness and generality of our QDLH framework, we evaluate several popular deep CNNs such as AlexNet, ResNet-18, VGGNet, and GoogLeNet as prototype models of the QDLH in the experiments. AlexNet was presented by Krizhevsky *et al.* [42] in 2012 and won the 2012 ILSVRC competition. AlexNet uses an eight-layer neural network which consists of five convolutional layers and three fully connected layers, including 630 million links, 60 million parameters, and 650 000 neurons. VGGNet was proposed by Simonyan and Zisserman [38] in 2014. VGGNet has two versions: VGG16 and VGG19. The only difference between them is the network depth. In our experiments, VGG16 is adopted, which consists of 16 layers (13 convolutional layers and three fully connected layers). GoogLeNet was proposed by Szegedy *et al.* [53] in 2014. Different from VGGNet which inherited the framework of AlexNet, GoogLeNet has made an innovative attempt on the network architecture. Although GoogLeNet has 22 layers, the model size is much smaller than AlexNet and VGGNet, and its performance is also superior. ResNet was presented by He *et Al.* [54] in 2015. It won first place in several computer competition vision. ResNet is mainly composed of residual blocks. The residual block is implemented by a shortcut connection. This operation does not add extra parameters and calculations to the network while both the training speed and performance are greatly increased. The residual network has a variety of structures and ResNet-18 is employed in our experiments. The ResNet-18 consists of a convolutional layer, a fully connected layer, and four residual blocks. Each residual block in ResNet-18 contains four convolutional layers, resulting in 18 layers in total.

The experiments are all run on a platform with Centos7 and tesla k80 graphic processing units. Our experiments are implemented based on the **Caffe** framework.[1] The publicly available pretrained network is as initialization for training. We fine-tune the Q-DFENet and HCLNet in our QDLH framework jointly via backpropagation to learn the model and hash codes for remote sensing images. The parameters for

[1] http://caffe.berkeleyvision.org/

network fine-tuning in the experiments are set as follows. We use minibatch stochastic gradient descent (SGD) with 0.9 momentum and the learning rate annealing strategy implemented in Caffe. The training policy is a "step" with a step size of 2000. We set the base learning rate to be 0.0001 and set the maximum number of iterations to be 10 000. The weight decay parameter is set to be 0.0002.

In scale-tanh optimization stage, the parameter of scale-tanh is updated by $\tau_{t+1} = \tau_t(1+G*\text{iter})^P$, where iter represents the number of training iterations. In experiments, we set $\tau_0 = 1$, $G = 0.005$, and $P = 0.5$. The optimal step size is set to 200. We can observe that τ is gradually increased in each step and finally the scale-tanh function comes very close to the sign function.

After generating the hash codes for all remote sensing images, the fast image retrieval is conducted by calculating the Hamming distance between different codes. In order to evaluate the retrieval performance, mean average precision (MAP) is used as the evaluation criterion and it is computed as follows:

$$\text{MAP} = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{n_i}\sum_{k=1}^{n_i}\text{precision}\,(R_{ik}) \tag{20}$$

where $q_i \in Q$ is a query image, and N_i is the number of images relevant to q_i in the retrieval database. Suppose that the relevant images are ordered as $\{r_i, r_2, \ldots, r_{N_i}\}$, R_{ik} is the set of ranked retrieval results from the top result until we get to point r_k. We report the MAP score of the top 50 images in the experimental results. In addition, the precision with respect to the top K retrieved examples (Precision@K) and precision-recall curves (PR-curves) are evaluated for the compared approaches. The t-distribution stochastic neighbor embedding (t-SNE) [55] is a method for dimensionality reduction. t-SNE has the ability to maintain the local structure of the original data. The low-dimensional data processed by t-SNE reflects the distance distribution of the original data in high-dimensional space. Therefore, t-SNE reflects the clustering characteristics of the generated hash codes. Typically, t-SNE transfers raw data into 2-D representations to produce an intuitive visualization. The lengths of hash codes are set to be 32, 64, and 96 separately in our experiments.

B. Comparison With the Full Precision Networks

In this section, we first evaluate the efficiency and effectiveness of the proposed QDLH approach by comparing it with full precision networks. Four different networks (AlexNet, VGGNet, GoogLeNet, and ResNet18) are tested in the experiments. We denote Alex-Q, VGG-Q, GoogLe-Q, and Res18-Q as the quantized deep hashing methods with our proposed framework. For comparison, we utilize Alex-F, VGG-F, GoogLe-F, and Res18-F to represent full precision deep hashing methods that replace the Q-DFENet in our framework with corresponding full precision nets. Tables II and III show the MAP scores on the two remote sensing data sets with both full precision and quantized DHNs. The deep hashing methods with full precision nets outperform the quantized deep hashing models. It is easy to understand that the full precision nets capture more information when extracting deep features from remote sensing images. On the UCMerced data set, all of the four quantized deep hashing approaches achieve very close retrieval accuracy to the full precision nets. On the AID data set, the performance degradation for GoogLe-Q is more obvious than the other three quantized nets. However, the results of GoogLe-Q are still competitive to those of the other methods. There is no significant degradation of MAP scores for the different quantized deep hashing models on the two data sets. We give visual examples to show the retrieved image results by both the quantized and full precision deep hashing nets with VGGNet on UCMerced data set in Fig. 7, from which we can make the same observations as in the aforementioned analysis.

TABLE II

MAP COMPARISON OF FULL PRECISION DEEP MODELS AND QUANTIZED DEEP MODELS FOR REMOTE SENSING IMAGE RETRIEVAL ON UCMERCED DATA SET WITH DIFFERENT HASH CODE LENGTHS

Nets	32 bit	64 bit	96 bit
Alex-F	0.9681	0.9764	0.9846
Alex-Q	0.9656	0.9749	0.9727
gap	**-0.0025**	**-0.0015**	**-0.0119**
Res18-F	0.9960	0.9964	0.9959
Res18-Q	0.9659	0.9770	0.9821
gap	**-0.0301**	**-0.0194**	**-0.0138**
VGG-F	0.9775	0.9801	0.9988
VGG-Q	0.9552	0.9698	0.9857
gap	**-0.0223**	**-0.0103**	**-0.0131**
GoogLe-F	0.9853	0.9940	0.9976
GoogLe-Q	0.9677	0.9721	0.9791
gap	**-0.0176**	**-0.0219**	**-0.0185**

TABLE III

MAP COMPARISON OF FULL PRECISION DEEP MODELS AND QUANTIZED DEEP MODELS FOR REMOTE SENSING IMAGE RETRIEVAL ON AID DATA SET WITH DIFFERENT HASH CODE LENGTHS

Nets	32 bit	64 bit	96 bit
Alex-F	0.8974	0.9402	0.9561
Alex-Q	0.8871	0.9175	0.9296
gap	**-0.0103**	**-0.0227**	**-0.0265**
Res18-F	0.8823	0.8988	0.9303
Res18-Q	0.8647	0.8921	0.9042
gap	**-0.0176**	**-0.0067**	**-0.0261**
VGG-F	0.9036	0.9363	0.9339
VGG-Q	0.8517	0.9086	0.9161
gap	**-0.0519**	**-0.0277**	**-0.0178**
GoogLe-F	0.9045	0.9425	0.9635
GoogLe-Q	0.8616	0.8982	0.9131
gap	**-0.0429**	**-0.0443**	**-0.0504**

The main advantages of the quantized DHNs are inference efficiency and memory economy. Therefore, we conduct an experiment to measure the time consumed by the full precision and quantized deep hashing models for image inference. We first ran the different trained models on the same testing set and obtained the total processing time for all the

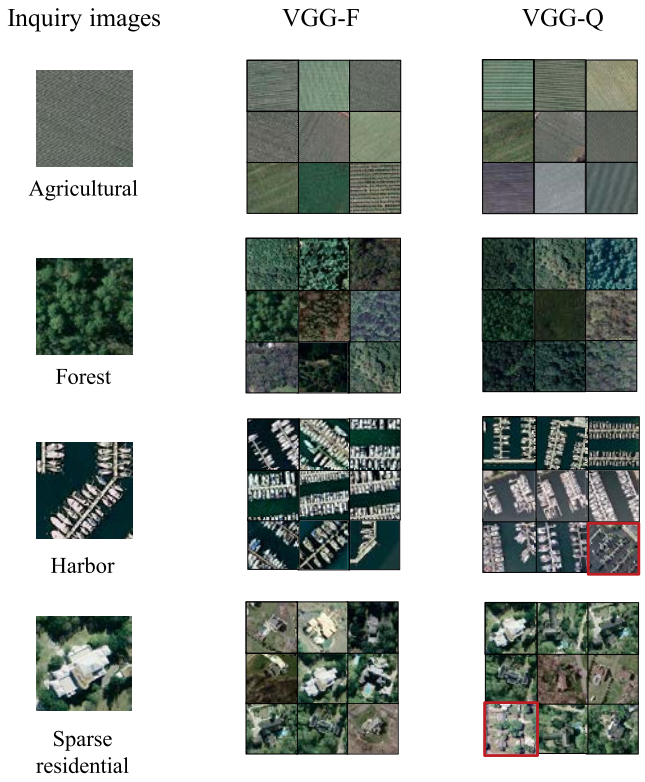


Fig. 7. Visual retrieval results of VGG-F and VGG-Q on UCMerced. The false results are marked with red rectangles.

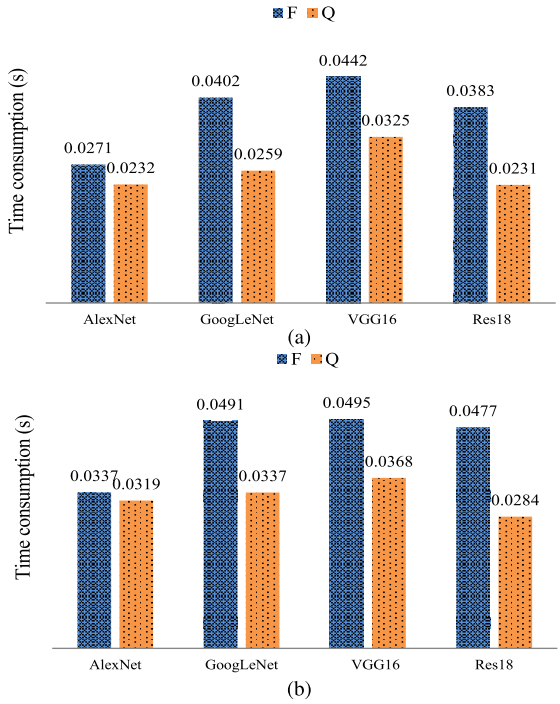


Fig. 8. Time(s) consumption of generating hash codes for the testing images of AID data set with different models. F: full precision networks. Q: quantized networks. We use the average time of an image to estimate time consumption. (a) UCMerced. (b) AID.

testing images. Then, the time consumption needed for one single image is calculated by averaging the running time on the size of the testing image set. The average time needed to generate the hash codes for testing images with different models is shown in Fig. 8. We observe that the quantized deep hashing methods are considerably faster than the full precision models from the obtained comparative results. Moreover, the efficiency improvement is more obvious especially for the deeper networks with more convolutional operations. For example, for ResNet-18 with 18 layers, the time required for generating 96-bit hash codes for one image has decreased by more than 40% on the two data sets. Furthermore, the binary weight parameters save 32× storage space and the 2-bit activated output features lead to about 16× reduction compared with original 32-bit float precision values. Therefore, with regard to the little accuracy loss, the overall performance improvement of our proposed QDLH framework is quite impressive for large-scale remote sensing image retrieval. This validates that the proposed quantized deep hashing framework is suitable to be deployed in the resource-limited remote sensing devices for on-orbit processing.

C. Comparison With State-of-the-Art Methods

In this section, we compare our proposed deep remote sensing image hashing framework with state-of-the-art algorithms. The competing approaches include hashing methods proposed for remote sensing image retrieval in recent years, such as partial randomness hashing (PRH) [19], kernel-based unsupervised hashing (KULSH) [17], kernel-based supervised hashing (KSLSH) [17], and DHNNs [35]. Moreover, we compare it against some representative hashing methods used in the computer vision field in the experiments, such as LSH [9], supervised discrete hashing (SDH) [56], column sampling-based discrete supervised hashing (COSDISH) [57], DHN [49], deep supervised hashing (DSH) [50], and deep pairwise-supervised hashing (DPSH) [58]. Among the various comparison methods, LSH, KULSH, and PRH are unsupervised hashing approaches that do not use label information for hash code generation, and the rest are supervised methods. Moreover, LSH, KULSH, PRH, KSLSH, SDH, and COSDISH are shallow methods and the remaining ones are deep hashing methods based on CNNs. For a fair comparison, we use the fc-7 features of a pretrained AlexNet as inputs for the shallow methods and the raw remote sensing images as inputs for the deep models.

The MAP scores of different hashing methods for remote sensing image retrieval on the two data sets are shown in Table IV. The results of the compared methods are produced by the public codes downloaded from the authors' website except DHNNs because we do not find its public source. The MAP results for DHNNs on the UCMerced data set are referenced from the article [35] with the same experimental settings but the results on the AID data set are not reported in the original article. We show the results of our proposed approach based on two deep nets: AlexNet and ResNet-18. Similarly, Alex-F and Res18-F represent the full precision models and Alex-Q and Res18-Q denote the quantized deep hashing methods. We observe in Table IV that PRH and DHNNs achieve the best performance among the traditional

TABLE IV
MAP OF PROPOSED HASHING MODELS AND EXISTING HASHING METHODS

Methods	Description	UCMerced			AID		
		32 bit	64 bit	96 bit	32 bit	64 bit	96 bit
LSH [9]	unsupervised / shallow	0.3886	0.5141	0.5540	0.2980	0.4058	0.4913
KULSH [17]	unsupervised / shallow	0.5379	0.6246	0.6566	0.4146	0.5085	0.5793
PRH [19]	unsupervised / shallow	0.5717	0.6561	0.6769	0.4754	0.5598	0.5968
KSLSH [17]	supervised / shallow	0.8874	0.9023	0.9128	0.7215	0.7614	0.7629
SDH [56]	supervised / shallow	0.9119	0.9342	0.9320	0.7704	0.8320	0.8386
COSDISH [57]	supervised / shallow	0.8713	0.8704	0.8776	0.8589	0.8744	0.8771
DSH [50]	supervised / deep	0.6317	0.6750	0.7502	0.4191	0.4585	0.4636
DHN [49]	supervised / deep	0.6707	0.7313	0.7707	0.6953	0.7464	0.7682
DPSH [58]	supervised / deep	0.7478	0.8174	0.8640	0.3008	0.3394	0.3546
DHNNs [35]	supervised / deep	0.9396	0.9718	0.9762	-	-	-
Ours(Alex-F)	supervised / deep	0.9681	0.9764	0.9846	0.8974	0.9402	0.9561
Ours(Alex-Q)	supervised / deep	0.9656	0.9749	0.9727	0.8871	0.9175	0.9296
Ours(Res18-F)	supervised / deep	0.9960	0.9964	0.9959	0.8823	0.8988	0.9503
Ours(Res18-Q)	supervised / deep	0.9659	0.9770	0.9821	0.8647	0.8921	0.9042

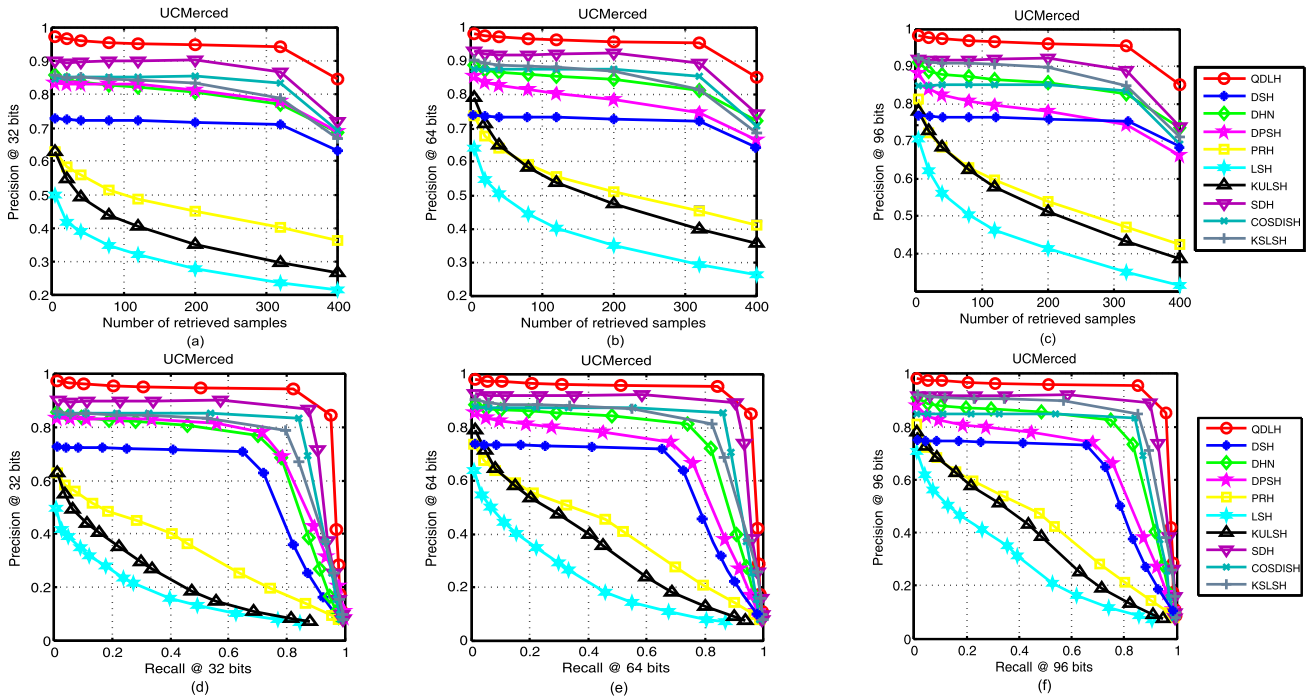


Fig. 9. Precision curves with respect to different number of retrieved images under (a) 32-bits, (b) 64-bits and (c) 96-bits, and precision-recall curves under (d) 32-bits, (e) 64-bits and (f) 96-bits, for different hashing approaches on the UC Merced data set.

unsupervised and supervised remote sensing image hashing methods, respectively. In general, the deep hashing methods outperform the shallow approaches. However, by employing the extracted deep features as inputs, the supervised shallow methods even achieve better results than some deep hashing approaches. Among all the compared methods, our proposed deep hashing approach obtains the highest MAP scores on the two data sets. By comparing our method Alex-F with the deep hashing methods such as DHN, DPSH, and DHNNs which use Alexnet for deep feature extraction, we can see that further improvements are achieved on the two data sets. This validates the effectiveness of our HCLNets because we employ a weighted pairwise loss function for hash code learning and the comparison methods treat each image pair equally in the learning procedure. More interestingly, the traditional deep hashing methods seem not to work well on the AID remote sensing data sets. The reason may be the data-imbalance problem arising in the data set. The number of images in the AID data set varies greatly between different classes, and the traditional deep hashing methods may not generate accurate hash codes by neglecting the data-imbalance problem. However, our proposed deep hashing framework achieves consistent performance on different data sets by introducing a class-intensive pairwise weight for different remote

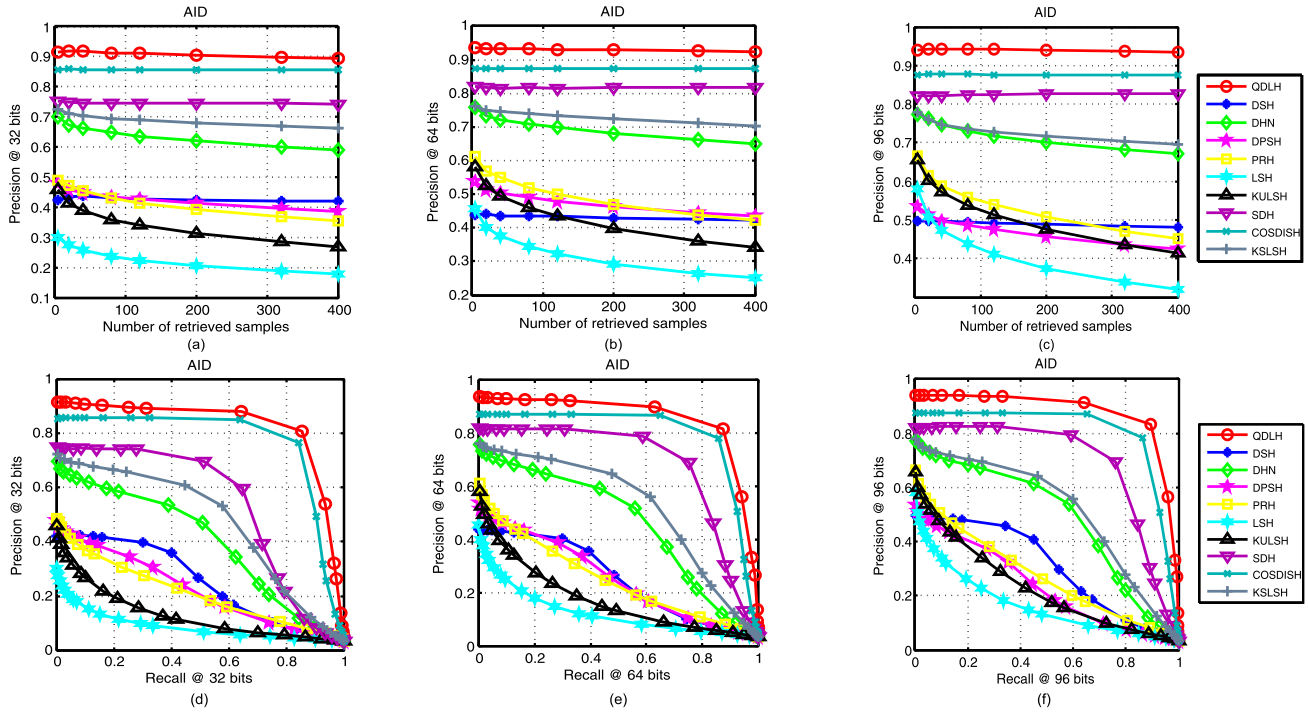


Fig. 10. Precision curves with respect to different number of retrieved images under (a) 32-bits, (b) 64-bits and (c) 96-bits, and precision-recall curves under (d) 32-bits, (e) 64-bits and (f) 96-bits, for different hashing approaches on the AID data set.

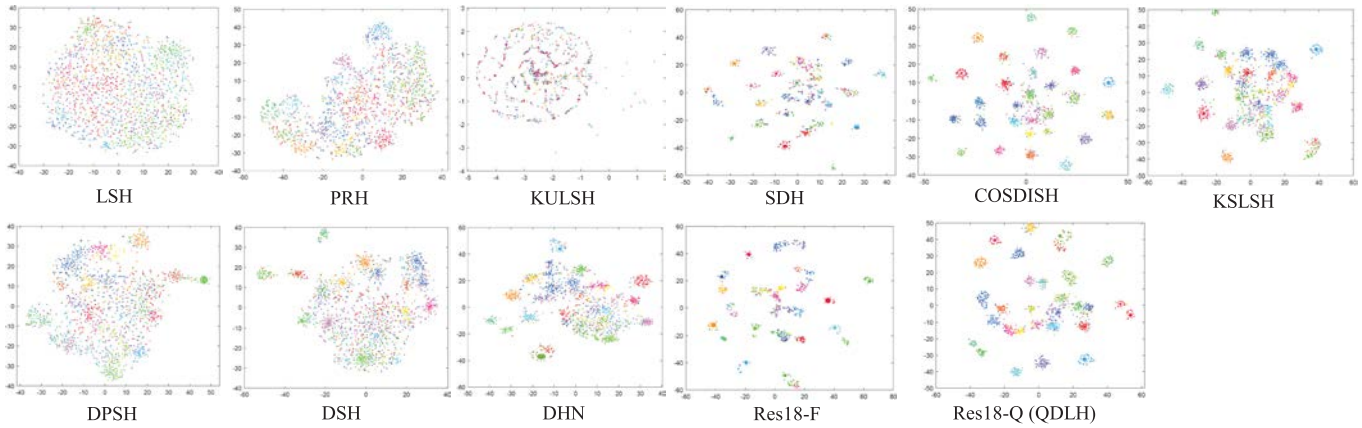


Fig. 11. t-SNE of hash codes generated by comparison methods and proposed QDLH on AID data set (32 bit).

sensing images. By comparing our proposed quantized deep hashing methods Alex-Q and Res18-Q with other traditional approaches, we can see that the proposed QDLH method consistently outperforms the others. More importantly, as shown in Fig. 8, the quantized deep hashing nets operate much faster than traditional deep hashing methods with full precision nets for remote sensing image processing.

To further illustrate the superiority of our proposed QDLH framework for remote sensing image retrieval, we illustrate the curves of precision and recall for different methods. Figs. 9 and 10 show the precision curves with different retrieved examples and overall PR curves on the two data sets, respectively. We report the curves of our QDLH method based on quantized AlexNet (Alex-Q) for comparison.

According to Figs. 9 and 10, by changing the number of retrieved images, the precision of our QDLH method consistently outperforms the alternatives. Compared with the other hashing methods, although our QDLH approach employs quantized deep nets instead of full-precision nets for image feature extraction, our retrieval accuracy is higher than 90% in most cases. Our method is superior to the compared methods in terms of both the retrieval precision and recall rate. This further validates the effectiveness of our hash code learning scheme with a weighted class-intensive objective function for remote sensing images. The PR curves reflect the overall image retrieval performance of the compared methods. The area under the PR curve is large when good performance is achieved. In addition, we visualize the t-SNE distance between

the generated hash codes of comparison methods and our proposed QDLH (Res18-Q) on AID data set in Fig. 11. The t-SNE results reflect the clustering distributions of hash codes and the different color dots represent hash codes from different classes. According to Fig. 11, the t-SNE results of hash codes generated by our proposed model QDLH show better intraclass cohesion than that from the other methods, which further validates that the class-intensive loss function in our proposed method is effective for gathering similar hash codes. The detailed results in Figs. 9–11 are consistent with the trends in the above experiments. This also shows the superiority of our QDLH method. In a nutshell, our QDLH approach is both an efficient and effective deep hashing framework for remote sensing image retrieval. It further shows significant potential for resource-limited remote sensing applications.

V. Conclusion

In this article, we have presented a novel quantized deep hashing model for remote sensing image retrieval. The proposed quantized hashing model is composed of two modules: the Q-DFENet and the HCLNet. The Q-DFENet is designed for high-level semantic feature representation of remote sensing images. However, different from existing deep hashing methods using full-precision DFENets, our Q-DFENet contains several quantized convolutional layers with binary filter weights and 2-bit activations. The Q-DFENet is quite efficient for inference and leads to significant memory and computation savings compared with the full-precision nets. The HCLNet converts the extracted high-level semantic features to binary hashing codes for fast remote sensing image retrieval. In our HCLNet, a class-intensive pairwise entropy loss and a scale-tanh activation function are used to handle the data-imbalance problem and generate more accurate hash codes. We have conducted extensive experiments on two public remote sensing image sets UCMerced and AID to evaluate the performance of our proposed framework. The experimental results have shown that our QDLH approach obtains significant memory and computation efficiency with small precision degradation compared with those used full-precision deep nets for hash code learning. In addition, compared with state-of-the-art methods, our proposed quantized deep hashing method has achieved promising retrieval performance for remote sensing images, which has further validated the superiority of our proposed approach. Therefore, our quantized deep hashing model is both efficient and effective for large-scale remote sensing image retrieval. It provides a possible technical solution for practical applications on resource-limited remote sensing devices. In future developments, we plan to deploy the proposed quantized deep hashing framework on mobile remote sensing devices (such as UAVs) to validate its performance in real applications.

References

[1] S. E. Cox and D. T. Booth, "Shadow attenuation with high dynamic range images: Creating RGB images that allow feature classification in areas otherwise obscured by shadow or oversaturation," *Environ. Monitor. Assessment*, vol. 158, pp. 231–241, Nov. 2009.

[2] X. Bai, H. Zhang, and J. Zhou, "VHR object detection based on structural feature extraction and query expansion," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 10, pp. 6508–6520, Oct. 2014.

[3] C. Wang, X. Bai, S. Wang, J. Zhou, and P. Ren, "Multiscale visual attention networks for object detection in VHR remote sensing images," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 2, pp. 310–314, Feb. 2019.

[4] L. Wu, R. Jin, and A. K. Jain, "Tag completion for image retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 3, pp. 716–727, Mar. 2013.

[5] A. W. M. Smeulders, T. S. Huang, and T. Gevers, "Content-based image retrieval," *Int. J. Comput. Vis.*, vol. 56, no. 1, pp. 5–6, 2004.

[6] Y. Yang and S. Newsam, "Geographic image retrieval using local invariant features," *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 2, pp. 818–832, Feb. 2013.

[7] S. Har-Peled and N. Kumar, "Approximate nearest neighbor search for low-dimensional queries," *SIAM J. Comput.*, vol. 42, no. 1, pp. 138–159, Jan. 2013.

[8] J. Wang, T. Zhang, J. Song, N. Sebe, and H. T. Shen, "A survey on learning to hash," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 769–790, Apr. 2018.

[9] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, "Locality-sensitive hashing scheme based on p-stable distributions," in *Proc. Symp. Comput. Geometry*, 2004, pp. 253–262.

[10] Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing," in *Proc. Neural Inf. Process. Syst.*, 2008, pp. 1753–1760.

[11] Y. Gong and S. Lazebnik, "Iterative quantization: A procrustean approach to learning binary codes," in *Proc. CVPR*, Jun. 2011, pp. 817–824.

[12] J. Wang, S. Kumar, and S.-F. Chang, "Semi-supervised hashing for large-scale search," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 12, pp. 2393–2406, Dec. 2012.

[13] L. Zhou, X. Bai, X. Liu, J. Zhou, and E. R. Hancock, "Learning binary code for fast nearest subspace search," *Pattern Recognit.*, vol. 98, Feb. 2020, Art. no. 107040.

[14] X. Bai, C. Yan, H. Yang, L. Bai, J. Zhou, and E. R. Hancock, "Adaptive hash retrieval with kernel based similarity," *Pattern Recognit.*, vol. 75, pp. 136–148, Mar. 2018.

[15] Z. Li and J. Tang, "Weakly supervised deep metric learning for community-contributed image retrieval," *IEEE Trans. Multimedia*, vol. 17, no. 11, pp. 1989–1999, Nov. 2015.

[16] N. Lukac, B. Zalik, S. Cui, and M. Datcu, "GPU-based kernelized locality-sensitive hashing for satellite image retrieval," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2015, pp. 1468–1471.

[17] B. Demir and L. Bruzzone, "Hashing-based scalable remote sensing image search and retrieval in large archives," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 2, pp. 892–904, Feb. 2016.

[18] T. Reato, B. Demir, and L. Bruzzone, "An unsupervised multicode hashing method for accurate and scalable remote sensing image retrieval," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 2, pp. 276–280, Feb. 2019.

[19] P. Li and P. Ren, "Partial randomness hashing for large-scale remote sensing image retrieval," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 3, pp. 464–468, Mar. 2017.

[20] P. Li, X. Zhang, X. Zhu, and P. Ren, "Online hashing for scalable remote sensing image retrieval," *Remote Sens.*, vol. 10, no. 5, pp. 709–723, 2018.

[21] D. Ye, Y. Li, C. Tao, X. Xie, and X. Wang, "Multiple feature hashing learning for large-scale remote sensing image retrieval," *ISPRS Int. J. Geo-Inf.*, vol. 6, no. 11, p. 364, 2017.

[22] C. Li, F. Wei, W. Dong, X. Wang, Q. Liu, and X. Zhang, "Dynamic structure embedded online multiple-output regression for streaming data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 2, pp. 323–336, Feb. 2019.

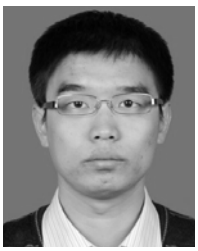
[23] C. Li, X. Wang, W. Dong, J. Yan, Q. Liu, and H. Zha, "Joint active learning with feature selection via CUR matrix decomposition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 6, pp. 1382–1396, Jun. 2019.

[24] C. Li, Q. Liu, J. Liu, and H. Lu, "Ordinal distance metric learning for image ranking," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 7, pp. 1551–1559, Jul. 2015.

[25] X. Zhu, J. Liu, J. Wang, C. Li, and H. Lu, "Sparse representation for robust abnormality detection in crowded scenes," *Pattern Recognit.*, vol. 47, no. 5, pp. 1791–1799, May 2014.

[26] X. Zhu, Z. Li, X.-Y. Zhang, C. Li, Y. Liu, and Z. Xue, "Residual invertible spatio-temporal network for video super-resolution," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, Jul. 2019, pp. 5981–5988.

[27] X.-Y. Zhang, C. Li, H. Shi, X. Zhu, P. Li, and J. Dong, "AdapNet: Adaptability decomposing encoder-decoder network for weakly supervised action recognition and localization," 2019, *arXiv:1911.11961*. [Online]. Available: http://arxiv.org/abs/1911.11961
[28] X.-Y. Zhang, S. Wang, and X. Yun, "Bidirectional active learning: A two-way exploration into unlabeled and labeled data set," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 26, no. 12, pp. 3034–3044, Dec. 2015.
[29] X. Zhang, H. Shi, C. Li, and P. Li, "Multi-instance multi-label action recognition and localization based on spatio-temporal pre-trimming for untrimmed videos," in *Proc. 34th AAAI Conf. Artif. Intell. (AAAI)*, New York, NY, USA, Feb. 2020.
[30] Z. Li, J. Tang, and T. Mei, "Deep collaborative embedding for social image understanding," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 9, pp. 2070–2083, Sep. 2019.
[31] Y. Kim, "Convolutional neural networks for sentence classification," in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1746–1751.
[32] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
[33] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. Comput. Vis. Pattern Recognit.*, 2015, pp. 3431–3440.
[34] J. Tang and Z. Li, "Weakly supervised multimodal hashing for scalable social image retrieval," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 10, pp. 2730–2741, Oct. 2018.
[35] Y. Li, Y. Zhang, X. Huang, H. Zhu, and J. Ma, "Large-scale remote sensing image retrieval by deep hashing neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 2, pp. 950–965, Feb. 2018.
[36] Y. Li, Y. Zhang, X. Huang, and J. Ma, "Learning source-invariant deep hashing convolutional neural networks for cross-source remote sensing image retrieval," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 11, pp. 6521–6536, Nov. 2018.
[37] L. Han, P. Li, X. Bai, C. Grecos, X. Zhang, and P. Ren, "Cohesion intensive deep hashing for remote sensing image retrieval," *Remote Sens.*, vol. 12, no. 1, p. 101, Dec. 2019.
[38] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent.*, 2015.
[39] S. Han, J. Pool, J. Tran, and W. J. Dally, "Learning both weights and connections for efficient neural network," in *Proc. Neural Inf. Process. Syst.*, 2015, pp. 1135–1143.
[40] D. D. Lin, S. S. Talathi, and V. S. Annapureddy, "Fixed point quantization of deep convolutional networks," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 2849–2858.
[41] M. Courbariaux, Y. Bengio, and J. David, "BinaryConnect: Training deep neural networks with binary weights during propagations," in *Proc. Conf. Workshop Neural Inf. Process. Syst.*, 2015, pp. 3123–3131.
[42] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Neural Inf. Process. Syst.*, 2012, vol. 141, no. 5, pp. 1097–1105.
[43] M. Rastegari, V. Ordonez, J. Redmon, and A. Farhadi, "XNOR-Net: ImageNet classification using binary convolutional neural networks," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 525–542.
[44] Z. Cai, X. He, J. Sun, and N. Vasconcelos, "Deep learning with low precision by half-wave Gaussian quantization," in *Proc. Comput. Vis. Pattern Recognit.*, 2017, pp. 5406–5414.
[45] S. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inf. Theory*, vol. IT-28, no. 2, pp. 129–137, Mar. 1982.
[46] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 1310–1318.
[47] Z. Cao, M. Long, J. Wang, and P. S. Yu, "HashNet: Deep learning to hash by continuation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5609–5618.
[48] P. Wang, Q. Hu, Y. Zhang, C. Zhang, Y. Liu, and J. Cheng, "Two-step quantization for low-bit neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4376–4384.
[49] H. Zhu, M. Long, J. Wang, and Y. Cao, "Deep hashing network for efficient similarity retrieval," in *Proc. AAAI Conf. Artif. Intell.*, 2016, pp. 2415–2421.
[50] H. Liu, R. Wang, S. Shan, and X. Chen, "Deep supervised hashing for fast image retrieval," in *Proc. Comput. Vis. Pattern Recognit.*, 2016, pp. 2064–2072.
[51] Y. Yang and S. D. Newsam, "Bag-of-visual-words and spatial extensions for land-use classification," in *Proc. Adv. Geograph. Inf. Syst.*, 2010, pp. 270–279.
[52] G. S. Xia *et al.*, "AID: A benchmark data set for performance evaluation of aerial scene classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 7, pp. 3965–3981, Apr. 2017.
[53] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.
[54] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
[55] J. Donahue *et al.*, "DeCAF: A deep convolutional activation feature for generic visual recognition," 2013, *arXiv:1310.1531*. [Online]. Available: http://arxiv.org/abs/1310.1531
[56] F. Shen, C. Shen, W. Liu, and H. T. Shen, "Supervised discrete hashing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 37–45.
[57] W. C. Kang, W. J. Li, and Z. H. Zhou, "Column sampling based discrete supervised hashing," in *Proc. AAAI Conf. Artif. Intell.*, 2016, pp. 1230–1236.
[58] W. Li, S. Wang, and W. C. Kang, "Feature learning based deep supervised hashing with pairwise labels," in *Proc. Int. Joint Conf. Artif. Intell.*, 2016, pp. 1711–1717.



Peng Li received the B.E. degree in automation from Shandong University, Jinan, China, in 2008, and the Ph.D. degree in pattern recognition and intelligent systems from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2013.

He is an Associate Professor with the College of New Energy, China University of Petroleum (East China), Qingdao, China. His research interests include machine learning methods and their applications in image processing and remote sensing.

Dr. Li was a recipient of the Prize Paper Award Honorable Mention from the IEEE TRANSACTIONS ON MULTIMEDIA in 2016.

Lirong Han received the B.Eng. degree from the Qingdao University of Science and Technology, Qingdao, China. She is pursuing the M.Eng. degree in information and communication engineering with the China University of Petroleum (East China), Qingdao.

Her research interests include machine learning and deep learning, with applications to remote sensing.

Xuanwen Tao (Student Member, IEEE) received the B.S. degree in electronic information science and technology from Tianjin Chengjian University, Tianjin, China, in 2016, and the M.Eng. degree in information and communication engineering from the China University of Petroleum (East China), Qingdao, China, in 2019. She is pursuing the Ph.D. degree with the Hyperspectral Computing Laboratory, University of Extremadura, Cáceres, Spain, supported by the China Scholarship Council.

Xiaoyu Zhang received the B.S. degree in computer science from the Nanjing University of Science and Technology, Nanjing, China, in 2005, and the Ph.D. degree in pattern recognition and intelligent systems from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2010.

He is an Associate Professor with the Institute of Information Engineering, Chinese Academy of Sciences, Beijing. His research interests include machine learning, data mining, and big data analysis.

Dr. Zhang's awards and honors include the Silver Prize of Microsoft Cup IEEE China Student Paper Contest in 2009, the Second Prize of the Wu Wen-Jun AI Science and Technology Innovation Award in 2016, the CCCV Best Paper Nominate Award in 2017, and the Third Prize of BAST Beijing Excellent S&T Paper Award in 2018.

Christos Grecos (Senior Member, IEEE) is the Vice Dean of post-graduate studies and research at the School of Computing, National College of Ireland, Dublin, Ireland.

Previously, he was the Chair and a tenured Full Professor of the Computer Science Department, Central Washington University, Ellensburg, WA, USA, and before that the Dean of the Faculty of Computing and Information Technology, Sohar University, Sohar, Oman. He was a Full Professor of Visual Communications Standards and the Head of the School of Computing, University of the West of Scotland, Glasgow, U.K., from 2009 to 2014. During that period, he was also the Faculty Lead for Research and Knowledge Exchange in the Faculty of Science and Technology. From 2001 to 2009, he worked in the University of Central Lancashire, Preston, U.K., and the University of Loughborough, Loughborough, U.K., as an Associate and Assistant Professor, respectively. His research interests include image/video compression standards, image/video processing and analysis, image/video networking, and computer vision. He has published over 190 research articles in top-tier international publications on these topics.

Dr. Grecos has received five best paper awards. He was the Vice Chair of the University Research and Knowledge Exchange Board from 2009 to 2014. He is on the editorial board or served as a guest editor for many international journals, and he has been invited to give talks in various international conferences. He has been the U.K. delegate on the NATO panel on Disadvantaged Networks, was involved in the Video Quality Experts Group (International Standardisation Organisation) in the area of High Definition Television, and is a High End Foreign Expert for the Chinese Government. He has graduated many Ph.D. students in a variety of universities. He has acted as an external examiner for more than 30 Ph.D. students across the U.K. and EU. He has obtained significant funding for his research from several national or international projects funded by the U.K. Engineering and Physical Sciences Research Council (EPSRC), U.K. TSB, and the European Union.

Antonio Plaza (Fellow, IEEE) received the M.Sc. and Ph.D. degrees in computer engineering from the Hyperspectral Computing Laboratory, Department of Technology of Computers and Communications, University of Extremadura, Cáceres, Spain, in 1999 and 2002, respectively.

He is the Head of the Hyperspectral Computing Laboratory, Department of Technology of Computers and Communications, University of Extremadura. He is one of the top-cited authors in Spain and in the University of Extremadura. He has guest edited ten special issues on hyperspectral remote sensing for different journals. His research interests include remotely sensed hyperspectral image analysis, signal processing, and efficient implementations of large-scale scientific problems on high-performance computing architectures, including commodity Beowulf clusters, heterogeneous networks of computers and clouds, and specialized computer architectures, such as field-programmable gate arrays or graphical processing units.

Dr. Plaza is a fellow of the IEEE for the contributions to hyperspectral data processing and parallel computing of Earth observation data. He was a recipient of a recognition as an Outstanding Associate Editor of the IEEE ACCESS in 2017. He was a member of the Editorial Board of the IEEE GEOSCIENCE AND REMOTE SENSING NEWSLETTER from 2011 to 2012 and the *IEEE Geoscience and Remote Sensing Magazine* in 2013. He was also a pixel of the Steering Committee of the IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING. He is a reviewer of 500 articles for over 50 different journals. He served as the Editor-in-Chief for the IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING from 2013 to 2017. He is an Associate Editor of the IEEE ACCESS. He served as the Director of Education Activities for the IEEE Geoscience and Remote Sensing Society (GRSS) from 2011 to 2012 and the President of the Spanish Chapter of the IEEE GRSS from 2012 to 2016.

Peng Ren (Senior Member, IEEE) received the B.Eng. and M.Eng. degrees in electronic engineering from the Harbin Institute of Technology, Harbin, China, and the Ph.D. degree in computer science from the University of York, York, U.K.

He is a Professor with the College of Oceanography and Space Informatics, China University of Petroleum (East China), Qingdao, China. His research interests include remote sensing and machine learning.

Dr. Ren was a recipient of the K. M. Scott Prize from the University of York in 2011 and the Eduardo Caianiello Best Student Paper Award at the 18th International Conference on Image Analysis and Processing in 2015, as one co-author.