Design Studies and Intelligence Engineering
L.C. Jain et al. (Eds.)

doi:10.3233/FAIA220729

Similar Image Retrieval Algorithm Based on Feature Fusion and Locality-Sensitive Hash

Kaiqiang Zhang[a],Wei Li[a,1], Shaoyong Yu[b], Shunyi Chen[c], Zhiqiang Zeng[a], Xiaoyu Ma[a], and Youmeng Luo[a]
[a] *School of Computer and Information Engineering, Xiamen University of Technology, China*
[b] *School of Mathematical and Information Engineering, Longyan University, China*
[c] *Center of Modern Engineering Training, Xiamen University of Technology, China*

Abstract. In this paper, we propose a similar image retrieval algorithm based on feature fusion and locality-sensitive hash to address the problems of inadequate representation of image content by individual features and long retrieval time for massive image data. The fusion of global features and attention features makes the image features have both color structure and semantic information, which can better characterize the image content. In the image retrieval stage, the locality-sensitive hash is used to hash encode the image features, the cosine similarity is used as the similarity measure, and finally, the index is built to improve the retrieval efficiency. The similar image retrieval algorithm proposed in this paper has improved the average finding accuracy and recall rate on Caltech 256 and Corel5k datasets compared with other methods, and the retrieval time is greatly reduced.

Keywords. similar image retrieval, feature fusion, attention mechanism, Arcface loss, locality-sensitive hash

1. Introduction

With the development of the Internet and emerging technologies, the Internet and various fields have generated a large amount of image data, and people have started to get used to using cell phones and other devices to take images and share or retrieve them through the Internet, such as the photo query expected product information function of Taobao APP, and the preliminary review function of offending images on social networking platforms. In other professional fields, image retrieval also exists in a wide range of applications, such as medical image retrieval, remote sensing image matching, etc. Usually, a query image is entered, a database is retrieved, and the query image results returned are as similar as possible, i.e., content-based image retrieval (CBIR), and it has become an urgent need to get the required data in the massive image data efficiently and with high quality [1]. This is a long-standing research topic in the field of computer vision. The content-based image retrieval task is mainly divided into two parts, one is the extraction of image features, and the other is the image similarity calculation and matching, where the extracted image features play a key role in the matching results, and

[1] Corresponding Author: Wei Li, Email: drweili@hotmail.com

in large-scale image retrieval tasks, the database usually stores a huge amount of data, which puts higher requirements on the fast response of image retrieval.

1.1. Related Work

In the early research, similar images were retrieved mainly by hand-made local features of images, such as SIFT, SURF, ORB, etc., using the approximate nearest neighbor search method of KD tree. Until now, many fields still use this technique of retrieval based on local features of images.

In recent years, many works have focused on the aggregation methods of local features, such as VLAD [2] and Fisher Vector (FV) [3]. By aggregating the local features of an image, it is possible to obtain image features with high characterization power.

With the rise of deep learning and the great success of convolutional neural networks in the field of computer vision, many people started to shift their attention to deep image features based on convolutional neural networks. Some retrieval algorithms using convolutional neural network-based retrieval combine the extracted deep local features with traditional feature aggregation techniques to construct descriptors of deep local features of images, such as NetVLAD [4], which achieves a large improvement in the performance of image retrieval compared to traditional local features such as SIFT.

Many computer vision research topics use convolutional neural networks based on attention mechanisms, including target detection, visual question and answer, and image text recognition. However, in the field of image retrieval, the use of attentional mechanisms to extract image features for image retrieval has not been sufficiently studied. Recently, researchers have proposed an image local feature descriptor DELF [5] designed for large-scale image retrieval applications that incorporates the attention mechanism in feature selection, and this feature descriptor can better represent image features and achieve improvement in retrieval accuracy. However, DELF suffers from a relatively high feature dimensionality, which poses certain challenges for image feature storage and similarity calculation. DELG [6] is the latest image retrieval model proposed by Google, which unifies global and local features into a single depth model, thus enabling accurate retrieval through effective feature extraction and introducing an auto-encoder-based local feature dimensionality reduction technique to improve training efficiency and retrieval performance, but there is still some room for improvement in retrieval speed for large-scale image retrieval.

1.2. Main contributions

To address the problems in the above research work, a similar image retrieval algorithm based on feature fusion and locality-sensitive hash is designed in this paper. The main contributions of this paper are as follows：

1) A feature aggregation network based on the ECA attention mechanism is proposed. The two-branch feature extraction network extracts the global features and attention features of the image separately, and the final features are obtained by cascading the two features to characterize the image.

2) The complexity of the training images is enhanced by data pre-processing, the number of samples is enriched, the feature extraction network is trained using the arcface loss function, and different pooling layers are used for different branches of the network

as a way to focus on different image features as much as possible so that the cascade can complement each other and can better represent the images.

3) After the image features are extracted, the hash function is used to hash encode the feature vectors and construct the index through LSH, which can effectively reduce the similarity calculation time of the feature vectors and alleviate the problem of large storage space occupied by the feature vector database.

2. Similar Image Retrieval Framework

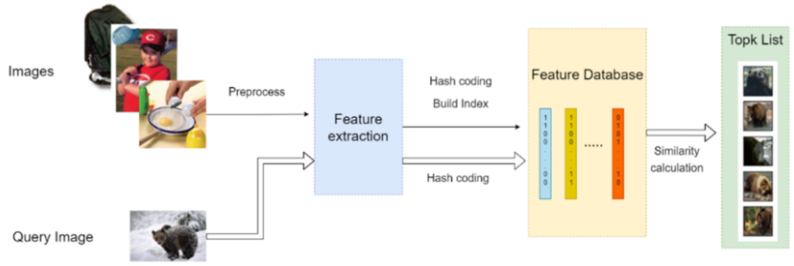

Figure 1. Algorithm framework

Combining feature fusion and locality-sensitive hash, a similar image retrieval algorithm is proposed in this paper, as shown in Figure 1, which is mainly divided into offline and online phases. In the offline phase, the processing of images includes image enhancement, feature extraction, feature hash coding and index construction, and in the online phase, the input images are subjected to feature extraction, feature similarity calculation and query index for similarity calculation.

In the offline stage, the images in the image database are first subjected to preprocessing operations, including uniform image size, cropping, blurring, adding noise, masking, and flipping operations to increase the number and complexity of the training images and improve the robustness of the image feature extraction model. Then, the images are fed into the convolutional neural network-based feature extraction module to obtain the image representation vector. Then, the fused feature vectors are hashed and encoded by a hash function, and the index is established using a locality-sensitive hash.

In the online stage, the user inputs a query image, which is hash coded after the feature extraction network, and the locality-sensitive hash makes cosine similarity calculation with the feature vectors in the image feature database to obtain k similarity equalities in descending order and return the corresponding image as the query result.

2.1. Image pre-processing

To make the feature extraction network more robust and generalizable, we performed image enhancement operations on each image in the image database, as shown in Figure 2, including uniform size to 224*224, random cropping, blurring, adding noise, random masking, and flipping operations.

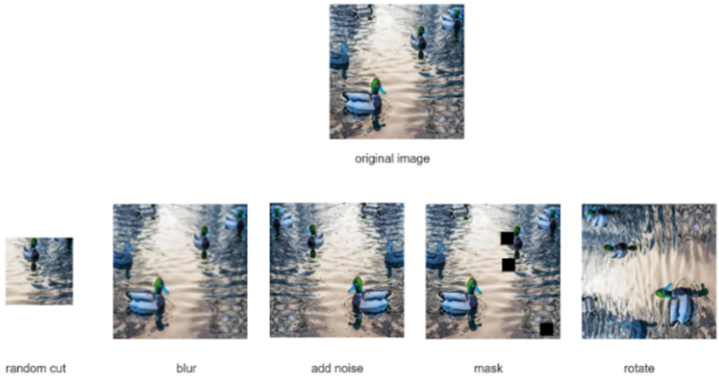

Figure 2. Image enhancement operation

2.2. Feature Extraction

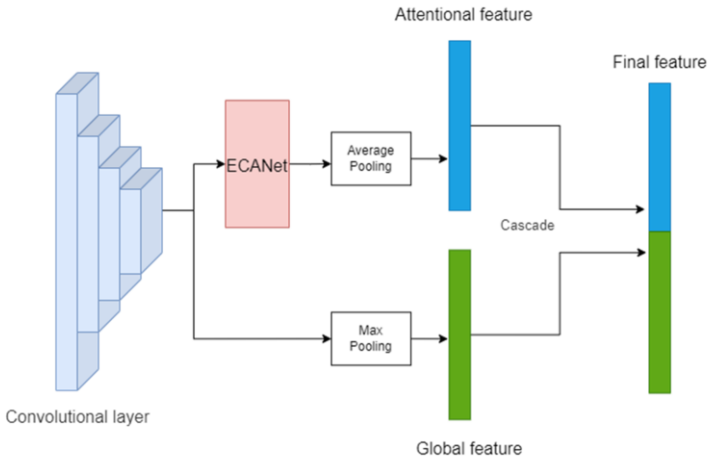

Figure 3. Feature extraction network

In this paper, we use a fully convolutional network as the backbone network for image feature extraction, as shown in Figure 3. The fully convolutional network is constructed using the feature extraction layer of a CNN trained with classification loss. We choose Resnet50 as the backbone network because Resnet50 has achieved very good performance on image classification tasks.

We use Average pooling and Max pooling for both branching networks. Average pooling can preserve background information and extract smoother features. Max pooling can capture local information and can better preserve features on the texture [7].

The image feature extraction module contains two extraction branches, the first branch of the first branch adds ECANet after layer4 convolution layer and uses the Average pooling layer to get the feature output of focused channel attention as attention image features, the second branch gets the feature output of layer4 convolution layer and

after Max pooling to get the global features of the image. Finally, the extracted features from the two branches are cascaded to get the final image features.

2.2.1. Efficient Channel Attention

SENet [8] is an attention mechanism proposed by Jie Hu et al. to improve the characterization of images by convolutional neural networks mainly by establishing relationships between image channels to correct the features of the channels and using global features to enhance the useful ones.

ECANet [9] is a channel attention mechanism proposed by Wang Qilong et al. Its main contribution is to demonstrate that dimensionality reduction in SENet has a negative impact on model performance, while bringing more parameters and increasing the computational effort of the model; it also demonstrates that proper cross-channel interaction has a positive effect on improving model performance. Therefore, ECANet improves on SENet by proposing a method that can autonomously choose the size of the convolution kernel, perform channel interaction through one-dimensional convolution, and extract the weights of each channel. The structure diagram of ECANet is shown in Figure 4.

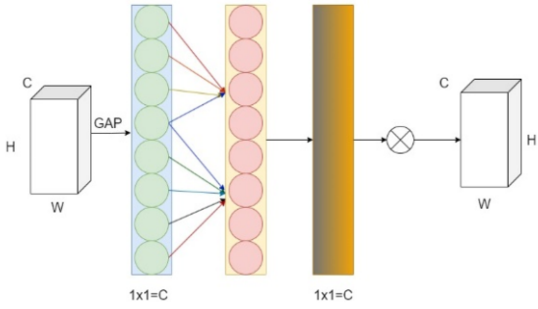

Figure 4. ECANet structure

For the feature layer with input size W×H and channel number C, as with SENet, the 1×1×C vector is also obtained by global averaging pooling, followed by discarding the series of operations of downscaling and upscaling to obtain the weights in SENet and performing a one-dimensional convolution of the vector with a convolution kernel of size k, and the value of k is related to the channel number c. The calculation is as follows.

$$k = f(c) = \left|\frac{log_2 c}{y} + \frac{b}{y}\right|_{o\,dd} \tag{1}$$

Finally, the ReLU function is used for activation to obtain the weight coefficients of each channel, and the feature layer with additional weights is obtained by the scale operation.

2.2.2. Loss function

Due to the existence of similar images in real scenes with large image background differences and high background similarity of heterogeneous images, which makes the inter-class distance of heterogeneous images small, and the standard cross-entropy loss function does not constrain the inter-class distance, the extracted features are easy to be

confused between classes. Therefore, this paper uses Arcface loss [10] as the loss function, which can achieve good results under weak labels [11], increases the inter-class distance and further converges the intra-class distance by adding an angle penalty term for the dissimilar classes in the angle domain [12]. The Arcface loss used for the loss function of the feature extraction network in this paper is as follows:

$$F_{Arc} = -\frac{1}{M}\sum_{i=1}^{M} log\frac{e^{s\left[\cos\left(\theta_{y_{i},i}+q\right)\right]}}{e^{s\left[\cos\left(\theta_{y_{i},i}+q\right)\right]}+\sum_{k\neq y_k} e^{s\cos(\theta_k,k)}} \tag{2}$$

2.2.3. Feature Fusion

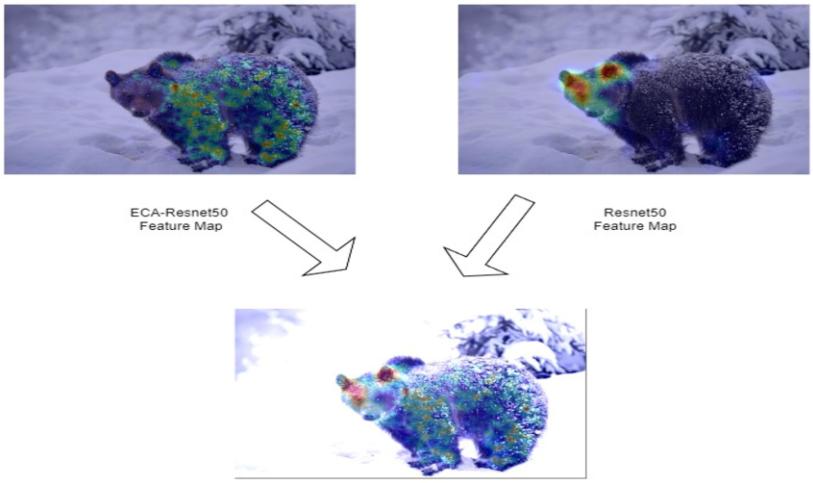

Figure 5. The two types of activation maps are complementary

The class activation map [13] is a weighted linear sum of visual patterns present at different spatial locations that can suggest which region of the image our view neural network is focusing on, and the feature heat map of the features output by the two-branch convolutional layer is shown in Figure 5, where it can be visualized that the two features can play a complementary role in the characterization of the image content.

The image feature extraction module contains two extraction branches, the first branch of the first branch adds ECANet after layer4 convolution layer, uses Average pooling, gets the feature output of focused channel attention as attention image feature, after squeeze operation, gets the feature vector f1 with length 2048; the second branch gets the feature output of layer4 convolution layer, after Max pooling, gets the global feature of image, after squeeze operation, gets the feature vector f2 with length 2048. The second branch obtains the feature output of layer4 convolution layer, and after Max pooling, gets the global features of the image, and after squeeze operation, gets the feature vector f2 of length 2048. finally, the features extracted from the two branches are cascaded:

$$F = [f_1, f_2] \tag{3}$$

F is the final image feature after feature fusion.

The length of the fused feature vector F is 4096, which will consume some storage space and is not conducive to the similarity calculation, so we will hash encode the features

and use a locality-sensitive hash for retrieval, which will be described in detail in the next section.

2.3. Similar image retrieval based on locality-sensitive hash

In traditional image retrieval, a linear search is used to find similar images. The feature extraction is completed for the image to be queried, and then the similarity is calculated with the image features so in the image feature database, and the corresponding n most similar images are returned according to the calculation result. In large-scale image retrieval scenarios, the time consumed by this matching strategy increases exponentially with the growth of data, and it is a challenge to the database storage space. The hash function is able to hash-code the data, which effectively reduces the dimensionality of the data and saves storage space [14].

2.3.1. Hash functions and feature encoding

We use the hash function to hash all the image data in the dataset. After the hash coding of two similar images in the database, the probability of staying adjacent in the new hash space is large, and the probability of staying non-adjacent for dissimilar images is large. By continuously dividing the hash space randomly, the probability of similar image data points falling in the same region is large.

Since the entire data set is divided using the hash function, the possible similar feature vectors are first obtained and then further filtered by the similarity calculation formula, so that the similarity calculation with all data points in the database can be avoided one by one and the efficiency of retrieval can be improved.

We set k hash function clusters to get k hash tables, each hash table corresponds to a function cluster H, each hash function cluster H contains b hash functions hash(), each hash function can generate a hash value, b hash functions finally generate a hash code of length b. As in Figure 6, the hash codes of the two images with similar semantic information are closer together, and the hash codes of the two images with dissimilar semantic information are further apart.

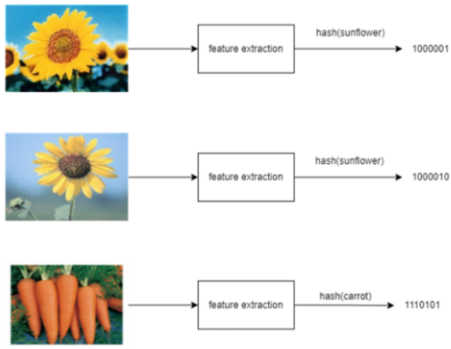

Figure 6. Hash encoding of features

2.3.2. Similarity calculation

When measuring the similarity of two features, we can use different metric distances, such as Jaccard distance, Hamming distance, Cosine distance, normal Euclidean distance,

etc. In this paper, we use cosine similarity to measure the similarity between different image features. The formula for calculating the cosine similarity is as follows:

$$\cos(\theta) = \frac{a \cdot b}{\| a \| \cdot \| b \|} \tag{4}$$

where a and b represent two data points that are projected into the hash space.

2.3.3. Build Index

Firstly, each hash-transformed data is projected into the same hash space, as shown in Figure 7, and then the hash space is divided randomly, and each data point will fall into one side of the hash space. After the hash space is partitioned several times, multiple regions are formed, and the data in each region are considered to be possibly adjacent, and then the hash table is constructed by hashing the data in each region into the corresponding Bucket by the hash function h(x). Each Bucket stores the similar feature hash codes that are partitioned into the same region.

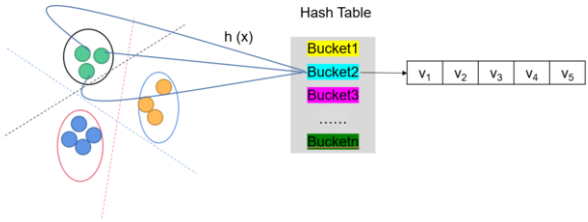

Figure 7. Building hash index

2.3.4. Online Search

In the online retrieval stage, the new query image first goes through the feature extraction network to get the corresponding representation vector, which is hash coded by the locality-sensitive hash and projected to be divided into a certain hash Bucket, and the similarity calculation is performed on all the feature vectors stored in this Bucket to return the first n similar image results of the user query.

3. Experiment

We conducted experiments on two publicly available datasets, Caltech 256 and Corel5k, to demonstrate the effectiveness of the method proposed in this paper, performing two sets of experiments, including comparing the search accuracy and retrieval time of different algorithms.

3.1. Dataset

The first dataset used for the experiments in this paper is the Caltech 256 dataset containing 20,607 images, with 256 categories and more than 80 images in each category. Some images of the dataset are shown in Figure 8.

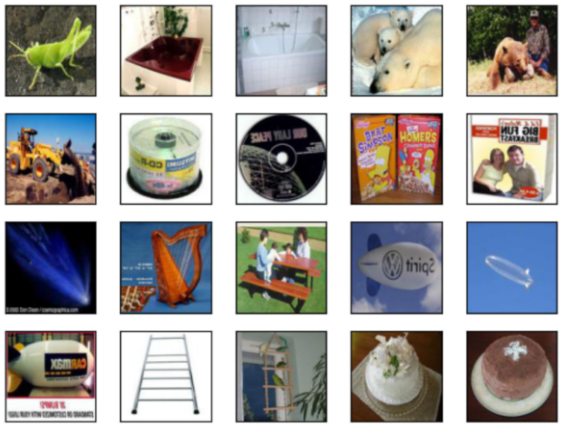

Figure 8. Partial images from the Caltech 256 dataset

The second dataset is Corel5k, which covers several topics, such as bridges, sunflowers, buildings, etc. Each topic contains 100 images of equal size. corel5k has become a standard dataset for image experiments since it was proposed for image annotation experiments and is widely used to compare the performance of annotation algorithms. Some of the images in the dataset are shown in Figure 9.

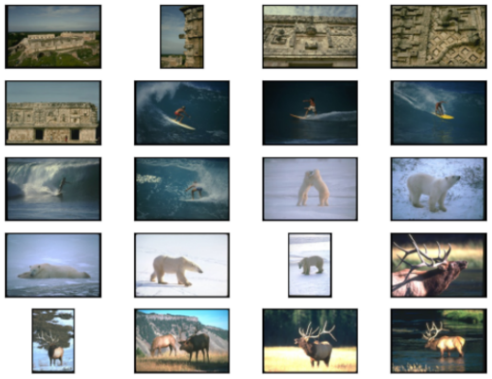

Figure 9. Partial images of the Corel5k dataset

3.2. Experimental environment

The experiments in this paper are conducted on Ubuntu 20.04.1 operating system, based on PyTorch 1.8 deep learning framework, Python version 3.8, hardware platform is Intel I5-12400F; Nvidia RTX3080 12GB; 64GB memory.

In the stage of building the feature database, the image is firstly preprocessed, the image size is resized to 112px*112px, and after fine tuning, the feature vector of layer4 of Resnet50 after maximum pooling is extracted and fused with the feature vector of layer4 of ECAnet-Resnet50 after maximum pooling, and the features are hash coded and indexed by the hash function.

In the similar image retrieval stage, feature extraction and hash coding are performed on the query image, and the cosine similarity degree is calculated for the

features in the image feature database by a locality-sensitive hash. The query and calculation returns the top M results of similarity degree, including the query sample itself, and the value of M is taken as 50 in the experiment.

3.3. Evaluation indicators and results

In order to quantitatively evaluate the superiority of this method, this experiment uses the mean lookup accuracy (MAP) and recall (Recall) as the metrics. Assume that the average lookup accuracy MAP is expressed as:

$$MAP = \frac{1}{x}\sum_{k=1}^{x}\left[\frac{1}{n}\sum_{i=1}^{n} f(i)\right] \tag{5}$$

The recall rate is calculated by the formula:

$$Racall = \frac{1}{m}\sum_{i=1}^{n} f(i) \tag{6}$$

f(i) is denoted as

$$f(i) = \begin{cases} 1, & similar \\ 0, & not\ similar \end{cases} \tag{7}$$

f(i) represents the similarity between two images, and the value of p is the total number of similar images.

We compare with three image retrieval algorithms, which are SIFT, NetVLAD, and Resnet.In traditional image retrieval, SIFT is often used to extract image features, and similar image retrieval is completed by using clustering algorithms such as k-means algorithm or kd-tree.CNN+NetVLAD retrieval algorithm CNN followed by NetVLAD layer makes the features more compact image features, neuralizes the traditional method and then matches similar images by linear search.Yan [15] et al. proposed to use Resnet as an image feature extraction network to calculate the similarity by linear search. Fusion of global and attentional features and retrieval using locality-sensitive hash is the method proposed in this paper. The results of MAP and recall for the four retrieval methods are shown in Table 1.

Table 1.MAP and Recall for different methods on dataset Caltech 256

Method	MAP	Recall	Top5	Top10	Top25
SIFT	0.573	0.338	0.782	0.641	0.568
CNN+NetVLAD	0.746	0.455	0.923	0.883	0.752
Resnet	0.893	0.476	1	0.933	0.903
Ours	0.942	0.573	1	1	0.952

As can be seen from Table 1, the MAP of the method proposed in this paper is improved by about 36% compared with the SIFT method, 19.6% compared with the CNN+NetVLAD method, and 4.9% compared with the Resnet method, while the recall rate is the highest among the four methods, which proves the effectiveness of the method.

We compared the retrieval performance of the four image retrieval methods on Caltech 256, Corel5k, and the results are shown in Table 2.

Table 2. Retrieval times of different methods on Caltech 256 and Corel5k

Method	Caltech 256	Corel5k
SIFT	3.1	3.7
CNN+NetVLAD	1.8	1.7
Resnet	1.6	1.8
Ours	0.34	0.39

From Table 4, we can see the performance of four different methods on the similar image retrieval task, the SIFT method takes the longest time, CNN + NetVLAD takes longer than both our methods, because the dimension of the image features extracted by these two methods is more than 512 dimensions, cut using linear search, which increases the time overhead of similarity calculation, while this paper gets the image through feature fusion with more features with semantic information, hash coding of features, effectively solves the problem of long feature vector length and high computational complexity, and avoids direct linear search through the index established in advance, and the retrieval time is greatly reduced from the experimental results.

To verify the effectiveness of feature fusion, a query case is shown in Figure 10, where the global feature extraction network, the feature extraction network with added attention mechanism, and the fused feature extraction network, each forming an image retrieval scheme, return the three most similar images, and it can be seen from the results that they are indeed highly complementary.

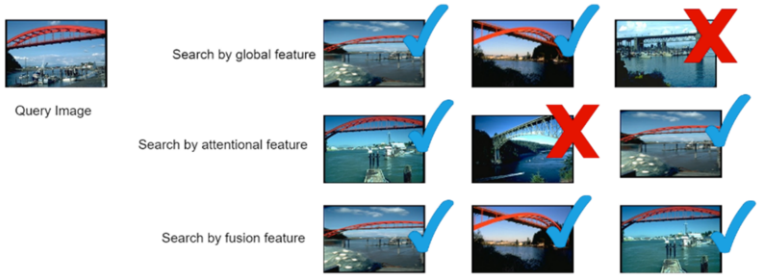

Figure 10. An example of a search based on Corel5k, returning the three most similar images

Figure 11 shows the query results returned by the four retrieval methods on the dataset Caltech 256. It can be intuitively seen that the SIFT method mainly focuses on the structural features of the images and ignores the semantic information of the images, and the retrieval results of CNN+NetVLAD and Resnet are greatly improved compared with SIFT, but there are still images with wrong matches, and the method proposed in this paper is matching accuracy and similarity ranking are better. However, it should be noted that since the feature extraction network in this paper is a two-branch feature fusion network, it takes more time to build the image feature database compared to the single feature extraction network, and we will subsequently optimize it in terms of improving the feature extraction time, such as using parallel computing.

Figure 11. Search results based on Caltech 256, comparing four different methods

4. Conclusion

The existing image retrieval algorithms, in the feature extraction stage, tend to focus on the global features of the image, the image attention mechanism can imitate the human eye observation and emphasize the important subject areas in the image, and the fusion of global and local features can better express the content of the image. In this paper, an image retrieval algorithm based on feature fusion and locality-sensitive hash is proposed. The final image features are obtained by fusing the global features based on neural network and the attention features based on attention mechanism. Experiments prove that the fused features are complementary and can better represent the image. In the image retrieval stage, the fused image features are encoded using a locality-sensitive hash, and the similarity of different images is measured using the cosine similarity measure, which reduces the similarity calculation and retrieval time by establishing an index and avoiding direct linear search. In the future, we will explore more feature fusion methods to improve the retrieval accuracy and consider combining PCA to improve the performance of image retrieval.

Acknowledgments: This research was funded by Fujian Natural Science Foundation of China (2022J011233) and Xiamen University of Technology (XPDKT20027).

References

[1] Javed A , Khan U A , Ashraf R . An effective hybrid framework for content based image retrieval (CBIR)[J]. Multimedia Tools and Applications, 2021.
[2] Jégou H, Douze M, Schmid C, et al. Aggregating local descriptors into a compact image representation[C]//2010 IEEE computer society conference on computer vision and pattern recognition. IEEE, 2010: 3304-3311.
[3] Sánchez J, Perronnin F, Mensink T, et al. Image classification with the fisher vector: Theory and practice[J]. International journal of computer vision, 2013, 105(3): 222-245.
[4] Arandjelovic R, Gronat P, Torii A, et al. NetVLAD: CNN architecture for weakly supervised place recognition[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 5297-5307.

[5] Cheng W, Shen Y, Zhu Y, et al. DELF: A Dual-Embedding based Deep Latent Factor Model for Recommendation[C]//IJCAI. 2018, 18: 3329-3335.
[6] Cao B, Araujo A, Sim J. Unifying deep local and global features for image search[C]//European Conference on Computer Vision. Springer, Cham, 2020: 726-743.
[7] Liu W J，Liang X J，Qu H C. Learning performance of convolutional neural networks with different pooling models［J］. Journal of Image and Graphics，2016，21(9) :1178-1190.
[8] Hu J, Shen L, Sun G. Squeeze-and-excitation networks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 7132-7141.
[9] Wang Q , Wu B , Zhu P , et al. ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks[C]// 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, 2020.
[10] Deng J, Guo J, Xue N, et al. Arcface: Additive angular margin loss for deep face recognition[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019: 4690-4699.
[11] Mao X Y and Peng Y B. 2020. Landmark recognition based on ArcFace loss and multiple feature fusion. Journal of Image and Graphics，25(08) : 1567-1577.
[12] Huang Z , Guan T , Qin W , et al. Gaussian-based probability fusion for person re-identification with Taylor angular margin loss[J]. Neural Computing and Applications, 2022:1-15.
[13] Zhou B, Khosla A, Lapedriza A, et al. Learning deep features for discriminative localization[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 2921-2929.
[14] Cao Y D,Liu Y Y,Sun F M,Jiang X.LSH with low space complexity for image retrieval[J].Computer Engineering And Science,2015,37(02):379-383.
[15] Yan L Q,Luo P R,Shi W,Liu X Q. Tangka Retrieval Based ResNet[J]. Journal of Ningxia University(Natural Science Edition),2021,42(03):257-262+269.