SSRG International Journal of Computer Science and Engineering
ISSN: 2348–8387 / https://doi.org/10.14445/23488387/IJCSE-V11I3P101

Volume 11 Issue 3, 1-8, March 2024

Original Article

Image Retrieval Based on Tree Structures from Hierarchical Data

Shihong Lu[1], Zhen Wang[2], Limeng Gao[3], Kaiyang Gong[4]

[1,2,3,4]*School of Computer Science and Technology, Shandong University of Technology, China.*

[1]*Corresponding Author : wzh@sdut.edu.cn*

Received: 15 January 2024 Revised: 26 February 2024 Accepted: 15 March 2024 Published: 28 March 2024

Abstract - *With the advent of the mobile network era, the number of images has increased explosively. In the context of mobile internet, image retrieval plays an irreplaceable role in our lives. Due to the continuous development of deep learning algorithms, researchers have introduced deep learning technology into the field of image retrieval for the generation of image hashes. However, most image hash algorithms only consider sample category loss and treat the category distance between different labels equally, thus ignoring the distance information between categories. To address the above issues, this paper proposes an image retrieval algorithm based on the path distance between categories in the sample category hierarchical structure. The Swin Transformer network is used to extract image features, and a similarity distance matrix is generated through the tree-like structure of image categories. The distance between the generated hash codes in the hash layer is consistent with the similarity distance matrix. In the Hamming space, similar images are relatively close, and completely dissimilar images have the greatest difference in hash codes. The distance between the hash centers of each category achieves a quantization effect. Experimental results on public datasets show that the introduction of sample category hierarchical structure and similarity distance loss significantly improves the accuracy of image retrieval.*

Keywords - *Image retrieval, Swin Transomfer, Similarity distance, Image hashing, Hamming space.*

1. Introduction

With the rapid development of computer technology and mobile internet, the importance of images in people's daily lives has been increasing. In various fields such as medical imaging, digital libraries, industrial production, security systems, transportation systems, and remote sensing systems, a large amount of image and video data is widely used. Therefore, fast retrieval of image and video data has become a challenging task. Image retrieval [1] is a method based on computer vision technology that aims to search and retrieve images by analyzing and comparing their features. The core of image retrieval is to extract the feature information of images, such as color, texture, shape, etc., and convert it into computationally processable data. Then, various similarity measurement methods are used to calculate the similarity between the query image and the images in the database, enabling image search and ranking.

In the 1970s, text-based image retrieval (TBIR) technology was proposed, which uses textual descriptions to search and retrieve images. TBIR utilizes textual semantic information for matching and provides semantically meaningful image search results. Although TBIR has the advantage of being fast and accurate, it also has some drawbacks. Firstly, text annotations of images cannot fully reflect the important information of the images themselves, resulting in insufficient richness of textual descriptions. Secondly, with the advent of the big data era, annotating massive images requires a significant amount of human effort and time. In the 1990s, content-based image retrieval (CBIR) technology emerged, which analyzes and queries images based on their content, such as color, texture, shape, and other low-level features. By mathematically describing the visual content of images using these low-level features, CBIR can reflect the visual content of the images themselves. The similarity measurement of image features is based on the extraction of image features and is calculated using a certain similarity calculation method (such as Euclidean distance). By sorting the similarity results, the desired images can be retrieved. The development of content-based image retrieval can be divided into two stages based on the emergence of deep learning: feature-based retrieval and deep feature-based retrieval. Traditional hashing methods use handcrafted features such as SIFT [2] (scale-invariant feature transform) and local feature descriptors [3] to solve the problem of poor invariance of global descriptors to brightness, transformations, occlusions, etc. However, the hash codes generated based on local feature descriptors do not consider the high-level semantic information of images, resulting in low retrieval accuracy.

When it comes to large-scale image retrieval, the combination of feature hashing and deep learning has become a trend. Deep hashing methods can be categorized into unsupervised, semi-supervised, and supervised deep hashing methods based on whether label information is used. Unsupervised and semi-supervised deep hashing methods further divide into those based on convolutional neural networks [4] (CNN), neural networks composed of self-attention mechanisms similar to Transformers, and unsupervised/semi-supervised deep hashing methods based on generative adversarial networks [5] (GAN). Supervised deep hashing methods can be further divided into triplet-based and pairwise-supervised deep hashing methods based on differences in data label information. Supervised deep hashing methods have achieved higher retrieval accuracy. Using deep hashing techniques for image retrieval is an effective method for the efficient retrieval of large-scale image data.

The earliest deep hashing methods used CNN as the backbone network for feature extraction, such as AlexNet [6], VGG [7], and ResNet [8], then mapped the continuous features of images to binary codes using sigmoid or ReLU nonlinear activation functions. Recently, Transformers have emerged as a new architecture that utilizes non-convolutional self-attention mechanisms. Transformers [9] have also been extended to computer vision tasks, and VIT [10] (Vision Transformer) is a hashing method based on Transformers for image retrieval. In 2014, CNNH [11] was proposed as the first deep neural network-based method for image retrieval. It can simultaneously learn feature representations and hash functions for images. The first stage of CNNH decomposes the similarity matrix into a low-dimensional hash matrix to obtain hash codes for each sample. The second stage trains hash functions using the obtained hash codes and class labels of each sample. NINH [12] network uses triplets of three images for training. In each triplet, the first two images are similar, while the first image and the third image are dissimilar. The objective of the triplet-based loss function is to ensure that the distance between similar samples in the obtained Hamming space is smaller than the distance between dissimilar samples. CSQ [13] image hashing algorithm introduces center similarity quantization, which encourages similar images to approximate a common hash center while different images converge to different hash centers. Hash centers are constructed using Hadamard matrices or Bernoulli distributions.

However, in the learning process of the above-mentioned deep supervised hashing methods, the general frameworks only utilize limited label information. It is generally assumed that images with the same label tend to converge to a hash center, and the distances between hash centers of different categories are equal. CSQ constructs a maximum hash center with mutually Hamming distances using label information directly. The Hamming distance between each class and other classes is the same. However, this approach ignores the similarity between image categories themselves. For data with hierarchical labels such as IAPRTC-12 and CIFAR-100 datasets, where the proximity between one category and another indicates their similarity level, and the farther the distance between categories, the lower their similarity.

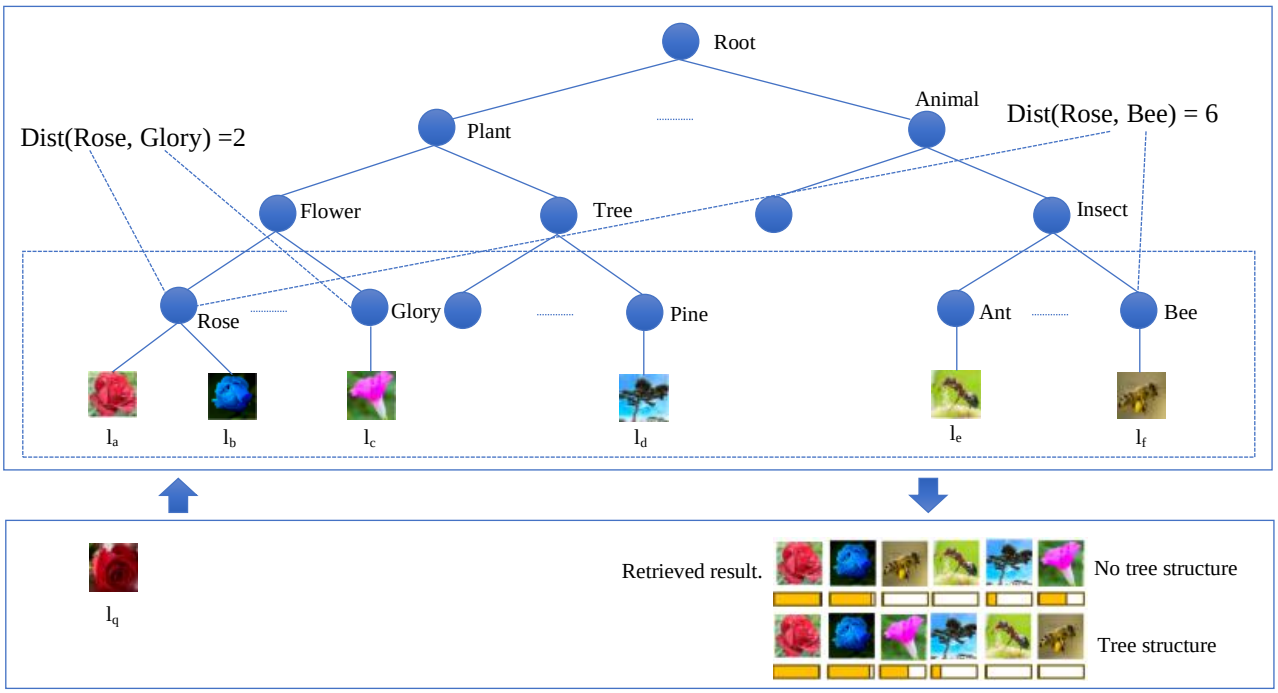

Fig. 1 Schematic diagram of retrieval results on a tree-structured data set

This hierarchical structure can be represented as a tree structure, with each category as a leaf node. By defining the distance between hash centers generated by the hash network based on the distance between leaf nodes, optimal performance can be achieved. Existing algorithms have not considered the hierarchical structure between categories. See Fig. 1, for example, for a query image labeled "rose," the retrieval results are ranked as "I_a, I_b, I_e, I_d, and I_c" in descending order. In this ranking, it is unreasonable for Ie and Id to rank ahead of I_c because although Ic is not a "rose," it shares a common parent class with Iq and belongs to flowers. In terms of similarity, I_c should be ranked higher than I_e and I_d.

2. Our Approach

2.1. Network Architecture

To address the above-mentioned problem, we propose a new supervised learning-based image hashing algorithm. In this paper, we utilize the Swin-Transformer [14] deep neural network for extracting image features.

We incorporate the path distance of the image category's hierarchical structure into the generation of hash codes, ensuring that the generated codes preserve the similarity.

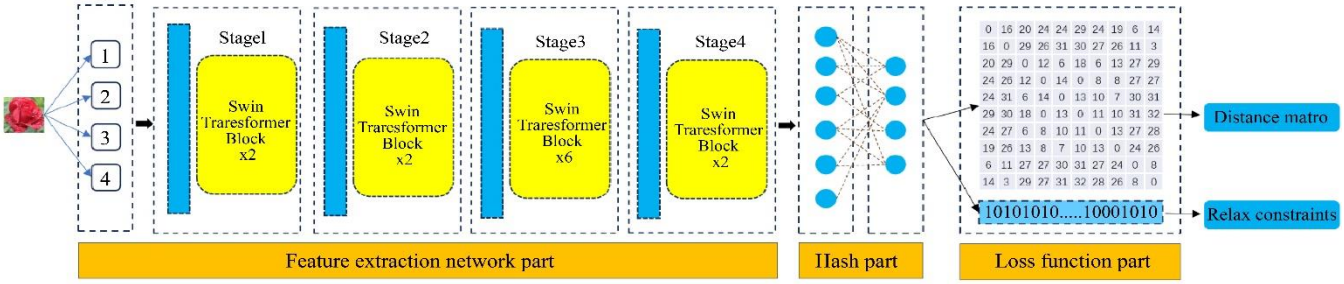

Fig. 2 The network structure diagram of this article

Figure 2 illustrates the structural model of the deep-supervised image retrieval method based on Swin Transformer. The model consists of three parts: the feature extraction network, the hash layer [15], and the loss function. The feature extraction part is based on the Swin Transformer architecture, which includes 1 patch partition and 4 stages. Each stage is composed of multiple stacked Swin Transformer blocks.

First, the input image is fed into the PatchPartition for a block operation, and then it is sent into the LinearEmbedding module to adjust the number of channels. Finally, through feature extraction and downsampling in stages 1, 2, 3, and 4, the final prediction result is obtained. It is worth noting that with each stage, the image size is reduced by half while the channel dimensions are doubled, similar to the ResNet network.

Each Swin Transformer Block in each stage consists of two connected Transformer Blocks based on W-MSA and SW-MSA (window-based multi-head self-attention) mechanisms, which improve computational performance. For the hashing task, a hash layer is added after Stage 4 to construct a hash feature extraction network. This network maps the output feature vectors to hash codes of different bit sizes. For a query image xi with a size of H×W, the feature extraction network can obtain the image's features.

$$zi = f(xi, \theta f) \tag{1}$$

The hash layer outputs:

$$\mathrm{Ci} = h(zi, \theta h) \tag{2}$$

In equations (1) and (2), f represents the feature extraction network, h represents the hash network, θ_f represents the parameters of the Swin Transformer feature extraction network, and θ_h represents the parameters of the hash layer.

2.2. Loss function

2.2.1. Hamming Distance Matrix

In a tree-like structure, the algorithm for calculating the path distance between two leaf nodes is as follows: Let *root* be the root node of the binary tree, n_1 and n_2 be two nodes in the given tree. lca is the lowest common ancestor of n_1 and n_2; $Dist(n_1,n_2)$ represents the distance between n_1 and n_2.

$$Dist(n1, n2) = Dist(root, n1) + Dist(root, n2) - 2 \times Dist(root, lca) \tag{3}$$

In Figure 1, the distance calculation for each category is as follows:*Dist(Rose, Glory)* = 2; *Dist (Rose, Pine)* = 4; *Dist(Rose, Ant)* = 6; *Dist (Rose, Bee)* = 6; If there are m categories in the samples, the size of the generated distance matrix is m×m. The elements of this matrix are:

$$M(i,j) = \begin{cases} 0, (i = j) \\ Dist(i,j), (i \neq j) \end{cases} \tag{4}$$

According to the given formula, for a dataset with m categories, we can obtain an m×m distance matrix M. Since the similarity of each category with itself is 100%, the diagonal of the matrix is all zeros. Taking K-bit hash codes as an example, to ensure a balanced hash distribution in the Hamming space, we set the maximum Hamming distance between hash centers of different categories as α×K. α is a hyperparameter, and experimental results show that when

α=0.5, the network converges fastest and achieves the highest accuracy in image retrieval. We can scale the category distance matrix accordingly.

$$D\ (i,j) = \frac{M(i,j)}{max(M)} \times \alpha \times K \quad (5)$$

Where *max(M)* is the maximum value in the distance matrix, the following loss function is used when generating hash codes.

$$\begin{cases} Loss1(b1,b2,d) = (d - H(b1,b2))^2 \\ s.t.bj \in \{+1,-1\}^k, j \in \{1,2\} \end{cases} \quad (6)$$

$H(b1,b2) = \frac{|b1-b2|}{2}$, $d = D(i,j)$, where i and j are indices representing categories. The relaxation constraint operation for hash codes is as follows.

$$Loss2 = \beta(||b1| - 1| + ||b2| - 1|) \quad (7)$$

The final loss function is:

$$Loss = (d - H(b1,b2))^2 + \beta(||b1| - 1| + ||b2| - 1|) \quad (8)$$

2.2.2. The Training Steps

For a dataset with a hierarchical structure, we can calculate the Euclidean distance between each category and other categories to obtain a similarity Euclidean distance matrix.

Then, by scaling the Euclidean distances, we can convert them into a similarity Hamming distance matrix. By constraining the loss function, the model can effectively encode the semantic information of images into hash codes.

Input: Training set X = {xi}, N, parameter α,β and the length K of hash codes.

Output: Parameters of the neural network and image feature codes Z.

Step 1: Calculate the similarity Hamming distance matrix based on the labels of the dataset using formula (4).
Step 2: Initialize the parameters of the Swin Transformer network and load pre-trained parameters. Add a hash layer on top of the Swin Transformer, consisting of two fully connected layers, with an output of K-bit binary codes C.
Step 3: Randomly select a batch and compute the continuous codes Z.
Step 4: Add a relaxation constraint Loss2 to the loss function Loss1 in the first training batch.
Step 5: Update the parameters of the hash layer in Step 2 using the backpropagation algorithm.
Step 6: Repeat Steps 2-5 until the network's output stabilizes and achieves the desired effect.

3. Results and Discussion

3.1. Experiment Parameter

All experiments were conducted using the PyTorch1.7 deep learning framework and a Geforce RTX 3060 graphics card. For data processing, the size of all images was first adjusted to 256×256. Then, for training images, standard image augmentation techniques, including random horizontal flipping and random cropping, were applied, with a random cropping size of 224. For test images, only center cropping with a cropping size of 224 was applied. In terms of parameter settings, BatchSize was set to 64, Adam optimizer was used, the learning rate was 0.0001, the weight decay value was 0.0001, and the number of training iterations was set to 150.

3.2. Datasets

This paper conducts experiments on CIFAR-100 and IAPR TC-12, two commonly used image retrieval datasets.

CIFAR-100 contains 60,000 images in 100 categories, with 600 images per category. There are 50,000 training sets, 500 for each category. Test set 10000, 100 for each category. The 100 categories of CIFAR-100 can be divided into 20 categories, and each image contains the exact category to which it belongs (that is, the category of the 100 categories), as well as the category of the category to which it belongs.

IAPRTC-12 contains a total of 20,000 images. Each image is manually segmented, and the resulting areas are annotated according to a predefined label vocabulary. The vocabulary is organized in a conceptual hierarchy. Visual features were extracted from each area. The annotation vocabulary has been organized on a conceptual level so that the IAPR TC-12 directory tree with root nodes has a total of seven levels. The dataset contains a total of six broad categories. This paper randomly selects 80% as the training set and the remaining 20% as the test set.

3.3. Evaluation Criterion

In this experiment, Normalized Discounted Cumulative Gain [16] (NDCG) is used as an evaluation index. Compared with mAP, NDCG has a feature that supports similarity measurement, while mAP [17] can only make binary judgments, i.e. similar or not similar. This feature is more reasonable when retrieving similar data. For the query sample q and the retrieval sequence V, the DCG formula is used for calculation.

$$DCG@k(q,V) = \sum_{i=1}^{k} G\,[rel(q,i)] \times D(i) \quad (9)$$

Where rel(q,i) represents the similarity between the i-th retrieved data and the query data, and the value ranges from 0 to 1. G(x) is the gain function, generally taken as $G(x)=2^x-1$,D(x) is the discount function, related to the position, generally taken as $D(i)=\log_2(1+i)$. DCG@k(q,V) is:

$$DCG@k(q,V) = \sum_{i=1}^{k} \frac{2^{rel(q,i)} - 1}{log_2(1+i)} \quad (10)$$

If the optimal retrieval sequence for q is I, then NDCG@k is:

$$NDGG@K(q) = \frac{DCG@K(q,V)}{DCG@K(q,I)} \quad (11)$$

It can be derived that the value range of NDCG is between 0 and 1. This article proposes a weighted recall rate to measure the recall rate in tree-structured data scenarios, defined as:

$$WeightRecall(q)@n = \frac{\sum_{i=1}^{n} rel(q,i)}{\sum_{i=1}^{N} rel(q,i)} \quad (12)$$

Where n is the number of top data points returned, and N is the length of the ranking list.

3.4. Experimental Results

The retrieval performance of different methods is shown in Table 1. By comparing the 32-bit, 48-bit, and 64-bit hash sizes with other methods, it can be seen that TSHH achieves higher NDCG@100 values on both CIFAR-100 and IAPRTC-12 datasets than other methods. On the CIFAR-100 dataset, as the encoding length increases from 32 bits to 64 bits, the NDCG@100 score calculated by TSHH increases from 0.6221 to 0.6586, significantly outperforming traditional hash methods based on deep learning features, especially on the IAPRTC-12 dataset. Compared to the highest value in the table for IAPRTC-12, VTS16-CSQ, TSHH shows an improvement of 6.39% at 32 bits, 4.05% at 48 bits, and 2.96% at 64 bits.

The reason for the more significant improvement in the IAPRTC-12 dataset compared to the CIFAR-100 dataset is mainly that the IAPRTC-12 dataset contains a more hierarchical structure of images than the CIFAR-100 dataset. The loss function utilizes the path distance information in the tree hierarchy of labels, allowing the hash code to have the distance semantic information of the tree hierarchy of labels, which further improves the retrieval performance in practical applications. Therefore, the TSHH method proposed in this paper is more effective on datasets with a deeper tree hierarchy.

In Figure 3, from (a) to (f), it can be observed that on the CIFAR-100 and IAPRTC-12 datasets, as the length of the hash code increases, the weighted recall score for images becomes higher. This indicates that the hash network proposed in this paper is capable of learning more semantic information within the tree-like structure of image categories. The enhanced recall curve shown in Figure (a) demonstrates that the TSHH model performs better than the baseline across all values of n from 0 to 5000.

3.5. Ablation experiment

This paper conducted ablation studies on the CIFAR-100 and IAPRTC-12 datasets to demonstrate the effectiveness of various components of the proposed deep hash model. The studies compared the retrieval metric (NDCG@100) using four different image hashing methods (CNNH, DCH, CSQ, and the hashing method proposed in this paper) with different feature extraction networks (AlexNet, VGGNet, ResNet50, VTS16, Our Backone).

Vertical comparisons (as shown in columns of Tables 2 and 3) proved that the feature extraction network used in this paper outperforms other backbone networks in retrieval performance across different hashing functions. On the CIFAR-100 dataset with 32-bit and 64-bit codes, it was observed that on CNNH, the NDCG increased by 8.0% and 3.5%, respectively, compared to AlexNet; 7.2% and 7.5%, respectively, compared to VGGNet; 1.1% and 0.8% respectively compared to ResNet50; and 0.4% and 0.7% respectively compared to VTS16. The backbone network proposed in this paper achieved the best retrieval results in vertical comparisons with DCH and CSQ. The effectiveness of the proposed backbone network on the IAPRTC-12 dataset is also proven in Table 3.

Table 1. Different NDCG@100 on two datasets

Method	CIFAR-100(NDCG@100)			IAPRTC-12(NDCG@100)		
	32bit	48bit	64bit	32bit	48bit	64bit
ITQ[18]	0.4197	0.4243	0.4272	0.6626	0.6633	0.6652
CNNH[11]	0.4413	0.4853	0.4921	0.6714	0.6822	0.6927
NINH[12]	0.5321	0.5559	0.5685	0.6881	0.6959	0.6985
DPSH[19]	0.5657	0.5693	0.5751	0.6853	0.6919	0.7034
DCH[20]	0.5872	0.5931	0.6117	0.6824	0.6962	0.7103
VTS16-CSQ[10]	0.5912	0.6066	0.6187	0.6932	0.7166	0.7287
TSHH	0.6221	0.6387	0.6586	0.7401	0.7571	0.7583

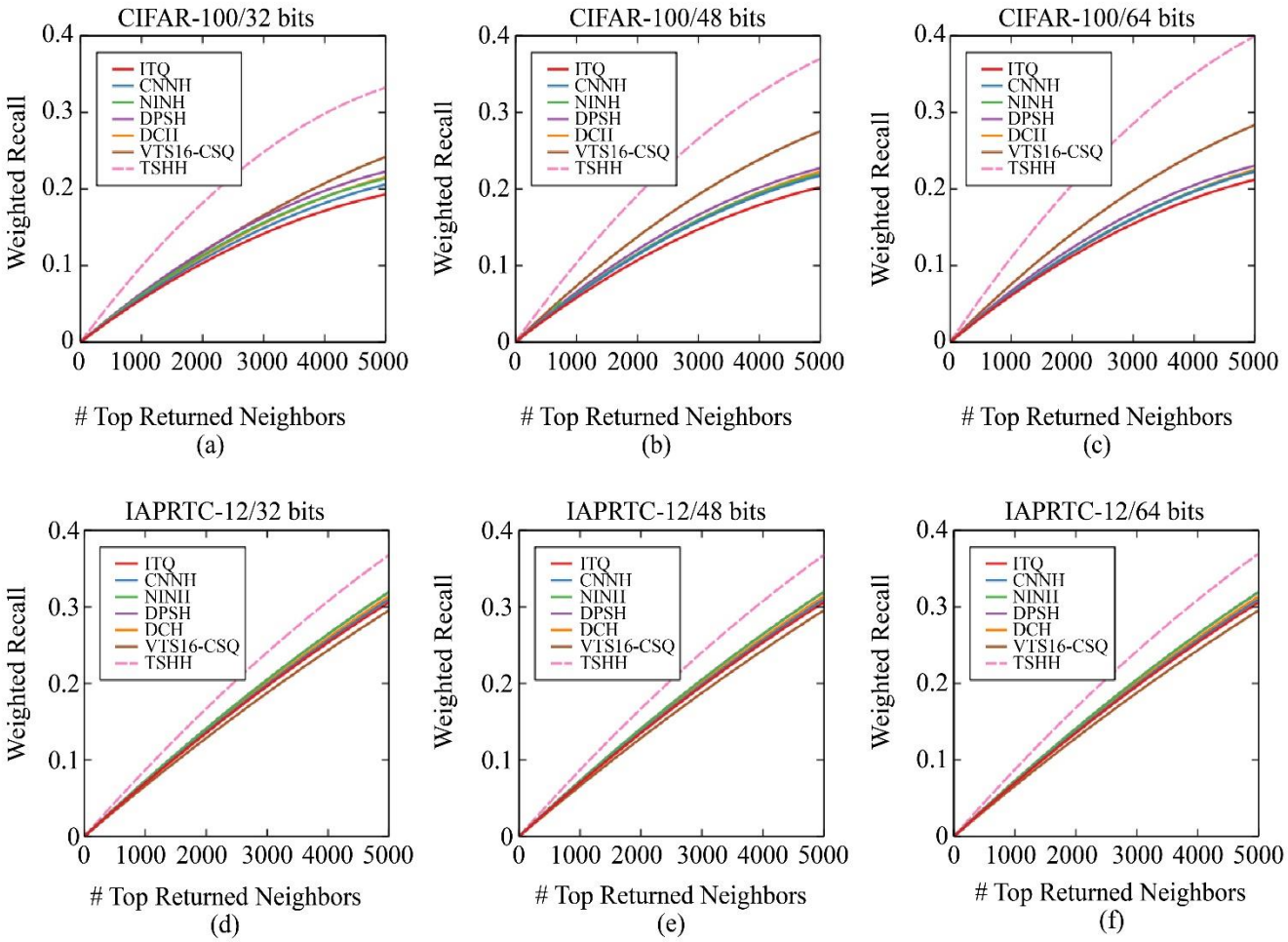

Fig. 3 Comparison of Weighted Recall@n values for different hash digits on CIFAR-100 and IAPRTC-12

Table 2. Comparison of retrieval metrics of different backbone networks on the CIFAR100 dataset

Backbone	CNNH		DCH		CSQ		Our loss	
	32bit	64bit	32bit	64bit	32bit	64bit	32bit	64bit
AlexNet[3]	0.441	0.492	0.547	0.553	0.523	0.535	0.590	0.598
VGGNet[7]	0.449	0.452	0.550	0.568	0.543	0.557	0.596	0.611
ResNet50[8]	0.510	0.519	0.587	0.591	0.582	0.588	0.603	0.614
VTS16[10]	0.517	0.520	0.580	0.588	0.591	0.607	0.614	0.630
Our backbone	0.521	0.527	0.592	0.695	0.610	0.615	0.622	0.659

Table 3. Comparison of retrieval metrics of different backbone networks on the IAPRTC-12 dataset

Backbone	CNNH		DCH		CSQ		Our loss	
	32 bit	64 bit	32 bit	64 bit	32 bit	64 bit	32 bit	64 bit
AlexNet[6]	0.671	0.693	0.677	0.690	0.662	0.678	0.683	0.690
VGGNet[7]	0.680	0.692	0.681	0.689	0.683	0.692	0.693	0.697
ResNet50[8]	0.688	0.692	0.682	0.710	0.682	0.710	0.711	0.723
VTS16[10]	0.691	0.694	0.688	0.715	0.693	0.717	0.745	0.753
Our backbone	0.694	0.699	0.695	0.717	0.701	0.718	0.757	0.758

Horizontal comparisons (as shown in rows of Tables 2 and 3) demonstrated that the retrieval performance using the same backbone network but with the loss function designed in this paper surpasses that of other loss functions. The experiments indicated that on the CIFAR-100 dataset with 32-bit and 64-bit codes using the AlexNet network, the loss function designed in this paper improved NDCG by 14.9% and 10.6%, respectively, compared to CNNH; 4.3% and 4.5% respectively compared to DCH; and 6.7% and 6.3% respectively compared to CSQ. The loss function proposed in this paper achieved the best retrieval results when compared with other feature networks. The effectiveness of the proposed loss function on the IAPRTC-12 dataset is also confirmed in Table 3.

Subsequent experiments were conducted to compare the effects of different values of parameters α and β on the retrieval results. These experiments were carried out on the IAPRTC-12 dataset with a hash size of 64 bits. The experiments concluded that the best results were achieved when the value of α was set to 0.5, and β was set to 0.01. Different values of α represent the maximum hash distance between different categories. For a K-bit hash encoding, when α is set to 0.5, the model converges the fastest and achieves the best performance.

Table 4. The impact of different α on the model

α	32bit-NDGG	64bit-NDGG
0.1	0.446	0.475
0.3	0.632	0.641
0.5	0.757	0.758
0.7	0.653	0.669
0.9	0.408	0.413

Table 5. The impact of different β on the model

β	32bit-NDGG	64bit-NDGG
0	0.382	0.411
0.001	0.532	0.558
0.01	0.754	0.761
0.1	0.723	0.756
1	0.458	0.503

Different values of β represent the proportion of quantization loss in the total loss. As β increases, the discrepancy between the network's discrete output values and the Hamming space can be reduced. However, an increase in β also leads to a decrease in the model's sensitivity to the distance loss among sample categories. Therefore, setting a reasonable value for β (such as 0.01) can significantly enhance the retrieval performance. (Tables IV and V).

4. Conclusion

To address the issue of consistent hash center distances in supervised image retrieval models, this paper introduces an image hashing model based on the hierarchical tree structure of samples. The model utilizes Swin Transformer as the network for extracting image features, combined with a custom hash module to generate hash codes. Experimental results show that on datasets where categories can form a hierarchical tree structure, the retrieval performance of the TSHH model is significantly improved. Particularly when the database lacks sufficient data of the same category as the queried image, images of similar categories are ranked higher, indicating that the model effectively leverages the hierarchical relationships between categories to enhance retrieval accuracy.

To further improve the model's applicability, future work will explore image similarity mining in datasets without a clear tree-structured relationship. The goal is to maintain the corresponding distance effect in the generated hash codes based on the degree of similarity between image categories, thereby improving image retrieval performance regardless of the dataset's structure. This will be challenging, as it requires the model to capture and utilize more subtle and implicit similarities between categories, potentially necessitating the development of new techniques or the improvement of existing ones to accommodate a wider range of application scenarios.

Funding Statement

The National Natural Science Foundation of China (61841602); The Natural Science Foundation of Shandong Province (ZR2020QF069).

References

[1] Homayoun Rastegar, and Davar Giveki, "Designing a New Deep Convolutional Neural Network for Content-Based Image Retrieval with Relevance Feedback," *Computers and Electrical Engineering*, vol. 106, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[2] Zahra Hossein-Nejad, and Mehdi Nasri, "An Adaptive Image Registration Method based on SIFT Features and RANSAC Transform," *Computers & Electrical Engineering*, vol. 62, pp. 524-537, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[3] Xiao Han et al., "SuperPointVO: A Lightweight Visual Odometry based on CNN Feature Extraction," *5th International Conference on Automation, Control and Robotics Engineering*, Dalian, China, pp. 685-691, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[4] Keiron O'Shea, and Ryan Nash, "An Introduction to Convolutional Neural Networks," *arXiv*, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[5] Antonia Creswell et al., "Generative Adversarial Networks: An Overview," *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 53-65, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[6] Cesare Alippi, Simone Disabato, and Manuel Roveri, "Moving Convolutional Neural Networks to Embedded Systems: The Alexnet and VGG-16 Case," *17th ACM/IEEE International Conference on Information Processing in Sensor Networks*, Porto, Portugal, pp. 212-223, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[7] Karen Simonyan, and Andrew Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *arXiv*, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[8] Fengxiang He, Tongliang Liu, and Dacheng Tao, "Why Resnet Works? Residuals Generalize," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 12, pp. 5349-5362, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[9] Ashish Vaswani et al., "Attention is All you Need," *Advances in Neural Information Processing Systems*, vol. 30, 2017. [Google Scholar] [Publisher Link]

[10] Alaaeldin El-Nouby et al., "Training Vision Transformers for Image Retrieval," *arXiv*, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[11] Rongkai Xia et al., "Supervised Hashing for Image Retrieval Via Image Representation Learning," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 28, no. 1, 2014. [CrossRef] [Google Scholar] [Publisher Link]

[12] Han Zhu et al., "Deep Hashing Network for Efficient Similarity Retrieval," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[13] Li Yuan et al., "Central Similarity Quantization for Efficient Image and Video Retrieval," *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3083-3092, 2020. [Google Scholar] [Publisher Link]

[14] Ze Liu et al., "Swin Transformer: Hierarchical Vision Transformer using Shifted Windows," *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10012-10022, 2021. [Google Scholar] [Publisher Link]

[15] Kevin Lin et al., "Deep Learning of Binary Hash Codes for Fast Image Retrieval," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 27-35, 2015. [Google Scholar] [Publisher Link]

[16] Yining Wang et al., "A Theoretical Analysis of Normalized Discounted Cumulative Gain (NDCG) Type Ranking Measures," *Proceedings of the 26th Annual Conference on Learning Theory*, vol. 30, pp. 25-54, 2013. [Google Scholar] [Publisher Link]

[17] Mascagni Pietro et al., "Artificial Intelligence for Surgical Safety: Automatic Assessment of the Critical View of Safety in Laparoscopic Cholecystectomy using Deep Learning," *Annals of Surgery*, vol. 275, no. 5, pp. 955-961, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[18] Yunchao Gong et al. "Iterative Quantization: A Procrustean Approach to Learning Binary Codes for Large-Scale Image Retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 2916-2929, 2012. [CrossRef] [Google Scholar] [Publisher Link]

[19] Richeng Xuan, Junho Shim, and Sang-Goo Lee, "Deep Semantic Hashing Using Pairwise Labels," *IEEE Access*, vol. 9, pp. 91934-91949, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[20] Yue Cao et al., "Deep Cauchy Hashing for Hamming Space Retrieval," *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1229-1237, 2018. [Google Scholar] [Publisher Link]