Computers, Materials & Continua CMC, vol.65, no.3, pp.2335-2350, 2020

Multi-Index Image Retrieval Hash Algorithm Based on Multi-View Feature Coding

Rong Duan[1], Junshan Tan[1, *], Jiaohua Qin[1], Xuyu Xiang[1], Yun Tan[1] and Neal N. Xiong[2]

Abstract: In recent years, with the massive growth of image data, how to match the image required by users quickly and efficiently becomes a challenge. Compared with single-view feature, multi-view feature is more accurate to describe image information. The advantages of hash method in reducing data storage and improving efficiency also make us study how to effectively apply to large-scale image retrieval. In this paper, a hash algorithm of multi-index image retrieval based on multi-view feature coding is proposed. By learning the data correlation between different views, this algorithm uses multi-view data with deeper level image semantics to achieve better retrieval results. This algorithm uses a quantitative hash method to generate binary sequences, and uses the hash code generated by the association features to construct database inverted index files, so as to reduce the memory burden and promote the efficient matching. In order to reduce the matching error of hash code and ensure the retrieval accuracy, this algorithm uses inverted multi-index structure instead of single-index structure. Compared with other advanced image retrieval method, this method has better retrieval performance.

Keywords: Hashing, multi-view feature, large-scale image retrieval, feature coding, feature matching.

1 Introduction

With the rise of smart phones and social tools, the rapid growth of image and video on the Internet has gradually formed a huge image database. In this era of big data, how to efficiently retrieve the images that users need is a research challenge in information retrieval. We hope to use some prediction [Wang, Kong, Guan et al. (2019)], feature fusion [Chen, Tao, Liu et al. (2020); Chen, Xu, Zuo et al. (2019); Liu, Zhu, Li et al. (2019)], coding [Tan, Qin, Xiang et al. (2019)] and recognition method [Wang, Qin, Xiang et al. (2019)] to effectively apply to image retrieval. Image retrieval is divided into two branches: text-based and content-based [Qin, Pan, Xiang et al. (2020)]. In recent years, content-based

[1] College of Computer Science and Information Technology, Central South University of Forestry & Technology, Changsha, 410004, China.

[2] Department of Mathematics and Computer Science, Northeastern State University, Tahlequah, OK 74464, USA.

[*] Corresponding Author: Junshan Tan. Email: T19852142@csuft.edu.cn.

Received: 17 June 2020; Accepted: 26 July 2020.

image retrieval technology has made great progress [Li, Qin, Xiang et al. (2018); Liu, Xiang, Qin et al. (2020); Qin, Li, Xiang et al. (2019); Xiang, Shen, Qin et al. (2019)], which aims to achieve effective feature matching of large-scale image retrieval.

The method based on bag-of-words (BoW) model is a common feature matching method [Zhou, Wu and Sun (2019)]. Based on this method, visual words are generated by the feature coding method for image feature, and then the quantified features are linked to the corresponding visual words to construct inverted index files, which match image features by searching. In order to generate the corresponding visual words, two feature coding methods are used in the BoW model: vector quantization based on K-means and scalar quantization. Vector quantization based on K-Means clusters features, each cluster center is regarded as a visual word, and all visual words constitute a codebook. However, visual codebook needs to be trained in advance, the calculation amount is large. When the data amount is large, the required memory space is also large. The method of scalar quantization quantifies the feature of each bin separately to generate visual words. Zhou et al. [Zhou, Li, Hong et al. (2015)] quantized each bin of SIFT feature to obtain 256-bit binary hash code. Zhou et al [Zhou, Yang, Wang et al. (2016)] proposed a dual cascade scalar quantization, which quantizes the feature of each bin at different resolutions. However, bin of similar features is not stable enough, they cannot be quantified as the same visual words sometimes. Therefore, in the feature matching stage, there will be a considerable lack of matching, which will affect the accuracy of image retrieval.

Data objects usually have multiple feature representations [Shen, Shen, Liu et al. (2018)] in some computer vision applications. In image retrieval, each database image can be described by different features, such as Gist feature, SIFT feature and so on. A feature is described as a view. The data that combines multiple view features is called multi-view, which describes image information more accurately [Liu, Wang, Zhang et al. (2014)]. Compared with single-view feature, multi-view feature has a higher level of semantic information. multi-view data through feature fusion has better feature expression than single-view data. Multi-view data has attracted more and more attention, which makes us also want to apply multi-view data to large-scale image retrieval to improve retrieval performance.

In image retrieval, feature matching between images can be regarded as the nearest neighbor (ANN) [Xue, Li, Peng et al. (2018)] to find features in a certain range. In some research, many hash algorithms about ANN search have been proposed. Hashing can be roughly divided into two categories: data-independent hashing and data-dependent hashing. Locality sensitive hashing (LSH) is a classical data-independent hash method. Some data-dependent hash methods have been proposed with better comparability, such as spectral hashing (SH), anchor graph hashing (AGH), iterative quantization (ITQ) [Gong, Lazebnik, Gordo et al. (2013)], convolutional neural network (CNN)-hash and so on. However, these hash methods can only map single-view to binary code, some multi-view hash methods came into being. Zhang et al. [Zhang, Wang and Si (2011)] proposed cross-modality search hashing (CHMIS), Liu et al. [Liu, Yu and Shao (2015)] proposed multi-view alignment hashing (MAH) and so on. However, the hash codes generated by these methods may not be optimal. In image retrieval, high-dimensional features are usually extracted to represent each image, while directly indexing these features and storing files will lead to high memory consumption. We hope to use multi-view data to describe the rich information of image, and use compressed hash code to describe high-dimensional feature efficiently.

In order to solve these problems, we propose a multi-index image retrieval hash algorithm based on multi-view feature coding. The purpose is to encode the image multi-view features to index files in a quantitative hash way, combine multiple index files for feature matching to reduce the matching error, thereby effectively apply to large-scale image retrieval. We summarize the main contributions of this work as follows:

1) This algorithm extracts many features of database image, such as Gist feature, SIFT feature and Densenet121 feature, and combines them to construct multi-view, which can describe the high-level information of image more comprehensively.

2) In this algorithm, K-means clustering is used to take the sequence number of clustering center as the index tag of index files, the hash codes of two different association features are linked to different inverted index files.

3) In the online query stage, the algorithm uses the distance between query image and cluster center constructed in the offline stage to find two candidate feature lists in two index files. The method of twice feature matching reduces the error of hash codes matching.

We have experimented with large open datasets. Experimental results show that the proposed method is superior to the advanced image retrieval method.

The rest of this paper is arranged as follows: Section 2 introduces the related work, Section 3 introduces our method, section 4 introduces the experiment evaluation, section 5 introduces the discussion, and the final conclusion is given in section 6.

2 DenseNet feature

Image retrieval technology based on deep learning mainly uses deep learning method in feature extraction module. The feature extracted by convolutional neural network represent the semantic information of high-level images, so it has good performance in image retrieval. So far, convolutional neural network has made many excellent achievements, such as AlexNet, VGGNet, ResNet and DenseNet [Luo, Qin, Xiang et al. (2020)]. In convolutional neural network, the pooling layer and convolution layer can extract the translation invariance of input features, and recognize the similar features of different spatial, which make CNN widely concerned in computer vision application.

DenseNet is a kind of convolutional neural network with dense connection. DenseNet solves the problem of gradient disappearance of deep network, strengthens the propagation of features, encourages feature reuse, reduces model parameters, it has better performance than other neural networks. In DenseNet, each dense block module of DenseNet makes use of the information of all the previous layers in the module. There are dense connections between any two layers, which realizes the reuse of features. Compared with the traditional neural network, the DenseNet network can extract higher-level image information and describe the image information more abundantly. Therefore, this paper uses the DenseNet121 network model for fine-tuning training and extracts DenseNet121 feature from the dataset to describe the image as one of the view features. We use DenseNet121 feature combined with BoW_SIFT or Gist feature to outperform low-level feature fusion.

3 Multi-index image retrieval hash algorithm based on multi-view feature coding

We define the features of database images, each of which corresponds to a view. $\{X^{(m)} = [X_1^{(m)}, \ldots, X_N^{(m)}]^T \in R^{N \times d_m}\}_m^M = 1$, Where *N* is the number of images in the database, d_m is the dimension of the *m-th* view, *M* is the number of views. In Tab. 1, we summarize the important notation in this article.

Table 1: Notations description

Notation	Description
$X^{(m)}$	Data matrix of the *m-th* view
d_m	Dimensionality of the *m-th* view
B	Hash code matrix
Y	Low-dimensional feature representation
R	Orthogonal transformation matrix
L	Graph Laplacian matrix
P	Transformation matrix
Kc	The number of cluster centers
N	The number of dataset images
M	The number of views
r	The length of hash code

3.1 Construction of multi-view feature

In order to use multiple features of the image to construct multi-view feature, we use a method based on random kernel canonical correlation analysis (RKCCA) [Lopez-Paz, Sra, Smola et al. (2014)] to obtain the maximum correlation between multiple views. Combine the analyzed feature vectors to obtain a feature vector that is more representative than a single feature vector. The RKCCA method relates the nonlinear relationship between two multidimensional random variables, which corresponds to the linear canonical correlation analysis on a pair of random nonlinear maps.

Firstly, according to the nonlinear random feature, the parameter $W_i \in R^d$ is randomly selected from the data independent distribution $p(w)$, construct a random feature map *z(x)* for the input data $X \in R^{N \times d}$, *m* is dimension.

$$z(X) := [z_1 \ \cdots \ z_m] \in R^{N \times m} \tag{1}$$

where $z_i := [cos(w_i^T x_1 + b_i), \ldots, cos(w_i^T x_N + b_i)] \in R^n , b_i \sim u(0,2\pi)$.

According to Bochner's theorem, the random nonlinear feature and the shift invariant kernel are connected. The *k (x, y)* is normalized to shift invariant kernel on $R^d \times R^d$.

$$k(x,y) = \int_{R^d} p(w) e^{-jw^T(x-y)} dw$$
$$\approx \sum_{i=1}^{m} \frac{1}{m} e^{-jw_i^T x} e^{jw_i^T y} = \langle \frac{1}{\sqrt{m}} z(x), \frac{1}{\sqrt{m}} z(y) \rangle \tag{2}$$

where $p(w)$ is defined as the inverse Fourier transform of k. Suppose K is the kernel matrix of data X, that is, $K_{ij} = k(x_i, y_j)$. When the kernel K is approximated by m random Fourier features, the kernel matrix K can be approximated,

$$K \approx \widehat{K} \coloneqq \frac{1}{m} z(X)z(X)^T = \sum_{i=1}^{m} \widehat{K}^{(i)} \tag{3}$$

The correlation between two views X^i, X^j *(1<i<M, 1<j<M)* is obtained by combining the kernel matrix and the linear canonical correlation analysis (CCA) method:

$$RKCCA(X^i, X^j) \coloneqq CCA(z_x(X^i), z_y(X^j)) \approx KCCA(X^i, X^j) \tag{4}$$

Through the analysis, the random weight, CCA projection matrix, feature average and other parameters of the two views are obtained respectively. The polynomial combination of the parameters and the original data matrix is used to obtain the fused association feature. Two new association features are connected in series to form a new feature vector, which can more accurately describe the images.

3.2 Hash code generation and optimization

The key of image retrieval is to keep the similarity of the original data. Laplacian mapping is used to map the data to a low dimensional space and try to keep the similarity relationship between the original data. The matrix form of the similarity preserving function is as follows:

$$\begin{gathered} \min_{B} Tr(B^T L B) \\ s.t.\ B^T B = N I_r \end{gathered} \tag{5}$$

In order to reduce the quantization loss of hash code, the objective function of similarity reservation of binary hash code is as follows:

$$\min_{Y} Tr(Y^T L Y) + \lambda \|YR - B\|_F^2 \tag{6}$$

where λ is the regularization parameter, Y is the feature representation of low dimension, L is the Laplacian matrix, and R is the rotation matrix. We use iterative quantization to quantize association feature into binary hash code with minimum quantization error.

According to the objective function of similarity preservation, we use the curvilinear search algorithm for optimization on Stiefel manifold to optimize the generation of hash codes. We use the association feature to perform singular value decomposition (SVD) to obtain the low-dimensional feature representation Y. Use feature Y as the input of the quantization hash method to solve the optimal rotation matrix R in the newly mapped hypercube space. Combined with the gradient descent of solving Y and iterative quantization, the generation of hash codes is optimized. The formula for solving the hash code is as follows,

$$B = sign(YR) \tag{7}$$

where *sign (·)* is a sign function.

3.3 Construction of inverted double index file

In image retrieval, feature matching between images can be regarded as the nearest or approximate nearest neighbor [Xue, Li, Peng et al. (2018)] to find features in a certain range. However, in large-scale image retrieval, such search is very expensive in calculation. In order to solve this problem, we use the number of cluster centers of K-means clustering as the index tag to link the image ID and the hash codes to construct inverted index files. The two different association features after feature fusion are respectively quantified to obtain the corresponding hash codes, so as to establish two inverted indexes to improve the retrieval accuracy during image matching.

Firstly, two association features are concatenated to construct multi-view data. We give the number of clustering centers Kc and performs K-means clustering on multi-view data. The purpose of K-means clustering is to minimize the sum of squares of the distance between all data points of multi-view feature $X = [X^{(1)}, X^{(2)}]$ and the cluster center μ_j.

$$dist = \sum_{j=1}^{Kc}(X_i - \mu_j)\text{，} \quad i=1,2,\dots,N \tag{8}$$

Through K-means clustering, each object is divided into clusters represented by the nearest cluster center. Then we solve the binary codes of the two association features and record the hash codes of the corresponding features in different index files. The cluster center number corresponding to the image is used as the index tag of the inverted index file, so that each inverted index file contains Kc index entries. Each index item links an item list, each index feature in the list records the r-bit hash code and corresponding image ID. The corresponding index entries of each index file store the same image IDs and different hash codes. Fig. 1 shows the structure of the inverted double index file. Algorithm 1 describes the construction process of inverted double index file.

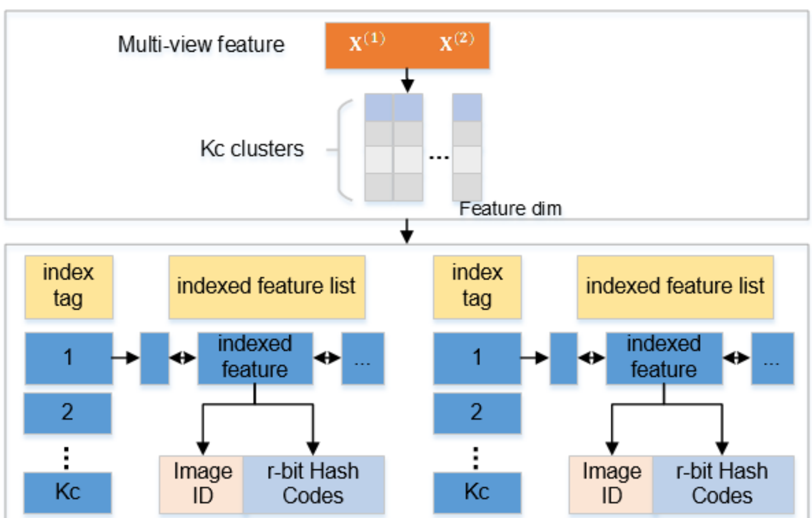


Figure 1: The construction process of inverted index files

Algorithm 1: Construction of inverted double index files

Input: Training set $\left\{X^{(m)} \in R^{N \times d_m}\right\}_{m=1}^{M}$, binary code length r, parameter λ, projection dimensionality *pd*, number of cluster centers *Kc*.

Output: Binary codes B, inverted index files.

1: Multiple view features $X^{(m)}$ are fused by RKCCA;

2: Connecting association features in series;

3: K-means clustering is used to separate the multi-view data into different clusters;

4: for i=1:m

5: Initialize Y by doing SVD on the weighted association features in series;

6: The initial *B* and *R* were obtained by iterative quantization of *Y*;

7: Update Y with curvilinear search algorithm using Eq. (6);

8: Update *B* and *R* were obtained by iterative quantization of *Y*;

9: Link the *B* and ID of each image to the index feature list of the *m-th* index file according to the number of clustering centers *Kc*;

10: end.

3.4 Image matching of large-scale image retrieval

In this section, we introduce the implementation of feature matching in the online query stage. Through the inverted index file established in the offline stage, query image can search for its matching object. The specific retrieval process framework of this scheme is shown in Fig. 2. In the offline stage, the inverted double index file is constructed by the method of Section 3.3. In the online query stage, the distance between the feature of query image and cluster center is calculated to obtain the index number of the cluster center corresponding to the minimum distance. The corresponding indexed feature list in the two index files is found as two candidate lists by index tag, and the Hamming distance is calculated by using the hash codes of the two association features of query image and the hash codes of corresponding candidate lists to match. Finally, the image corresponding to the first *k* minimum distances is returned as the retrieval results.

3.4.1 Hash code generation of query image

Assuming that there is a linear mapping between the original feature data and hash code, the method of out-of-sample expansion [Xiang, Shen, Qin et al. (2019)] is used to generate the hash code of the query image. By solving the following problems.

$$\min_{P} \|B - XP\|_F^2 + \delta\|P\|_F^2 \tag{9}$$

The transformation matrix *P* can be obtained as follows, the regularization parameter δ is set to 0.001 in the experiment.

$$P = (X^T X + \delta I_D)^{-1} X^T B \tag{10}$$

For any query image $x_q = \left[x_q^{(1)T}, \dots, x_q^{(M)T}\right]$, the binary hash code of x_q is obtained from the following formula.

$$b_q = sign(x_q P) \tag{11}$$

where $sign(\cdot)$ is a sign function.

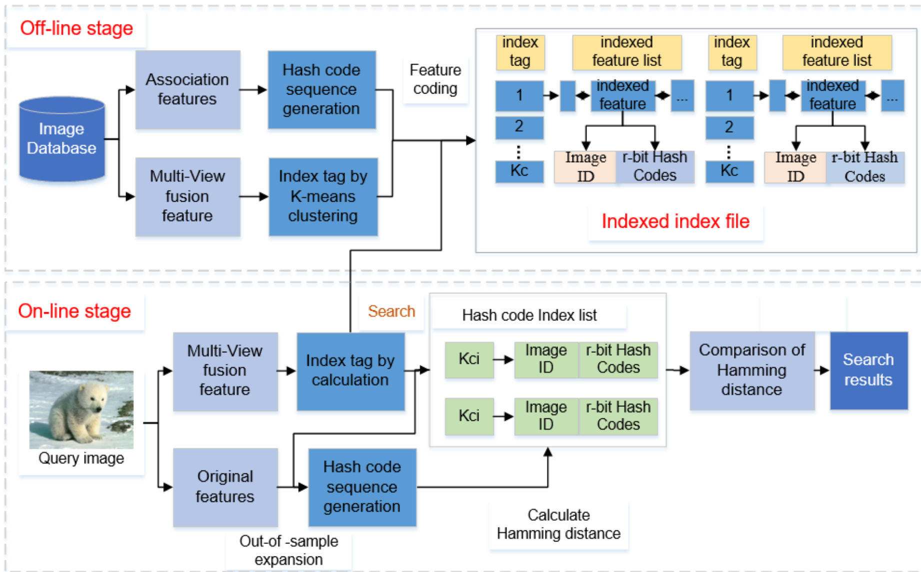


Figure 2: The retrieval process framework of the proposed method

3.4.2 Feature filtering and matching

The single-view feature is extracted from the query image, two association features are obtained by using the weight of the corresponding view, CCA projection matrix and other parameters obtained in the offline stage. In this paper, we calculate the Euclidean distance between the multi-view feature of the query image and the clustering center obtained in the offline stage, so as to obtain the sequence number of the nearest clusters (index tag in the inverted index file). Two candidate feature lists are obtained from the corresponding two inverted index files by index tag, different candidate feature lists are matched with different original view data features of the query image.

After obtaining the two candidate feature lists and hash codes corresponding to the query image, the Hamming distance is calculated to measure the similarity between the image in the candidate lists and the query image.

We compare the bit vectors of query feature b_q and database feature b_d as follows:

$$dist = hammingDist(b_q\,,\ b_d) \tag{12}$$

Finally, we sort the Hamming distance between the two candidate feature lists and the query image, and return the images corresponding to the first *k* minimum distances as the retrieval results.

Algorithm 2 describes the feature matching process in the online query stage.

Algorithm 2: Feature matching in online query stage

Input: Query set $\left\{X_q^{(m)} \in R^{N\times d_m}\right\}_{m=1}^{M}$, binary code length r, feature correlation parameters, cluster center matrix, inverted index files.

Output: Hamming distance, search results.

1: Using the original feature $X_q^{(m)}$ and the feature correlation parameters obtained in the off-line stage to obtain the association features;

2: Connecting association features in series;

3: Calculate the Euclidean distance between the multi-view data of X_q and the clustering center obtained in the off-line phase;

4: Using the sequence number of the clustering center of the minimum Euclidean distance as an index tag;

5: Obtain two candidate feature lists corresponding to the two inverted index files through the index tag;

6: for i=1:m

7: Using Eq. (11) to get the binary hash code of $X_q^{(m)}$;

8: Using Eq. (12) to get Hamming distance;

9: Record the smaller Hamming distance of the same image;

10: end.

4 Experiments

In this section, we evaluate the performance of this algorithm in large-scale image retrieval. The specific experimental environment and datasets are described as follows.

Experimental environment: Intel (R) core i5-9300H @ 2.40 GHz, 8 GB ram and NVIDIA GeForce GTX 1650 GPU. All experiments were carried out on MATLAB r2017a. We use widely used benchmark datasets, namely Caltech-256 and Caltech-101.

Caltech-256 dataset: a total of 29780 images, including 256 categories, each category has about 80 to 800 images. We randomly selected 1000 pictures as the query set, and all 29780 pictures as the database to build the inverted index file.

Caltech-101 dataset: a total of 9144 images, including 102 categories (one of which is background), each category has about 40 to 800 images. We randomly selected 3019 pictures as the query set, and all 9144 pictures to build the inverted index file.

For each image in different datasets, we extract 512 dimensions Gist feature, 500 dimensions BoW_SIFT feature and 1024 dimensions DenseNet121 feature. We select two features to learn the maximum correlation between each other, set the feature projection dimension to 512 dimensions, and concatenate two association features as multi-view data.

4.1 Average accuracy analysis

In this section, we test the performance of this algorithm on Caltech-256 and Caltech-101 datasets. Compared with the existing multi-view hash method based on the RKCCA [Tan, Duan, Qin et al. (2020)] (hereinafter referred to as "FFRK") in different view feature. In the average accuracy experiment, we set the value of the parameter λ in the similarity preservation function to 10^{-2}, and select different view weights to test.

We perform image retrieval based on different queries on different datasets, and match the images by comparing the hash codes of query image and the binary inverted index file. This algorithm compares the Top-k results returned from the specified query set with real nearest neighbors on dataset, and evaluates the average mean accuracy (mAP).

4.1.1 Average accuracy analysis of multi-view features

We use three features extracted from dataset images to test the average accuracy of our method on the multi-view combination of high-level feature DenseNet121 and the other two low-level features. The two sets of data features are as follows: (1) Gist feature and DenseNet121 feature to construct multi-view; (2) BoW_SIFT feature and DenseNet121 feature to construct multi-view. The statistics of the Top-50 mAP test results of different hash code bits on the Caltech-256 and Caltech-101 datasets are shown in Tab. 2, 3 below. The *Kc* value of Caltech-256 is set to 1024, while that of Caltech-101 is 1536. The effectiveness of this algorithm can be seen by comparing the results of Tab. 2, 3.

Table 2: Top-50 mAP comparison with respect to different number of bits and different features on Caltech-256 dataset

Feature	16		32		64		128	
	FFRK	OURS	FFRK	OURS	FFRK	OURS	FFRK	OURS
Gist+ DenseNet	0.6115	0.9169	0.7965	0.9255	0.8646	0.9530	0.8978	0.9585
BoW_SIFT+ DenseNet	0.6617	0.9016	0.8069	0.9054	0.8756	0.9329	0.9102	0.9441

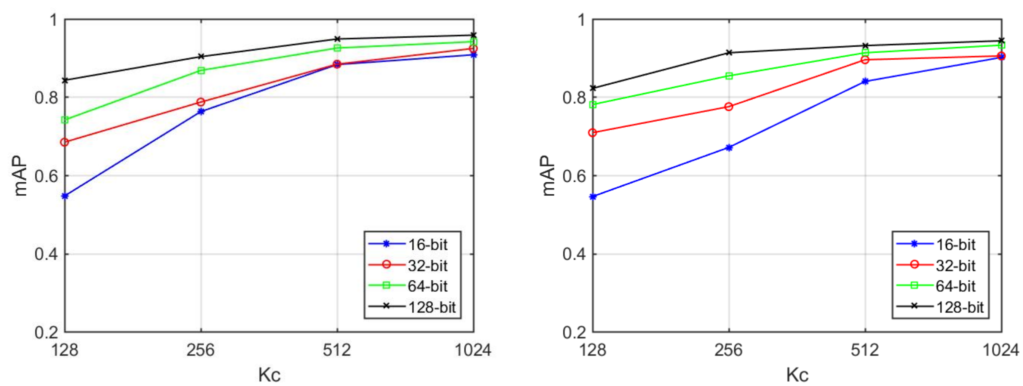
Table 3: Top-50 mAP comparison with respect to different number of bits and different features on Caltech-101 dataset

Feature	16		32		64		128	
	FFRK	OURS	FFRK	OURS	FFRK	OURS	FFRK	OURS
Gist+ DenseNet	0.8483	0.9547	0.9595	0.9681	0.9793	0.9716	0.9834	0.9719
BoW_SIFT+ DenseNet	0.9284	0.9501	0.9735	0.9613	0.9801	0.9717	0.9815	0.9839

Compared with the FFRK method, our method also performs experiments on two sets of feature combinations. As shown in Tab. 2, on the Caltech-256 dataset, the Top-50 results of different multi-view combinations under different hash code lengths have all been greatly improved. The mAP results shown in Tab. 3 show that on the Caltech-101 dataset, the Top-50 results of different multi-view combinations under different hash code lengths have been partially improved, and the mAP results are generally higher. Experimental results show that: (1) Using dense convolutional neural network DenseNet121 feature and low-level features (such as BoW_SIFT feature, Gist feature) combined with random kernel canonical correlation analysis method can maintain the efficiency of image retrieval. (2) The feature fusion method based on RKCCA combined with the multi-index retrieval method can improve the retrieval accuracy of image retrieval. These results prove that the Top-k retrieval performance of this algorithm is better than the FFRK method.

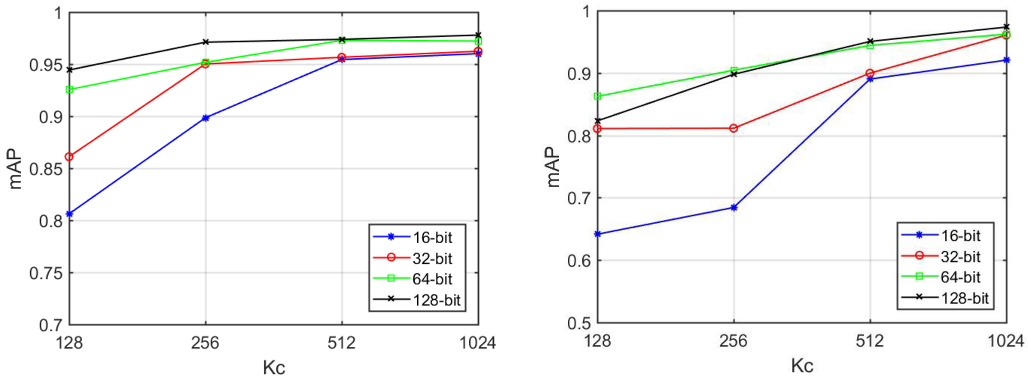
4.1.2 Average accuracy analysis under different Kc

In this section, we will test the impact of different *Kc* values on image retrieval results under different hash code lengths on different datasets. Fig. 3 shows the Top-50 results of different *Kc* values on the Caltech-256 dataset, where the multi-view combination is (a) Feature combination: Gist feature and DenseNet121 feature, (b) Feature combination: BoW_SIFT feature and DenseNet121 feature. It can be seen from the figure that under each hash code length, with the increase of the number of cluster centers *Kc*, the value of mAP increases gradually. The longer the hash code is, the larger the *Kc* is, the better the effect of image retrieval. In Fig. 3(a), when the hash code length reaches 128-bit and *Kc* value is 1024, the mAP result of top-50 is 0.9585. In Fig. 3(b), when the hash code length reaches 128-bit and *Kc* value is 1024, the mAP result of top-50 is 0.9441.



(a) (b)

Figure 3: The mAP results of different *Kc* in different bits under different multi-view feature of Caltech-256 dataset



(a) (b)

Figure 4: The mAP results of different *Kc* in different bits under different multi-view feature of Caltech-101 dataset

Fig. 4 shows the mAP results of different *Kc* values on the Caltech-101 dataset, where the multi-view combination the same as that used in Fig. 3. As can be seen from the figure, under each hash code length, with the increase of the *Kc*, the value of mAP also increases almost gradually. In Fig. 3(a), when the hash code length reaches 128-bit and *Kc* value is 1024, the mAP result of Top-50 is 0.9779. In Fig. 3(b), when the hash code length reaches 128-bit and *Kc* value is 1024, the mAP result of Top-50 is 0.9735.

4.1.3 Average accuracy analysis of Top-k

In this section, we will test the mAP of our method and the FFRK method under different values of Top-k. The experiments in this section are conducted on the Caltech-256 dataset. The multi-view feature is a combination of BoW_SIFT and DenseNet121 features. The hash code length is set to 128-bit, the *Kc* value is set to 1024, and the *k* value of Top-k is set to 5, 10, 20, 30, 40, 50. From the result curve in the figure below, we can see the retrieval accuracy of our method in the Top-k is better than that of FFRK method.

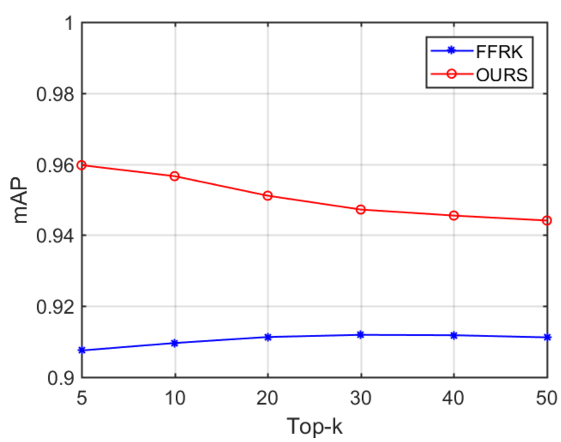


Figure 5: The mAP results of Caltech-256 dataset in different Top-k

4.2 Efficiency analysis

In this section, we evaluate the time efficiency of the algorithm on different datasets. On Caltech-256, we set the value of *Kc* is 1024, multi-view is a combination of BoW_SIFT feature and DenseNet feature. On Caltech-101, the *Kc* value is 1536, and the multi-view is a combination of Gist feature and DenseNet121 feature. Our method does not require an expensive visual codebook training process, saving a lot of computing and memory resources. Our method uses simple binary hash codes to construct inverted indexes and match features, so the construction time of the entire index file and the average retrieval time are relatively fast. It can be seen from Tabs. 4, 5, the index construction time increases with the number of dataset pictures and the length of the hash code. Under different datasets and different hash code lengths, the average retrieval time does not change much.

Table 4: Time performance of different bits on Caltech-256

bits	16	32	64	128
The indexing time (seconds)	26.41	43.78	59.95	123.38
Average search time (seconds)	0.037	0.041	0.039	0.049

Table 5: Time performance of different bits on Caltech-101

bits	16	32	64	128
The indexing time (seconds)	14.02	20.15	29.10	75.40
Average search time (seconds)	0.039	0.041	0.041	0.044

5 Discussion

This section will discuss the experimental results of this work and future plans. In this work, we test and measure the performance of the algorithm by using the average accuracy mAP of Top-k. In the online query stage, the increase of the number of the nearest cluster centers will increase the range of preliminary candidate features to a certain extent. However, how to ensure that most of the added candidate features are the nearest neighbors of the query image is an important problem. It is not enough to rely on the classification of K-means alone. In the next step, we will continue to study how to effectively increase the candidate features and improve the overall mAP of retrieval. The adaptive selection of the number of clustering centers is also our next research direction.

6 Conclusion

In this paper, we proposed a multi-index hash method based on multi-view feature coding for large-scale image retrieval. This algorithm uses the fusion of multiple features to obtain multi-view data with higher-level information for image matching to improve retrieval accuracy. This algorithm quantifies the association feature to generate binary hash code, and links to index tag generated by K-means clustering to construct inverted index files. In order to improve the accuracy of image matching, we designed an inverted double index structure for double feature matching, which improves search efficiency while maintaining high accuracy. Through experiments on public datasets, the effectiveness of this method relative to existing large-scale image retrieval methods is verified.

Acknowledgement: The author would like to thank the equipment support of Central South University of Forestry & Technology and the support of National Natural Science Fund of China.

Funding Statement: This work was supported in part by the National Natural Science Foundation of China under Grant 61772561, author J. Q, http://www.nsfc.gov.cn/; in part by the Key Research and Development Plan of Hunan Province under Grant 2018NK2012, author J. Q, http://kjt.hunan.gov.cn/; in part by the Key Research and Development Plan of Hunan Province under Grant 2019SK2022, author Y. T, http://kjt.hunan.gov.cn/; in part by the Science Research Projects of Hunan Provincial Education Department under Grant 18A174, author X. X, http://kxjsc.gov.hnedu.cn/; in part by the Science Research Projects of Hunan Provincial Education Department under Grant 19B584, author Y. T, http://kxjsc.gov.hnedu.cn/; in part by the Degree & Postgraduate Education Reform Project of Hunan Province under Grant 2019JGYB154, author J. Q, http://xwb.gov.hnedu.cn/; in part by the Postgraduate Excellent teaching team Project of Hunan Province under Grant [2019]370-133, author J. Q, http://xwb.gov.hnedu.cn/, in part by the Postgraduate Education and Teaching Reform Project of Central South University of Forestry & Technology under Grant 2019JG013, author X. X, http://jwc.csuft.edu.cn/ in part by the Natural Science Foundation of Hunan Province (No. 2020JJ4140), author Y. T, http://kjt.hunan.gov.cn/; and in part by the Natural Science Foundation of Hunan Province (No. 2020JJ4141), author X. X, http://kjt.hunan.gov.cn/.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

Chen, Y. T.; Tao, J. J.; Liu, L. W.; Xiong, J.; Xia, R. L. et al. (2020): Research of improving semantic image segmentation based on a feature fusion model. *Journal of Ambient Intelligence and Humanized Computing*.

Chen, Y. T.; Xu, W. H.; Zuo, J. W.; Yang, K. (2019): The fire recognition algorithm using dynamic feature fusion and IV-SVM classifier. *Cluster Computing*, vol. 22, no. S3, pp. 7665-7675.

Gong, Y.; Lazebnik, S.; Gordo A.; Perronnin, F. (2013): Iterative quantization: a procrustean approach to learning binary codes for large-scale image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 12, pp. 2916-2929.

Li, H.; Qin, J. H.; Xiang, X. Y.; Pan, L. L.; Ma, W. T. et al. (2018): An efficient image matching algorithm based on adaptive threshold and RANSAC. *IEEE Access*, vol. 6, no. 1, pp. 66963-66971.

Liu, L.; Yu, M. Y.; Shao, L. (2015): Multiview alignment hashing for efficient image search. *IEEE Transactions on Image Process*, vol. 24, no. 3, pp. 956-966.

Liu, Q.; Xiang, X. Y.; Qin, J. H.; Tan, Y.; Tan, J. S. et al. (2020): Coverless steganography based on image retrieval of DenseNet features and DWT sequence mapping. *Knowledge-Based Systems*, vol. 192, no. 2020, pp. 105375-105389.

Liu, X. W.; Zhu, X. Z.; Li, M. M.; Wang, L.; Tang, C. et al. (2019): Late fusion incomplete multi-view clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. vol. 41, no. 10, pp. 2410-2423.

Liu, X.; Wang, L.; Zhang, J.; Yin, J.; Liu, H. (2014): Global and local structure preservation for feature selection. *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 6, pp. 1083-1095.

Lopez-Paz, D.; Sra, S.; Smola, A.; Ghahramani, Z.; Schölkopf, B. (2014): Randomized nonlinear component analysis. *Proceedings of the 31st International Conference on Machine Learning*, Beijing, China, vol. 4, pp. 1359-1367.

Luo, Y. J.; Qin, J. H.; Xiang, X. Y.; Tan, Y.; Liu, Q. et al. (2020): Coverless real-time image information hiding based on image block matching and Dense Convolutional Network. *Journal of Real-Time Image Processing*, vol. 17, no. 1, pp. 125-135.

Qin, J. H.; Li, H.; Xiang, X. Y.; Tan, Y.; Pan, W. Y. et al. (2019): An encrypted image retrieval method based on harris corner optimization and LSH in cloud computing. *IEEE Access*, vol. 7, no. 1, pp. 24626-24633.

Qin, J. H.; Pan, W. Y.; Xiang, X. Y.; Tan, Y.; Hou, G. M. (2020): A biological image classification method based on improved CNN. *Ecological Informatics*, vol. 58, no. 2020, pp. 1-8.

Shen, X. B.; Shen, F. M.; Liu, L.; Yuan, Y. H.; Liu, W. W. et al. (2018): Multi-view discrete hashing for scalable multimedia search. *ACM Transactions on Intelligent Systems and Technology*, vol. 9, no. 5, pp. 1-21.

Tan, J. S.; Duan, R.; Qin, J. H.; Xiang, X. Y.; Tan, Y. (2020): Feature fusion multi-view hashing based on random kernel canonical correlation analysis. *Computers, Materials & Continua*, vol. 63, no. 2, pp. 675-689.

Tan, Y.; Qin, J. H.; Xiang, X. Y.; Ma, W. T.; Pan, W. Y. et al. (2019): A robust watermarking scheme in YCBCR color space based on channel coding. *IEEE Access*, vol. 7, no.1, pp. 25026-25036.

Wang, B. W.; Kong, W. W.; Guan, H.; Xiong, N. N. (2019): Air quality forecasting based on gated recurrent long short-term memory model in internet of things. *IEEE Access*, vol. 7, no. 1, pp. 69524-69534.

Wang, J.; Qin, J. H.; Xiang, X. Y.; Tan, Y.; Pan, N. (2019): CAPTCHA recognition based on deep convolutional neural network. *Mathematical Biosciences and Engineering*, vol. 16, no. 5, pp. 5851-5861.

Xiang, L. Y.; Shen, X. B.; Qin, J. H.; Hao, W. (2019): Discrete multi-graph hashing for large-scale visual search. *Neural Processing Letters*, vol. 49, no. 3, pp. 1055-1069.

Xue, W. Z.; Li, H.; Peng, Y. G.; Cui, J. T.; Shi, Y. (2018): Secure k nearest neighbors query for high-dimensional vectors in outsourced environments. *IEEE Transactions on Big Data*, vol. 4, no. 4, pp. 586-599.

Zhang, D.; Wang, F.; Si, L. (2011): Composite hashing with multiple information sources. *Proceedings of ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 225-234.

Zhou, W.; Li, H.; Hong, R.; Lu, Y.; Tian, Q. (2015): BSIFT: Toward data-independent codebook for large scale image search. *IEEE Transactions on Image Processing*, vol. 24, no. 3, pp. 967-979.

Zhou, W.; Yang, M.; Wang, X.; Li, H.; Lin, Y. et al. (2016): Scalable feature matching by dual cascaded scalar quantization for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 1, pp. 159-171.

Zhou, Z. L.; Wu, J. Q. M.; Sun, X. M. (2019): Multiple distance-based coding: toward scalable feature matching for large-scale web image search. *IEEE Transactions on Big Data*, pp. 1.