ICIHRN: An Interpretable Multilabel Hash Retrieval Method for Satellite Cloud Images

Wei Jin, Zhoutao Cai, Yukai Pan, and Randi Fu

***Abstract*—Observing clouds to understand the weather is a crucial method for people to forecast upcoming conditions. Utilizing content-based satellite cloud image retrieval allows for the swift discovery of comparable historical cloud images, significantly aiding meteorologists in their advanced investigations. Nevertheless, satellite cloud images often present complexities due to their inclusion of diverse cloud types, leading to inadequate retrieval outcomes when relying on conventionally employed single-label retrieval techniques. Despite notable accomplishments in cloud image retrieval applications utilizing deep neural networks, concerns regarding network interpretability undermine confidence in the model's deductive outcomes. This article introduces the interpretable cloud image hash retrieval network, a framework that employs a singular object-level global unit alongside multiple local feature units for the purpose of generating hash codes tailored to cloud image retrieval. Furthermore, an attention branching network is incorporated to enhance the model's focus on discriminative regions within the image. In addition, a suppression module is implemented to progressively uncover complementary regions through the suppression of prominent areas in preceding layers and the amalgamation of relationships among activated regions. This ensures that each feature unit is endowed with distinctive semantic information, thereby imparting a level of interpretability to the retrieval outcomes. On this foundation, multilabel supervision is seamlessly integrated into the deep hash learning framework. This integration not only enhances the depiction of intricate semantic contents within cloud images but also boosts retrieval efficiency. Comprehensive experimental outcomes, grounded in the publicly accessible satellite cloud map dataset LSCIDMR-V2, demonstrate superior performance relative to other methods.**

***Index Terms*—Cloud image retrieval, feature extraction, feature fusion, hash learning, interpretability.**

I. Introduction

WEATHER conditions play an important role in our daily lives. Clouds are crucial players in weather systems, with various cloud types, phases, and heights exerting a profound influence on the formation and evolution of such systems. They have long been at the forefront of meteorological observation and research. Satellite imagery serves as a potent tool for monitoring clouds and weather systems. By utilizing satellite imagery, we can gain insights into diverse weather conditions, evaluate their intensity and potential future trajectories, and establish a reliable foundation for all-weather forecasting and disaster weather prediction. In this article, we aim to explore the monitoring of tropical cyclones, temperate cyclones, and other potential cloud and weather systems through the task of satellite image retrieval. Through the satellite cloud image retrieval algorithm, similar historical cloud images can be quickly found, offering invaluable assistance to meteorologists in their subsequent research endeavors.

Content-based image retrieval (CBIR) has gained increasing attention as a prominent branch within the field of image retrieval. In general, CBIR mainly consists of two parts: feature extraction and similarity measurement [1]. Feature extraction aims to extract representative features of an image, whereas similarity measurement quantifies the resemblance between the query image and the target image, ultimately locating the most similar images within the database. Most of the traditional feature extraction methods use manually designed features. For example, Chandraprakas and Narayana [2] extract texture features for cloud image retrieval using the grayscale covariance matrix method. Xia et al. [3] propose a grid-based inner circle method to extract region-based features of cloud images for retrieval. Nevertheless, due to the complexity of cloud image content and the high degree of similarity among diverse image types, relying on manual features falls short in accurately unveiling the meteorological semantic information embedded within cloud images, ultimately culminating in subpar retrieval outcomes.

With the rapid advancements in deep learning techniques, convolutional neural networks (CNNs) have been extensively utilized for the extraction of high-level semantic information features from remote sensing images [4], [5], [6], yielding remarkable success. However, the majority of methodologies devised for remote sensing image retrieval predominantly employ single-label retrieval [7], [8], [9], thereby limiting the model's learning to only the most prominent labels. In recent years, research efforts have shifted toward multilabel retrieval of remote sensing images, aiming to surmount the constraints associated with single-label remote sensing image retrieval [10], [11]. Despite the widespread application of these deep learning techniques in remote sensing imagery, their application in satellite cloud image retrieval remains scarcely reported. Furthermore, remote sensing imagery encompasses vast amounts of data, leading traditional methods to extract high-dimensional image features for accurate representation of semantic information. Consequently, this results in increased time and space costs. Deep hashing algorithms

Manuscript received 1 January 2024; revised 1 March 2024; accepted 11 April 2024. Date of publication 16 April 2024; date of current version 25 April 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 42071323 and Grant 42371331 and in part by the Joint Funds of the Zhejiang Provincial Natural Science Foundation of China under Grant LZJMZ24D050002. *(Corresponding author: Randi Fu.)*

The authors are with the Faculty of Electrical Engineering and Computer Science, Ningbo University, Ningbo 315211, China (e-mail: jinwei@nbu.edu.cn; stormczt@outlook.com; panyuk@outlook.com; furandi@nbu.edu.cn).

Digital Object Identifier 10.1109/JSTARS.2024.3389065

compress high-dimensional image feature vectors into binary hash codes to represent image content and employ a simple dissimilarity operation to calculate the Hamming distance for retrieval, thereby significantly conserving computation time and memory usage. Typically, current hashing techniques rely on the output from the final CNN feature layer to produce binary hash codes; the retrieval process based on these hash codes offers no compelling explanation for the ultimate decision outcomes and is deficient in a certain level of interpretability. For meteorologists, the results output from a black-box model cannot be fully trusted. Consequently, it is imperative for us to ensure that the learned hash codes possess unambiguous meaning and interpretability.

To address the challenges while accounting for the varying cloud types and weather systems present in distinct target regions of satellite cloud images, we propose an interpretable cloud image hash retrieval network (ICIHRN). The proposed network comprises two branch structures: a global feature learning branch responsible for acquiring global feature units that depict the primary semantic objects within cloud images, and the other branch, the local feature learning branch, incorporates multiple local feature units dedicated to representing diverse local cloud type targets within the image. Within each local feature unit learning module, an attention branching network is employed to concentrate on salient regions of the image and incorporates a suppression module subsequent to each subbranch. This module identifies complementary regions by suppressing the most prominent features within the preceding layer of the feature map, ensuring that each feature unit possesses its own distinctive semantic information. Ultimately, we produce hash codes for retrieval purposes by amalgamating individual object-level feature units with multiple local feature units within the hash learning module, thus imparting a degree of interpretability to the retrieval results. In addition, we integrate multilabel supervision into the deep hashing framework, facilitating the portrayal of intricate semantic content within cloud images. The main contributions of this article are as follows.

1) We propose an ICIHRN. The network comprises a feature learning module and a hash learning module, designed to efficiently achieve content-based multilabel retrieval for satellite cloud images. To the best of our knowledge, there are very few works focusing on deep-learning-based satellite cloud image retrieval.
2) We propose a structure for interpretable hash codes. By utilizing the global learning branch and local learning branch of the network, we obtain a single feature unit with object-level significance and multiple local feature units, enriching each unit with semantic information. Once these feature units are combined, we employ a deep hashing method to produce a hash code that is semantically informative and interpretable, enhancing efficiency while also imparting a degree of interpretability to the model.
3) The proposed method was evaluated on a public dataset. The experimental results show that the method achieves 94.39% mean average precision (mAP) on LSCIDMR-V2, which is better than other existing methods.

II. Related Works

A. Multilabel Remote Sensing Image Retrieval

Multilabel remote sensing image retrieval involves utilizing multilabel labeled training images to identify remote sensing images resembling a specific query image. Chaudhuri et al. [12] develop a multilabel remote sensing dataset based on the UCMD remote sensing dataset and proposed a semisupervised graph-theoretic approach for remote sensing image retrieval. Dai et al. [13] propose a method for efficiently modeling and exploiting sparsity in remote sensing image descriptors for the supervised retrieval method. Shao et al. [11] employ a fully convolutional network to extract regional convolutional features of the image for multilabel remote sensing image retrieval. Sumbul and Demir [14] propose a new graph-theoretic deep representation learning method, which utilizes a graph structure to offer a region-based image representation that combines local information and relevant spatial organization to effectively solve the problem of remote sensing image retrieval. Sumbul et al. [15] propose a deep-learning-based triple sampling method to learn the metric space for multilabel labeled images, thereby achieving effective results.

However, research on multilabel remote sensing image retrieval has been constrained by the scarcity of large-scale datasets with multiple labels. To address this problem, Sumbul et al. [16] propose BigEarthNet, a practical benchmark dataset for large-scale multilabel remote sensing, comprising 590 326 Sentinel-2 image tiles. Subsequently, they present an enhanced version, BigEarthNet-mm [6]. However, this remote sensing dataset does not satisfy the need for weather systems and cloud-type-related applications. To fill this data gap, Bai et al. [17] propose LSCIDMR, a dataset focused on weather and cloud systems, utilizing three channels of data from the Himawari-8 satellite, and each image in LSCIDMR is cropped to dimensions of 1000×1000 pixels. LSCIDMR comprises LSCIDMR-S and LSCIDMR-M, designed for processing single and multiple labels, respectively. The labels in LSCIDMR are broadly grouped into three categories: weather systems, cloud systems, and terrestrial systems. However, these datasets provide only a crude classification of cloud systems into two phases: high ice clouds and low water clouds. Subsequently, Zhao et al. [18] expand this dataset by refining the two cloud phase labels into nine distinct cloud types, resulting in the creation of LSCIDMR-v2. This updated version incorporates all 16 channels of satellite data. The availability of this dataset facilitates our investigation into large-scale multilabel satellite cloud map retrieval. We will employ this dataset in our research presented in this article.

B. Interpretability of Deep Neural Networks

In recent years, driven by the success of deep learning, significant progress has been made in various computer vision tasks. Although this method possesses remarkable discriminative inference capabilities, its lack of interpretability, being a black-box model, remains a major critique and a potentially fatal drawback. In recent times, there has been a growing emphasis on enhancing the interpretability of deep learning models. Existing

research in this area can be broadly categorized into two main approaches: 1) explaining existing models through visualization techniques and diagnostic analysis of their deep features and 2) incorporating prior knowledge to modify models and enhance their ability to provide interpretable feature representations. For example, Zhou et al. [19] propose a class activation mapping approach to identify important regions in the input image for inference by analyzing image class features. Gradient-weighted class activation mapping (Grad-CAM) [20] improves the method by incorporating gradient computation in the last convolutional layer. Bau et al. [21] propose a comprehensive framework for network profiling, which aims to measure the interpretability of a neural network's underlying representations through assessments of consistency between individual hidden units and a set of semantic concepts, thus providing a more objective evaluation criterion. Apart from their emphasis on interpreting and analyzing trained models, these methodologies develop visually interpretable models by making structural modifications to conventional deep learning architectures, thereby enhancing transparency and trustworthiness. For example, Zhang et al. [22] design interpretable CNNs by forcing each filter to represent a specific portion of the object. Zhang et al. [23] propose an attention-based CNN structure, IA-CNN, such that each feature map in the last convolutional layer has only one response from the target object. This design enables automatic extraction of key points from images and significantly enhances the interpretability of the CNN model, facilitating a deeper understanding of its decision-making processes. Meqdad et al. [24] create interpretable features for CNN models encoded as evolutionary trees for genetic planning algorithms, and these trees make the models interpretable by learning the process of extracting deep structural features.

In satellite cloud image retrieval, our primary focus lies in ensuring that the hash code features employed for retrieval possess explicit meaning and interpretability, rather than solely relying on modifications to traditional models. Hence, in this research, we propose a semantically rich hash code that combines a single object-level hash code with multiple local hash codes, aimed at facilitating interpretable image retrieval.

C. Deep Hash Learning

Similarity retrieval is a fundamental problem in information retrieval and data mining applications [25]. With the rapid growth of image data, retrieving similar images is relatively costly. In deep learning, hashing techniques have become one of the most popular and effective methods. The aim of hashing is to acquire a set of hash functions that can transform each image into a concise binary code via a hashing procedure. Existing methods for hash learning can be broadly classified into two categories: unsupervised hashing and supervised hashing.

Supervised hashing aims to derive hash codes through the utilization of supervised information, with representative supervised hashing methods including kernel-supervised hashing [26] and minimum-loss hashing [27]. Unsupervised hashes do not require labeled datasets to learn mapping functions. They depend on the inherent structure of the image data itself to learn the mapping function. Typical methods within unsupervised hashing include spectral hashing [28], iterative quantization [29], and density-sensitive hashing [30]. With the great success of deep learning in computer vision, combining CNNs with hashing techniques has become a mainstream method for image retrieval. For example, deep supervised hashing [31] uses pairs of images as training input to learn compact binary codes for retrieval. Deep Cauchy hashing [32] generates compact hash codes by designing a pairwise cross-entropy loss based on the Cauchy distribution, which significantly penalizes similar pairs of images if they are greater than a threshold value.

In recent years, there has been continuous development of algorithms that integrate hashing techniques into remote sensing image retrieval. For example, Liu et al. [33] propose a feature and hash (FAH) algorithm, which consists of a deep feature learning module and an adversarial hash learning module for generating hash codes with balanced distributions. Song et al. [34] propose an asymmetric hash code learning method, designed to generate hash codes for querying database images in an asymmetric manner. This approach enhances the representation of both deep features and hash codes. Tang et al. [35] develop Meta-Hashing, a technique in which hash learning is represented in a meta manner, to improve the generalization capability of the hash network when limited labeled training samples are available. Experimental results demonstrate its effectiveness in large-scale retrieval of remote sensing images. Sun et al. [36] propose an unsupervised deep hashing method that relies on soft pseudo-labels. This method leverages a deep autoencoding network to autonomously acquire soft pseudo-labels and a local similarity matrix, enabling the learning of similarity between remote sensing images and leading to satisfactory retrieval outcomes.

III. Method

In this section, we describe our proposed method in detail. We propose an ICIHRN. The ICIHRN consists of two main parts: 1) feature representation learning, focused on obtaining feature units consisting of a single global feature unit and multiple local feature units; and 2) hash learning, which optimizes the network by employing a hash loss function to transform the features into distinct hash codes. Ultimately, the resulting hash codes are utilized to assess the similarity between images, enabling precise retrieval of satellite cloud images. Later, we describe all these components in detail.

A. Feature Representation Learning

Satellite imagery serves as a potent tool for monitoring cloud and weather patterns, with content-based satellite cloud retrieval aiming to expeditiously locate historical cloud images resembling the target image. This facilitates further research for meteorologists. Moreover, we require the model to possess a level of interpretability, fostering trust in the model's inferential judgments among meteorologists. Fig. 1 illustrates the overarching structure of the ICIHRN. Since the CNN model has a stronger hierarchical architecture and is easy to capture local features in the image [37], we adopt it as the backbone

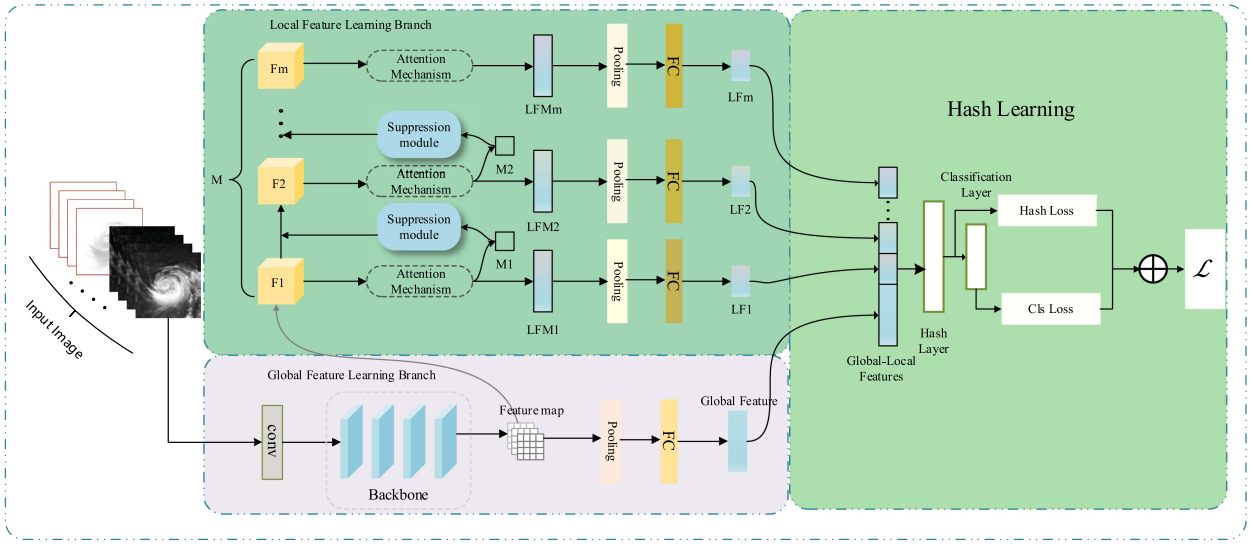

Fig. 1. Proposed framework of ICIHRN. It comprises two primary branches: the feature representation learning module and the hash learning module. Within the feature representation branch, there are two subbranches: the global feature learning branch and the local feature learning branch. In the global feature learning branch, images are inputted into the backbone network to generate feature maps, which are then processed through pooling and fully connected layers to extract global features. Simultaneously, the feature map is passed to the local feature learning branch, where multiple local feature units are derived using an attention mechanism coupled with a suppression module. Following this, we concatenate the global and local features before feeding them into the hash learning module to learn the interpretable hash code.

part of the network structure. Unlike general natural images, satellite cloud image is usually a multispectral image whose different channels will reflect different physical properties. We chose the first, second, third, and fifth band data from the dataset as inputs for our model. Specifically, the first, second, and third channels consist of visible band data offering insights into aerosol physical properties, while the fifth channel represents near-infrared band data, providing information on cloud physical parameters. Considering the varying resolution sizes of the input images and aiming to preserve intricate details while bolstering the robustness of batch processing, we resize the input images to 256 × 256 and normalize the input data.

1) Global Feature Learning Branch: We denote $X = \{x_1, x_2, \ldots, x_N\}$ as the set of input satellite cloud images for each batch, where N is the size of each batch. The corresponding image labels are denoted as $Y = \{y_1, y_2, \ldots, y_N\}$, where $y_i = \{z_1, z_2, \ldots, z_C\}$, C is the total number of categories of the image, and $z_i \in \{0, 1\}$, with 1 and 0 denoting that the ith label of this image is positive or negative, respectively. Initially, we extract the global semantic information from the image. Specifically, since ResNet50 [38] increases the flexibility and stability of the network by introducing residuals and has been effective in several image retrieval methods [7], [36], we utilize ResNet50, the backbone of our network, to process the input image X and obtain its deep feature map F:

$$F = f_\omega(X) \in R^{C \times H \times W}. \tag{1}$$

To obtain the global features, we perform an average pooling operation on the feature graph F and obtain our global semantic features GF after feature dimensionality reduction through a fully connected layer f with the addition of the parameter θ

$$\mathrm{GF} = \mathrm{f}_\theta(\mathrm{GAP}(F)) \tag{2}$$

where GAP denotes a global average operation.

2) Local Feature Learning Branch: Satellite cloud images often contain multiple cloud-type targets, and we need to capture these local-level targets to generate more meaningful hash codes. In human perception, attention mechanisms enable selective focus on salient object parts, thereby facilitating improved capture of visual structure [39]. In the feature learning process, in addition to considering multiscale information, the diversity of objects is also considered. Toward this objective, we devise m mini-branches for local feature extraction, to capture the objects' information based on F. The main idea is to utilize an attention branching network [40] to generate an attention graph to highlight the discriminative regions corresponding to each category. Specifically, taking the first small branch as an example, we initially send the feature map F through a 1 × 1 convolution into the local feature learning module to obtain $F1$. Subsequently, $F1$ is passed through 3 × 3 and 1 × 1 convolutional layer and batch normalization (BN) layer, followed by activation using the rectified linear unit (ReLU) function to generate F_m. Finally, F_m is again passed through a 1 × 1 convolution, which is aggregated and normalized, and then, a Sigmoid function is chosen to generate the attention map $M1$. The process is shown as follows:

$$F_m = \mathrm{ReLu}(\mathrm{BN}(\mathrm{Conv}(\mathrm{Conv}(F1)))) \tag{3}$$

$$M1 = \mathrm{Sigmoid}(\mathrm{Conv}(\mathrm{ConvConv}(F_m))). \tag{4}$$

After that, we perform the Hadamard product between elements of $F1$ and the attention map $M1$ to obtain the local feature map LFM1. The specific process is shown in Fig. 2. However, despite its ability to identify the most distinguishable regions, this method occasionally overlooks the remaining object-specific regions within the complementary regions of the image. Given

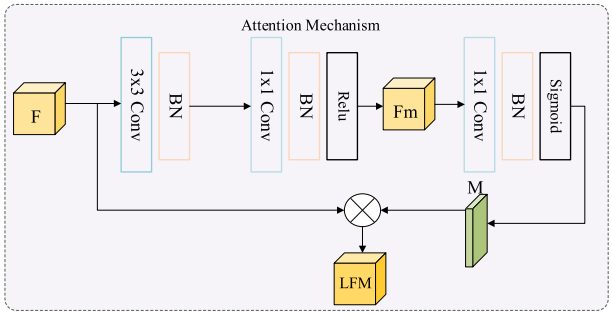

Fig. 2. Attention mechanism.

the contextual relationship between various cloud class regions in satellite cloud maps, erasing the most distinguishable regions in a simplistic manner may impede the extraction of other object-specific regions. Consequently, we introduce a suppression module designed to amplify other object-specific regions while simultaneously suppressing the most discriminative ones. Specifically, we initially input the attention map M into the suppression module and record the standard deviation and mean of all elements in the attention graph M as μ^{std} and μ^{mean}. For each element within M $\mu^k \in \{\mu^1, \mu^2, \ldots, \mu^{H\times W}\}$, do the following:

$$\mu^k = 1 - \frac{\mu^k - \mu^{\text{mean}}}{(\mu^{\text{std}})^a} \tag{5}$$

where a is a hyperparameter that distinguishes the degree of suppression ratio of the salient region from the degree of enhancement ratio of the other activated regions; the higher a is, the higher the degree of suppression and enhancement. Through this operation, which relies on enhancing suppression, some of the most discriminative regions from the previous stage will be inhibited. Simultaneously, other complementary regions that have been activated will experience enhancement alongside. Consequently, the connection between the activated regions from the previous stage and those generated in later stages is preserved. Subsequently, we obtain new local features by multiplying the suppressed attention map with the layer's feature F. This process is repeated to acquire multiple local features. Likewise, we perform average pooling on these local features and undergo dimensionality reduction through a fully connected layer to derive the local feature unit $LF = \{LF_i\}_{i=1}^m$. Finally, a novel global–local fusion feature is created by concatenating the local feature unit LF with the global feature unit GF and then send this feature to the hash learning module for learning.

B. Hash Learning

To enhance the precision of image retrieval through the generation of compact binary hash codes, we append a hash network following the backbone network for the purpose of mapping global–local fusion features to binary hash codes. We input this feature into the hash layer and proceed to directly quantize the output into the desired discrete values, thereby facilitating the subsequent retrieval task. Specifically, we set the number of nodes on the hash layer to K, where K is the length of the desired hash code. For each hash code $i = 1, 2, 3, \ldots, K$, we calculate it using the subsequent equation:

$$b_i = \text{sgn}(d_i) \tag{6}$$

$$\text{sgn}(x) = \begin{cases} 1, & x \geq 0.5 \\ 0, & x < 0.5 \end{cases} \tag{7}$$

where d_i is the hash-like code obtained by passing the input image i through the hash layer. To minimize the quantization error arising during the conversion of the true feature representation into a binary hash code [41], [42], [43], we introduce a quantization loss function to bring the binary code closer to the desired hash code

$$L_q = -\frac{1}{K}\sum_{i=1}^{N} \|d_i - 0.5e\|^2 \tag{8}$$

where e is a K-dimensional vector with value 1. In addition to the quantization loss, we define a bit balancing loss L_b to efficiently create hash codes. This loss is used to ensure that each hash code has a 50% probability of being 0 or 1. This loss function is defined as

$$L_b = \sum_{i=1}^{N} (\text{mean}(d_i) - 0.5)^2 \tag{9}$$

where $\text{mean}(d_i)$ denotes the calculation of the mean of the values in d_i.

Meanwhile, the classification labels of an image often contain the overall semantic information of the image, especially for multilabel images, and we can use these labels as semantic clues to aid network training, enhance the precision of hash codes, and endow hash codes with richer semantic meanings. Therefore, we introduce a classification layer after the hash network layer to improve the feature extraction capability and at the same time enhance the class discrimination capability of the model. In single-label image classification scenarios [44], [45], we often use the classical cross-entropy (CE) loss, which is denoted as

$$L_{\text{CE}}(P) = \sum_{i=1}^{m} -y_i \log P \tag{10}$$

where N is the number of the batch sizes for training, y_i is the one-hot label of x_i, and P is the probability that a sample is entered into that class. In contrast to single-label images, multilabel images frequently encompass multiple classes. Therefore, we can view all possible labels contained in an image as a binary classification problem. That is, an input image corresponds to multiple labels, and each label corresponds to a binary classification (yes or no), so we can design a multilabel classification loss L_{cls} as follows:

$$L_{\text{cls}} = \frac{1}{N}\sum_{n=1}^{N} l_n \tag{11}$$

$$l_n = \frac{1}{C}\sum_{i=1}^{C} -w_i \begin{pmatrix} y_n^i * \log\left(\sigma\left(x_n^i\right)\right) + \\ \left(1 - y_n^i\right) * \log\left(1 - \sigma\left(x_n^i\right)\right) \end{pmatrix} \tag{12}$$

where C denotes the total number of all possible labels contained in the image, x_n^i denotes the output of the model corresponding to the ith label of the nth sample in a batch, y_n^i denotes the true value corresponding to the ith label of the nth sample, and

$\sigma(x) = \frac{1}{1+\exp(-x)}$ is denoted as a sigmoid activation function used to map x into the (0,1) interval.

In summary, when designing the loss function, the labeling information of the image and the output information of the hash layer are fully utilized to generate a hash code rich in semantic information and discriminative properties. Consequently, the total loss function is expressed as follows:

$$L_{\text{total}} = \lambda L_{\text{cls}} + \mu L_q + \nu L_b \tag{13}$$

where λ, μ, and ν are penalty parameters to better balance the individual losses.

Finally, the model is sufficiently trained so that the hash codes of all the samples in the database can be obtained via the model output. Then, we compute the Hamming distance between the query samples and all the database samples and sort them to get the desired retrieval results.

IV. Experiment

To assess the efficacy of the proposed method, we conduct multiple experiments in this section. Initially, we introduce the experimental dataset and experimental environment, then present the implementation details of the experiments, and ultimately showcase the experimental results and providing analysis and discussion.

A. Dataset and Settings

In this experiment, we utilize a publicly accessible satellite cloud image dataset, LSCIDMR-V2 [18]. The source data are provided by the P-Tree system of the Japan Aerospace Exploration Agency. The entire dataset covers the northern hemisphere of the western Pacific Ocean. It contains a total of 104 390 images with 17 labels, each with a pixel size of 300 × 300 and a spatial resolution of 2.0 km.

In order to compare the effectiveness of different retrieval methods, we use mAP, P@k, average cumulative gain (ACG), weighted mean average precision (WAP), and normalized discounted cumulative gains (NDCG) as evaluation criteria, where ACG, WAP, and NDCG serve as performance metrics specifically tailored for multilabel retrieval; higher values of mAP and P@k signify better retrieval performance. Specifically, given a query image of quantity Q, the value of mAP can be calculated by the following equation:

$$\text{mAP} = \frac{1}{Q} \sum_{r=1}^{Q} \text{AP}(r) \tag{14}$$

where AP denotes average precision, defined as

$$\text{AP} = \frac{1}{R} \sum_{k=1}^{K} P(k)\, r(k) \tag{15}$$

where $P(k)$ denotes the precision of the first k images retrieved, and $r(k)$ is an indicator function that specifies whether the kth image is relevant to the query image: the value is 1 when it is relevant to the query image, and 0 when it is not relevant. k denotes the number of images retrieved, and R is the number of ground truths retrieved.

The metric ACG describes the similarity of the shared labels between the query image and the corresponding first k retrieved images. For the query image q, its ACG score is

$$\text{ACG@}k = \frac{1}{k} \sum_{i}^{k} S(q,i) \tag{16}$$

where $S(q,i)$ is the similarity of the common label shared between images q and i.

WAP is an average score similar to mAP and refers to the average of the ACG scores of the first n retrieved images. WAP is calculated as

$$\text{WAP} = \frac{1}{Q} \sum_{r=1}^{Q} \left(\frac{1}{R} \sum_{k}^{R} r(k) \times \text{ACG@}k \right). \tag{17}$$

In general, the precision of retrieval is the ratio of retrieved correct results to retrieved results obtained. We use P@k as an auxiliary performance measure, which represents the precision when the number of returned results is K. It can be calculated by the following formula, where $R_k(r)$ denotes the number of images related to the query image among the first k images retrieved

$$P@k = \frac{1}{Q} \sum_{r=1}^{Q} \frac{R_k(r)}{k}. \tag{18}$$

NDCG is a normalized discounted cumulative gain score. Given a query image q, the discounted cumulative gain (DCG) score of top k retrieved images is calculated by

$$\text{DCG@}k = \sum_{i}^{k} \frac{2^{S(q,i)-1}}{\log(1+i)}. \tag{19}$$

Then, the NDCG score of top k retrieved images is calculated as

$$\text{NDCG@k} = \frac{\text{DCG@k}}{Z_k} \tag{20}$$

where Z_k is the maximum value of DCG@k to constrain the value of NDCG in the range of [0,1].

We use a pretrained ResNet50 network as the backbone network to extract the deep embedding of the images. We replace the three-channel convolutional input at the beginning of the pretraining ResNet50 with a four-channel convolutional input. Subsequently, we fine-tune the network parameters using our custom loss function and dataset, with the aim of enhancing its adaptability to our specific retrieval requirements for cloud map images. Concurrently, to bolster the model's generalization capabilities and mitigate overfitting, we normalize the input images. The spatial resolution size of the input image is 256 × 256. The penalty coefficients λ, μ, and ν of the loss function are set to 0.5, 0.5, and 0.0002, respectively. M in the local feature learning branch is set to 3. The initial learning rate is set to 0.0005, with a decay of 50% after every 30 epochs. The optimizer for the experiment is set to Adam, where β_1 is 0.9 and β_2 is 0.999, and the training batch is set to 64. The hardware environment for the experiments in this article is Intel Core i5-10600KF CPU, NVIDIA GeForce GTX RTX3060 32-GB RAM.

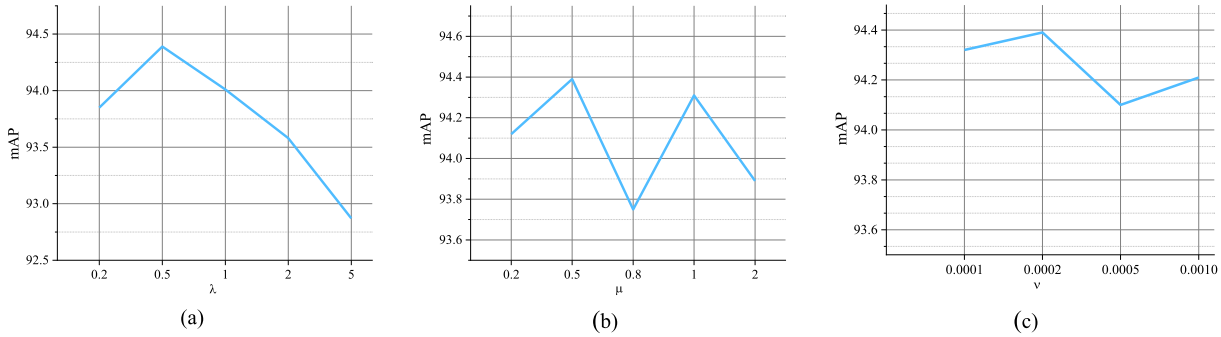

Fig. 3. Effects of different penalty coefficient on mAP. (a) Effect of λ on mAP. (b) Effect of μ on mAP. (c) Effect of ν on mAP.

TABLE I
RETRIEVAL PERFORMANCE OF DIFFERENT HASH BITS CODES

Bits	mAP	P@50	P@100	P@200	ACG@50	WAP	NDCG
36	92.76	92.88	92.58	92.26	78.98	78.95	96.01
66	93.26	93.36	93.11	92.75	81.42	81.35	96.58
132	**94.39**	**94.49**	**94.28**	**93.92**	**82.25**	**82.23**	**98.03**
264	93.82	93.84	93.70	93.37	81.16	81.13	97.25

The bold values represent the best values in a column of data.

TABLE II
RETRIEVAL PERFORMANCE OF NUMBER OF LOCAL BRANCHES

M	mAP	P@50	P@100	P@200	ACG@50	WAP	NDCG
2	92.10	92.12	92.03	91.89	76.11	76.09	96.64
3	**94.39**	**94.49**	**94.28**	**93.92**	**82.25**	**82.23**	**98.03**
4	92.93	93.01	92.66	92.26	80.65	80.63	96.83
5	92.66	92.71	92.56	92.54	78.91	78.84	97.04

The bold values represent the best values in a column of data.

B. Experimental Result

This section will give the experimental results on LSCIDMR-V2 and analyze the results. We randomly split the dataset into a training set and a test set in a ratio of 8:2, where the training set contains 83 227 images and the test set contains 21 163 images.

1) Penalty Coefficient Analysis of the Loss Function: As shown in (12), our loss function has three penalty coefficients, i.e., λ, μ, and ν. Specifically, λ is used to control the contribution of image label information to overall similarity. μ and ν are, respectively, used to control the contribution of hash quantization and bit balance to the overall objective function. We designed a series of experiments for analyzing the effects of all the above parameters on the retrieval results, and it should be noted that when analyzing one parameter, the other two parameters are set to fixed values.

The ultimate outcomes of our experimentation are presented in Fig. 3. Specifically, Fig. 3(a) illustrates the evolution of mAP values as the λ parameter increases across the dataset. The graph reveals that as $\lambda < 0.5$, the mAP value experiences an upward trajectory. Conversely, as λ continues to rise, the mAP value demonstrates a descending pattern. It is worth noting that the retrieval performance reaches its optimal value when $\lambda = 0.5$. Fig. 3(b) depicts the fluctuation of mAP values corresponding to increasing values of μ on the dataset. The peak MAP is obtained when $\mu = 0.5$, while the lowest value of map is obtained when $\mu = 0.8$. In addition, Fig. 3(c) showcases the variation of mAP values in response to escalating ν values on the dataset. We can observe that the impact of this penalty coefficient on the map value does not change significantly, and the change is stable. In conclusion, after careful consideration, we have opted to set the penalty coefficients as follows: λ at 0.5, μ at 0.5, and ν at 0.0002.

2) Impact of Hash Size on Retrieval Performance: In order to facilitate the subsequent experiments, the effect of {36,66,132,264}on the model retrieval performance is first investigated for models with different length hash codes, where the global hash unit occupies $K/2$ bits and the local hash unit occupies $K/2M$ bits. The experimental results are shown in Table I. We observe an improvement in retrieval performance as the number of hash code bits increases, with optimal performance achieved at 132 bits, and beyond this point, performance begins to decline. When the number of bits is 132, its mAP value reaches 94.39%, which improves by 1.76% (36 bits), 1.21% (66 bits), and 0.61% (264 bits), respectively, compared to other bits. In addition, its WAP value reaches an optimal 82.23%, surpassing the performance of other bit lengths. Therefore, for the purpose of subsequent experiments, we finally chose to set the hash size to 132 bits.

3) Impact of the Number of Local Branches M *on Retrieval Performance:* In the proposed model, the local learning module comprises multiple local learning branches. Therefore, we investigate the effect of the number of local branching modules *M* on the overall retrieval performance. The experimental results are shown in Table II. We observe that the model's retrieval performance peaks when *M* is set to 3, achieving an mAP of 94.39%. This represents an improvement of 2.48%, 1.57%, and 1.87% compared to other configurations, respectively. In addition, the WAP value reaches an optimal 82.23%, representing improvements of 8.07%, 1.98%, and 4.29% compared to other configurations, respectively. A plausible explanation for this observation is that when *M* equals 3, each local feature unit occupies a sufficient number of bits to effectively convey the semantic information of its respective part. Conversely, in

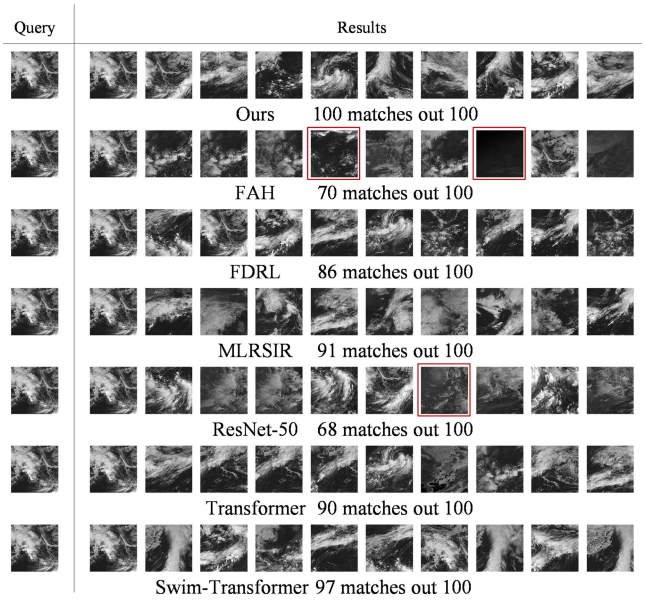

Fig. 4. Retrieval examples of different methods on dataset LSCIDMR-V2.

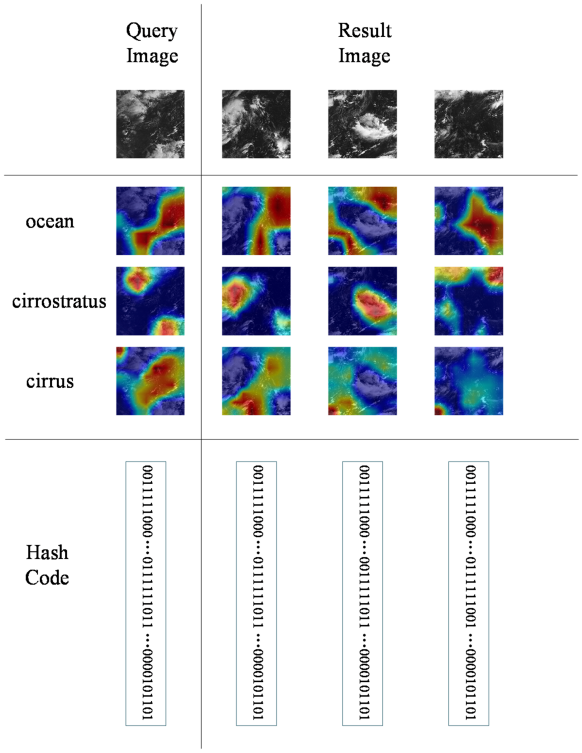

Fig. 5. Visualization of the query image and results.

other configurations, variations in the number of bits allocated to feature units lead to a reduction in their ability to express semantic information accurately, thereby compromising performance. However, it is worth noting that the overall performance variations are not substantial, which underscores the general efficacy of our proposed method. For the purpose of subsequent experiments, we finally choose to set the number of local branches M as 3.

4) Comparison Experiments With State-of-the-Art Methods: In order to verify the validity of the proposed model, we further compared the performance of our method with the existing methods. The method chosen for comparison include ResNet-50, Transformer [48], Swim-Transformer [49], FAH [33], FDRL [46], and MLRSIR-NET [47]. Swim-Transformer adopts a hierarchical structure for adapting images of different scales and implements a linear complexity attention computation using a sliding window approach to optimize the Transformer. FAH consists of two modules: deep feature learning model (DFLM) and adversarial hash learning model (AHLM). It improves the retrieval efficiency by mapping dense features onto compact hash codes and combining it with an adversarial regularization submodel to make the hash codes discretely and uniformly distributed. FDRL is a label-guided similarity-based feature decomposition and interactive learning approach for solving multilabel image retrieval. MLRSIR-NET is a framework for multilabel remote sensing image retrieval, which consists of two main subnetworks, i.e., multilevel feature extraction and deep hashing, which achieves effective multilabel retrieval of remote sensing images by encoding feature vectors into a compact hash code. To make a fair comparison, we set the length of the feature vectors of each model to 132 bits uniformly and learn them under the same training environment.

The experimental results are presented in Table III. From the experimental data, it is shown that the ICIHRN model achieves the most superior results on the dataset. The ICIHRN model has the best mAP value of 94.39%, which is 22.28% (ResNet50), 14.54% (FAH), 8.29% (FDRL), 5.95% (MLRSIR-NET), 3.85% (Transformer), and 3.40% (Swim-Transformer) higher than the comparison models, respectively. The proposed model's superior results compared to the Transformer class can be attributed to the fact that Transformers lack local receptive fields and translation invariance, which are characteristic of CNNs. This limitation reduces their ability to extract features specific to image perception, as they primarily focus on global features and struggle to effectively capture local image features. In contrast, most comparative methods are tailored for natural or remote sensing image datasets, leading to subpar performance. Conversely, our proposed method is specifically designed to address the complexities of satellite cloud imagery and diverse image scene content. By extracting both global features and multiple local feature units, and subsequently combining them to generate interpretable hash codes, our method enhances feature extraction capabilities and delivers exceptional performance. Furthermore, our model demonstrates consistent stability in terms of P@K value, while achieving optimal ACG and WAP scores. This signifies that our model excels not only in retrieving similar images but also in accurately gauging the similarity between multiple tags within those retrieved images, further validating the efficacy of our proposed approach.

In order to visually demonstrate the retrieval effect of the proposed method on the dataset, we show some retrieval examples in Fig. 4. Specifically, we randomly select an image as the query object, and retrieval results are obtained by sorting the similarity measure between the query object and the target image. In the first column, we display the query image, while the

TABLE III
COMPARISON ON LSCIDMR-V2 WITH STATE-OF-THE-ART METHODS

Methods	mAP(%)	P@50	P@100	P@200	ACG@50	WAP	NDCG
ResNet50	77.19	77.21	76.36	75.22	68.18	67.42	82.29
Transformer	90.89	90.40	89.87	88.40	75.89	75.73	95.69
SwimTransFormer	91.29	91.62	91.13	90.35	77.55	77.48	96.38
FAH	82.41	82.15	84.11	83.88	58.18	57.33	88.86
FDRL	87.16	87.28	86.93	86.48	80.18	80.14	94.86
MLRSIR-NET	89.09	89.15	88.95	88.60	80.74	80.52	95.43
ours	**94.39**	**94.49**	**94.28**	**93.92**	**82.25**	**82.23**	**98.03**

The bold values represent the best values in a column of data.

TABLE IV
COMPARISON OF RETRIEVAL TIME (IN SECONDS) OF DIFFERENT METHODS

Methods	ResNet50	Transformer	SwimTrans Former
	0.3320	0.3436	0.4153
FAH	FDRL	MLRSIR-NET	ours
0.3071	0.3177	0.3286	0.3145

TABLE V
RESULTS OF AN ABLATION EXPERIMENT ON LSCIDMR-V2

Methods	mAP	P@50	P@100	P@200
LF	89.18	89.30	89.12	88.48
LWS	91.99	92.01	91.76	91.61
LWSA	92.84	92.97	92.26	91.85
LWSAH_no_L_q	93.71	93.78	93.23	93.16
LWSAH_no_L_b	93.43	93.52	93.34	93.21
ICIHRN(without hash)	93.25	93.36	93.05	92.83
ICIHRN	**94.39**	**94.49**	**94.28**	**93.92**

The bold values represent the best values in a column of data.

subsequent columns showcase the retrieval results. In addition, images enclosed in a red box indicate incorrect retrieval results. Because of space limitations, we can only present the first ten retrieval results and provide a summary of the correct results among the initial 100. These retrieval examples also validate the effectiveness of the method.

In addition to the quantitative metrics, retrieval efficiency is also an important factor when designing a retrieval algorithm. We conduct a comparative analysis of the proposed method and other methodologies, with a particular focus on retrieval time. Given that the model training phase is an offline procedure that requires only a one-time execution, the time investment for various models is deemed reasonable. Our primary consideration here lies in assessing the time expenditure associated with retrieving images through these models. Specifically, we carry out ten experiments for each method, randomly selecting 50 images to serve as the query set and retrieving the top 100 images from the dataset. The final result is determined by calculating the average value of these experiments.

The experimental results are shown in Table IV. We can find that the hash retrieval methods are generally faster than other methods for retrieval. Our proposed method can achieve 0.3145 s in retrieval speed, which is acceptable in terms of time performance. Besides, our method has high retrieval accuracy, which, taken together, verifies the effectiveness of the proposed method.

5) Ablation Study: In this section, in order to investigate the contribution of each component to the ICIHRN model, we conduct ablation experiments here. In particular, the proposed network structure mainly consists of a feature learning module and a hash learning module. The feature learning module encompasses a global feature learning branch and a local feature learning branch with a suppression module and an attention mechanism integrated within. To demonstrate the efficacy of these modules and branches, we incrementally incorporate them into the baseline backbone (i.e., ResNet-50). Therefore, we constructed the following methods for the ablation experiments: 1) LocalFeature (LF), which involves adding the local feature branch to the backbone network without including the suppression module and attention mechanism; 2) local with suppression (LWS), which adds the suppression module to method 1; 3) local with suppression attention (LWSA), which incorporates the attention mechanism module into method 2; 4) local with suppression and hash (LWSAH), which appends the hash learning module to method 3. We also split the method into LWSAH_no_L_q and LWSAH_no_L_b in order to verify the validity of our loss function; 5) The ICIHRN (without hash) indicates the method without the hashing module; and 6) the ICIHRN, representing our proposed comprehensive model in its entirety. We choose mAP to evaluate these methods numerically, and the final experimental results are shown in Table V.

By observing these results, we can easily find that the ICIHRN achieves the best performance, and each component of the model contributes positively to the final retrieval results. Specifically, when incorporating the suppression module and attention mechanism, our model's mAP value reaches 94.39%, exhibiting a noteworthy increase of 5.84% compared to its performance without these components. Moreover, adding the suppression module and the attention mechanism module alone improves 2.61% and 1.67%, respectively, which proves the effectiveness of the module to enhance the model's ability to extract image features and improve the performance of retrieval. Likewise, we observe that the introduction of L_q and L_b positively impacts the model's

retrieval performance. Precisely, the individual inclusion of L_q and L_b leads to enhancements of 0.73% and 1.03%, respectively. Moreover, when we combine these two loss functions and utilize them concurrently, our results exhibit a notable improvement of 1.67%. The outcomes unequivocally demonstrate that our proposed feature learning component effectively enhances the model's ability to extract pertinent features, while our hash learning component ensures that the learned hash codes are both accurate and compact, collectively validating the effectiveness of our proposed method.

6) Interpretability: In order to better understand how our method works, we provide some visualizations in Fig. 5. We visualize the query images and the retrieved images using the Grad-CAM tool and show the hash codes of these images for observation. The three tags with the highest category probability in our images are "ocean," "cirrostratus," and "cirrus." It can be seen that the retrieved images and the original images all have the contents in the image labels, and these contents are effectively displayed in the heat map. This shows that the extraction of image features by our model is well founded and interpretable, and the retrieved images also find a basis for determining the existence of the category, enabling meteorologists to trust the model output. At the same time, we show a part of the hash code of the image; it can be seen that the query image and the retrieved image have roughly the same hash code, only in very few positions there are differences, and these hash codes by the features of the image for the hash transformation. We can think that these hash codes can be able to display the semantic part of the display with a certain degree of interpretability.

V. Conclusion

In this article, we propose an ICIHRN for satellite cloud image retrieval. Satellite cloud images, as a type of remote sensing imagery, differ from natural images in their composition of multiple data channels, with each channel representing distinct physical properties and containing a wealth of information. Furthermore, the complexity of cloud image content often results in a single image encompassing diverse cloud class compositions, which cannot be adequately captured by a simplistic single-label annotation. Therefore, we adopt a multilabel image retrieval method to extract effective features by designing a multilabel classification loss to train the model. Given the significance of cloud mapping research in meteorological work, it is imperative that our model is interpretable, ensuring that meteorologists can place trust in the model's outputted results. Accordingly, we have designed our network architecture to consist of a global feature learning branch and a local feature learning branch. The global feature branch learns a single global hash unit that represents the object level, while the local branch generates multiple local hash units that represent different local cloud types in the image by adding a suppression module to dynamically localize the image one by one to distinguish between different types of cloud regions in the image, and we combine the two to make the hash code rich in semantic information, which gives it a certain degree of interpretability. By integrating these two components, we enrich the hash code with semantic information, thereby enhancing its interpretability. For the purpose of enhancing retrieval efficiency, we have introduced a dedicated hash learning branch tasked with learning more precise hash codes. The effectiveness of the proposed method is demonstrated through extensive experiments on the publicly available dataset LSCIDMR-V2.

The research presented in this article has some shortcomings; for example, we simply selected four of the satellite cloud image channel data as input and ignored the data from multiple other channels. In addition, our approach necessitates a substantial quantity of labeled data for effective training, procuring which can often be costly, particularly for satellite cloud images that demand manual annotation. In our future endeavors, we aim to explore the integration of data from various channels with metric learning techniques to enhance the optimization of the loss function, thereby ensuring the compactness of the generated image features, and to further increase the interpretability of image retrieval by starting from the perspective of image similarity metrics.

References

[1] B. Demir and L. Bruzzone, "A novel active learning method for content based remote sensing image retrieval," in *Proc. 23nd Signal Process. Commun. Appl. Conf.*, 2015, pp. 2130–2133.

[2] D. Chandraprakas and M. Narayana, "Content based satellite cloud image retrieval using texture features," *Int. J. Eng. Res. Appl.*, vol. 7, no. 6, pp. 11–18, 2017.

[3] S. Xia, Q. Li, and J. Zhang, "Satellite cloud image retrieval with grid based inscribed circle method," in *Proc. Int. Conf. Machinery, Mater. Inf. Technol. Appl.*, 2015, pp. 351–356.

[4] L. Zhang, L. Zhang, and B. Du, "Deep learning for remote sensing data: A technical tutorial on the state of the art," *IEEE Geosci. Remote Sens. Mag.*, vol. 4, no. 2, pp. 22–40, Jun. 2016.

[5] Z. Yuan et al., "A lightweight multi-scale crossmodal text-image retrieval method in remote sensing," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5612819.

[6] G. Sumbul et al., "BigEarthNet-MM: A large-scale, multimodal, multilabel benchmark archive for remote sensing image classification and retrieval [software and data sets]," *IEEE Geosci. Remote Sens. Mag.*, vol. 9, no. 3, pp. 174–180, Sep. 2021.

[7] Y. Liu, L. Ding, C. Chen, and Y. Liu, "Similarity-based unsupervised deep transfer learning for remote sensing image retrieval," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 11, pp. 7872–7889, Nov. 2020.

[8] 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.

[9] J. Chu, L. Li, and X. Xiao, "Remote sensing image retrieval by multi-scale attention-based CNN and product quantization," in *Proc. 40th Chin. Control Conf.*, 2021, pp. 8292–8297.

[10] J. Kang, R. Fernandez-Beltran, D. Hong, J. Chanussot, and A. Plaza, "Graph relation network: Modeling relations between scenes for multilabel remote-sensing image classification and retrieval," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 5, pp. 4355–4369, May 2021.

[11] Z. Shao, W. Zhou, X. Deng, M. Zhang, and Q. Cheng, "Multilabel remote sensing image retrieval based on fully convolutional network," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 318–328, 2020.

[12] B. Chaudhuri, B. Demir, S. Chaudhuri, and L. Bruzzone, "Multilabel remote sensing image retrieval using a semisupervised graph-theoretic method," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 2, pp. 1144–1158, Feb. 2018.

[13] O. E. Dai, B. Demir, B. Sankur, and L. Bruzzone, "A novel system for content-based retrieval of single and multi-label high-dimensional remote sensing images," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 7, pp. 2473–2490, Jul. 2018.

[14] G. Sumbul and B. Demir, "A novel graph-theoretic deep representation learning method for multi-label remote sensing image retrieval," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2021, pp. 266–269.
[15] G. Sumbul, M. Ravanbakhsh, and B. Demir, "A relevant, hard and diverse triplet sampling method for multi-label remote sensing image retrieval," in *Proc. IEEE Mediterranean Middle-East Geosci. Remote Sens. Symp.*, 2022, pp. 5–8.
[16] G. Sumbul, M. Charfuelan, B. Demir, and V. Markl, "BigEarthNet: A large-scale benchmark archive for remote sensing image understanding," in *Proc. IEEE Int. Geosci. Remote Sens. Symp.*, 2019, pp. 5901–5904.
[17] C. Bai, M. Zhang, J. Zhang, J. Zheng, and S. Chen, "LSCIDMR: Large-scale satellite cloud image database for meteorological research," *IEEE Trans. Cybern.*, vol. 52, no. 11, pp. 12538–12550, Nov. 2022.
[18] D. Zhao, Q. Wang, J. Zhang, and C. Bai, "Mine diversified contents of multispectral cloud images along with geographical information for multilabel classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 4102415.
[19] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proc. IEEE Conf. Comput. Vis. pattern Recognit.*, 2016, pp. 2921–2929.
[20] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 618–626.
[21] D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba, "Network dissection: Quantifying interpretability of deep visual representations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3319–3327.
[22] Q. Zhang, Y. N. Wu, and S. C. Zhu, "Interpretable convolutional neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 8827–8836.
[23] Z. Zhang, Y. Chen, H. Li, and Q. Zhang, "IA-CNN: A generalised interpretable convolutional neural network with attention mechanism," in *Proc. Int. Joint Conf. Neural Netw.*, 2021, pp. 1–8.
[24] M. N. Meqdad, F. Abdali-Mohammadi, and S. Kadry, "Meta structural learning algorithm with interpretable convolutional neural networks for arrhythmia detection of multisession ECG," *IEEE Access*, vol. 10, pp. 61410–61425, 2022.
[25] J. Wang, H. T. Shen, J. Song, and J. Ji, "Hashing for similarity search: A survey," 2014. [Online]. Available: https://arvix.org/abs/1408.2927
[26] W. Liu, J. Wang, R. Ji, Y. G. Jiang, and S. F. Chang, "Supervised hashing with kernels," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 2074–2081.
[27] M. E. Norouzi and D. J. Fleet, "Minimal loss hashing for compact binary codes," in *Proc. 28th Int. Conf. Mach. Learn.*, 2011, pp. 353–360.
[28] Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2009, pp. 1753–1760.
[29] Y. Gong, S. Lazebnik, A. Gordo, and F. Perronnin, "Iterative quantization: A Procrustean approach to learning binary codes for large-scale image retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 12, pp. 2916–2929, Dec. 2013.
[30] Z. Jin, C. Li, Y. Lin, and D. Cai, "Density sensitive hashing," *IEEE Trans. Cybern.*, vol. 44, no. 8, pp. 1362–1371, Aug. 2014.
[31] H. Liu, R. Wang, S. Shan, and X. Chen, "Deep supervised hashing for fast image retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2064–2072.
[32] Y. Cao, M. Long, B. Liu, and J. Wang, "Deep Cauchy hashing for hamming space retrieval," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 1229–1237.
[33] C. Liu, J. Ma, X. Tang, F. Liu, X. Zhang, and L. Jiao, "Deep hash learning for remote sensing image retrieval," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 4, pp. 3420–3443, Apr. 2021, doi: 10.1109/TGRS.2020.3007533.
[34] W. Song, Z. Gao, R. Dian, P. Ghamisi, Y. Zhang, and J. A. Benediktsson, "Asymmetric hash code learning for remote sensing image retrieval," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5617514, doi: 10.1109/TGRS.2022.3143571.
[35] X. Tang et al., "Meta-hashing for remote sensing image retrieval," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5615419, doi: 10.1109/TGRS.2021.3136159.
[36] Y. Sun et al., "Unsupervised deep hashing through learning soft pseudo label for remote sensing image retrieval," *Knowl.-Based Syst.*, vol. 239, Mar. 2022, Art. no. 107807, doi: https://doi.org/10.1016/j.knosys.2021.107807.
[37] H. Wu et al., "CvT: Introducing convolutions to vision transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 22–31.
[38] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[39] H. Larochelle and G. E. Hinton, "Learning to combine foveal glimpses with a third-order Boltzmann machine," in *Proc. Int. Conf.. Neural Inf. Process. Syst.*, 2010, pp. 1243–1251.
[40] H. Fukui, T. Hirakawa, T. Yamashita, and H. Fujiyoshi, "Attention branch network: Learning of attention mechanism for visual explanation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 10705–10714.
[41] S. Roy, E. Sangineto, B. Demir, and N. Sebe, "Metric-learning-based deep hashing network for content-based retrieval of remote sensing images," *IEEE Geosci. Remote Sens. Lett.*, vol. 18, no. 2, pp. 226–230, Feb. 2021.
[42] X. Luo et al., "A survey on deep hashing methods," *ACM Trans. Knowl. Discov. Data*, vol. 17, no. 1, 2023, Art. no. 15.
[43] Z. Yang, O. I. Raymond, W. Sun, and J. Long, "Asymmetric deep semantic quantization for image retrieval," *IEEE Access*, vol. 7, pp. 72684–72695, 2019.
[44] X. Liu, Y. Wu, W. Liang, Y. Cao, and M. Li, "High resolution SAR image classification using global-local network structure based on vision transformer and CNN," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, 2022, Art. no. 4505405.
[45] X. Wang, H. Xu, L. Yuan, W. Dai, and X. Wen, "A remote-sensing scene-image classification method based on deep multiple-instance learning with a residual dense attention ConvNet," *Remote Sens.*, vol. 14, no. 20, Art. no. 5095, 2022.
[46] Y. Dai, W. Song, Y. Li, and L. D. Stefano, "Feature disentangling and reciprocal learning with label-guided similarity for multi-label image retrieval," *Neurocomputing*, vol. 511, pp. 353–365, Oct. 2022, doi: https://doi.org/10.1016/j.neucom.2022.09.007.
[47] M. Mostafa, S. Sayed, and A. Hamed, "Learning to hash with convolutional network for multi-label remote sensing image retrieval," *Int. J. Intell. Eng. Syst.*, vol. 13, pp. 539–547, Jan. 2020, doi: 10.22266/ijies2020.1031.47.
[48] A. Vaswani et al., "Attention is all you need," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 6000–6010.
[49] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 10012–10022.