# Feature Fusion Multi-View Hashing Based on Random Kernel Canonical Correlation Analysis

**Junshan Tan[1], Rong Duan[1], Jiaohua Qin[1, *], Xuyu Xiang[1] and Yun Tan[1]**

**Abstract:** Hashing technology has the advantages of reducing data storage and improving the efficiency of the learning system, making it more and more widely used in image retrieval. Multi-view data describes image information more comprehensively than traditional methods using a single-view. How to use hashing to combine multi-view data for image retrieval is still a challenge. In this paper, a multi-view fusion hashing method based on RKCCA (Random Kernel Canonical Correlation Analysis) is proposed. In order to describe image content more accurately, we use deep learning dense convolutional network feature DenseNet to construct multi-view by combining GIST feature or BoW_SIFT (Bag-of-Words model+SIFT feature) feature. This algorithm uses RKCCA method to fuse multi-view features to construct association features and apply them to image retrieval. The algorithm generates binary hash code with minimal distortion error by designing quantization regularization terms. A large number of experiments on benchmark datasets show that this method is superior to other multi-view hashing methods.

**Keywords:** Hashing, multi-view data, random kernel canonical correlation analysis, feature fusion, deep learning.

## 1 Introduction

With the rise of various social networks and the growth of image and video data in the network, a powerful image retrieval database is gradually formed. In view of these massive pictures, how to effectively retrieve the pictures which users need from the huge image database has become a research challenge in the field of information retrieval. We look forward to using machine learning methods [Ma, Qin, Xiang et al. (2019); Tan, Qin, Xiang et al. (2019)] in this field, and have recently made some efforts in image retrieval [Li, Qin, Xiang et al. (2018); Qin, Li, Xiang et al. (2019)]. Hashing technology has been widely used in large-scale image retrieval due to its significant advantages in computing and storage. Hashing maps raw data (such as images and text) to low-dimensional Hamming space, and uses the similarity-preserved binary code for similarity retrieval. Multi-view data carries more comprehensive and rich information than single-view data,

[1] College of Computer Science and Information Technology, Central South University of Forestry & Technology, Changsha, 410114, China.

[*] Corresponding Authors: Jiaohua Qin. Email: qinjiaohua@163.com;

Rong Duan. Email: RongDuan126@163.com.

Received: 22 June 2019; Accepted: 24 July 2019.

so it is more efficient to image retrieval by using multi-view data combined with the hashing method.

Hashing can be divided into two categories: data-independent and data-related. Locality sensitive hashing (LSH) [Gionis, Indyk and Motwani (1999)] is one of the well-known hashing methods and it is data-independent. LSH generates hash functions by random projection and embeds similar data into similar binary code. However, LSH needs long hash code to achieve good performance. So some data-related hashing methods have been proposed with shorter hash code and better comparability. For example, spectral hashing (SH) [Weiss, Torralba and Fergus (2009)] introduces unsupervised graph hashing, anchor graph hashing (AGH) [Liu, Wang, Kumar et al. (2011)] uses anchor graphs to solve SH, iterative quantization (ITQ) [Gong, Lazebnik, Gordo et al. (2013)] uses orthogonal rotation based on PCA. The rotation matrix optimizes the initial projection matrix, and discrete graph hashing (DGH) [Liu, Mu, Kumar et al. (2014)] uses discrete optimization to solve SH and so on. However, these hashing methods can only learn binary code from single-view data, and cannot be directly used for data described by multi-view.

In computer vision applications, the data objects involved usually have multiple features [Shen, Liu, Tsang et al. (2018); Shen, Shen, Sun et al. (2018)]. For example, each image can be described by different features in image retrieval, such as SIFT feature, GIST feature, HOG feature and so on. This data is called multi-view data, each view corresponds to a feature, reflecting the different characteristics of the data, we need to adapt to the selection of view features [Liu, Wang, Zhang et al. (2014)]. Compared with single-view data, multi-view data is more abundant and comprehensive, which makes multi-view learning receive more and more attention. Some literatures [Xiang, Shen, Qin et al. (2019); Shen, Shen, Sun et al. (2018)] show that multi-view features usually have better performance than single-view feature in image retrieval. Therefore, we expect to use multi-view features to learn more compact hash code for image retrieval.

Multi-view hashing generates compact binary code from multi-view data. Some representative hashing algorithms such as composite hashing with multiple information sources (CHMIS) [Zhang, Wang, Li et al. (2011)], multi-view spectral hashing (SU-MVSH) [Kim, Kang and Choi (2012)], multi-view aligned hashing (MAH) [Liu, Yu, Shao et al. (2015)] and so on. However, these methods may have limitations. They give up binary constraints and continuous thresholding features to obtain hash code, but the algorithm may be affected by distortion errors. Studies Liu et al. [Liu, Mu, Kumar et al. (2014)] have shown that simple schemes can cause large quantization errors and lead to low quality hash code. Secondly, their training time complexity is high, and it is difficult to carry out large-scale computation in the case of large data sets. Therefore, how to effectively combine multi-view data with hashing method in large-scale image retrieval is still a challenging research topic.

Based on the above considerations, in this paper, we propose a multi-view hashing method based on Random Kernel Canonical Correlation Analysis (RKCCA). This algorithm aims to learn compact hash code by integrating multi-view. We summarize the main contributions of this work as follows:

(1) This algorithm extracts the DenseNet feature of the deep learning dense convolutional network. The advantages of the DenseNet feature can be used to

describe the image information more accurately and combine it with the GIST feature or the BoW_SIFT (Bag-of-Words model+SIFT feature) feature to construct a multi-view to get deeper image information.

(2) This algorithm uses RKCCA method to find the feature with the greatest correlation among multi-view, and fuses these features to get the association feature with higher layers picture information, and then uses the association feature as the input of the hashing method to describe the image information.

(3) The algorithm designs the regularization term in the quantization stage, minimizes the quantization error in the iterative optimization, ensures that the algorithm reaches the convergence state, and finds the best binary code.

We did a lot of experimentation with large benchmark data sets. The experimental results show that the proposed method is superior to the most advanced multi-view hashing method.

The rest of the paper is organized as follows: Section 2 describes the related work, Section 3 presents the proposed method, Section 4 presents the experimental evaluation, and the final conclusion is given in Section 5.

## 2 Related work

### *2.1 DenseNet feature of dense convolutional network*

With the introduction of deep learning, some achievements Wang et al. [Wang, Qin, Xiang et al. (2019)] have been made in convolutional neural networks. Image retrieval technology based on deep learning mainly applies in the feature extraction module of image retrieval, and uses a convolutional neural network to extract image features. Since the development of the convolutional neural network, many excellent results have been achieved. Convolution layer and pooling layer in convolution neural network can extract translation invariance of input features and identify similar features in different spatial locations, which makes convolution neural network widely used in computer vision. In convolution neural networks, convolution kernels are used to extract features. These initialized convolution kernels are constantly updated in the process of back propagation and iteration, making them infinitely close to the real solution. The essence of this method is to initialize a set of eigenvectors that conform to a certain distribution, and then update the set of eigenvectors infinitely in reverse propagation for feature extraction.

DenseNet is a convolutional neural network with dense connections. It was proposed to win at the Computer Vision and Pattern Recognition Conference in 2017. DenseNet has better performance than other neural networks, such as residual neural network (ResNet), convolutional neural network (CNN) and so on. In DenseNet network, the input of each layer of the network is the union of the output of all the previous layers, and the feature maps learned by this layer of the network will also be used as the input of all the later layers. There is a direct link between any two layers of the network to realize the reuse of features. Compared with traditional neural networks, deep learning convolutional neural networks have more training layers and higher complexity, which can extract higher level image information. Therefore, we use a convolution neural network to extract image features. In this paper, we use DenseNet121 network model to train DenseNet feature extraction from benchmark data set as one of the view data to describe image information and use

DenseNet feature to construct multi-view combining BoW_SIFT feature or GIST feature.

## 3 Feature fusion multi-view hashing based on random kernel canonical correlation analysis

Firstly, we give several view features of dataset objects. $\{X^{(m)}=[X_1^{(m)},\dots,X_N^{(m)}]^T \in R^{N\times d_m}\}_m^M=1$, where $d_m$ is the dimension of the *mth* view and *M* is the number of views and *N* is the number of dataset objects. Hash code matrix is expressed as $B=[b_1,\dots,b_N]^T \in \{-1,1\}^{N\times r}$, where $b_i \in \{-1,1\}^{r\times 1}$ is the hash code corresponding to each database object. *r* is the length of the hash code. Tab. 1. describes the important symbols used in this article.

**Table 1:** Important notations used in this paper

| Notation | Description |
|---|---|
| $X^{(m)}$ | Data matrix of the *mth* view |
| $L^{(m)}$ | Graph Laplacian matrix of the *mth* view |
| $\theta_m$ | Weight of the *mth* view |
| $d_m$ | The dimensionality of the *mth* view |
| B | Hash code matrix |
| Y | Continuous low-dimensional embedding |
| R | Orthogonal transformation matrix in the quantization stage |
| Z | The similarity matrix |
| M | The number of views |
| N | The number of datasets object |
| r | The length of the hash code |

### *3.1 Reservation of similarity*

The key step of image retrieval is to compare and measure the similarity between images. In this paper, image data is mapped by Laplacian mapping. Laplacian feature mapping will map the data on the manifold to low-dimensional space, while retaining the similarity between the original data as much as possible. First, we define the similarity of objective preservation function.

$$\min_{b_i} \sum_{i,j=1}^N S_{ij} \left\|b_i-b_j\right\|_2^2 \tag{1}$$

A total of *N* samples, where $S_{ij}$ tables the similarity of the original space samples. Assuming that the new mapping space has *k* dimensions and $b_i$ has *k* dimensions, since the minimization of all dimensions in the new space equals the minimization of one dimension, the similarity preservation function equation can deduce the following equations:

$$\min_{B} Tr(B^T L^{(m)} B)$$
$$s.t.\ B \in \{-1,+1\}^{N\times r}, B^T B = NI_r \tag{2}$$

The above equation can represent all samples of a certain dimension in a new space. Neighborhood graph is constructed after similarity preservation objective function is determined. Traditional neighborhood graph construction methods such as KNN method, because of its high time complexity $O(N^2)$ and poor computational efficiency, we choose litekmeans method to construct anchor graph instead of the neighbor graph (the nearest neighbor graph between the anchor point and each data sample point is used to approximate the nearest neighbor graph between the data sample point and sample point). Taking the *mth* view as an example, the features of this view are clustered to obtain $k(k \ll N)$ clustering centers, each of which is called an anchor point $\{\mu_j^{(m)}\}_{j=1}^{k}$. The similarity matrix $Z^{(m)} \in R^{N \times k}$ between training data and anchor points is defined as follows:

$$Z_{ij}^{(m)} = \begin{cases} \frac{exp\ (D^2(x_i^{(m)},\mu_j^{(m)})/\sigma)}{\sum_{j\in[i]} exp\ (D^2(x_i^{(m)},\mu_j^{(m)})/\sigma)}, & \forall\, j \in [i] \\ 0, & otherwise \end{cases} \tag{3}$$

where $[i]$ represents the exponent of $s(s \ll k)$ closest to the $x_i^{(m)}$ anchor, $D^2\left(x_i^{(m)}, \mu_j^{(m)}\right)$ is Euclidean the distance between $x_i^{(m)}$ and $\mu_j^{(m)}$, $\sigma$ is usually predefined with parameters. Each row in $Z^{(i)}$ contains only non-zero terms that sum to 1. We summarize all *M* views features with the following objective functions:

$$\min_{B\ ,\ \Theta} \sum_{m=1}^{M} \theta_m Tr(B^T L^{(m)} B)$$

$$s.t.\ B \in \{-1,1\}^{N\times r}, B^T B = N I_r\ ,$$

$$and\ \sum_{m=1}^{M} \theta_m = 1, \theta_m \geq 0, m = 1, \dots, M \tag{4}$$

where $\Theta = [\theta_1, \dots, \theta_M]^T$, $\theta_m$is a variable weighing the proportion of view features. By fusing multi-view features by weighting, considering the similarity structure and quantization loss of binary code, we have the following objective function.

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

$$s.t. Y^T Y = I_r \tag{5}$$

where $\lambda$ is the regularization parameter, $L$ is the Laplace matrix, and $YR$ is the continuously rotating low-dimensional embed. The rotation of the matrix can reduce the quantization error. We can get the binary code of the eigenvectors under the optimal rotation matrix by looking for the rotation matrix with the smallest quantization error.

### *3.2 Canonical correlation analysis*

Canonical correlation analysis (CCA) [Hotelling (1936)] is a method to correlate the linear relationship between two multidimensional random variables. Assumed characteristic data $X \in R^{d_{mx}}$, $Y \in R^{d_{my}}$, Where *X* is a sample matrix of $N \times d_{mx}$, *Y* is a sample matrix of $N \times d_{my}$, while $d_{mx}$ and $d_{my}$ are the characteristic dimensions of *X* and *Y* respectively. We projected *X* and *Y* respectively. The corresponding projection vectors were *a* and *b*. The corresponding projection data became $U=a^T X$ and $V=b^T Y$.

The definitions of $X$ and $Y$ correlation coefficients are as follows:

$$\rho(X,Y)=\frac{cov(X,Y)}{\sqrt{DX}\sqrt{DY}} \tag{6}$$

According to the purpose of CCA canonical correlation analysis and Eq. (6), our optimization goal is to maximize $\rho(U,V)$ and obtain the corresponding projection vectors $a$ and $b$, namely

$$\underset{a,b}{argmax}\ \frac{cov(U,V)}{\sqrt{D(U)}\sqrt{D(V)}} \tag{7}$$

Before projection, we standardized the original data, make $S_{XX}=Var(X)$, $S_{YY}=Var(Y)$, $S_{XY}=cov(X,Y)$, $S_{YX}=cov(Y,X)$,and $Var(X)$ and $Var(Y)$ represent the covariance matrix of $X$ and $Y$ respectively, and $cov(X,Y)$ represents the covariance matrix between them. Through these formulas and the derivation of the projection data $U$ and $V$ in the projection direction of $X$ and $Y$, we use $Var(U)=a^TS_{XX}a,\ Var(V)=b^TS_{YY}b,\ cov(U,V)=a^TS_{XY}b$ to transform the optimization objective into:

$$\underset{a,b}{argmax}\ \frac{a^TS_{XY}b}{\sqrt{a^TS_{XX}a}\sqrt{b^TS_{YY}b}} \tag{8}$$

We use two coefficients to scale the size of $a$ and $b$, and when the numerator and denominator change by the same multiple, the optimization target result remains unchanged. We adopted an optimization method similar to SVM, fixed the denominator, added restrictions, and optimized the numerator. The specific optimization objectives were converted into:

$$\underset{a,b}{argmax}\ a^TS_{XY}b$$

$$s.t.\ a^TS_{XX}a=1, b^TS_{YY}b=1 \tag{9}$$

By using the Lagrange function, the optimization objective is converted into maximize the following equation:

$$L=a^TS_{XY}b-\frac{\lambda}{2}(a^TS_{XX}a-1)-\frac{\theta}{2}(b^TS_{YY}b-1) \tag{10}$$

By solving Eq. (10), $a$ and $b$ can be found by eigenvectors corresponding to the maximum eigenvalue of the generalized eigenvalue problem.

$$S_{XX}^{-1}S_{XY}S_{YY}^{-1}S_{YX}a=\lambda^2a \tag{11}$$

By the same method, we can find the eigenvector corresponding to the maximum eigenvalue which is the linear coefficient $b$ of $Y$.

### *3.3 Feature fusion based on random kernel canonical correlation analysis*

Feature fusion based on canonical correlation analysis is aimed at different views of data. By calculating the maximum correlation of two views, they are integrated into a subspace. But the traditional canonical correlation analysis can only explore the linear relationship between two groups of random variables. In practice, the relationship between data variables is often non-linear, and it is very difficult to calculate on large-scale data. So we use the random kernel canonical correlation analysis (RKCCA) [Lopez-Paz, Sra, Smola et al. (2014)] method to deal with the problem of feature fusion of non-linear data. The combination of the randomized method and the linear algorithm is helpful to reveal the features of non-linear patterns in data. For regression or classification problems, random features show little or no performance loss, while greatly saving computational requirements.

According to the non-linear random features, we can randomly extract the parameter $W_i \in R^d$ from the data-independent distribution $p(w)$, and construct an *M*-dimensional random features map $z(X)$ for the input data $X \in R^{n\times d}$ which obeys the following structure.

$$w_1, \dots, w_m \sim p(w),$$
$$z_i := [cos(w_i^T x_1 + b_i), \dots, cos(w_i^T x_n + b_i)] \in R^n ,$$
$$z(X) := [z_1 \cdots z_m] \in R^{n\times m} \tag{12}$$

Bochner's theorem helps to join shift-invariant kernels and random nonlinear features. Let $k(x,y)$ be a real value and normalize, and become a shift-invariant kernel on $R^d \times R^d$. then,

$$k(x,y) = \int_{R^d} p(w) e^{-jw^T(x-y)} dw$$
$$\approx \sum_{i=1}^{m} \frac{1}{m} e^{-jw_i^T x} e^{jw_i^T y}$$
$$= \sum_{i=1}^{m} \frac{1}{m} cos(w_i^T x + b_i)\, cos(w_i^T y + b_i)$$
$$= \langle \frac{1}{\sqrt{m}} z(x), \frac{1}{\sqrt{m}} z(y) \rangle, \tag{13}$$

where $p(w)$ is set to *k*'s inverse Fourier transform, $b_i \sim u(0,2\pi)$. Let $K \in R^{n\times n}$ be the kernel matrix of data $X \in R^{n\times d}$, i.e., $K_{ij} = k(x_i, y_j)$. When $m$ random Fourier features are used to approximate the kernel *k*, we can approximate the kernel matrix $K \approx \widehat{K}$, where

$$\widehat{K} := \frac{1}{m} z(X) z(X)^T = \frac{1}{m} \sum_{i=1}^{m} z_i z_i^T = \sum_{i=1}^{m} \widehat{K}^{(i)} \tag{14}$$

We fuse the obtained kernel matrix with the linear canonical correlation method on the similarity matrix obtained by the feature. This fusion method can be understood as the linear canonical correlation analysis performed on a pair of random non-linear maps.$z_x : R^{n\times p} \rightarrow R^{n\times d_{mx}}, z_x : R^{n\times q} \rightarrow R^{n\times d_{my}}$ of the data $X \in R^{n\times p}$ and $Y \in R^{n\times q}$. Schematically,

$$RKCCA(X,Y) := CCA(z_x(X), z_y(Y)) \approx KCCA(X,Y) \tag{15}$$

Consistent with the above-mentioned canonical correlation analysis, we find projection vectors a and b according to the given view features, extract the canonical correlation features *U* and *V* between multi-view data, and assign weight fusion to these correlation features as the association features of multi-view after projection. And the correlation feature is a more recognizable feature vector than any input feature vector. In this way, the correlation feature vectors learnt from multiple views can better represent the features, and the recognition rate can be higher when applied to image retrieval.

### *3.4 Quantitative hashing method*

According to the objective matrix of Eq. (5), we use gradient descent optimization and Stiefel manifold optimization with curvilinear search to find the local optimal solution to ensure convergence of iteration. *Y* is a continuous low-dimensional embedding obtained by singular value decomposition after feature fusion. As an input to solve the hash code value, we express $G = \nabla F(Y)$ as a gradient relative to *Y*. The calculation equation is as follows.

$$G = 2\big((1+\lambda)I - Z^T \wedge^{-1} Z\big)Y - 2BR^T \tag{16}$$

where $Z = \left[Z^{(1)}, \dots, Z^{(m)}\right]$, $\varLambda = diag(\varLambda^{(1)}, \dots, \varLambda^{(m)})$.

We use *Y* to solve the hash code. In the iterative process, the optimal rotation *R* is obtained in the newly mapped hypercube space, so that the continuous rotation low-dimensional embedding is close to the hash code *B*, so that the quantization error is minimized. We solve instead the problem

$$\underset{B,R}{min} \|YR - B\|_F^2$$

$$s.t.\ B \in \{-1, +1\}^{N\times r}, R^T R = I_r \tag{17}$$

The random matrix is decomposed by singular value decomposition (SVD) and the corresponding orthogonal matrix is obtained as the initial value of *R*.

(1) Given fixed *R* to Solve *B*: The problem of solving *B* can be expressed as

$$B = sign(YR) \tag{18}$$

where *sign( • )* is the sign function.

(2) Given fixed *B* to Solve *R*: The singular value decomposition (SVD) of $B^TY$ is defined as $B^TY = P\sum Q^T = \sum_{k=1}^{r} \sigma_k p_k q_k^T$, where *r* is the rank of $B^TY$, $\sigma_1, \dots, \sigma_r$ are the positive singular, $P = [P_1, \dots, P_r]$ and $Q = [q_1, \dots, q_r]$ are left and right singular vectors respectively. The solution of R is $(PQ)^T$, and 50 iterations are performed alternately until the optimal rotation continuous low-dimensional embedding is found.

Algorithm 1: Feature Fusion Multi-view Hash Based on Random Kernel Canonical Correlation Analysis

```
Input: training set $\{X^{(m)} \in R^{N\times d_m}\}_{m=1}^{M}$; binary code length $r$; parameter $\lambda$.
Output: binary code $B$.
1: Obtain an anchor graph and a similarity matrix $Z$ for the view features respectively;
2: The $Z$ of views is fused by RKCCA;
3: Weighting the fused association features；
4: Initialize $Y$ by doing SVD on the weighted association features;
5: Random initialization R is$\{-\mathrm{r}, \mathrm{r}\}^{\mathrm{N\times r}}$ for SVD (Singular Value Decomposition);
6: Random initialization $B$ is$\{-1,1\}^{N\times r}$;
```

```
7: Update $R$ to $R = (PQ)^T$;
8: Repeat
9: Update$Y$using curvilinear search;
10: Update $B$using Eq. (18);
11: Calculate the singular value decomposition of $Y$, $B^TY = P\sum Q^T$;
12: Update $R$ with $R = (PQ)^T$;
13: Until convergence
```

## 4 Experiments

### *4.1 Datasets*

In this section, we test and evaluate the performance of the algorithm. We use widely used benchmark data sets, namely Caltech-256, Caltech-101. We select two view features for data set to construct multi-view data.

The Caltech-256 dataset consists of 29780 images, including 256 categories, each of which contains about 80 to 800 images. We randomly selected 1,000 pictures as the test set and the remaining 28,780 pictures as the training set.

The Caltech-101 dataset consists of 9144 images, including 102 categories (one of which is the background) with 40 to 800 images per category. We randomly selected 3019 pictures as the test set and 6125 pictures as the training set.

We extract 512-dimensional GIST feature, 500-dimensional BoW_SIFT (Bag-of-Words model+SIFT feature) feature and 1024-dimensional DenseNet feature for each image of the above two datasets, and randomly select two feature combinations each time as multi-view data for experimental testing.

### *4.2 Average accuracy experiments*

In terms of the validity verification of this algorithm, we compared it with the existing discrete multi-graph hashing method DMGH [Xiang, Shen, Qin et al. (2019)], and carried out experimental tests by comparing the data features of different views. The accuracy and recall of this algorithm are tested on large datasets Caltech-256 and Caltech-101.The range of parameter $\lambda$ in similarity reserved function as$[10^{-8}, 10^{-6}, \ldots, 10^{0}, 10^{2}]$, and test by cross-validation on the training set by selecting different weights on the similarity matrix.

In this paper, the mean average precision (mAP) and precision-recall curve (P-R curve) are used to test the retrieval performance of the algorithm on the benchmark dataset. the mAP is the mean of the average precision (AP) of all training samples.

We use BoW_SIFT feature, GIST feature and DenseNet feature extracted from datasets to cross-combine for the test in different feature fusion methods. Data features: (1) Single-view feature: BoW_SIFT feature or GIST feature or DenseNet feature; (2) BoW_SIFT feature and GIST feature to build multi-view; (3) BoW_SIFT feature or GIST feature combines DenseNet feature to build multi-view. The results of the mAP test for different hash code length on dataset Caltech-256 and Caltech-101 are shown in Tabs. 2-3, respectively.

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

| | 16 | | 32 | | 64 | | 128 | |
|---|---|---|---|---|---|---|---|---|
| Feature | DMGH | OURS | DMGH | OURS | DMGH | OURS | DMGH | OURS |
| BoW_SIFT +GIST | 0.0566 | 0.0518 | 0.0596 | 0.0577 | 0.0608 | **0.0611** | 0.0650 | **0.0671** |
| GIST+DenseNet | **0.2623** | **0.5404** | **0.4363** | **0.7022** | **0.5904** | **0.7940** | **0.7421** | **0.8480** |
| BoW_SIFT+DenseNet | **0.2391** | **0.5305** | **0.4328** | **0.7113** | **0.5758** | **0.7954** | **0.7316** | **0.8389** |

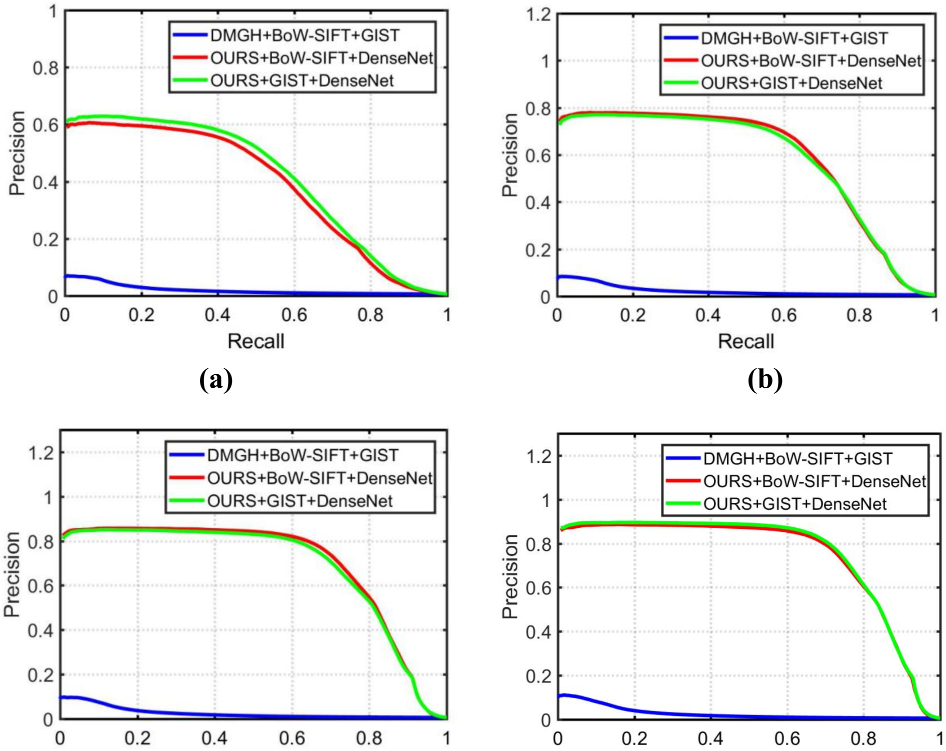
Bold data means better results than the comparison method, and the best results for different bits are red bold.

Compared with the DMGH algorithm proposed in the literature [Xiang, Shen, Qin et al. (2019)], which combines BoW_SIFT feature with GIST feature in multi-view data for image retrieval, we select multi-view data of BoW_SIFT feature and DenseNet feature combination, multi-view data of GIST features and DenseNet feature combination to perform image retrieval with our method, the mAP results shown in the examples in Tabs. 2, 3. As can be seen from the above results: (1) Using DenseNet feature of deep learning dense convolution neural network combined with other features (such as BoW_SIFT feature and GIST feature) to construct multi-view and apply it to multi-view hashing method can improve the results of mAP. (2) On the whole, the mAP results of the DenseNet feature combined with this method are better than the DMGH method used for comparison. The precision–recall curves of different hash code lengths (16 bit, 32 bit, 64 bit, 128 bit) on the datasets Caltech-256 and Caltech-101 are shown in Figs. 1 and 2, respectively. From the results of Figs. 1 and 2. we can also see that the precision-recall curve of this algorithm is higher than that of DMGH. These results confirm that the proposed algorithm is superior to discrete multi-view hashing method based on large-scale image retrieval (DMGH) proposed by literature [Xiang, Shen, Qin et al. (2019)].

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

| | 16 | | 32 | | 64 | | 128 | |
|---|---|---|---|---|---|---|---|---|
| Feature | DMGH | OURS | DMGH | OURS | DMGH | OURS | DMGH | OURS |
| BoW_SIFT +GIST | 0.2131 | 0.1843 | 0.2065 | 0.2030 | 0.2200 | 0.2033 | 0.2228 | 0.1963 |
| GIST+DenseNet | **0.5296** | **0.7081** | **0.7845** | **0.8413** | **0.8535** | **0.8940** | **0.8554** | **0.8362** |
| BoW_SIFT+DenseNet | **0.6380** | **0.7342** | **0.8434** | **0.8768** | **0.9043** | **0.9097** | **0.8632** | **0.8406** |

Bold data means better results than the comparison method, and the best results for different bits are red bold.



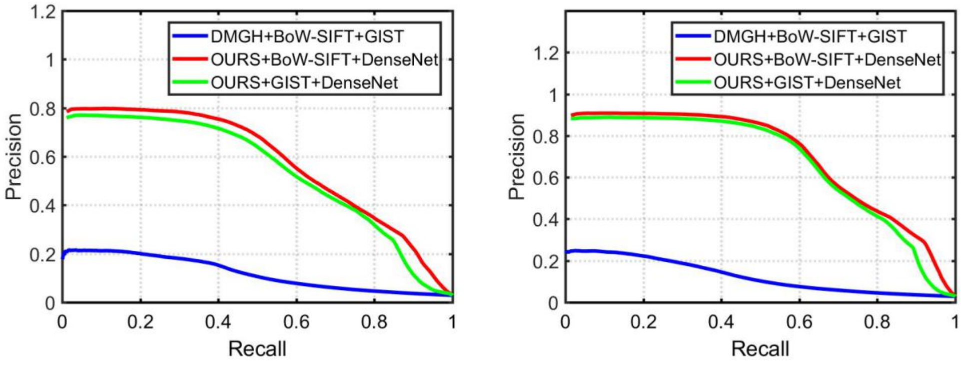
(a) (b)

(c) (d)

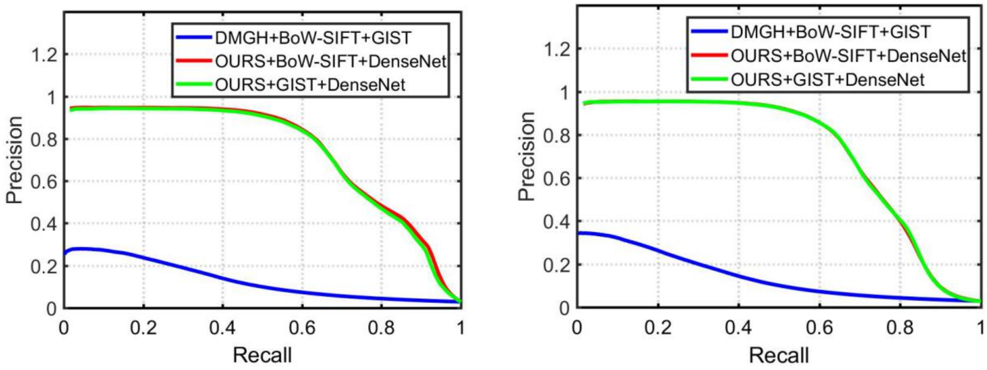
**Figure 1:** Precision recall curves on Caltech-256 dataset with respect to a different number of bits. **(a)** 16 bit, **(b)** 32bit, (**c**) 64 bit and (**d**) 128 bit

### *4.3 Convergence analysis*

In this section, we analyze the convergence performance of our method. We take Caltech-256 as an example, Fig. 3. shows the convergence curve of our method under 16 bit hash code length. From Fig. 3. we can see that our method can converge quickly.



(a) (b)



**(c)** **(d)**

**Figure 2:** Precision recall curves on Caltech-101 dataset with respect to different number of bits. **(a)** 16 bit, **(b)** 32bit, **(c)** 64 bit and **(d)** 128 bit

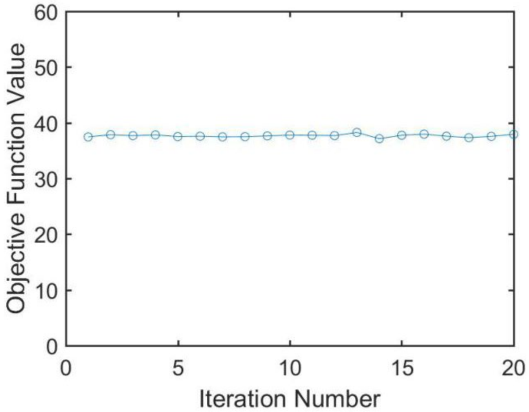


**Figure 3:** Convergence analysis of our method on the Caltech-256 dataset

### *4.4 Comparison with our methods using single-view*

In this section, we will analyze the performance of single-view data and multi-view data in image retrieval using our method. We experimented on the benchmark dataset Caltech-256. The view data were BoW_SIFT feature and DenseNet feature respectively. We experimented on the two features in a single-view and multi-view respectively. The results of the mAP experiment are shown in Fig. 4. From the results of Fig. 4, we can see that the retrieval performance of the multi-view data combined with our method is better than that of the single-view data method. Our method can explore the complementarity between multi-view data to improve the performance of image retrieval.

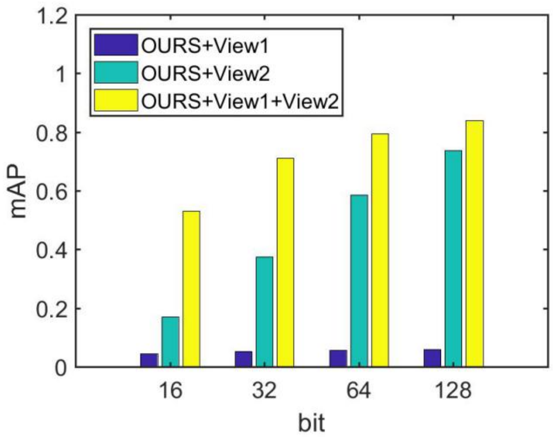


**Figure 4:** Performance comparison of this algorithm on Caltech-256 dataset in multi-view and single-view

## 5 Conclusion

This paper studies the efficient application of learning compact binary hash codes in image retrieval from multi-view data. We propose a multi-view hashing method based on feature fusion of RKCCA (Random Kernel Canonical Correlation Analysis), which effectively integrates multiple views according to feature fusion. This algorithm extracts features of deep learning dense convolution network and constructs multi-view features combining with other different features to describe image data better. This algorithm constructs anchor graphs for each view to obtain similarity matrix, and then finds the most relevant features among multiple views through RKCCA and fuses them to get the correlation features, which are used as input of hashing method. In this algorithm, a regularization term is designed to reduce the distortion error in the hash quantization stage. Through a large number of experiments on large benchmark datasets, it is proved that this method is superior to the existing multi-view hashing method.

**Acknowledgment:** This work is supported by the National Natural Science Foundation of China (No. 61772561), the Key Research & Development Plan of Hunan Province (No. 2018NK2012), the Science Research Projects of Hunan Provincial Education Department (Nos. 18A174,18C0262), and the Science & Technology Innovation Platform and Talent Plan of Hunan Province (2017TP1022).

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

## References

**Gionis, A.; Indyk, P.; Motwani, R.** (1999): Similarity search in high dimensions via hashing. *Proceedings of International Conference on Very Large Data Bas*es, pp. 518-529.

**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.

**Hotelling, H.** (1936): Relations between two sets of variates. *Biometrika*, vol. 28, no. 3-4, pp. 321-377.

**Kim, S.; Kang, Y.; Choi, S.** (2012): Sequential spectral learning to hash with multiple representations. *Proceedings of European Conference on Computer Vision*, pp. 538-551.

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

**Liu, W.; Wang, J.; Kumar, S.; Chang, S. F.** (2011): Hashing with graphs. *Proceedings of International Conference on Machine Learning*, pp. 1-8.

**Liu, W.; Mu, C.; Kumar, S.; Chang, S. F.** (2014): Discrete graph hashing. *Proceedings of Advances in Neural Information Processing Systems*, pp. 3419-3427.

**Liu, X. W.; Wang, L.; Zhang, J.; Yin, J. P.; 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. *Computer Science*, vol. 4, pp. 1359-1367.

**Ma, W. T.; Qin, J. H.; Xiang, X. Y.; Tan, Y.; Luo, Y. J. et al.** (2019): Adaptive median filtering algorithm based on divide and conquer and its application in CAPTCHA recognition. *Computers, Materials & Continua*, vol. 58, no. 3, pp. 665-677.

**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.

**Shen, X. B.; Liu, W. W.; Tsang, I. W.; Sun, Q. S.; Ong, Y. S.** (2018): Multilabel prediction via cross-view search. *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 9, pp. 4324-4338.

**Shen, X. B.; Shen, F. M.; Sun, Q. S.; Liu, L.; Yuan, Y. H. 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, 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, 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.

**Weiss, Y.; Torralba, A.; Fergus, R.** (2009): Spectral hashing. *Proceedings of Advances in Neural Information Processing Systems*, pp. 1753-1760.

**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.

**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.