Received May 12, 2020, accepted May 23, 2020, date of publication May 29, 2020, date of current version June 11, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2998524

Supervised Matrix Factorization Hashing With Quantitative Loss for Image-Text Search

HUAN ZHAO[1], SONG WANG[1], XIAOLIN SHE[1], AND CHENGHUI SU[2]
[1]College of Computer Science and Electronic Engineering, Hunan University, Hunan 410082, China
[2]Higher Research Institute, Southwest University of Political Science & Law, Chongqing 401120, China

Corresponding author: Huan Zhao (hzhao@hnu.edu.cn)

This work was supported by the National Key Research and Development Program of China under Grant 2018YFC0831800.

ABSTRACT Image-text hashing approaches have been widely applied in large-scale similarity search applications due to their efficiency in both search speed and storage efficiency. Most recent supervised hashing approaches learn a hash function by constructing a pairwise similarity matrix or directly learning the hash function and hash code (i.e.,1 or −1) procedure based on class labels. However, the former suffers from high training complexity and storage cost, and the latter ignores the semantic correlation of the original data, both of which prevent discriminative hash codes. To this end, we propose a novel discrete hashing algorithm called supervised matrix factorization hashing with quantitative loss (SMFH-QL). The proposed SMFH-QL first generates hash codes via the class label, avoiding the construction of a pairwise similarity; then, matrix factorization is used to design hash codes from original image-text data, thereby eliminating the impact of class labels and reducing the quantization error. Moreover, we introduce a quantitative loss function term to learn hash codes by incorporating class labels and the original data information, facilitating learning a similarity-preserving hash function in image-text search. Extensive experiments show that SMFH-QL outperforms several existing hashing methods on three representative datasets.

INDEX TERMS Image-text search, supervised hashing, hash function, hash codes, quantitative loss function.

I. INTRODUCTION

Image-text search has attracted much attention due to the explosive growth of data in search engines and social networks in recent years. Image-text search plays an important role in many scenarios in the fields of target monitoring and object tracking [1]–[3], video surveillance [4], [5], audio-text recognition [6], face and saliency detection [7], [8], human computer interaction [9], [10] and multimodal modelling [11], [12], etc. Given an image query, the task of image-text search is to retrieve the most relevant texts in a text dataset, and vice versa. However, performing accurate and efficient image-text similarity searches on large-scale datasets is challenging when faced with limited storage resource and search ability. To address this challenge, many hashing-based methods have been proposed to transform image-text data in original feature space into compact binary codes(e.g., hash codes) in low-dimensional Hamming space. The crucial problem of hashing-based image-text search is how to preserve the intermodal and intramodal similarity correlation of the original image-text data for hash codes in Hamming space.

Generally, according to whether label information is utilized, existing hashing-based image-text methods can be classified as unsupervised methods and supervised methods. Unsupervised methods [13]–[22] utilize only the image-text pair, including co-occurrence information, to explore their semantic correlation in the shared image-text feature representation space. However, these methods cannot take advantage of semantic label information, i.e., cannot exploit class labels to preserve the intermodal and intramodal correlations of image-text data from the original feature space, which deteriorates the image-text search performance. By contrast, supervised methods [23]–[32] attempt to preserve correlation by exploiting the semantic labels to learn more consistent hash codes for the original image-text data, resulting in efficient retrieval performance. In this paper, we focus on supervised hashing-based methods for similarity image-text search.

Although many supervised hashing-based methods have achieved promising results, they still confront some common

The associate editor coordinating the review of this manuscript and approving it for publication was Fuhui Zhou.

drawbacks in learning common feature representations or hash codes and controlling the quantization error. (1) Most hashing-based image-text methods directly quantize the common potential real number representation into hash code in the process of binary quantization. Due to the inherent discrete characteristics of hash code, the constraints are relaxed to continuous values to obtain a solution, resulting in a large quantization error, which makes the generated hash code suboptimal. Consequently, this process substantially affects the retrieval accuracy of image-text search. (2) Most existing hashing approaches learn a hash function using the class label to construct a pairwise similarity matrix. However, these approaches entail high computational complexity and storage cost. (3) Another approach is to obtain hash codes directly by class label, which ignores the influence of the original image-text data. Moreover, this approach cannot take advantage of the semantic correlation of intramodal and intermodal (e.g., image and text) information to learn discriminative hash codes.

In light of the aforementioned problems, we propose a novel supervised hashing method called supervised matrix factorization hashing-based quantitative loss (SMFH-QL). The main contributions of the proposed method are summarized as follows:

- We propose a SMFH-QL algorithm for image-text searching by combining class labels with a matrix factorization strategy. SMFH-QL introduces class labels to generate hash codes directly and exploits matrix factorization to learn the hash function. Based on this fusion strategy, the proposed approach maintains the well-learned similarity correction of the original image-text data, generating more discriminative hash codes.
- We introduce a quantization loss function term to constrain the objective function of SMFH-QL to achieve a closer correlation between the common representation and hash codes. Thus, we can eliminate redundant feature by label information, which eventually reduces the quantitative loss and improves the quality of hash function when performing quantization.
- We evaluate the proposed SMFH-QL in an extensive experiment on three image-text retrieval datasets. The results demonstrate that SMFH-QL obtains the best retrieval performance over the state-of-the-art hashing methods.

The rest of this paper is organized as follows. Section II introduces related work. Section III presents the details of our method, including matrix factorization, the hash function, and the optimization algorithms. Section IV presents the experimental results, followed by conclusions and directions for future work in Section IV.

II. RELATED WORK

As mentioned previously, a variety of unsupervised and supervised image-text hashing methods have been proposed. The following review of related work focuses on these two aspects.

A. UNSUPERVISED IMAGE-TEXT HASHING METHODS

Unsupervised image-text hashing methods exploit only image-text original data to find intramodal and intermodal semantic correlations when learning the hash function. Kumar and Udupa [33] proposed cross-view hashing (CVH), which is an extension of the spectral hashing (SH) [34] method, that minimizes the weighted average distance between hash codes to preserve the similarity between data modalities. Zhu *et al*. [35] proposed linear cross-modal hashing (LCMH), which adopts anchor maps to preserve the similarity between the same modality, avoiding the construction of similarity maps for all training datasets and thus greatly improving the hash search efficiency. Zhou *et al*. [36] proposed latent semantic sparse hashing (LSSH), which first performs sparse coding and matrix factorization to capture the latent semantic features of images and texts and then projects features into a latent semantic space quantized to obtain an optimized hash code. Ding *et al*. [37] presented collective matrix factorization hashing (CMFH). This method was the first attempt to project data of different modalities into a common subspace by exploiting collective matrix factorization hashing to learn unified hash codes. Unsupervised methods cannot use class label information to guide the similarity correlation of original image-image, text-text and image-data data, leading to degradation of retrieval performance.

B. SUPERVISED IMAGE-TEXT HASHING METHODS

Supervised image-text hashing methods can improve upon the retrieval performance of unsupervised methods by considering label information. Therefore, supervised methods can be applied in more scenarios than can unsupervised approaches. In general, supervised image-text hashing methods can be further categorized into supervised nonmatrix factorization schemes and supervised matrix factorization schemes.

1) SUPERVISED NONMATRIX FACTORIZATION METHODS

Supervised nonmatrix factorization schemes employ given class labels to guide the hash code or hash function learning procedure. Bronstein *et al*. [38] proposed a sensitive image-text hashing method called CMSSH, which was the first attempt to apply the binary classification to hash code learning, and then applied a boosting algorithm to the learn hash codes of different modalities. Lin *et al*. [39] proposed semantics-preserving hashing (SePH), which first adopts the semantic correlation matrix of the training data as the supervised information, then transforms the semantic correlation matrix and the learned hash code into a probability distribution, and finally minimizes the Kullback-Leibler (KL) divergence of the two probability distributions. In the process of hash code learning, the method performs nonlinear projection via kernel logistic regression (KLR) and maps data features into hash codes, which are then utilized to

learn hash functions. Zhang and Li [40] proposed semantic correlation maximization (SCM), which measures the semantic similarity between data modalities by multiplying the class matrix of the training data. Xu *et al*. [41] proposed discriminative cross-modal hashing (DCH). DCH directly generates discriminative binary hash codes via a discrete coordinate descent strategy and then learns modality-specific hash functions based on the learned binary codes. However, these supervised nonmatrix factorization schemes ignore the co-correlation of the original image-text features and cannot mine and employ the semantic similarity correlation of image-text modalities.

2) SUPERVISED MATRIX FACTORIZATION METHODS

Supervised matrix factorization schemes apply matrix factorization to mine the intramodal and intermodal semantic correlations from the original image-text modality, which improves the hash function or hash code. For instance, Liu *et al*. [42] proposed supervised matrix factorization hashing (SMFH), which can utilize the label information to preserve the semantic similarity between different data modalities. Moreover, the adjacent structure is used to maintain the similarity between data of different modalities based on the CMFH method, which further enhances retrieval accuracy. Tang *et al*. [28] proposed a similar supervised matrix factorization method based on the class label matrix. However, the above approaches require a pairwise semantic similarity matrix, which leads to a large storage and computational cost. Wang *et al*. [43] proposed label-consistent matrix factorization hashing (LCMFH), which directly guides the process of learning the hash function via the label information of the training data. Therefore, LCMFH achieves a short training time and high search precision by avoiding the construction of a pairwise semantic similarity matrix. However, this method focuses on the similarity between hash codes of the same type of data, which ignores the feasibility of different data hash codes.

By reviewing the aforementioned methods, supervised approaches can achieve better retrieval performance than unsupervised approaches by considering class labels. Some supervised nonmatrix factorization hashing methods ignore the co-correlation of original image-text features and do not consider the semantic similarity correlation of image-text modalities. Meanwhile, supervised matrix factorization approaches are designed to preserve semantic relationships through class labels to construct pairwise similarity matrices. However, pairwise similarity matrices entail high training complexity and storage costs. In this paper, the SMFH-QL algorithm is proposed to address the two drawbacks and to optimize the hash codes and hash function directly via the class label, taking into account the intermodal and intramodal semantic correlations in the original image-text feature space and avoiding the construction of pairwise similarity matrices. The main differences between the proposed SMFH-QL and several state-of-the-art hashing methods are summarized in Table 1.

TABLE 1. Characteristics of existing image-text hashing methods (N is the number of training samples, relaxation denotes "relaxing+rounding" strategy for hash optimization, relaxing the discrete constraints and simply acquiring the hash codes by threshold function).

Method	Learning strategy	Optimization	Complexity
LSSH [36]	unsupervised	relaxation	$O\left(N^2\right)$
CMFH [37]	unsupervised	relaxation	$O\left(N\right)$
SMFH-Liu [42]	supervised	relaxation	$O\left(N\right)$
SMFH-Tang [28]	supervised	relaxation	$O\left(N\right)$
SCM-seq [40]	supervised	relaxation	$O\left(N^2\right)$
SePH [39]	supervised	relaxation	$O\left(N^2\right)$
DCH [41]	supervised	discrete	$O\left(N\right)$
LCMFH [43]	supervised	discrete	$O\left(N\right)$
SMFH-QL	supervised	discrete	$O\left(N\right)$

III. THE PROPOSED SMFH-QL

Fig. 1 shows an overview of our proposed SMFH-QL. It includes collective matrix factorization to produce common representation, a hash function to learn mapping matrices for guiding the query procedure, a classification loss function to directly generate a unified hash code via the class label matrix, and a quantitative loss function to maintain the well-learned similarity correlation between the hash code and common representation when performing quantization. These parts are combined with a training procedure for learning the common representation for each image or text modality to calculate image-text similarities.

A. MODELS AND PROBLEM FORMULATION

Table 2 lists the notations and corresponding definitions used in the study. SMFH-QL accepts original paired image-text data as the input and processes them through common representation learning and hash coding. The ultimate goal of the proposed approach is to retrieve the most relevant texts given an image query and vice versa.

TABLE 2. Notation used in the study of SMFH-QL.

Notations	Definition
$\mathbf{O}$	training instances
$\mathbf{X}, \mathbf{Y}$	image, text feature matrices, respectively
$\mathbf{T}$	semantic label matrix
$\mathbf{H}$	hash code matrix of training instances
$\mathbf{U_1}, \mathbf{U_2}$	basic matrices for matrix factorization
$\mathbf{W_1}, \mathbf{W_2}$	mapping matrices for the hash function
$\mathbf{V}$	shared latent representation
$\mathbf{Z}$	projection matrix for the classification loss function
$\phi\left(\bullet\right)$	kernel function
n	number of training instances
d_1, d_2	dimensions of image, text feature matrices
c	number of label classes
l	length of hash codes
ε	width of kernel

1) COLLECTIVE MATRIX FACTORIZATION

Collective matrix factorization is commonly used in low-rank representation learning. Ding *et al*. [37] demonstrated that

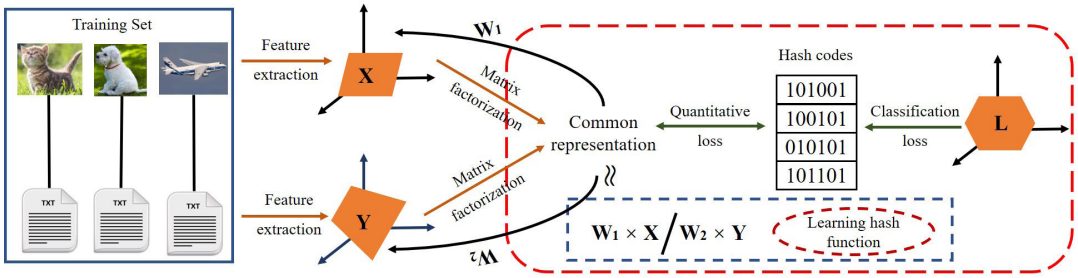


FIGURE 1. An overview of our proposed SMFH-QL, which consists of four parts that employ a collective matrix factorization strategy to produce a common representation from the features of image-text modality to learn the hash function and generate hash codes based on the semantic label. We also exploit a quantitative loss function term to strengthen the semantic correlation between the common representation and hash codes when performing quantization.

collective matrix factorization is effective in mining the relationships between multimodal data and shallow semantics in image-text retrieval. Given the image and text modalities X and Y, the proposed method needs to find the basic matrices $U_1 \in \mathbb{R}^{d_1 \times l}$,$U_2 \in \mathbb{R}^{d_2 \times l}$ and the shared latent representation $V \in \mathbb{R}^{l \times n}$ via matrix factorization.

Although the features extracted from different data of an image-text instance are heterogeneous, the two modalities still have common semantic information because they describe the contents of the identical instance. Hence, we assume that different modalities can share a common latent space generated by collective matrix factorization. Considering this, we define the following formulation:

$$J_1(U_1, U_2, V) = \|X - U_1 V\|_F^2 + \|Y - U_2 V\|_F^2 . \quad (1)$$

2) HASH FUNCTION

As mentioned above, we have obtained the unified representation V by Eq. (1), but it cannot be used for similarity query owing to its high dimensionality. Given out-of-sample instances, we need to learn two mapping functions of image and text modalities for hash code. Therefore, the corresponding transformation is defined as follows:

$$J_2(W_1, W_2, V) = \|V - W_1 X\|_F^2 + \|V - W_2 Y\|_F^2 , \quad (2)$$

where W_1 and W_2 are the mapping matrices.

3) CLASSIFICATION LOSS FUNCTION

To maintain the consistency between the learned hash code and the original semantic label, we utilize a shared label matrix to produce a unified hash code directly. That is, a classification error term is introduced to constrain the hash code, and a classifier is trained via minimizing the classification error in the process of learning the corresponding objective function, which improves the discrimination of learned hash codes; i.e.,

$$J_3(H, Z) = \left\| T - Z^T H \right\|_F^2 , \quad (3)$$

subject to $H \in \{-1, 1\}^{l \times n}$, where $Z \in \mathbb{R}^{l \times c}$ is a projection matrix.

4) QUANTITATIVE LOSS FUNCTION

In order to avoid a continuous relaxation strategy, we directly generate binary hash codes. Hence in the sequel, we construct a connection between H and V, where the matrix V can be accurately represented via the learned hash codes. In this paper, we minimize the difference between a square loss measure H and V to eliminate the large quantitative loss:

$$J_4(H, V) = \|H - V\|_F^2 , \quad (4)$$

subject to $H \in \{-1, 1\}^{l \times n}$.

5) THE KERNEL METHOD

In this work, to address the linear indivisibility of data in low-dimensional space, we adopt a nonlinear embedding method for all training samples, which ensures that the data are linearly separable in the high-dimensional kernel space [44], [45]. The corresponding formulation is listed as follows:

$$\phi(S^{(t)}) = \left\{ \phi\left(s_i^t\right) \right\}_{i=1}^n , \quad (5)$$

where $S^{(t)} = \{X, Y\}$ denotes the aggregation of image and text modalities.

$$\phi\left(x^{(t)}\right) = [A_1, \ldots A_m]^T , \quad (6)$$

where $A_1 = \exp\left(\frac{-\left\| x^{(t)} - b_1^{(t)} \right\|^2}{\varepsilon} \right)$, $A_m = \exp\left(\frac{-\left\| \mathbf{x}^{(t)} - b_m^{(t)} \right\|^2}{\varepsilon} \right)$, and $\left\{ b_j^{(t)} \right\}_{j=1}^m$ are m randomly selected anchor points.

6) OVERALL OBJECTIVE FUNCTION

Integrating Eq. (1), Eq. (2), Eq. (3), Eq. (4) with Eq. (5), we establish a unified hashing framework and express the overall objective function of the proposed SMFH-QL. However, the excessive number of parameters in the above five formulations increase the complexity of the proposed model and may lead to overfitting. Thus, we further consider introducing a regularization term to solve the overfitting problem, which keeps the model simple and constrains the characteristics of the model. Concretely, the regularization term is defined as

follows:

$$J_5(U_1, U_2, W_1, W_2, Z, V) = \|U_1\|_F^2 + \|U_2\|_F^2 + \|W_1\|_F^2 + \|W_2\|_F^2 + \|Z\|_F^2 + \|V\|_F^2 . \quad (7)$$

Based on the above illustration, the final objective function of SMFH-QL is as follows:

$$\arg\min\ J = \lambda J_1 + \beta J_2 + \mu J_3 + \alpha J_4 + \gamma J_5, \quad (8)$$

subject to $H \in \{-1, 1\}^{l\times n}$, where λ, β, μ, α, γ are the trade-off coefficients. A detailed definition of Eq. (8) is given by the following:

$$\begin{aligned} J = {} & \mu \left\| T - Z^T H \right\|_F^2 + \alpha \|H - V\|_F^2 \\ & + \lambda \|\phi(X) - U_1 V\|_F^2 + \lambda \|\phi(Y) - U_2 V\|_F^2 \\ & + \beta \|V - W_1 \phi(X)\|_F^2 + \beta \|V - W_2 \phi(Y)\|_F^2 \\ & + \gamma Re(U_1, U_2, W_1, W_2, V, Z), \end{aligned} \quad (9)$$

subject to $H \in \{-1, 1\}^{l\times n}$. The objective function for SMFH-QL is formulated to generate discriminative hash codes by preserving the label consistency and distribution of features. Therefore, our final aim is to optimize and minimize Eq. (9).

B. OPTIMIZATION ALGORITHM OF THE PROPOSED SMFH-QL

Eq. (9) is a nonconvex problem on matrix variables U_1, U_2, W_1, W_2, V, Z. Fortunately, the problem is convex with respect to any one of the six matrix variables if the other variables are fixed [43]. Therefore, an iterative optimization strategy is adopted to approach the optimal solution gradually. We adopt this strategy to optimize the objective function of SMFH-QL. Then the optimization problem in Eq. (9) can be solved by updating the following steps iteratively.

Step 1: Updating U_1. Fixing U_2, W_1, W_2, V, Z, we learn basic matrices U_1 as below:

$$\min_{U_1} \lambda \|\phi(X) - U_1 V\|_F^2 + \gamma \|U_1\|_F^2 . \quad (10)$$

Let $\frac{\partial J}{\partial U_1} = 0$; we then have

$$U_1 = \lambda \phi(X) V^T \left(\lambda V V^T + \gamma I\right)^{-1}. \quad (11)$$

Step 2: Updating U_2. Similarly, we can obtain

$$U_2 = \lambda \phi(Y) V^T \left(\lambda V V^T + \gamma I\right)^{-1}. \quad (12)$$

Step 3: Updating W_1. Fixing U_1, U_2, W_2, V, Z, we learn mapping matrices W_1 as follows:

$$\min_{W_1} \beta \|V - W_1 \phi(X)\|_F^2 + \gamma \|W_1\|_F^2 . \quad (13)$$

Let $\frac{\partial J}{\partial W_1} = 0$; we then have

$$W_1 = \beta V(\phi(X))^T \left(\beta \phi(X) (\phi(X))^T + \gamma I\right)^{-1}. \quad (14)$$

Step 4: Updating W_2. Similarly, we can obtain

$$W_2 = \beta V(\phi(Y))^T \left(\beta \phi(Y) (\phi(Y))^T + \gamma I\right)^{-1}. \quad (15)$$

Step 5: Updating Z. Fixing U_1, U_2, W_1, W_2, V, the optimization for projection matrix Z is:

$$\min_{Z} \mu \left\| T - Z^T H \right\|_F^2 + \gamma \|Z\|_F^2 . \quad (16)$$

Let $\frac{\partial J}{\partial Z} = 0$; we then have

$$Z = \left(\mu H H^T + \gamma I\right)^{-1} \mu H T^T. \quad (17)$$

Step 6: Updating V. Fixing U_1, U_2, W_1, W_2, Z and setting $\frac{\partial J}{\partial V} = 0$, we have

$$\begin{aligned} V = {} & \left[\lambda (U_1)^T U_1 + \lambda (U_2)^T U_2 + (\alpha + 2\beta) I\right]^{-1} \\ & \times \begin{bmatrix} \lambda (U_1)^T \phi(X) + \lambda (U_2)^T \phi(Y) \\ + \beta W_1 \phi(X) + \beta W_2 \phi(Y) + \alpha H \end{bmatrix}. \end{aligned} \quad (18)$$

Step 7: Updating H. The hash code H is optimized by fixing other variables as follows:

$$\min_{H} \mu \left\| T - Z^T H \right\|_F^2 + \alpha \|H - V\|_F^2 , \quad (19)$$

subject to $H \in \{-1, 1\}^{l\times n}$. The two terms in Eq. (19) are expanded into the following two formulas:

$$\begin{aligned} \left\| T - Z^T H \right\|_F^2 &= tr\left(T^T T\right) - 2tr\left(T^T Z^T H\right) \\ &\quad + tr\left(H^T Z Z^T H\right) \\ &= const1 - 2tr\left(T^T Z^T H\right), \end{aligned} \quad (20)$$

and

$$\begin{aligned} \|H - V\|_F^2 &= tr\left(H^T H\right) - 2tr\left(V^T H\right) + tr\left(V^T V\right) \\ &= const2 - 2tr\left(V^T H\right), \end{aligned} \quad (21)$$

where $tr(\bullet)$ is the trace of the matrix and

$$\begin{aligned} const1 &= tr\left(T^T T\right) + tr\left(H^T Z Z^T H\right), \\ const2 &= tr\left(H^T H\right) + tr\left(V^T V\right). \end{aligned}$$

Updating H by fixing the other variables, we regard $const1$ and $const2$ as constants. By combining Eq. (18) and Eq. (19), we see that Eq. (20) is equivalent to the following problems:

$$\min_{H} -\mu tr\left(T^T Z^T H\right) - \alpha tr\left(V^T H\right), \quad (22)$$

subject to $H \in \{-1, 1\}^{l\times n}$. Therefore, we can obtain the final closed-form solution for H, i.e.,

$$H = sign(\alpha V + \mu Z T), \quad (23)$$

where $sign(\cdot)$ is the element-wise sign function. According to Eq. (23), we can directly obtain a closed-form solution for H. The optimization strategy can discretely generate all bits of the hash code via the class label and common representation.

Here, SMFH-QL avoids the large quantitative loss suffered by relaxation and reduces the time consumption of a bit-by-bit optimization scheme. Meanwhile, the discrete optimization strategy accelerates the training process.

C. COMPUTATIONAL COMPLEXITY ANALYSIS

In this section, we theoretically analyze the computational complexity of SMFH-QL. Suppose that $d = \{d_1, d_2\}$ is the feature dimension of the modalities and c, l, n and w are the length of the class label, the length of the hash codes, the number of training instances, and the number of iterations, respectively. The computational complexity of each step of the optimization process for SMFH-QL is shown in Table 3. During the training of the SMFH-QL algorithm, because $l, d, c \ll n$, the overall computational complexity is $O(nw)$. In the query procedure, the computational complexity of SMFH-QL is $O(dl)$, which is also linear with respect to the query complexity of the proposed method. In summary, SMFH-QL is highly scalable for large-scale datasets due to the linear training complexity and query complexity.

TABLE 3. The detailed computational complexity of SMFH-QL.

Equation	Computational complexity
Eq. (11)	$O\left(\left(d_1 l + l^2\right) n + l^3 + l^2 + d_1 l^2\right)$
Eq. (12)	$O\left(\left(d_2 l + l^2\right) n + l^3 + l^2 + d_2 l^2\right)$
Eq. (14)	$O\left(\left(d_1 l + d_1^2\right) n + d_1^3 + d_1^2 + l d_1^2\right)$
Eq. (15)	$O\left(\left(d_2 l + d_2^2\right) n + d_2^3 + d_2^2 + l d_2^2\right)$
Eq. (17)	$O\left(l\left(l + c\right) n + l^2 + l^3 + l^2 c\right)$
Eq. (18)	$O\left(\left(d_1 l + d_2 l + l\right) n + l^2 + l^3 + d_1 l^2 + d_2 l^2\right)$
Eq. (23)	$O\left(\left(l + lc\right) n\right)$

D. OUT-OF-SAMPLE EXTENSION

For a new query instance x or y, its corresponding hash codes b can be generated by the trained mapping matrices W_1, W_2. We can obtain the hash functions $h(x)$ and $h(y)$ for out-of-sample extension: $b = h(x) = sign(W_1 x)$ and $b = h(y) = sign(W_2 y)$.

IV. EXPERIMENTS

To demonstrate the effectiveness of SMFH-QL, we conduct extensive experiments on three widely used datasets. First, we introduce the three representative datasets and the evaluation and comparison methods. Then, we perform comparative experiments using SMFH-QL and other methods. Finally, to further validate the efficacy of SMFH-QL, we evaluate the experiments via empirical analysis, including convergence analysis, training time results and parameter analysis.

A. DATASETS

We evaluate the SMFH-QL on the three representative image-text search datasets: Wiki, MIRFLICKR-25K and NUS-WIDE. The details of the three datasets are shown in Table 4.

TABLE 4. The details of the three evaluated datasets.

Dataset	Wiki	MIRFILICKR-25K	NUS-WIDE
Total	2866	20,015	186,577
Query Set	693	2000	1867
Training Set	2173	5000	5000
Image Feature	128-D SIFT	512-D GIST	500-D BOVW
Text Feature	10-D LDA	1386-D BOW	1000-D BOW

- **Wiki** [46]: This dataset consists of 2,866 image-text pairs from Wikipedia. Each instance contains ten topics, such as warfare, art, and sky, and each sample is an image-text pair. Each image is represented as a 128-D SIFT feature vector, and each text is represented by a 10-D LDA topics vector. Following the experimental protocol in SePH [39] and DLFH [30], we randomly select 2173 pairs as a training set and the rest as the query set.
- **MIRFILCKR-25K** [47]: The dataset contains approximately 25K instances, and each image is annotated by several user-assigned tags selected from 24 labels. Each image is represented as a 512-dimension GIST feature vector. Each text is represented by a 1386-D BOW vector. Following the experimental protocol, we randomly sample 5000 instances as the training set and select 2000 instances as the query set.
- **NUS-WIDE** [48]: This dataset contains approximately 270K images with annotated tags from 81 semantic concepts. Following DLFH, we choose the 10 most frequent concepts consisting of 186,577 images as the experimental data. Each image is a 500-D bag-of-visual words (BOVW) vector, and each text is represented as a 1000-D BOW vector.

B. EVALUATION METRIC

The mean Average Precision (mAP) is a common evaluation metric. The mAP is the mean of the average precision (AP), and the AP of the top R instances is defined as:

$$AP(q) = \frac{1}{L} \sum_{r=1}^{R} P_q(r) \xi(r), \tag{24}$$

where q is a query instance, R is the number of instances and N is the number of queries. L is the number of relevant instances in the retrieved set, and $P(r)$ represents the precision of the top r retrieved instances. $\xi(r)$ is an indicator function, and $\xi(r) = 1$ if the rth instance is relevant to the query and $\xi(r) = 0$ otherwise. Therefore, mAP can be computed by:

$$mAP = \frac{1}{N} \sum_{i=1}^{N} AP(q_i), \tag{25}$$

where R is the size of the query set in the following experiments.

Moreover, we adopt two other criteria, i.e., the Precision-Recall curve and topN-Precision curve [49], which are frequently used in image-text searches.

C. BASELINE METHODS AND EXPERIMENTAL SETUP

Our method is compared against several state-of-the-art hashing methods: LSSH, CMFH, SMFH, SCM-seq, DCH and LCMFH. The first two are unsupervised methods and the last four are supervised methods.

- **LSSH** [36] extracts an image representation via sparse coding and a text representation via matrix factorization and conducts unified optimization of the objective function by means of hash code learning.
- **CMFH** [37] learns a unified hash code via matrix factorization with a latent factor model, which can decompose the characteristics of different modes of samples into the same space.
- **SMFH** [42] constructs a similarity matrix with class label information to strengthen the constraint of data similarity between modalities based on the CMFH method.
- **SCM-seq** [40] utilizes the label information of samples to maximize the semantic correlation between modalities and proposes two learning models as optimization algorithms. In this paper, we adopt the superior approach SCM-seq, which implements sequential learning.
- **DCH** [41] directly generates discriminative hash codes via discrete coordinate descent and then learns modality-specific hash functions based on the learned binary codes.
- **LCMFH** [43] directly guides the procedure of hash function learning based on the semantic labels of the training data. Therefore, LCMFH avoids the construction of a pairwise similarity matrix, which reduces the number of calculations.
- **SMFH-NQL** is a version of SMFH-QL that lacks a quantitative loss function term, i.e., $\alpha = 0$ in Eq. (9). Here, SMFH-NQL is used to demonstrate the influence of quantitative loss and the retrieval performance of the proposed SMFH-QL. For fairness, we adopt the same parameters used for SMFH-QL.

To validate the superiority of SMFH-QL, several state-of-the-art hashing methods are used for comparisons, including unsupervised hashing (LSSH, CMFH), supervised nonmatrix factorization hashing (SCM-seq, DCH) and supervised matrix factorization (SMFH, LCMFH). The source codes of the baseline methods were kindly provided by the authors. All parameters in their objective functions are set according to their original papers.

All baseline methods and our SMFH-QL are implemented in MATLAB (64 bit). We perform two image-text search tasks: I→T and T→I. I→T task represents image retrieval from relevant texts, and T→I utilizes text querying of relevant images. The experiments are conducted on a personal computer with an Intel (R) Core (TM) CPU i7-8550U @1.80 GHz and 8 GB RAM and 64-bit Windows 10 operating system.

D. RESULTS AND DISCUSSION

1) ASSSESSMENT OF SMFH-QL'S QUALITY ON WIKI

The first experiment compares the baseline approaches and SMFH-QL on the Wiki dataset. The parameters $\{\lambda, \beta, \alpha, \mu, \gamma\}$ for Eq. (9) are $\{0.5, 10, 10, 10000, 0.1\}$. The mAP values of all methods on Wiki are shown in Table 5. The Precision-Recall curves and the topN-precision curves for all compared methods are plotted in Fig. 2 and 3, respectively.

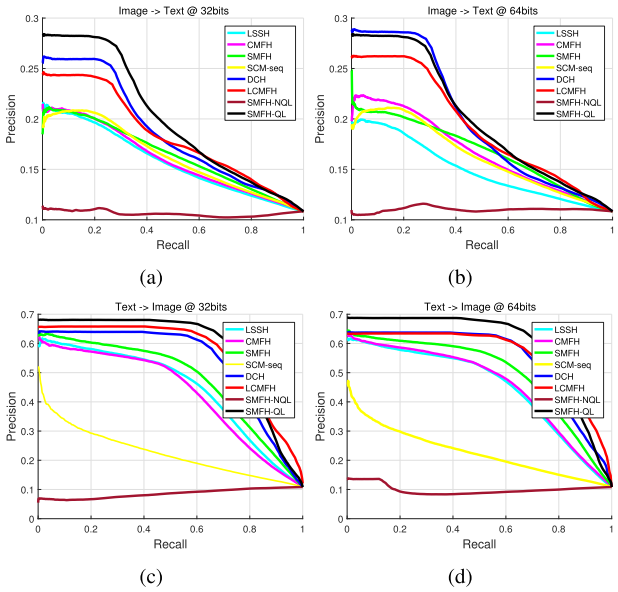


FIGURE 2. Precision-Recall curves on Wiki dataset when the hash code length is set to 32 bits or 64 bits.

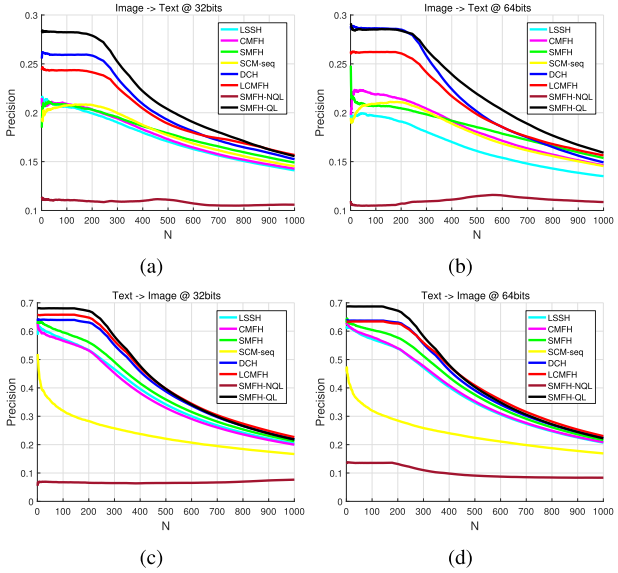


FIGURE 3. topN-Precision curves on Wiki dataset when the number of hash bits is 32 or 64.

From Table 5, we have the following observations: (1) SMFH-QL achieves the best results, which confirms the

TABLE 5. The mAP results on Wiki, MIRFLICKR-25K and NUS-WIDE datasets with different hash code length.

Task	Methods	Wiki				MIRFILCKR-25K				NUS-WIDE			
		16bits	32bits	64bits	128bits	16bits	32bits	64bits	128bits	16bits	32bits	64bits	128bits
I→T	LSSH [36]	0.2297	0.2532	0.2406	0.2336	0.6269	0.6394	0.6350	0.6476	0.5047	0.5176	0.5229	0.5439
	CMFH [37]	0.2454	0.2529	0.2572	0.2613	0.6465	0.6462	0.6478	0.6444	0.4966	0.5150	0.5146	0.5086
	SMFH [42]	0.2276	0.2514	0.2533	0.2564	0.6087	0.6255	0.6596	0.6960	0.5670	0.6031	0.6117	0.6282
	SCM-seq [40]	0.2474	0.2363	0.2589	0.2596	0.6756	0.6891	0.6828	0.6922	0.5496	0.5642	0.5425	0.5369
	DCH [41]	0.3423	0.3599	0.3806	0.3848	0.6694	0.6869	0.6987	0.7014	0.5686	0.6189	0.6118	0.6270
	LCMFH [43]	0.3114	0.3337	0.3455	0.3607	0.6721	0.6710	0.6695	0.6897	0.6191	0.6166	0.6203	0.6295
	SMFH-NQL	0.1830	0.2072	0.2184	0.2135	0.5625	0.5813	0.5787	0.5830	0.4038	0.3834	0.3940	0.3884
	SMFH-QL	**0.3541**	**0.3858**	**0.3924**	**0.3926**	**0.7036**	**0.7203**	**0.7168**	**0.7308**	**0.6119**	**0.6252**	**0.6203**	**0.6358**
T→I	LSSH [36]	0.6134	0.6296	0.6349	0.6304	0.6918	0.7187	0.7381	0.7596	0.6200	0.6494	0.6840	0.7055
	CMFH [37]	0.6105	0.6281	0.6385	0.6468	0.6977	0.7114	0.7433	0.7559	0.4827	0.4909	0.5024	0.4947
	SMFH [42]	0.5590	0.6473	0.6678	0.6617	0.7112	0.7242	0.7434	0.7552	0.5834	0.6522	0.6750	0.7163
	SCM-seq [40]	0.3819	0.4479	0.4418	0.4405	0.6890	0.6957	0.7040	0.7166	0.5496	0.5691	0.5755	0.5955
	DCH [41]	0.6999	0.7051	0.7117	0.7222	0.7459	0.7622	0.7812	0.7928	0.7376	0.7815	0.7898	0.8036
	LCMFH [43]	0.6969	0.7042	0.7094	0.7408	0.7016	0.7061	0.7125	0.7320	0.7052	0.6960	0.7074	0.7178
	SMFH-NQL	0.1732	0.1584	0.2246	0.2426	0.5580	0.5717	0.5614	0.5974	0.4003	0.3746	0.4078	0.3892
	SMFH-QL	**0.7478**	**0.7564**	**0.7669**	**0.7653**	**0.7712**	**0.7904**	**0.7909**	**0.8082**	**0.7935**	**0.7958**	**0.7955**	**0.8077**

efficiency of the proposed algorithm. (2) The mAP values of LCMFH, DCH and SMFH-QL are much better than those of the other baseline methods because generating hash codes directly from the class label matrix improves the performance of hashing-based methods. SMFH-QL again achieves the best results, followed by DCH, possibly because it introduces a quantization loss term, which can learn discriminative common representation via the generated hash code. (3) As the hash code length increases, the performance of SMFH-QL improves, which indicates that longer hash codes embed more semantic information. (4) The mAP values of most algorithms are better in T→I than in I→T because the features of the text modality can better express the semantic information from an original instance. (5) SMFH-NQL achieves much worse mAP scores than the proposed SMFH-QL because the quantitative loss term is not included in SMFH-QL, which leads to large quantization error. These experimental results illustrate the importance of the quantitative loss term and demonstrate the effectiveness of SMFH-QL.

The Precision-Recall curves and the topN-Precision curves on the Wiki dataset are shown in Figures 2 and 3 from 16 bits to 128 bits, respectively. We make the following observations from Fig. 2 and 3. (1) SMFH-QL has the highest precision in image-text searching, which is consistent with the mAP results. (2) SMFH-QL achieves higher precision than SMFH-NQL because of the missing quantitative loss term, which further confirms that our proposed algorithm is superior.

2) ASSESSMENT OF SMFH-QL'S QUALITY ON MIRFLICKR-25K

In this part, we conduct the same experiments as those for the Wiki on MIRFLICKR-25K, where the length of the hash code is 16, 32, 64, and 128 bits. The parameters $\{\lambda, \beta, \alpha, \mu, \gamma\}$ for Eq. (9) are $\{0.5, 10, 100, 1000, 0.1\}$ on MIRFLICKR-25K. The mAP scores of SMFH-QL and the baseline methods are reported in Table 5. From Table 5, we can observe the following results: (1) SMFH-QL outperforms the other methods in terms of mAP on MIRFLICKR-25K, confirming that our proposed algorithm is effective for large-scale datasets. (2) As the hash code length increases, the mAP values of all comparison methods increase because the longer the code length is, the more semantic information the hash code contains. (3) The results of I→T and T→I differ greatly on Wiki, whereas they are similar on MIRFLICKR-25K. There are two reasons for this observation. First, the quality of images on Wiki is poor, and the correlation with semantic tags is inferior. By contrast, the text on Wiki is well edited at the beginning of collection and more tag relevant. Second, there are 25k images on MIRFLICKR-25K with a corresponding tag and annotation attached to each image, which greatly reduces the semantic gap of heterogeneous data and results in inferior retrieval performance.

The Precision-Recall curves and the topN-Precision curves are illustrated in Fig. 4 and 5, respectively. Clearly, SMFH-QL achieves the best results, consistent with the mAP values. In addition, all methods have higher precision on MIRFLICKR-25K than on Wiki because the semantic gap is much smaller for MIRFLICKR-25K. Finally, SMFH-QL achieves higher precision than SMFH-NQL, which confirms the influence of the quantitative loss term and the effectiveness of SMFH-QL.

3) ASSESMENT OF SMFH-QL'S QUALITY ON NUS-WIDE

In this part, we compare SMFH-QL and other methods on the NUS-WIDE dataset. The parameters $\{\lambda, \beta, \alpha, \mu, \gamma\}$ for Eq. (9) are $\{0.5, 10, 100, 1000, 0.1\}$. Table 5 shows the mAP values of all baseline methods on NUS-WIDE. Fig. 6 and 7 plot the Precision-Recall curves and the topN-Precision curves for all methods, respectively. According to the experimental results from Table 5 and Fig. 6 and 7, the proposed SMFH-QL outperforms the other methods in image-text search tasks because SMFH-QL introduces a quantization loss function term to constrain the objective function,

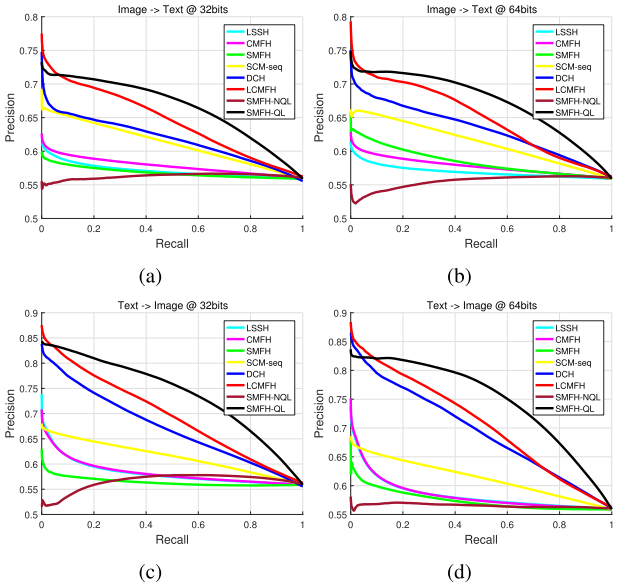


FIGURE 4. Precision-Recall curves on MIRFLICKR-25K dataset when the hash code length is set to 32 bits or 64 bits.

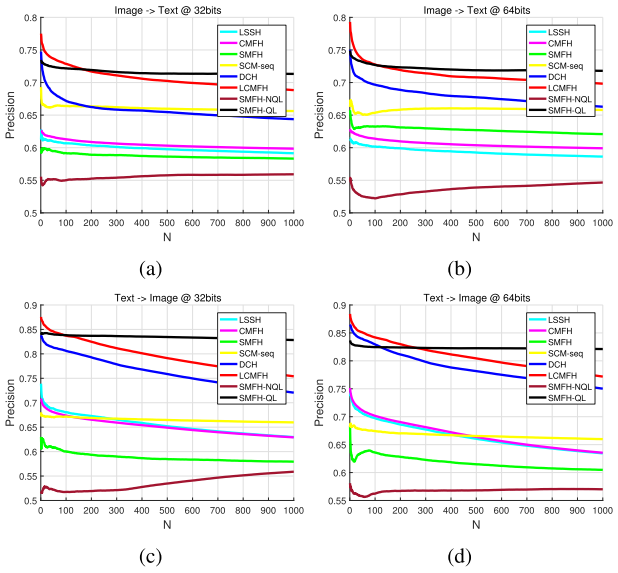


FIGURE 5. topN-Precision curves on MIRFLICKR-25K dataset when the number of hash bits is 32 or 64.

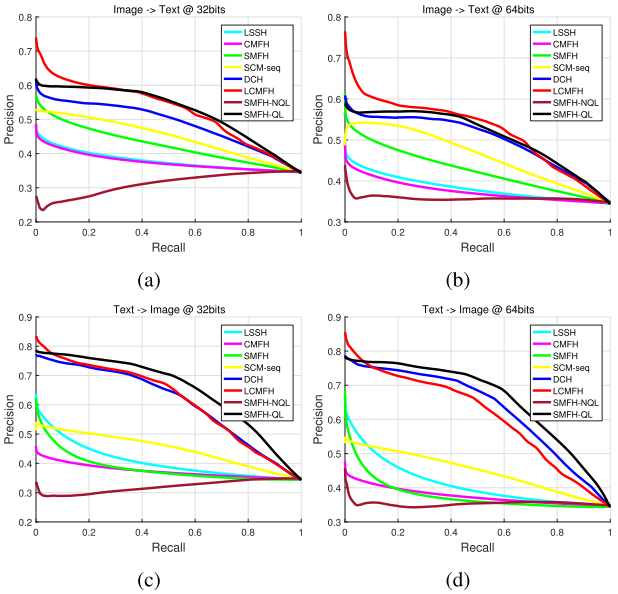


FIGURE 6. Precision-Recall curves on NUS-WIDE dataset when the hash code length is set to 32 bits or 64 bits.

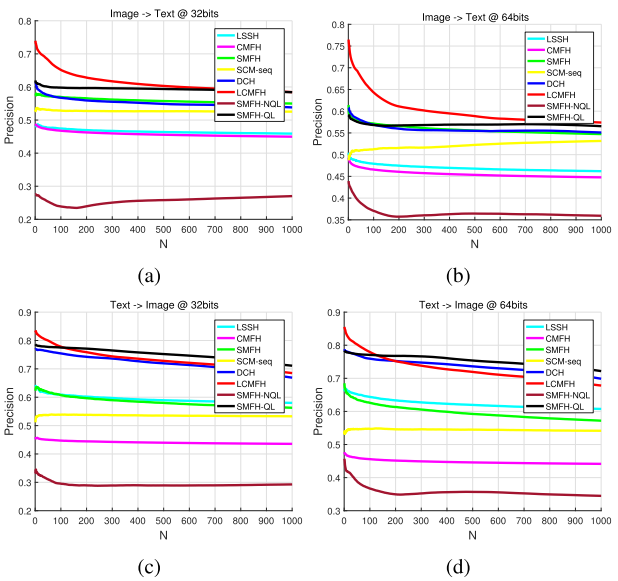


FIGURE 7. topN-Precision curves on NUS-WIDE dataset when the number of hash bits is 32 or 64.

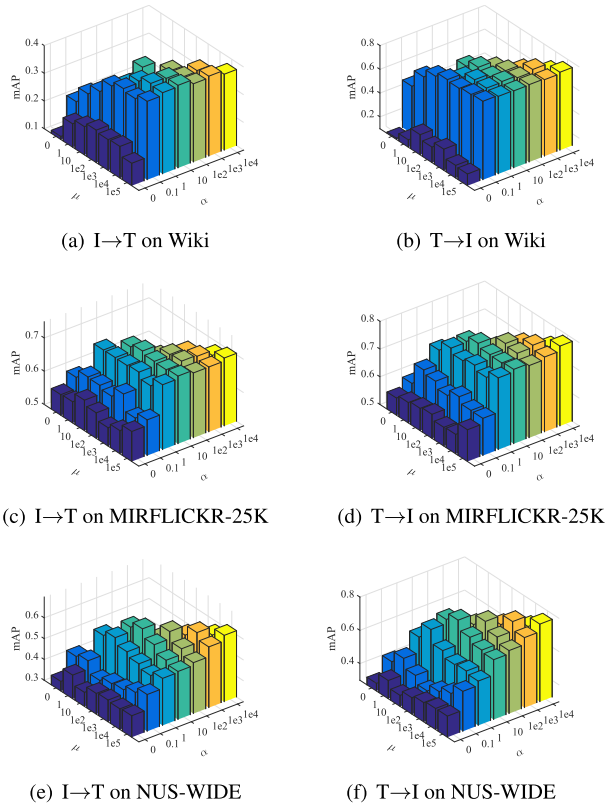
which yields a closer correlation between the shared semantic representation and hash codes. In addition, SMFH-NQL performs much worse than SMFH-QL, which is consistent with evaluation of Wiki and MRIFLICKR-25K. These experimental results demonstrate the superior retrieval performance of SMFH-QL.

4) EVALUATION OF THE QUANTITATIVE LOSS TERM

The parameter μ controls the importance of label consistency and hash codes, α influences the quantitative loss function term, and β influences the performance of the learning hash function. Here, we analyze the effect of the quantization loss term on generating hash code and learning the hash function during the training procedure. After fixing the code length to 64 bits, we vary $\alpha \in \{0, 0.1, 1, 10, 1e2, 1e3, 1e4\}$, $\mu \in \{0, 1, 10, 1e2, 1e3, 1e4, 1e5\}$ and $\beta \in \{0, 0.01, 0.1, 1, 10, 1e2, 1e3\}$. Then, we assess α and μ, α and β on two retrieval tasks on the three benchmark datasets.

The six subfigures in Fig. 8 illustrate the mAP values on the two retrieval tasks with the different settings of α and μ on the three image-text datasets. The results shown in Fig. 8 yield the following observations.

Observation 1: Performance on Wiki. (1) SMFH-QL can achieve stable performance when α varies in [10,1e4] and



(a) I→T on Wiki (b) T→I on Wiki

(c) I→T on MIRFLICKR-25K (d) T→I on MIRFLICKR-25K

(e) I→T on NUS-WIDE (f) T→I on NUS-WIDE

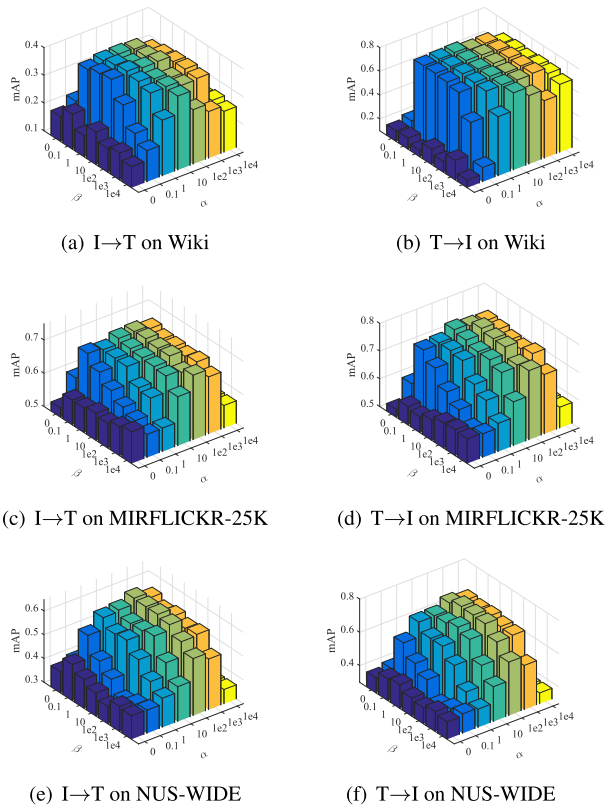
FIGURE 8. Performance variation for mAP with respect to α and μ @ 64 bits on the three datastes.

μ varies in [1e3,1e5]. (2) When α is excessively small, e.g., α is in [0,1], the mAP scores of SMFH-QL deteriorate because the learned hash codes do not consider the common latent features, which makes the hash function poor. When α is excessively large, e.g., α is in [1e3,1e4], the mAP scores of SMFH-QL start to decline because the learned hash codes consider more common latent features, which makes the hash function imprecise. Therefore, a suitable score for α can decide the range of μ. (3) In addition, when α is in [1,100], SMFH-QL achieves the best performance on the three datasets for μ in [1e3,1e4]. Thus, as for Wiki, if the value of α is excessively large or small, it can affect the value of μ, which prevents the generated hash codes from learning a discriminative common feature representation. In addition, the imprecise common feature representation worsens the to-be-learned hash function and ultimately degrades the image-text search performance.

Observation 2: Performance on MIRFLICKR-25K. (1) For $\alpha = 0$, the mAP values are the worst result on both retrieval tasks, regardless of the value of μ. The main reason is that SMFH-QL can only learn the hash function from the common feature representation obtained by matrix factorization of the original image-text data. Thus, in this case, the label information does not effect common feature representation, which makes the to-be-learned hash function weak and consequently affects the efficiency of the image-text search. (2) When the values of α vary in the range of [1e3, 1e4], the mAP values corresponding to μ are unstable. (3) When the value of α is in [10,100], SMFH-QL achieves the best performance on the three datasets for μ in [1e3,1e4].

Observation 3: Performance on NUS-WIDE. Similarly, the same observations can be found on NUS-WIDE. Thus, the quantitative loss term has a substantial effect on the classification loss term, which demonstrates the importance of the quantitative loss function term.

The six subfigures in Fig. 9 illustrate the mAP scores in the two retrieval tasks under different settings of α and β on the three benchmark datasets. From Fig. 9, we have the following observations. (1) When α is in the range of [0,1], the results of mAP corresponding to parameter β are sensitive and unstable for the three datasets. For $\alpha = 0$, the mAP corresponding to β achieves the worst scores because the generated hash code cannot be associated with common feature representation. That is, the label information does not guide the process of learning the hash function, which impacts the retrieval performance. (2) From Figure 8, except for Wiki, if α is in the range of [1e3,1e4], the mAP corresponding to β obtains worse scores when α is in [0,1]. One potential reason is that the quantitative loss term generates a large value of matrix V, which leads to redundant features in label information during learning of the hash function. (3) Another phenomenon is



(a) I→T on Wiki (b) T→I on Wiki

(c) I→T on MIRFLICKR-25K (d) T→I on MIRFLICKR-25K

(e) I→T on NUS-WIDE (f) T→I on NUS-WIDE

FIGURE 9. Performance variation for mAP with respect to α and β @ 64 bits on the three datastes.

that the mAP of β in the range of [10,100] can obtain the best scores when the values of α is in the range of [10,100], i.e., SMFH-QL achieves optimal performance.

In summary, the quantitative loss function term establishes similarity correlation of hash codes H and common feature representation V, resulting in learning of a better discriminative hash function during the training procedure. In light of the aforementioned analysis, we confirm that the quantitative loss term yields a close connection between the classification loss term and the hash function term; i.e., the label information indirectly guides the procedure of learning the hash function. Finally, Fig. 8 and 9 directly verify the importance of the quantitative loss term and indirectly demonstrate the effectiveness of SMFH-QL in image-text searching.

E. EMPIRICAL ANALYSIS

In the previous section, we conducted extensive experiments on the three datasets and employed three most common evaluation metrics, i.e., mAP, Precision-Recall curve and topN-Precision curve. In this section, we perform an empirical analysis to validate the retrieval performance of the proposed SMFH-QL algorithm.

1) CONVERGENCE ANALYSIS

The objective function of SMFH-QL is optimized via an iterative strategy in Algorithm 1. The convergence rate of this strategy is significant for the retrieval performance of SMFH-QL. Hence, we perform additional experiments on the same three datasets with the length of hash code fixed to 64 bits. The convergence curves are shown in Fig. 10. According to Figure 10, the following observations can be obtained.

Algorithm 1 SMFH-QL Training Procedure

Input:

Feature matrices X, Y, semantic label matrix T, parameters λ, μ, α, β, γ, number of iterations w;

Output:

Hash code matrix H, mapping matrices W_1, W_2;

1: Randomly initialize H, Z, U_1, U_2, W_1, W_2;
2: Randomly select anchor point sets in image-text modality;
3: **for** $j = 1$ to w **do**
4: Calculate U_1, U_2 using Eq. (11), Eq. (12), respectively;
5: Calculate W_1, W_2 using Eq. (14), Eq. (15), respectively;
6: Calculate Z using Eq. (17);
7: Calculate V using Eq. (18);
8: Calculate H using Eq. (23);
9: **end for**
10: **return** H, W_1, W_2;

(1) We conduct 100 iterations for convergence on the Wiki, MIRFLICKR-25K and NUS-WIDE datasets, respectively. The objective function values of SMFH-QL converge in approximately 10 iterations, which guarantees the effectiveness of the proposed method. (2) For Wiki and NUS-WIDE, the objective function values for SMFH-QL decrease faster, indicating that the proposed method has better convergence on these datasets.

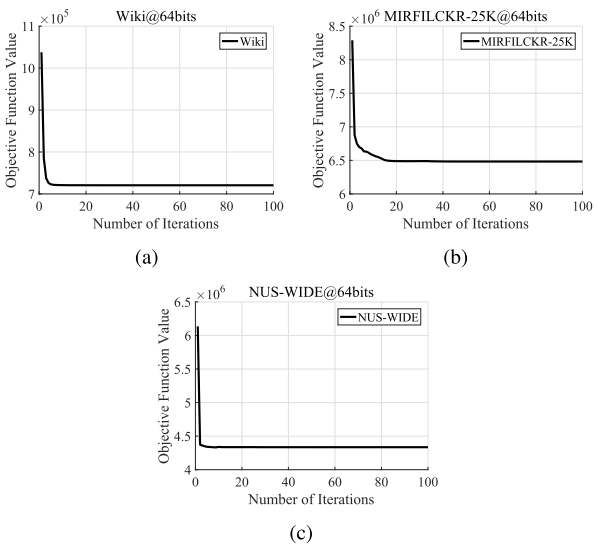


FIGURE 10. Convergence curves on Wiki, MIRFLICKR-25K and NUS-WIDE datasets when the hash code length is set to 64 bits.(a) Wiki. (b) MIRFLICKR-25K. (c) NUS-WIDE.

2) TRAINING TIME ANALYSIS

To demonstrate the efficiency of SMFH-QL on large-scale datasets, we analyze and compare the training time (in seconds) of all comparison methods with different code lengths on MIRFLICKR-25K and NUS-WIDE. The experimental results are shown in Tables 6 and 7. For all comparison approaches, the training time includes the time for learning the hash function and hash code.

TABLE 6. Training time comparison on MIRFLICKR-25K (in seconds) with code from 16 bits to 128 bits.

Method	16 bits	32 bits	64 bits	128 bits
LSSH [36]	115.40	119.42	127.88	163.08
CMFH [37]	51.95	57.12	61.85	67.60
SMFH [42]	27.36	27.71	33.29	40.91
SCM-seq [40]	35.76	57.41	113.10	218.36
DCH [41]	7.16	8.15	9.36	12.28
LCMFH [43]	6.97	7.85	9.53	11.11
SMFH-NQL	6.17	9.59	8.84	**12.83**
SMFH-QL	**6.06**	**6.96**	**8.66**	12.87

(1) As shown in Table 6, LSSH takes the most time in different bits during the training procedure. SCM-seq consumes more and more time as the code length increases. CMFH and SMFH require less time than LSSH and SCM-seq but much more time than DCH, LCMFH,SMFH-NQL and SMFH-QL because these four methods directly guide the hash code or hash function procedure based on the semantic label, thereby avoiding the construction of pairwise similarity matrices that requires a large quantity of calculation.

TABLE 7. Training time comparison on NUS-WIDE (in seconds) with code from 16 bits to 128 bits.

Method	16 bits	32 bits	64 bits	128 bits
LSSH [36]	762.61	895.10	878.10	889.17
CMFH [37]	546.07	591.25	522.80	703.03
SMFH [42]	518.27	1848	5188	12234
SCM-seq [40]	118.54	176	276	433
DCH [41]	12.82	12.63	14.18	17.93
LCMFH [43]	**5.44**	**5.18**	**6.69**	**8.92**
SMFH-NQL	6.13	7.54	10.19	12.67
SMFH-QL	6.25	6.82	8.77	12.44

(2) From Table 7, we see that LSSH, CMFH, SMFH, and SCM-seq require much more training time than DCH, LCMFH,SMFH-NQL and SMFH-QL. Additionally, the latter four methods have similar training times, which is the same as the results for MIRFLICKR-25K. Besides, LCMFH has achieved the best reesults than other methods. However, in summary, SMFH-QL possesses the best retrieval performance when keeping similar computational efficiency than several state-of-the-art hashing-based image-text methods.

3) PARAMETER ANALYSIS

In the previous experiments, we empirically set all values of parameters based on three datasets. In this section, we conduct confirmatory experiments to demonstrate the influence of changing parameters $\lambda, \beta, \alpha, \mu, \gamma$ on the proposed SMFH-QL. All parameter experiments are completed on the same three datasets with 64 bit hash codes, where "I→T" and "I→T" denote querying text by image and querying image by text, respectively. As shown in Fig. 11, we have the following observations.

- **Parameter** λ controls the influence of image-text modalities during the procedure of matrix factorization. We consider that the original image-text data play an identical role in matrix decomposition and keep common importance in SMFH-QL. Therefore, we set the value $\lambda = 0.5$ without concrete analysis.
- **Parameter** β influences the performance of learning the hash function term. From Fig. 11, it can be observed that SMFH-QL achieves the best results when $\beta = 10$ on the three datasets.
- **Parameter** μ controls the importance of label consistency and the generated hash codes in the final objective function. If μ is too small, it cannot make full use of label information to generate a discriminative hash code, which reduces the performance of SMFH-QL. If the value of μ is too large, the redundant features in the label matrix will be brought into the procedure of learning the hash code, reducing the quality of the hash code. Fig. 11 illustrates that SMFH-QL achieves the best results when $\mu = 10000$ on Wiki and when μ is 1000 on MIRFLICKR-25K and NUS-WIDE.
- **Parameter** α influences the quantitative loss function term. If μ is too small or large, it degrades the performance of SMFH-QL. The main reason is that the common semantic representation V cannot be accurately represented via the learned binary code H. Therefore, we obtain $\alpha = 10$ on Wiki and $\alpha = 100$ on MIRFLICKR-25K and NUS-WIDE based on Fig. 11.
- **Parameter** γ controls the weight of the regularization term. Thus, when the value of γ is too small, the effect of the regularization term will be reduced, which makes the training process of SMFH-QL overfitted. By contrast, when γ is too small, underfitting may occur. In addition, Fig. 11 shows that the mAP of SMFH-QL remains stable on MIRFLICKR-25K and NUS-WIDE but tends to decline on Wiki when γ is more than 0.1. One possible reason is that the training sets in Wiki are small, which results in overfitting.

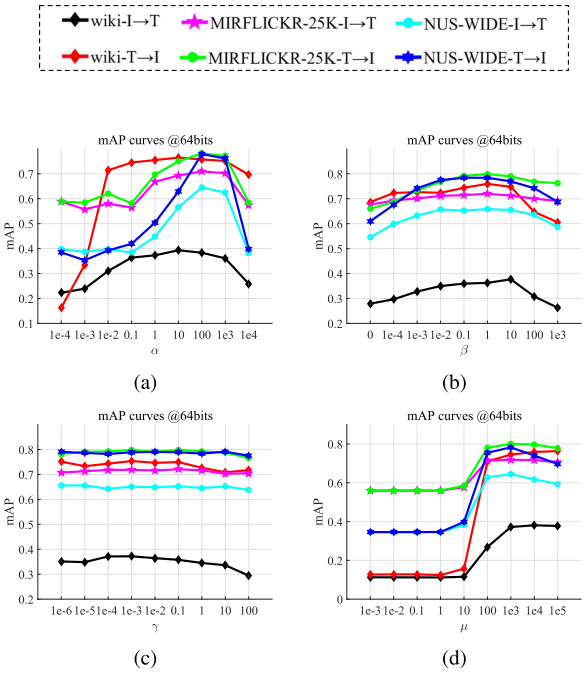


FIGURE 11. Parameter analysis curves on the three datasets when the hash code length is set to 64 bits.

From Fig. 11, we observe that $\alpha \in [10, 1000]$, $\beta \in [10, 1000]$, $\mu \in [0.01, 10]$, and $\gamma \in [0.001, 1]$ are insensitive, and SMFH-QL achieves satisfactory results with a wide range of parameter values.

V. CONCLUSION

In this paper, we propose a novel discrete supervised matrix factorization hashing-based method for image-text searching. The proposed SMFH-QL can learn discriminative hash codes because of two contributions: (1) it directly generates hash codes and a common feature representation by employing semantic class and matrix factorization, respectively; (2) it constructs a strong similarity correlation between the common feature representation and hash codes when performing quantization. Extensive experiments on three widely used benchmark datasets demonstrate that SMFH-QL substantially outperforms several state-of-the-art hashing-based

image-text search methods. In future work, we plan to integrate the proposed SMFH-QL with manifold embedding learning to capture the real local structure in the original image-text data, which can be used to generate more compact hash codes.

REFERENCES

[1] K. Nai, Z. Li, G. Li, and S. Wang, "Robust object tracking via local sparse appearance model," *IEEE Trans. Image Process.*, vol. 27, no. 10, pp. 4958–4970, Oct. 2018.

[2] G. Li, M. Peng, K. Nai, Z. Li, and K. Li, "Multi-view correlation tracking with adaptive memory-improved update model," *Neural Comput. Appl.*, no. 4, pp. 1–17, Aug. 2019, doi: 10.1007/s00521-019-04413-4.

[3] Z. Li, S. Gao, and K. Nai, "Robust object tracking based on adaptive templates matching via the fusion of multiple features," *J. Vis. Commun. Image Represent.*, vol. 44, pp. 1–20, Apr. 2017.

[4] K. Xie, X. Li, X. Wang, G. Xie, J. Wen, J. Cao, and D. Zhang, "Fast tensor factorization for accurate Internet anomaly detection," *IEEE/ACM Trans. Netw.*, vol. 25, no. 6, pp. 3794–3807, Dec. 2017.

[5] W. E. A. Chen, "A novel fuzzy deep-learning approach to traffic flow prediction with uncertain spatial–temporal data features," *Future Gener. Comput. Syst.*, vol. 89, pp. 78–88, Dec. 2018.

[6] S. He and H. Zhao, "Automatic syllable segmentation algorithm of chinese speech based on MF-DFA," *Speech Commun.*, vol. 92, pp. 42–51, Sep. 2017.

[7] X. Zhao, Y. Lin, and J. Heikkila, "Dynamic texture recognition using volume local binary count patterns with an application to 2D face spoofing detection," *IEEE Trans. Multimedia*, vol. 20, no. 3, pp. 552–566, Mar. 2018.

[8] C. Xia, H. Zhang, and X. Gao, "Combining multi-layer integration algorithm with background prior and label propagation for saliency detection," *J. Vis. Commun. Image Represent.*, vol. 48, pp. 110–121, Oct. 2017.

[9] G. Xie, G. Zeng, J. Jiang, C. Fan, R. Li, and K. Li, "Energy management for multiple real-time workflows on cyber–physical cloud systems," *Future Gener. Comput. Syst.*, vol. 105, pp. 916–931, Apr. 2020.

[10] B. Yang, Z. Li, S. Jiang, and K. Li, "Envy-free auction mechanism for VM pricing and allocation in clouds," *Future Gener. Comput. Syst.*, vol. 86, pp. 680–693, Sep. 2018.

[11] L. Nie, X. Wang, J. Zhang, X. He, H. Zhang, R. Hong, and Q. Tian, "Enhancing micro-video understanding by harnessing external sounds," in *Proc. ACM Multimedia Conf. (MM)*, 2017, pp. 1192–1200.

[12] L. Nie, M. Liu, and X. Song, "Multimodal learning toward micro-video understanding," *Synth. Lectures Image, Video, Multimedia Process.*, vol. 9, no. 4, pp. 1–186, Sep. 2019.

[13] J. Youn, J. Shim, and S.-G. Lee, "Efficient data stream clustering with sliding windows based on locality-sensitive hashing," *IEEE Access*, vol. 6, pp. 63757–63776, 2018.

[14] S. Su, Z. Zhong, and C. Zhang, "Deep joint-semantics reconstructing hashing for large-scale unsupervised cross-modal retrieval," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Seoul, South Korea, Oct. 2019, pp. 3027–3035.

[15] J. Zhang and Y. Peng, "Multi-pathway generative adversarial hashing for unsupervised cross-modal retrieval," *IEEE Trans. Multimedia*, vol. 22, no. 1, pp. 174–187, Jan. 2020.

[16] X. Qi, X. Zeng, and H. Tang, "Cross-modal hashing retrieval based on density clustering," *IEEE Access*, early access, Mar. 6, 2020, doi: 10.1109/ACCESS.2020.2978876.

[17] Y. Zhang, J. Cao, and X. Gu, "Learning cross-modal aligned representation with graph embedding," *IEEE Access*, vol. 6, pp. 77321–77333, 2018.

[18] X. Li, L. Gao, X. Xu, J. Shao, F. Shen, and J. Song, "Kernel based latent semantic sparse hashing for large-scale retrieval from heterogeneous data sources," *Neurocomputing*, vol. 253, pp. 89–96, Aug. 2017.

[19] J. Yu and X.-J. Wu, "Unsupervised multimodal hashing for cross-modal retrieval," 2019, *arXiv:1904.00726*. [Online]. Available: http://arxiv.org/abs/1904.00726

[20] T. Yao, Z. Zhang, L. Yan, J. Yue, and Q. Tian, "Discrete robust supervised hashing for cross-modal retrieval," *IEEE Access*, vol. 7, pp. 39806–39814, 2019.

[21] F. Shen, Y. Xu, L. Liu, Y. Yang, Z. Huang, and H. T. Shen, "Unsupervised deep hashing with similarity-adaptive and discrete optimization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 12, pp. 3034–3044, Dec. 2018.

[22] C. Huang, C. C. Loy, and X. Tang, "Unsupervised learning of discriminative attributes and visual representations," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 5175–5184.

[23] Z. Ji, W. Yao, W. Wei, H. Song, and H. Pi, "Deep multi-level semantic hashing for cross-modal retrieval," *IEEE Access*, vol. 7, pp. 23667–23674, 2019.

[24] Z. Chen, F. Zhong, G. Min, Y. Leng, and Y. Ying, "Supervised Intra- and inter-modality similarity preserving hashing for cross-modal retrieval," *IEEE Access*, vol. 6, pp. 27796–27808, 2018.

[25] C.-X. Li, T.-K. Yan, X. Luo, L. Nie, and X.-S. Xu, "Supervised robust discrete multimodal hashing for cross-media retrieval," *IEEE Trans. Multimedia*, vol. 21, no. 11, pp. 2863–2877, Nov. 2019.

[26] L. Zhu, X. Lu, Z. Cheng, J. Li, and H. Zhang, "Flexible multi-modal hashing for scalable multimedia retrieval," *ACM Trans. Intell. Syst. Technol.*, vol. 11, no. 2, pp. 1–20, Mar. 2020.

[27] S. Chen, F. Shen, Y. Yang, X. Xu, and J. Song, "Supervised hashing with adaptive discrete optimization for multimedia retrieval," *Neurocomputing*, vol. 253, pp. 97–103, Aug. 2017.

[28] J. Tang, K. Wang, and L. Shao, "Supervised matrix factorization hashing for cross-modal retrieval," *IEEE Trans. Image Process.*, vol. 25, no. 7, pp. 3157–3166, Jul. 2016.

[29] Y. Wu, X. Luo, X.-S. Xu, S. Guo, and Y. Shi, "Dictionary learning based supervised discrete hashing for cross-media retrieval," in *Proc. ACM Int. Conf. Multimedia Retr.*, Seoul, South Korea, Jun. 2018, pp. 222–230.

[30] Q.-Y. Jiang and W.-J. Li, "Discrete latent factor model for cross-modal hashing," *IEEE Trans. Image Process.*, vol. 28, no. 7, pp. 3490–3501, Jul. 2019.

[31] J. Gui, T. Liu, Z. Sun, D. Tao, and T. Tan, "Fast supervised discrete hashing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 2, pp. 490–496, Feb. 2018.

[32] Y. Luo, Y. Yang, F. Shen, Z. Huang, P. Zhou, and H. T. Shen, "Robust discrete code modeling for supervised hashing," *Pattern Recognit.*, vol. 75, pp. 128–135, Mar. 2018.

[33] S. Kumar and R. Udupa, "Learning hash functions for cross-view similarity search," in *Proc. 20th Int. Joint Conf. Artif. Intell.*, Barcelona, Spain, Jul. 2011, pp. 1360–1365.

[34] Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing," in *Proc. 23rd Neural Inf. Process. Syst.*, Vancouver, BC, Canada, Dec. 2009, pp. 1753–1760.

[35] X. Zhu, Z. Huang, H. T. Shen, and X. Zhao, "Linear cross-modal hashing for efficient multimedia search," in *Proc. 21st ACM Int. Conf. Multimedia (MM)*, Barcelona, Spain, 2013, pp. 143–152.

[36] J. Zhou, G. Ding, and Y. Guo, "Latent semantic sparse hashing for cross-modal similarity search," in *Proc. 37th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, Gold Coast, QLD, Australia, 2014, pp. 415–424.

[37] G. Ding, Y. Guo, and J. Zhou, "Collective matrix factorization hashing for multimodal data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Columbus, OH, USA, Jun. 2014, pp. 2083–2090.

[38] M. M. Bronstein, A. M. Bronstein, F. Michel, and N. Paragios, "Data fusion through cross-modality metric learning using similarity-sensitive hashing," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, San Francisco, CA, USA, Jun. 2010, pp. 3594–3601.

[39] Z. Lin, G. Ding, J. Han, and J. Wang, "Cross-view retrieval via probability-based semantics-preserving hashing," *IEEE Trans. Cybern.*, vol. 47, no. 12, pp. 4342–4355, Dec. 2017.

[40] D. Zhang and W.-J. Li, "Large-scale supervised multimodal hashing with semantic correlation maximization," in *Proc. 28th AAAI Conf. Artif. Intell.*, Quebec City, QC, Canada, Jul. 2014, pp. 2177–2183.

[41] X. Xu, F. Shen, Y. Yang, H. T. Shen, and X. Li, "Learning discriminative binary codes for large-scale cross-modal retrieval," *IEEE Trans. Image Process.*, vol. 26, no. 5, pp. 2494–2507, May 2017.

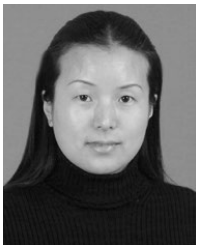
[42] H. Liu, R. Ji, Y. Wu, and G. Hua, "Supervised matrix factorization for cross-modality hashing," in *Proc. 25th Int. Joint Conf. Artif. Intell.*, New York, NY, USA, Jul. 2016, pp. 1767–1773.

[43] D. Wang, X. Gao, X. Wang, and L. He, "Label consistent matrix factorization hashing for large-scale cross-modal similarity search," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 10, pp. 2466–2479, Oct. 2019.

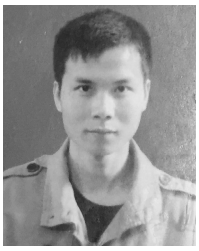
[44] W. Liu, J. Wang, R. Ji, Y.-G. Jiang, and S.-F. Chang, "Supervised hashing with kernels," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2074–2081.

[45] H.-J. Huang, R. Yang, C.-X. Li, Y. Shi, S. Guo, and X.-S. Xu, "Supervised cross-modal hashing without relaxation," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2017, pp. 1159–1164.

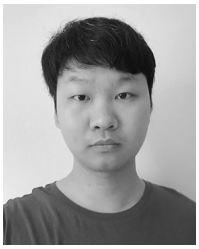
[46] J. Costa Pereira, E. Coviello, G. Doyle, N. Rasiwasia, G. R. G. Lanckriet, R. Levy, and N. Vasconcelos, "On the role of correlation and abstraction in cross-modal multimedia retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 3, pp. 521–535, Mar. 2014.
[47] M. J. Huiskes and M. S. Lew, "The MIR flickr retrieval evaluation," in *Proc. 1st ACM Int. Conf. Multimedia Inf. Retr. (MIR)*, Vancouver, BC, Canada, 2008, pp. 39–43.
[48] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y. Zheng, "NUS-WIDE: A real-world Web image database from National University of Singapore," in *Proc. ACM Int. Conf. Image Video Retr.*, Santorini Island, Greece, 2009, pp. 48–56.
[49] R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*, vol. 463. New York, NY, USA: ACM, 1999.



HUAN ZHAO received the B.S., M.S., and Ph.D. degrees in computer science and technology from Hunan University, Changsha, China, in 1989, 2004, and 2010, respectively. She is a currently a Professor with the School of Information Science and Technology, Hunan University. Her current research interests mainly include speech signal processing, cross-media retrieval, and natural language processing.



SONG WANG received the M.S. degree from the Changsha University of Science and Technology, China, in 2017. He is currently pursuing the Ph.D. degree in computer science and technology with Hunan University, Changsha, China. His research interests include multimedia analysis and retrieval.



XIAOLIN SHE received the B.S. degree in computer science and technology from the Changsha University of Science and Technology, China, in 2018. He is currently pursuing the M.S. degree in computer technology with Hunan University, China. His research interests include machine learning and multimedia retrieval.

CHENGHUI SU received the B.S. degree in law from Dali University, in 2012, and the M.S. and Ph.D. degrees in law from the Southwest University of Political Science & Law, in 2016 and 2019, respectively. She is currently a Lecturer with the Southwest University of Political Science & Law and a full-time Researcher with the Higher Research Institute, Southwest University of Political Science & Law. Her research interests include network information law and data law.