Knowledge-Based Systems 204 (2020) 106188

Contents lists available at ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Discriminative deep asymmetric supervised hashing for cross-modal retrieval

Haopeng Qiang [a], Yuan Wan [a,*], Ziyi Liu [a], Lun Xiang [b], Xiaojing Meng [a]

[a] Mathematical Department, School of Science, Wuhan University of Technology, Wuhan, 430070, China
[b] Statistics Department, School of Science, Wuhan University of Technology, Wuhan, 430070, China

ARTICLE INFO

Article history:
Received 12 March 2020
Received in revised form 24 May 2020
Accepted 22 June 2020
Available online 29 June 2020

Keywords:
Asymmetric hashing learning
Discrete optimization
Discriminative
Cross-modal retrieval

ABSTRACT

Due to the advantages of low storage cost and high retrieval efficiency, cross-modal hashing has received considerate attention. Most existing deep cross-modal hashing adopt a symmetric strategy to learn same deep hash functions for both query instances and database instances. However, the training of these symmetric deep cross-modal hashing methods is time-consuming, which makes them hard to effectively utilize the supervised information for cases with large-scale datasets. Inspired by the latest advance in the asymmetric hashing scheme, in this paper, we propose a discriminative deep asymmetric supervised hashing (DDASH) for cross-modal retrieval. Specifically, asymmetric hashing only learns hash codes of query instances by deep hash functions while learning the hash codes of the database instances by hand-crafted matrices. It cannot only make full use of the information in large-scale datasets, but also reduce the training time. Besides, we introduce discrete optimization to reduce the binary quantization error. Furthermore, a mapping matrix which maps generated hash codes into the corresponding labels is introduced to ensure that the hash codes are discriminative. We also calculate the level of similarity between instances as supervised information. Experiments on three common datasets for cross-modal retrieval show that DDASH outperforms state-of-the-art cross-modal hashing methods.

1. Introduction

With the explosive growing of multimedia data on the Internet, retrieval between instances of different modalities has attracted much attention. For example, given an image query, one may want to obtain all semantically related images, videos and audios from the database. Cross-modal retrieval, which takes one type of data as the query and returns the relevant data of other types, has incrementally received attention since it is a natural way to search for multi-modal data [1]. However, due to the inconsistent distribution and representation, there are heterogeneity gaps between different modalities. So it is difficult to return correct semantic results on the large-scale cross-modal retrieval task. Finding solutions that effectively bridge the semantic gap and achieve fast retrieval for cross-modal retrieval has become a hot research spot.

In big data applications, the searching time for exact nearest neighbor (NN) [2,3] is typically expensive for the given queries [4]. Hence, approximate nearest neighbor (ANN) search [5] has become more and more popular. Hashing, as one of the most widely used techniques for ANN search, aims to encode the data points from original space into hamming space. Thanks to the binary hash code representation, hashing methods provide constant or sub-linear search time and dramatically reduce the storage cost for the data points [6], thus cross-modal hashing methods have attracted much attention from both academia and industry [2,7–14]. In the early cross-modal hashing methods [7,8,15,16], feature extracting and hash coding are two separate steps, which means that the hand-crafted features might not be optimally compatible with the hash coding procedure. Hence, these existing cross-modal hashing methods with hand-crafted features may not achieve satisfactory performance in real applications [11]. With the great development of deep learning, deep neural network has been widely incorporated in cross-modal hashing [11,17–20]. Compared with early methods, these cross-modal hashing models utilize semantic labels to enhance the correlation of cross-modal data while the modality features and hash functions can be jointly learned in an end-to-end framework.

Most existing deep cross-modal hashing methods adopt a symmetric strategy to learning the same deep hash functions for both query and database instances. As described in [21], the crucial disadvantage of these symmetric hashing strategy is that the symmetric discrete constraint might be difficult to optimize. In image retrieval, recent works [4,21,22] have exhibited that

* Corresponding author.
E-mail addresses: qianghaopengshuxuejd@whut.edu.cn (H. Qiang), wanyuan@whut.edu.cn (Y. Wan), liuziyi@whut.edu.cn (Z. Liu), xianglun@whut.edu.cn (L. Xiang), mxj@whut.edu.cn (X. Meng).

https://doi.org/10.1016/j.knosys.2020.106188
0950-7051/

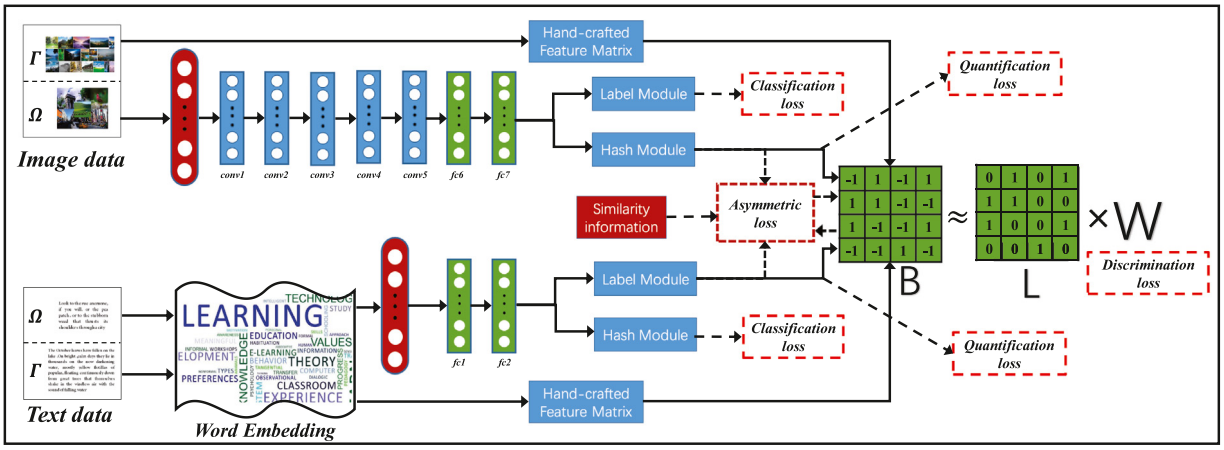


Fig. 1. The framework of discriminative deep asymmetric supervised hashing (DDASH). An Image CNN network for learning image representations and a Text neural network for learning text representations, respectively. DDASH encompasses three steps: (1) feature extracting networks extract the features of different modalities (2) Jointly optimize the asymmetric loss and classification loss to seek the optimal parameters of these networks and those of hashing functions. (3) Optimize the discrimination loss to ensure that the generated hash code is discriminative.

the asymmetric hashing, which uses different hash functions for query instances and database instances, can preserve more similarity (label) information, which means better retrieval accuracy. However, few works [23] employ asymmetric hashing in cross-modal retrieval. Besides, the training of these deep symmetric cross-modal hashing methods is typically time-consuming. In order to reduce the consumption of training time and make the training practicable, most existing deep cross-modal hashing methods have to sample a small subset from the whole dataset to construct a training set for deep hash functions learning, so that many instances in dataset may be discarded during training [4]. Besides, these methods choose to first solve a relaxed problem to simplify the optimization involved in the binary code learning procedure, and then quantize the solved continuous solution to achieve the approximate binary solution. The binary quantization error is also difficult to optimize when dealing with large scale with training data points. Hence, it is hard for these methods to make the search performance satisfactory in the cases with large-scale datasets. Furthermore, the hash codes generated by these methods may be weak in discrimination. A high-quality hash code should be discriminative, which means that it can represent the features of the corresponding instance. And these methods also fail to do not take the level of similarity between data points into account.

In this paper, we propose a novel method named *discriminative deep asymmetric supervised hashing* (DDASH) for cross-modal retrieval. Fig. 1 shows the framework of DDASH. In order to make full use of supervised information in the dataset and make the training practicable, DDASH divides the dataset into two parts: query instances and database instances. The number of query instances is much less than the number of database instances. DDASH treats the query instances and database instances in an asymmetric way. Specifically, DDASH learns deep hash functions only for query instances, while the binary hash codes for database instances are learned by hand-crafted feature matrices. In addition, DDASH adopts a discrete optimization strategy to reduce the binary quantization error, so that the search performance of our method in dealing with large-scale datasets is satisfactory. Furthermore, in order to make the generated hash codes discriminative, we introduce a simple linear mapping that can map the generated hash codes into the corresponding labels. And the level of similarity between instances is also considered in DDASH instead of simply dividing them into similar or dissimilar.

The main contributions of DDASH are outlined as follows:

- We propose a novel discriminative deep asymmetric supervised hashing (DDASH) method for cross-modal retrieval. DDASH learns deep hash functions for query instances and learns the binary hash codes for database instances through hand-crafted feature matrices. This asymmetric training method is more efficient when dealing with large scale dataset.
- By utilizing discrete optimization, the unified discrete binary codes can be solved without relaxation. It cannot only reduce the binary quantization error, but also effectively reduce the consumption of time and computing resource.
- We introduce a mapping matrix to help our networks generate discriminative binary hash codes, and the level of similarity between instances is also considered.
- Experiments on three common open datasets for cross-modal retrieval show that our method outperforms state-of-the-art cross-modal hashing methods.

The remaining part of this paper is organized as follows. In Section 2, several related works are introduced. Our proposed method and optimization are presented in Section 3. Experiments are shown in Section 4 and Section 5 concludes this work.

2. Related work

Cross-modal hashing methods can be divided into unsupervised hashing methods and supervised hashing methods, based on whether supervised information is used for learning or not [24]. Unsupervised hashing methods learn hash functions by mining the information of training data without supervised information. Representative unsupervised cross-modal hashing methods include collective matrix factorization hashing (CMFH) [25], latent semantic sparse hashing (LSSH) [26], alternating co-quantization (ACQ) [27] and unsupervised generative adversarial cross-modal hashing (UGACH) [28]. CMFH and LSSH both adopt collective matrix factorization to learn cross-view hash functions. ACQ generates high-quality hash codes by minimizing the binary quantization error. UGAN uses generative model [29] to fit the distribution over the manifold structure, and selects informative data of another modality to challenge the discriminative model.

Supervised cross-modal hashing methods learn the hash functions by utilizing supervised information. Representative supervised cross-modal hashing methods include cross view hashing (CVH) [7], semantic correlation maximization (SCM) [8], semantics-preserving hashing (SePH) [9], linear subspace ranking

hashing (LSRH) [30] and asymmetric discrete cross-modal hashing (ADCH) [23]. CVH proposes mapping similarity samples from different views into similar hash codes based on the inspiration of single-view spectral hash. SCM learns two hash functions by integrating semantic labels into the learning procedure. SePH minimizes the Kullback–Leibler divergence of the probability distributions of original data and generated hash codes. In order to avoid the binary quantization error, LSRH uses softmax function instead of the symbolic function. ADCH leverages the collective matrix factorization technique to learn the common latent representations while preserving not only the cross-correlation from different modalities but also the semantic similarity. Due to the excellent performance of deep learning in image classification [31–34] and image retrieval [35–37], more and more deep cross-modal methods are proposed, such as deep cross-modal hashing (DCMH) [11], deep visual-semantic hashing (DVSH) [10], pairwise relationship guided deep hashing (PRDH) [38], triplet-based deep hashing (TDH) [19] and self-supervised adversarial hashing (SSAH) [20]. DCMH trains the feature extraction module and hash learning module jointly in an end-to-end framework. DVSH utilizes convolutional neural networks (CNNs) and long short-term memory (LSTM) to separately learn unified binary codes for each modality. PRDH integrates different types of pairwise constraints to encourage the similarities of the hash codes from an intra-modal view and an inter-modal view, respectively. TDH proposes a deep hashing method based on triplet to make up for the single similarity of a cross-modal pair. SSAH builds a label network as the supervised network of other modalities networks.

Typically, deep cross-modal methods get better retrieval results. The extracted features reflect the information of each modality data better, and the feature extraction module and hash learning module are trained at the same time. The training of deep neural networks often consumes a great deal of time. To reduce the training time, most deep cross-modal methods use a small set randomly selected from the original data as the training set. Hence, these methods can't fully utilize dataset and supervised information, and can't deal with large-scale retrieval well. So we propose an asymmetric cross-modal hashing method to solve these problems.

3. The proposed approach

3.1. Notations and problem definition

We use bold uppercase letters like $\mathbf{X}$ to represent matrices, and bold lowercase letters like $\mathbf{y}$ to represent vectors. $\mathbf{F}_{i*}$ denotes the ith row of $\mathbf{F}$, and $\mathbf{G}^T$ denotes the transpose of $\mathbf{G}$. $\|\cdot\|_F$ is the Frobenius norm of a matrix, and $sign(x)$ is a symbolic function. Suppose that there are N training instances $\mathbf{O} = \{o_i\}_{i=1}^N$. Each instance has features from image modality and text modality and corresponding label. $\mathbf{x}$ and $\mathbf{y}$ represent image modality and text modality, respectively. $\mathbf{L} \in \{0, 1\}^{N\times c}$ denotes multi-label matrix, and c is the number of class. $\mathbf{S} \in [-1, 1]^{N\times N}$ is a semantic similarity matrix which reflects the level of similarity between instances. The goal of our method is to learn two deep hash functions $h_x(\mathbf{x}_q) \in \{-1, +1\}^r$ and $h_y(\mathbf{y}_q) \in \{-1, +1\}^r$ respectively for image modality x and text modality y, where r is the length of the hash codes. So we can generate binary hash codes for any new query instance.

3.2. Deep architecture

Considering the favorable ability of feature expression, two deep neural networks are established to train image modality and text modality, as shown in Fig. 1.

For the image modality network, we use the deep neural network named CNN-F [39] which is trained on the ImageNet dataset [40] to extract features. The original CNN-F model consists of five convolution layers ($conv$) and three fully-connected layers (fc). The $fc8$ layer is replaced with a fully-connected layer with R hidden nodes, $R = c + r$. c nodes are used to predict labels through features and r nodes are used to learn hash codes. From $conv1$ to $fc7$, we use *relu* as the activation function of each layer. For the last layer, *tanh* is used as the activation function in the hash module, and *sigmoid* is used as the activation function in the label module.

For the text modality network, text data are denoted as vectors by using bag-of-words(BOW). A deep feed-forward neural network which consists of three fully-connected layers is introduced to encode text data. The BOW is used as the input to the text modality network. The first two layers with 4096 hidden nodes, and the last layer has R hidden nodes in which r nodes are used for encoding and c nodes are used for label prediction.

3.3. Asymmetric loss

3.3.1. Semantic similarity information

The level of similarity between instances is simply divided into similarity or dissimilarity in most deep cross-modal methods. In order to reflect the level of similarity between instances, we follow SCM [8] in which the similarity between the ith instance and the jth instance is defined as follow:

$$\mathbf{S}_{ij} = 2\frac{\mathbf{L}_{i*}\mathbf{L}_{j*}^T}{\|\mathbf{L}_{i*}\|_F\|\mathbf{L}_{j*}\|_F} - 1 \tag{1}$$

Thus, the semantic similarity matrix $\mathbf{S} \in [-1, 1]^{N\times N}$ can be obtained.

3.3.2. Asymmetric loss

Most deep symmetric cross-modal hashing approaches randomly sample a small part of data from the original dataset as training set since training deep neural network is time-consuming. However, when dealing with large-scale data, it often leads to the difficulty of training sets containing fewer categories of instances, resulting in insufficient retrieval performance. Inspired by asymmetric deep supervised hashing (ADSH) [4] in image retrieval, we propose an asymmetric loss function. All instances in the dataset constitute the training set $\mathbf{O} = \{o_i\}_{i=1}^N$. In order to make the training practicable, we can randomly sample m instances from the training set $\mathbf{O}$ to construct the query instances $\mathbf{O}^\Omega$, where $m \ll N$. More specifically, we set $\boldsymbol{\Phi} = \mathbf{O}^\Omega$ where $\mathbf{O}^\Omega$ denotes the instances indexed by Ω. Here, we use $\Omega = \{i_1, i_2, \ldots, i_m\}$ to denote the indices of the sampled query instances and $\Gamma = \{j_1, j_2, \ldots, j_n\}$ where $n = N - m$ to denote the indices of the database instances. $\boldsymbol{\Psi} = \mathbf{O}^\Gamma$ represents the database instances. Hash codes of each modality of instances in $\boldsymbol{\Phi}$ are generated by deep neural networks. For the ith instances, hash codes for image modality and text modality are obtained as follows

$$\begin{cases} \mathbf{U}_i = F(\mathbf{x}_i; \theta_x) \\ \mathbf{V}_i = G(\mathbf{y}_i; \theta_y) \end{cases} \tag{2}$$

where $\mathbf{U} \in [-1, +1]^{m\times r}$ and $\mathbf{V} \in [-1, +1]^{m\times r}$ are the hash codes of image modality and text modality of query instances, which are generated by deep neural networks. F and G denote image network and text network, respectively. θ_x and θ_y are parameters of image network and text network.

For database instances $\boldsymbol{\Psi}$, hash codes of each modality of instances are learned by hand-crafted feature matrices. We can

get the binary codes by solving the following equation.

$$\min_{\mathbf{H}_i,\mathbf{B}^\Gamma} \sum_{i=1}^{2} \|\mathbf{F}_i - \mathbf{B}^\Gamma \mathbf{H}_i^T\|_F^2 + \|\mathbf{H}_i^T\|_F^2 \quad s.t. \quad \mathbf{B}^\Gamma \in \{-1,+1\}^{n\times r} \tag{3}$$

where $\mathbf{F}_1$ is the hand-crafted feature matrix of image modality and $\mathbf{F}_2$ is that of text modality. $\mathbf{B}$ is the binary code matrix of the whole data and $\mathbf{B}^\Gamma$ is that of database instances. When Eq. (3) achieves the minimum value, the binary code obtains matrix the optimal solution.

When using image modality instances in $\boldsymbol{\Phi}$ to query text modality instances in $\boldsymbol{\Psi}$, the hash codes of image modality instances should keep the similarity which preserved in semantic similarity matrix with the hash codes of text modality instances. In addition, the hash codes of image modality instances should also keep the similarity with the hash codes of text modality instances in $\boldsymbol{\Phi}$. Furthermore, in order to ensure that the generated features can represent the training data accurately, the features are not only used to generate hash codes but also used to make classification which can ensure that the obtained features are discriminative. So we can obtain the loss function of hash learning as:

$$\begin{aligned} J_1 = &\sum_{i=1}^{m}\sum_{j=1}^{n}(\mathbf{U}_{i*}(\mathbf{B}_{j*}^\Gamma)^T - r\dot{\mathbf{S}}_{ij})^2 + \|\mathbf{F}_2 - \mathbf{B}^\Gamma\mathbf{H}_2^T\|_F^2 \\ &+ \|\mathbf{H}_2^T\|_F^2 + \alpha\sum_{i=1}^{m}\sum_{k=1}^{m}(\mathbf{U}_{i*}\mathbf{V}_{k*}^T - r\ddot{\mathbf{S}}_{ik})^2 \\ &+ \beta\|\mathbf{U} - \mathbf{B}^\Omega\|_F^2 + \gamma\|\hat{\mathbf{L}}_x - \mathbf{L}^\Omega\|_F^2 \\ &s.t. \quad \mathbf{B} \in \{-1,+1\}^{N\times r} \end{aligned} \tag{4}$$

where α, β and γ are three hyper-parameters, $\mathbf{B}^\Omega \in \{-1,+1\}^{m\times r}$ is the binary code matrix of the query instances $\boldsymbol{\Phi}$. $\dot{\mathbf{S}} \in [-1,1]^{m\times n}$ is the semantic similarity matrix between query instances $\boldsymbol{\Phi}$ and database instances $\boldsymbol{\Psi}$. $\ddot{\mathbf{S}} \in [-1,1]^{m\times m}$ is between query instances $\boldsymbol{\Phi}$. $\hat{\mathbf{L}}_x$ is the label matrix predicted by image network, and $\mathbf{L}^\Omega$ is the label matrix of query instances $\boldsymbol{\Phi}$.

By optimizing the first four terms in Eq. (4), the hash codes satisfy the semantic similarity information of the original space. The fifth term is introduced to reduce the binary quantization error. And the last term is classification loss, which is used to ensure that the features extracted by image network are discriminative. Similarly, When using text modality instances in $\boldsymbol{\Phi}$ to query image modality instances in $\boldsymbol{\Psi}$, we can also obtain a loss function as:

$$\begin{aligned} J_2 = &\sum_{i=1}^{m}\sum_{j=1}^{n}(\mathbf{V}_{i*}(\mathbf{B}_{j*}^\Gamma)^T - r\dot{\mathbf{S}}_{ij})^2 + \|\mathbf{F}_1 - \mathbf{B}^\Gamma\mathbf{H}_1^T\|_F^2 \\ &+ \|\mathbf{H}_1^T\|_F^2 + \alpha\sum_{i=1}^{m}\sum_{k=1}^{m}(\mathbf{V}_{i*}\mathbf{U}_{k*}^T - r\ddot{\mathbf{S}}_{ik})^2 \\ &+ \beta\|\mathbf{V} - \mathbf{B}^\Omega\|_F^2 + \gamma\|\hat{\mathbf{L}}_y - \mathbf{L}^\Omega\|_F^2 \\ &s.t. \quad \mathbf{B} \in \{-1,+1\}^{N\times r} \end{aligned} \tag{5}$$

Thus, the asymmetric loss is defined as:

$$J_{asy} = J_1 + J_2 \tag{6}$$

It is evident that optimizing the above loss function will preserve the semantic correlation among instances from different modalities.

3.4. Discrimination loss

Most existing deep cross-modal hashing methods do not guarantee whether the hash codes generated by hash functions are discriminative or not. We hope that the generated hash codes of instances which classified into different categories can be clearly distinguished. Inspired by some Autoencoder methods [41–43], in order to get discriminative high-quality hash codes, we hope that generated hash codes can get the corresponding labels only through simple linear transformation. So DDASH learns a mapping matrix $\mathbf{M}$ that maps hash code matrix to label matrix, i.e. $\mathbf{BM} \approx \mathbf{L}$, where $\mathbf{M} \in \mathbb{R}^{r\times c}$. In order to facilitate the calculation in the following discrete optimization process, we rewrite the formula as $\mathbf{B} \approx \mathbf{LW}$, where $\mathbf{W} = \mathbf{M}^{-1}$. So we contain the discrimination loss as:

$$J_{dis} = \|\mathbf{B} - \mathbf{LW}\|_F^2 + \|\mathbf{W}\|_F^2 \tag{7}$$

3.5. Objection function and learning algorithm

As mention above, the overall objective function can be defined as follow:

$$\begin{aligned} \min_{\theta_x,\theta_y,\mathbf{B},\mathbf{W},\mathbf{H}_i^T} J &= J_{asy} + \eta J_{dis} \\ &= \|\mathbf{U}(\mathbf{B}^\Gamma)^T - r\dot{\mathbf{S}}\|_F^2 + \|\mathbf{V}(\mathbf{B}^\Gamma)^T - r\dot{\mathbf{S}}\|_F^2 \\ &+ \|\mathbf{F}_1 - \mathbf{B}^\Gamma\mathbf{H}_1^T\|_F^2 + \|\mathbf{F}_2 - \mathbf{B}^\Gamma\mathbf{H}_2^T\|_F^2 \\ &+ \|\mathbf{H}_1^T\|_F^2 + \|\mathbf{H}_2^T\|_F^2 + 2\alpha\|\mathbf{UV}^T - r\ddot{\mathbf{S}}\|_F^2 \\ &+ \beta(\|\mathbf{U} - \mathbf{B}^\Omega\|_F^2 + \|\mathbf{V} - \mathbf{B}^\Omega\|_F^2) \\ &+ \gamma(\|\hat{\mathbf{L}}_x - \mathbf{L}^\Omega\|_F^2 + \|\hat{\mathbf{L}}_y - \mathbf{L}^\Omega\|_F^2) \\ &+ \eta(\|\mathbf{B} - \mathbf{LW}\|_F^2 + \|\mathbf{W}\|_F^2) \\ s.t. \quad &\mathbf{B} \in \{-1,+1\}^{N\times r} \end{aligned} \tag{8}$$

where η is a hyper-parameter. This is the final objective function of our DSSAH for learning. J should be minimized. By optimizing Eq. (8), we can generate the binary codes of any new instances. We learn θ_x and θ_y by using Stochastic Gradient Descent (SGD) following alternated learning strategy. Since $\mathbf{B}$ is discrete, it is not convex and hard to solve. So we use discrete optimization algorithm. The iterative optimization steps of our model are briefly summarized in Algorithm 1. The detailed derivation process is described below.

3.5.1. θ-Step

Update θ_x. With other parameters fixed, the Eq. (8) can be rewritten as

$$\begin{aligned} \min_{\theta_x} \quad &\|\mathbf{U}(\mathbf{B}^\Gamma)^T - r\dot{\mathbf{S}}\|_F^2 + \alpha\|\mathbf{UV}^T - r\ddot{\mathbf{S}}\|_F^2 \\ &+ \beta\|\mathbf{U} - \mathbf{B}^\Omega\|_F^2 + \gamma\|\hat{\mathbf{L}}_x - \mathbf{L}^\Omega\|_F^2 \end{aligned} \tag{9}$$

where the parameters θ_x can be learned by utilizing the stochastic gradient descent (SGD) with the back-propagation (BP) algorithm.

Update θ_y. With other parameters fixed, we rewrite the Eq. (8) as follows

$$\begin{aligned} \min_{\theta_y} \quad &\|\mathbf{V}(\mathbf{B}^\Gamma)^T - r\dot{\mathbf{S}}\|_F^2 + \alpha\|\mathbf{VU}^T - r\ddot{\mathbf{S}}\|_F^2 \\ &+ \beta\|\mathbf{V} - \mathbf{B}^\Omega\|_F^2 + \gamma\|\hat{\mathbf{L}}_y - \mathbf{L}^\Omega\|_F^2 \end{aligned} \tag{10}$$

where the parameters θ_y can be learned the same as θ_x.

3.5.2. $\mathbf{H}_i$-Step

With other parameters fixed, the Eq. (8) is reduced as

$$\min_{\mathbf{H}_i} \sum_{i=1}^{2} \|\mathbf{F}_i - \mathbf{B}^\Gamma\mathbf{H}_i^T\|_F^2 + \|\mathbf{H}_i^T\|_F^2 \quad s.t. \quad \mathbf{B}^\Gamma \in \{-1,+1\}^{n\times r}. \tag{11}$$

Then, we calculate the derivation of Eq. (11) with respect to $\mathbf{H}_i$ and the closed-form solution of $\mathbf{H}_i$ can be obtained by setting the derivation as 0

$$\mathbf{H}_i = \mathbf{F}_i^T \mathbf{B}^\Gamma ((\mathbf{B}^\Gamma)^T \mathbf{B}^\Gamma + \mathbf{I})^{-1}. \tag{12}$$

where $\mathbf{I}$ is an identity matrix.

3.5.3. B-Step

With other parameters fixed, we rewrite the Eq. (8):

$$\begin{aligned} \min_{\mathbf{B}} \quad & \|\mathbf{U}(\mathbf{B}^\Gamma)^T - r\dot{\mathbf{S}}\|_F^2 + \|\mathbf{V}(\mathbf{B}^\Gamma)^T - r\dot{\mathbf{S}}\|_F^2 \\ & + \|\mathbf{F}_1 - \mathbf{B}^\Gamma \mathbf{H}_1^T\|_F^2 + \|\mathbf{F}_2 - \mathbf{B}^\Gamma \mathbf{H}_2^T\|_F^2 \\ & + \beta(\|\mathbf{U} - \mathbf{B}^\Omega\|_F^2 + \|\mathbf{V} - \mathbf{B}^\Omega\|_F^2) \\ & + \eta\|\mathbf{B} - \mathbf{L}\mathbf{W}\|_F^2 \\ s.t. \quad & \mathbf{B} \in \{-1, +1\}^{N \times r} \end{aligned} \tag{13}$$

Since $\mathbf{B}^\Omega$ and $\mathbf{B}^\Gamma$ represent the hash codes of query instances and database instances respectively, the Eq. (13) can be divided into the following two subproblems to solve the optimal value:

$$\begin{aligned} \min_{\mathbf{B}^\Gamma} \quad & \|\mathbf{U}(\mathbf{B}^\Gamma)^T - r\dot{\mathbf{S}}\|_F^2 + \|\mathbf{V}(\mathbf{B}^\Gamma)^T - r\dot{\mathbf{S}}\|_F^2 \\ & + \|\mathbf{F}_1 - \mathbf{B}^\Gamma \mathbf{H}_1^T\|_F^2 + \|\mathbf{F}_2 - \mathbf{B}^\Gamma \mathbf{H}_2^T\|_F^2 \\ & + \eta\|\mathbf{B}^\Gamma - \mathbf{L}^\Gamma \mathbf{W}\|_F^2 \\ s.t. \quad & \mathbf{B}^\Gamma \in \{-1, +1\}^{n \times r} \end{aligned} \tag{14}$$

$$\begin{aligned} \min_{\mathbf{B}^\Omega} \quad & \beta(\|\mathbf{U} - \mathbf{B}^\Omega\|_F^2 + \|\mathbf{V} - \mathbf{B}^\Omega\|_F^2) \\ & + \eta\|\mathbf{B}^\Omega - \mathbf{L}^\Omega \mathbf{W}\|_F^2 \\ s.t. \quad & \mathbf{B}^\Omega \in \{-1, +1\}^{m \times r} \end{aligned} \tag{15}$$

These both two problems with discrete constraints are NP hard.

For Eq. (14), we introduce the discrete cyclic coordinate descent (DCC) algorithm [44]. We rewrite the first term of Eq. (14) as follows:

$$\begin{aligned} \|\mathbf{U}(\mathbf{B}^\Gamma)^T - r\dot{\mathbf{S}}\|_F^2 = & tr(\mathbf{B}^\Gamma \mathbf{U}^T \mathbf{U}(\mathbf{B}^\Gamma)^T) \\ & + tr(\mathbf{B}^\Gamma \mathbf{Q}_1^T) + const \end{aligned} \tag{16}$$

where $\mathbf{Q}_1 = -2r\dot{\mathbf{S}}^T\mathbf{U}$. *const* is a constant independent of $\mathbf{B}^\Gamma$. Similarly, following four formulas can be obtained.

$$\begin{aligned} \|\mathbf{V}(\mathbf{B}^\Gamma)^T - r\dot{\mathbf{S}}\|_F^2 = & tr(\mathbf{B}^\Gamma \mathbf{V}^T \mathbf{V}(\mathbf{B}^\Gamma)^T) \\ & + tr(\mathbf{B}^\Gamma \mathbf{Q}_2^T) + const \end{aligned} \tag{17}$$

$$\begin{aligned} \|\mathbf{F}_1 - \mathbf{B}^\Gamma \mathbf{H}_1^T\|_F^2 = & tr(\mathbf{B}^\Gamma \mathbf{H}_1^T \mathbf{H}_1(\mathbf{B}^\Gamma)^T) \\ & + tr(\mathbf{B}^\Gamma \mathbf{Q}_3^T) + const \end{aligned} \tag{18}$$

$$\begin{aligned} \|\mathbf{F}_2 - \mathbf{B}^\Gamma \mathbf{H}_2^T\|_F^2 = & tr(\mathbf{B}^\Gamma \mathbf{H}_2^T \mathbf{H}_2(\mathbf{B}^\Gamma)^T) \\ & + tr(\mathbf{B}^\Gamma \mathbf{Q}_4^T) + const \end{aligned} \tag{19}$$

$$\|\mathbf{B}^\Gamma - \mathbf{L}^\Gamma \mathbf{W}\|_F^2 = tr(\mathbf{B}^\Gamma \mathbf{Q}_5^T) + const \tag{20}$$

where $\mathbf{Q}_2 = -2r\dot{\mathbf{S}}^T\mathbf{V}$, $\mathbf{Q}_3 = -2\mathbf{F}_1\mathbf{H}_1$, $\mathbf{Q}_4 = -2\mathbf{F}_2\mathbf{H}_2$ and $\mathbf{Q}_5 = -2\mathbf{L}^\Gamma\mathbf{W}$. Hence, the Eq. (14) can be rewritten as follow:

$$\begin{aligned} \min_{\mathbf{B}^\Gamma} \quad & tr(\mathbf{B}^\Gamma \mathbf{U}^T \mathbf{U}(\mathbf{B}^\Gamma)^T) + tr(\mathbf{B}^\Gamma \mathbf{V}^T \mathbf{V}(\mathbf{B}^\Gamma)^T) \\ & + tr(\mathbf{B}^\Gamma \mathbf{H}_1^T \mathbf{H}_1(\mathbf{B}^\Gamma)^T) + tr(\mathbf{B}^\Gamma \mathbf{H}_2^T \mathbf{H}_2(\mathbf{B}^\Gamma)^T) \\ & + tr(\mathbf{B}^\Gamma \mathbf{Q}_1^T) + tr(\mathbf{B}^\Gamma \mathbf{Q}_2^T) + tr(\mathbf{B}^\Gamma \mathbf{Q}_3^T) \\ & + tr(\mathbf{B}^\Gamma \mathbf{Q}_4^T) + \eta tr(\mathbf{B}^\Gamma \mathbf{Q}_5^T) + const \\ s.t. \quad & \mathbf{B}^\Gamma \in \{-1, +1\}^{n \times r} \end{aligned} \tag{21}$$

Then, we can update $\mathbf{B}^\Gamma$ bit by bit. Each time we update one column of $\mathbf{B}^\Gamma$ with other columns fixed. Let $\mathbf{B}_{*k}^\Gamma$, $\mathbf{U}_{*k}$, $\mathbf{V}_{*k}$, $(\mathbf{H}_1)_{*k}$ and $(\mathbf{H}_2)_{*k}$ denote the kth column of $\mathbf{B}^\Gamma$, $\mathbf{U}$, $\mathbf{V}$, $\mathbf{H}_1$ and $\mathbf{H}_2$, respectively. Let $(\mathbf{Q}_1)_{*k}$, $(\mathbf{Q}_2)_{*k}$, $(\mathbf{Q}_3)_{*k}$, $(\mathbf{Q}_4)_{*k}$ and $(\mathbf{Q}_5)_{*k}$ denote the kth column of $\mathbf{Q}_1$, $\mathbf{Q}_2$, $\mathbf{Q}_3$, $\mathbf{Q}_4$ and $\mathbf{Q}_5$, respectively. Let $\hat{\mathbf{B}^\Gamma}_k$ denote the matrix $\mathbf{B}^\Gamma$ excluding $\mathbf{B}_{*k}^\Gamma$, $\hat{\mathbf{U}}_k$ denote the matrix $\mathbf{U}$ excluding $\mathbf{U}_{*k}$, $\hat{\mathbf{V}}_k$ denote the matrix $\mathbf{V}$ excluding $\mathbf{V}_{*k}$, $(\hat{\mathbf{H}}_1)_k$ denote the matrix $\mathbf{H}_1$ excluding $(\mathbf{H}_1)_{*k}$ and $(\hat{\mathbf{H}}_2)_k$ denote the matrix $\mathbf{H}_2$ excluding $(\mathbf{H}_2)_{*k}$. To optimize $\mathbf{B}_{*k}^\Gamma$, we can get the objective function:

$$\begin{aligned} J(\mathbf{B}_{*k}^\Gamma) = & tr(\mathbf{B}_{*k}^\Gamma[2\mathbf{U}_{*k}^T \hat{\mathbf{U}}_k(\hat{\mathbf{B}^\Gamma}_k)^T + 2\mathbf{V}_{*k}^T \hat{\mathbf{V}}_k(\hat{\mathbf{B}^\Gamma}_k)^T \\ & + 2(\mathbf{H}_1)_{*k}^T(\hat{\mathbf{H}}_1)_k(\hat{\mathbf{B}^\Gamma}_k)^T + 2(\mathbf{H}_2)_{*k}^T(\hat{\mathbf{H}}_2)_k(\hat{\mathbf{B}^\Gamma}_k)^T \\ & + (\mathbf{Q}_1)_{*k}^T + (\mathbf{Q}_2)_{*k}^T + (\mathbf{Q}_3)_{*k}^T \\ & + (\mathbf{Q}_4)_{*k}^T + \eta(\mathbf{Q}_5)_{*k}^T]) + const \end{aligned} \tag{22}$$

Then, we need to deal with the following problem:

$$\begin{aligned} \min_{\mathbf{B}_{*k}^\Gamma} J(\mathbf{B}_{*k}^\Gamma) = & tr(\mathbf{B}_{*k}^\Gamma[2\mathbf{U}_{*k}^T \hat{\mathbf{U}}_k(\hat{\mathbf{B}^\Gamma}_k)^T + 2\mathbf{V}_{*k}^T \hat{\mathbf{V}}_k(\hat{\mathbf{B}^\Gamma}_k)^T \\ & + 2(\mathbf{H}_1)_{*k}^T(\hat{\mathbf{H}}_1)_k(\hat{\mathbf{B}^\Gamma}_k)^T + 2(\mathbf{H}_2)_{*k}^T(\hat{\mathbf{H}}_2)_k(\hat{\mathbf{B}^\Gamma}_k)^T \\ & + (\mathbf{Q}_1)_{*k}^T + (\mathbf{Q}_2)_{*k}^T + (\mathbf{Q}_3)_{*k}^T \\ & + (\mathbf{Q}_4)_{*k}^T + \eta(\mathbf{Q}_5)_{*k}^T]) \\ s.t. \quad & \mathbf{B}_{*k}^\Gamma \in \{-1, +1\}^n \end{aligned} \tag{23}$$

So, we can get the optimal solution of Eq. (23) as follow:

$$\begin{aligned} \mathbf{B}_{*k}^\Gamma = -sign(2\hat{\mathbf{B}^\Gamma}_k[\hat{\mathbf{U}}_k^T \mathbf{U}_{*k} + \hat{\mathbf{V}}_k^T \mathbf{V}_{*k} + (\hat{\mathbf{H}}_1)_k^T(\mathbf{H}_1)_{*k} \\ + (\hat{\mathbf{H}}_2)_k^T(\mathbf{H}_2)_{*k}] + (\mathbf{Q}_1)_{*k} + (\mathbf{Q}_2)_{*k} \\ + (\mathbf{Q}_3)_{*k} + (\mathbf{Q}_4)_{*k} + \eta(\mathbf{Q}_5)_{*k}) \end{aligned} \tag{24}$$

For Eq. (15), we rewrite the problem in the same way as Eq. (14). Hence, the Eq. (15) can be rewritten as follow:

$$\begin{aligned} \min_{\mathbf{B}^\Omega} \quad & tr(\mathbf{B}^\Omega[-2\beta\mathbf{U}^T - 2\beta\mathbf{V}^T - 2\eta\mathbf{W}^T(\mathbf{L}^\Omega)^T]) \\ s.t. \quad & \mathbf{B}^\Omega \in \{-1, +1\}^{m \times r} \end{aligned} \tag{25}$$

Then, we can get the optimal solution of Eq. (25) as follow:

$$\mathbf{B}^\Omega = sign(2\beta\mathbf{U} + 2\beta\mathbf{V} + 2\eta\mathbf{L}^\Omega\mathbf{W}) \tag{26}$$

3.5.4. W-Step

With other parameters fixed, the objective function Eq. (8) are written as follow:

$$\begin{aligned} \min_{\mathbf{W}} \quad & \eta(\|\mathbf{B} - \mathbf{L}\mathbf{W}\|_F^2 + \|\mathbf{W}\|_F^2) \\ s.t. \quad & \mathbf{W} \in \mathbb{R}^{c \times r} \end{aligned} \tag{27}$$

Then, we calculate the derivation of Eq. (27) with respect to $\mathbf{W}$ and the closed-form solution of $\mathbf{W}$ can be obtained by setting the derivation as 0

$$\mathbf{W} = \eta(\mathbf{L}^T\mathbf{L} + \mathbf{I})^{-1}\mathbf{L}^T\mathbf{B} \tag{28}$$

3.6. Out-of-sample extension

After training DDASH, we can use the deep neural networks to generate binary codes for query instances including unseen query instances during training. Specifically, we use the following equations to generate binary codes for instance o_q:

- **for image modality**

$$u_q = h(\mathbf{x}_q; \theta_x) = sign(F(\mathbf{x}_q; \theta_x)) \tag{29}$$

- **for text modality**

$$v_q = h(\mathbf{y}_q; \theta_y) = sign(G(\mathbf{y}_q; \theta_y)) \tag{30}$$

Algorithm 1 iterative optimization steps of our model

Input:
Image set **X**, text set **Y**
label matrix **L**, Similarity matrix **S**
Ouput:
Parameters θ_x and θ_y of the deep neural networks, and binary code matrix **B**.
Initialization
Initialize parameters θ_x, θ_y, α, β, γ, η.
mini-batch size N_x, N_y, maximum iteration number T_{max}, image network iteration number T_x and text network iteration number T_y.
for iter $= 1,2,\cdots T_{max}$ **do**
  Randomly generate index set Ω and Γ. Divide the data into Φ and Ψ. Calculate $\dot{\mathbf{S}}$ and $\ddot{\mathbf{S}}$ using Eq. (1).
  for iter $= 1,2,\cdots T_x$ **do**
    Randomly sample N_x data from Φ^x to construct a mini-batch.
    Update parameter θ_x using Eq. (9).
  end for
  for iter $= 1,2,\cdots T_y$ **do**
    Randomly sample N_y data from Φ^y to construct a mini-batch.
    Update parameter θ_y using Eq. (10).
  end for
  Update $\mathbf{H}_i$ according to update rule in Eq. (12).
  for $k = 1 \rightarrow r$ **do**
    Update $\mathbf{B}^{\Gamma}_{*k}$ according to update rule in Eq. (24).
  end for
  Update $\mathbf{B}^{\Omega}$ according to update rule in Eq. (26).
  Update **W** according to update rule in Eq. (28).
end for

4. Experiments

We evaluated DDASH on three common datasets: ***MIRFLICKR-25K***[45], ***NUS-WIDE*** [46], and ***Wiki*** [47]

4.1. Datasets

- ***MIRFLICKR-25K***[1][45]: This dataset contains 25,000 instances collected from Flickr. Each instance corresponds to a multi-labeled image associated with several text tags. For a fair comparison, we follow the setting of DCMH [11] to select instances which have at least 20 textual tags. For each instance, the image is represented by a 512-dimensional SIFT feature vector, while the text is represented as a 1386-dimensional bag-of-words vector. Each instance is annotated with one or more labels, from a total of 24 semantic labels.
- ***NUS-WIDE***[2][46]: This dataset consists of 260,648 web images with 81 concept labels and associated textual tags. We also follow the setting of DCMH [11] to select 195,834 instances which belong to the 21 most frequent concept labels as our dataset. For each instance, the image is a 500-dimensional bag-of-visual words(BOVW) vector. Furthermore, the text is represented as a 1000-dimensional bag-of-words vector.
- ***Wiki***[3][47]: This dataset is made up of 2866 image–text pairs collected from Wikipedia and every paired instances is classified into one of 10 categories. Each image is represented by a 128-dimensional SIFT feature vector and a 10-dimensional topic vector is given to describe the text.

4.2. Evaluation protocol

Cross-modal retrieval aims to take data points of one modality as the query and obtain relevant points of other modality, such as using images to retrieve text, and vice versa. We adopt two widely used evaluation methods in retrieval protocols: **hamming ranking** and **hash lookup** to evaluate our method DDASH.

According to the hamming distance between the given query instances, the hamming ranking protocol ranks the instances in the retrieval dataset in an increasing order. Mean average precision(MAP) [48] is used widely to measure the accuracy of Hamming ranking protocol. In order to calculate the MAP value, we first introduce the Average Precision(AP). For ith query instance, $AP(i)$ is defined as:

$$AP(i) = \frac{1}{N}\sum_{r=1}^{R} p(r)rel(r) \tag{31}$$

where N is the number of relevant instances in the retrieved database. $p(r)$ is the precision of the top r returned instances. If the rth returned result is relevant to the query instance, $rel(r) = 1$; otherwise $rel(r) = 0$. MAP is obtained by calculating the average AP of all queries.

The hash lookup protocol builds a lookup table and returns all instances within a certain Hamming radius of the query instance. The precision–recall curve is widely used to measure the accuracy of the hash lookup protocol [48].

4.3. Baselines and settings

We compare DDASH with seven state-of-the-art cross-modal hashing methods, including CVH [7], STMH [16], SCM [8], SePH [9], ADCH [23], DCMH [11] and SSAH [20]. The first five methods are shallow-structure-based methods, DCMH and SSAH are deep-structure-based methods, in which SSAH introduces generative adversarial network (GAN).

For our method, we build the training set and test set. For MIRFLICKR-25K and NUS-WIDE, we randomly sample 2000 instances as test set and remaining instances as training set and retrieval set. For Wiki, 600 instances are randomly sampled as test set, and remaining instances as training set and retrieval set.

In DDASH, for MIRFLICKR-25K and NUS-WIDE, 3000 instances are randomly sampled as query instances and remaining instances of training set as database instances. For Wiki, we randomly sample 1800 instances as query instances and remaining instances of training set as database instances. Hyper-parameters named α, β, γ and η are selected based on a validation set, and we set $\alpha = 100$, $\beta = 200$, $\gamma = 300$, and $\eta = 100$. We will further demonstrate the insensitivity of these parameters in the following subsection. The image network is initialized by the CNN-F network [39] which is pre-trained on the ImageNet dataset [40], and the text network is randomly initialized. The input of image network is the raw pixels of images, and the BOW vectors are input of text network. The mini-batch size is set as 64 and the number of iterations is 100. Learning rates of the image network are chosen from $10^{-5.5}$ to 10^{-9}, and the learning rates of the text network are from $10^{-4.5}$ to 10^{-9}. We update the image network and the text network alternately 20 times, and then update $\mathbf{H}_i$, **B** and **W**. In this paper, we construct the deep architectures using pytorch [49] and all experiments are run for three times, and the mean performance is reported.

[1] http://press.liacs.nl/mirflickr/
[2] http://lms.comp.nus.edu.sg/research/NUS-WIDE.htm
[3] http://www.svcl.ucsd.edu/projects/crossmodal/

Table 1
MAP. The best accuracy is shown in boldface. The baselines are based on CNN-F features.

Task	Method	MIRFLICKR-25K			NUS-WIDE			Wiki		
		16 bits	32 bits	64 bits	16 bits	32 bits	64 bits	16 bits	32 bits	64 bits
Image-query-Text	CVH [7]	0.557	0.554	0.554	0.374	0.366	0.361	0.178	0.164	0.153
	STMH [16]	0.613	0.621	0.627	0.471	0.486	0.494	0.243	0.246	0.248
	SCM [8]	0.671	0.682	0.685	0.540	0.548	0.555	0.234	0.241	0.244
	SePH [9]	0.712	0.719	0.723	0.603	0.613	0.621	0.283	0.300	0.304
	ADCH [23]	0.713	0.719	0.723	0.613	0.628	0.634	0.321	0.342	**0.351**
	DCMH [11]	0.741	0.746	0.749	0.590	0.603	0.609	0.264	0.269	0.279
	SSAH [20]	0.782	0.790	0.800	0.642	0.636	0.639	0.332	0.335	0.339
	DDASH	**0.830**	**0.843**	**0.850**	**0.735**	**0.739**	**0.743**	**0.341**	**0.345**	0.344
Text-query-Image	CVH [7]	0.574	0.571	0.571	0.361	0.349	0.339	0.119	0.115	0.113
	STMH [16]	0.607	0.615	0.621	0.447	0.467	0.478	0.595	0.615	0.6269
	SCM [8]	0.693	0.701	0.706	0.534	0.541	0.548	0.226	0.246	0.248
	SePH [9]	0.721	0.726	0.731	0.598	0.602	0.610	0.631	0.655	0.664
	ADCH [23]	0.798	0.804	0.813	0.710	0.708	0.728	0.621	0.642	0.667
	DCMH [11]	0.782	0.790	0.793	0.638	0.651	0.657	0.621	0.628	0.638
	SSAH [20]	0.791	0.795	0.803	0.669	0.662	0.666	0.627	0.628	0.632
	DDASH	**0.835**	**0.846**	**0.853**	**0.731**	**0.736**	**0.741**	**0.658**	**0.664**	**0.674**

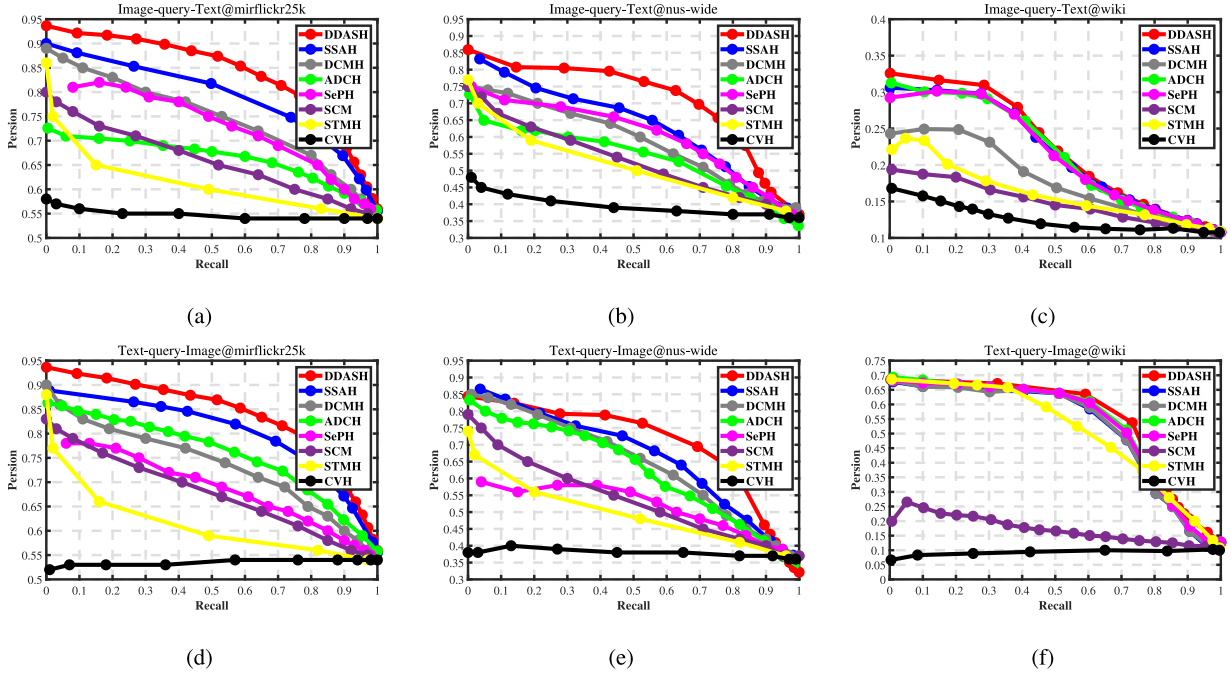


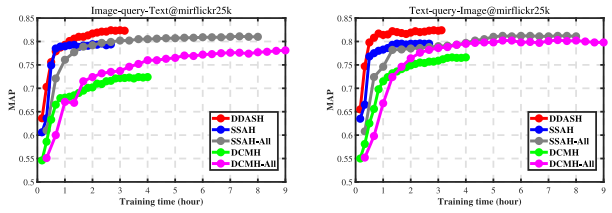
Fig. 2. The precision–recall curves of cross-modal retrieval on MIRFLICKR-25K, NUS-WIDE and Wiki. The baselines are based on CNN-F features. The code length is 16 bits. (a), (b), (c) Image-query-Text. (d), (e), (f) Text-query-Image.

4.4. Performance comparisons and discussions

4.4.1. Hamming ranking

The MAP results obtained by different methods on the MIRFLICKR-25K, NUS-WIDE and Wiki datasets are reported in Table 1. "Image-query-Text" denotes that the query is image and the database is text-based, and "Text-query-Image" denotes that the query is text and the database is image-based. We can find that in most cases the deep-structure-based methods can outperform the shallow-structure-based methods. Furthermore, it is easy to observe that DDASH achieves a remarkable improvement in MAP compared with all the other baselines, including deep-structure-based baselines and shallow-structure-based hashing baselines.

On the large-scale datasets, because of making full use of the supervised information of all the data, DDASH achieves the best performance compared with other baselines. Specifically, on the MIRFLICKR-25K dataset, in comparison to the shallow-structure-based methods CVH, STMH, SCM, SePH and ADCH. DDASH achieves 10%–20% improvement in MAP. Specifically, DDASH obtains at least 83.0% (Image-query-Text) and 82.4% (Text-query-Image) MAP scores and reaches as high as 85.0% and 83.9% when the code length is 64 bits. Compared with the deep-structure-based methods DCMH and SSAH, DDASH also has a improvement. When using image to query text, the presented approach gains about more than 5% enhancement. When using text to query image, the presented approach also gains about more than 3% enhancement. On the NUS-WIDE dataset, in regard to shallow-structure-based methods, DDASH achieves absolute more than a 10% increase on MAP. Compared with DCMH and SSAH, it can be seen that DDASH can achieve more than a 5% increase. In particular, the MAP of Image-query-Text has increased by nearly 10%. On the small-scale dataset, although Wiki has only 2866 instances and all baselines can make full use of the data information, DDASH also achieves the best performance in most cases with different values of the code length. Especially when using text to query image, DDASH achieves more than a 3% increase.



(a) (b)

Fig. 3. Training time on MIRFLICKR-25K. The code length is 16 bits.

4.4.2. Hash lookup

In the hash lookup protocol, for a given Hamming radius, we calculate the precision and recall based on the returned instances and plot the precision–recall curve, as shown in Fig. 2. In Fig. 2, we plot precision–recall curves of state-of-the-art cross-modal hashing methods and DDASH on these three common datasets when the length of hash code is 16 bits. From the precision–recall curves of the first two large-scale datasets, we can find that DDASH significantly outperforms other methods. The curves of DDASH on the Wiki dataset are also at the top of the figures.

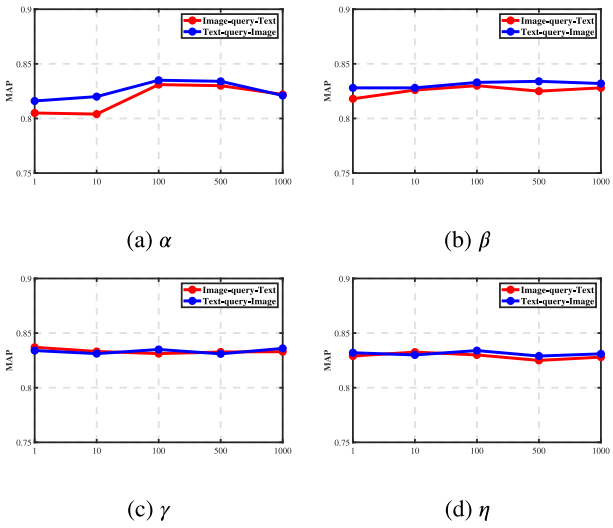
4.4.3. Time complexity

In addition, we compare our DDASH with deep-structure-based baselines by adopting the whole database as the training set on MIRFLICKR-25K dataset. The results are shown in Fig. 3. Here, SSAH and DCMH denote the deep structure-based baselines with 10,000 sampled points for training. SSAH-All and DCMH-All denote the counterparts of the corresponding deep-structure-based baselines which adopt the whole database for training. We can find that for a dataset of 20,000 instances, if the whole database is used for training, it needs more than 6 h for most baselines to converge. Hence, we have to sample a subset of the large-scale datasets for training. From Fig. 3, it is easy to find that to achieve similar accuracy, DDASH is much faster than all the baselines, either with sampled training points or with the whole database. Furthermore, DDASH can achieve a higher accuracy than all baselines with much less time.

There are two reasons why the training time of DDASH is much shorter than that of DCMH and SSAH. Firstly, the training of deep neural networks is typically time-consuming. The time complexity of the deep symmetric supervised hashing method is at least $O(n^2)$ when all database points are used for training, while the DDASH is $O(n)$. Secondly, due to the asymmetric strategy adopted by DDASH, it is not necessary to use all data as the training dataset of the neural networks. We can use a dataset that is much less than the original dataset as the training set of the neural networks, and the remaining data as the database set. The hash codes of the instances in the database set are learned by hand-crafted feature matrices. Therefore, the training time of the model will be further shortened.

4.4.4. Sensitivity to parameters

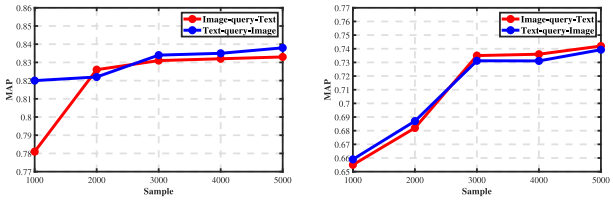
Four hyper-parameters named α, β, γ and η are set in the Eq. (8). So we explore the influence of these four hyper-parameters. The MAP scores under different values of α, β, γ and η are shown in Fig. 4, where the training set is MIRFLICKR-25K dataset and the length of hash code is 16 bits. We set α, β, γ and η with $1 \leq \alpha, \beta, \gamma, \eta \leq 1000$. As can be seen, DDASH is insensitive to the parameters. Specifically, α, β, γ and η have a wide range in $[100, 500]$, $[1, 1000]$, $[1, 1000]$ and $[1, 1000]$, respectively. This relatively demonstrates the robustness and effectiveness of DDASH.



(a) α (b) β

(c) γ (d) η

Fig. 4. A sensitivity analysis of four hyper-parameters. The dataset is MIRFLICKR-25K and the code length is 16 bits.



(a) MIRFLICKR-25K (b) NUS-WIDE

Fig. 5. MAPS on MIRFLICKR-25K and NUS-WIDE by varying the size of query instances. The code length is 16 bits.

4.4.5. Ablation study of DDASH

To go deeper with the effectiveness of DDASH, we design some variants of our proposed approach. We construct our DDASH-I with a symmetric strategy instead of the asymmetric strategy. Like most existing deep cross-modal hashing methods, it learns same deep hash functions for both query and database points. Following the settings of DCMH and SSAH, 10,000 instances are randomly sampled as training set and remaining instances of training set as retrieval set. The DDASH-II utilizes discrete semantic similarity matrix $\mathbf{S}$ which has values -1 or 1 rather than continues matrix whose values vary from -1 to 1. For all datasets, when the image i and text j share at least one common label, $\mathbf{S}_{ij} = 1$. Otherwise, $\mathbf{S}_{ij} = -1$. The variant 3 is recorded as DDASH-III which only considering the asymmetric loss, without discrimination loss. We use 16 bits hash codes to test image-query-text and text-query-image on the MIRFLICKR-25K and NUS-WIDE, and then get the MAP of each variant. Table 2 shows the value of MAP on dataset MIRFLICKR-25k and NUS-WIDE. Compared DDASH-I with DDASH, it is easy to find that our proposed asymmetric hashing has a higher improvement in retrieval performance. Due to the consideration of discrimination loss, the MAP of DDASH is also higher than that of DDASH-III.

4.4.6. Effect of sample size

Moreover, the variations on the MAP results are evaluated when using different numbers of instances as query instances in the hash learning, as shown in Fig. 5. Specifically, we report the results on MIRFLICKR-25K and NUS-WIDE at 16 bits for cross-modal retrieval. Due to the small amount of the Wiki dataset, this dataset was not tested. As can be seen, better retrieval accuracy

Table 2
The MAP Comparison of DDASH Variants on MIRFLICKR-25K and NUS-WIDE. The best accuracy is shown in boldface.

Tasks	Datasets	DDASH-I	DDASH-II	DDASH-III	DDASH
Image-query-Text	MIRFLICKR-25K	0.777	0.821	0.810	**0.830**
	NUS-WIDE	0.650	0.730	0.698	**0.735**
Text-query-Image	MIRFLICKR-25K	0.785	0.835	0.830	**0.835**
	NUS-WIDE	0.649	0.731	0.709	**0.731**

can be achieved with larger number of sampled query instances. Larger the number of sampled query instances, larger the calculation amount. From Fig. 5, we can see that after the number of sampled instances reaches 3000, the rising speed of the MAP slows down. So we sample 3000 instances as query instances in our experiments.

5. Conclusion

In this paper, we propose a novel discriminative deep asymmetric supervised hashing (DDASH) for cross-modal retrieval. Asymmetric hashing only generates hash codes of query instances by deep hash functions, and learns the hash codes of the database instances by hand-crafted matrices. DDASH can solve the problem that existing deep symmetric cross-modal hashing methods cannot fully utilize the supervised information for cases with large-scale databases due to the time-consuming of deep neural network training. In addition, we use the discrete optimization to solve the overall objective function which is NP-hard. It efficiently reduces the binary quantization error and training time. Furthermore, we introduce a sample mapping matrix which maps hash codes to the corresponding labels to ensure that the generated hash codes are discriminative. We also calculate the level of similarity between instances as supervised information to better reflect the semantic similarity between instances, instead of simply dividing them into similar or dissimilar. Experiments on three common open datasets for cross-modal retrieval show that DDASH outperforms the other competitive cross-modal hashing methods. In the future, we plan to further analyze the semantic similarity information and discrete optimization to improve the retrieval performance of our method.

CRediT authorship contribution statement

Haopeng Qiang: Conceptualization, Methodology, Software. **Yuan Wan:** Writing - review & editing. **Ziyi Liu:** Writing - review & editing, Software. **Lun Xiang:** Software, Validation. **Xiaojing Meng:** Visualization, Investigation.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work is supported by the Fundamental Research Funds for the Central Universities, China under Grant 2019ZY232 and 2019IB010. This work is also supported by Excellent Dissertation Cultivation Funds of Wuhan University of Technology, China under Grant 2018YS076.

References

[1] X. Zhang, H. Lai, J. Feng, Attention-aware deep adversarial hashing for cross-modal retrieval, in: Proceedings of the European Conference on Computer Vision, ECCV, 2018, pp. 591–606.
[2] A. Gionis, P. Indyk, R. Motwani, et al., Similarity search in high dimensions via hashing, in: Vldb, Vol. 99, No. 6, 1999, pp. 518–529.
[3] A. Andoni, P. Indyk, Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions, in: 2006 47th Annual IEEE Symposium on Foundations of Computer Science, FOCS'06, IEEE, 2006, pp. 459–468.
[4] Q.-Y. Jiang, W.-J. Li, Asymmetric deep supervised hashing, in: Thirty-Second AAAI Conference on Artificial Intelligence, 2018, pp. 3342–3349.
[5] A. Andoni, I. Razenshteyn, Optimal data-dependent hashing for approximate near neighbors, in: Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing, ACM, 2015, pp. 793–801.
[6] Y. Gong, S. Lazebnik, A. Gordo, F. Perronnin, Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval, IEEE Trans. Pattern Anal. Mach. Intell. 35 (12) (2012) 2916–2929.
[7] S. Kumar, R. Udupa, Learning hash functions for cross-view similarity search, in: Twenty-Second International Joint Conference on Artificial Intelligence, 2011, pp. 1360–1365.
[8] D. Zhang, W.-J. Li, Large-scale supervised multimodal hashing with semantic correlation maximization, in: Twenty-Eighth AAAI Conference on Artificial Intelligence, 2014, pp. 2177–2183.
[9] Z. Lin, G. Ding, M. Hu, J. Wang, Semantics-preserving hashing for cross-view retrieval, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3864–3872.
[10] Y. Cao, M. Long, J. Wang, Q. Yang, P.S. Yu, Deep visual-semantic hashing for cross-modal retrieval, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2016, pp. 1445–1454.
[11] Q.-Y. Jiang, W.-J. Li, Deep cross-modal hashing, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 3232–3240.
[12] J.T. Zhou, H. Zhao, X. Peng, M. Fang, Z. Qin, R.S.M. Goh, Transfer hashing: From shallow to deep, IEEE Trans. Neural Netw. Learn. Syst. 29 (12) (2018) 6191–6201.
[13] P. Hu, D. Peng, X. Wang, Y. Xiang, Multimodal adversarial network for cross-modal retrieval, Knowl.-Based Syst. 180 (2019) 38–50.
[14] L. Zhu, X. Lu, Z. Cheng, J. Li, H. Zhang, Deep collaborative multi-view hashing for large-scale image search, IEEE Trans. Image Process. 29 (2020) 4643–4655.
[15] H. Hotelling, Relations between two sets of variates, in: Breakthroughs in Statistics, Springer, 1992, pp. 162–190.
[16] D. Wang, X. Gao, X. Wang, L. He, Semantic topic multimodal hashing for cross-media retrieval, in: Twenty-Fourth International Joint Conference on Artificial Intelligence, 2015, pp. 3890–3896.
[17] Y. Cao, M. Long, J. Wang, S. Liu, Collective deep quantization for efficient cross-modal retrieval, in: Thirty-First AAAI Conference on Artificial Intelligence, 2017, pp. 3974–3980.
[18] L. Jin, K. Li, Z. Li, F. Xiao, G.-J. Qi, J. Tang, Deep semantic-preserving ordinal hashing for cross-modal similarity search, IEEE Trans. Neural Netw. Learn. Syst. 30 (5) (2018) 1429–1440.
[19] C. Deng, Z. Chen, X. Liu, X. Gao, D. Tao, Triplet-based deep hashing network for cross-modal retrieval, IEEE Trans. Image Process. 27 (8) (2018) 3893–3903.
[20] C. Li, C. Deng, N. Li, W. Liu, X. Gao, D. Tao, Self-supervised adversarial hashing networks for cross-modal retrieval, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4242–4251.
[21] F. Shen, X. Gao, L. Liu, Y. Yang, H.T. Shen, Deep asymmetric pairwise hashing, in: Proceedings of the 25th ACM International Conference on Multimedia, 2017, pp. 1522–1530.
[22] J. Li, B. Zhang, G. Lu, D. Zhang, Dual asymmetric deep hashing learning, IEEE Access 7 (2019) 113372–113384.
[23] X. Luo, P.-F. Zhang, Y. Wu, Z.-D. Chen, H.-J. Huang, X.-S. Xu, Asymmetric discrete cross-modal hashing, in: Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval, 2018, pp. 204–212.

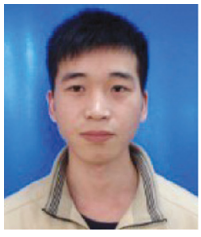
[24] Q.-Y. Jiang, W.-J. Li, Discrete latent factor model for cross-modal hashing, IEEE Trans. Image Process. 28 (7) (2019) 3490–3501.
[25] G. Ding, Y. Guo, J. Zhou, Collective matrix factorization hashing for multimodal data, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 2075–2082.
[26] J. Zhou, G. Ding, Y. Guo, Latent semantic sparse hashing for cross-modal similarity search, in: Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval, ACM, 2014, pp. 415–424.
[27] G. Irie, H. Arai, Y. Taniguchi, Alternating co-quantization for cross-modal hashing, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1886–1894.
[28] J. Zhang, Y. Peng, M. Yuan, Unsupervised generative adversarial cross-modal hashing, in: Thirty-Second AAAI Conference on Artificial Intelligence, 2018, pp. 539–546.
[29] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Advances in Neural Information Processing Systems, 2014, pp. 2672–2680.
[30] K. Li, G.-J. Qi, J. Ye, K.A. Hua, Linear subspace ranking hashing for cross-modal retrieval, IEEE Trans. Pattern Anal. Machine Intell. 39 (9) (2016) 1825–1838.
[31] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in Neural Information Processing Systems, 2012, pp. 1097–1105.
[32] M. Oquab, L. Bottou, I. Laptev, J. Sivic, Learning and transferring mid-level image representations using convolutional neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 1717–1724.
[33] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the inception architecture for computer vision, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2818–2826.
[34] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[35] W.-J. Li, S. Wang, W.-C. Kang, Feature learning based deep supervised hashing with pairwise labels, in: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, 2016, pp. 1711–1717.
[36] H. Liu, R. Wang, S. Shan, X. Chen, Deep supervised hashing for fast image retrieval, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2064–2072.
[37] H. Zhu, M. Long, J. Wang, Y. Cao, Deep hashing network for efficient similarity retrieval, in: Thirtieth AAAI Conference on Artificial Intelligence, 2016, pp. 2415–2421.
[38] E. Yang, C. Deng, W. Liu, X. Liu, D. Tao, X. Gao, Pairwise relationship guided deep hashing for cross-modal retrieval, in: Thirty-First AAAI Conference on Artificial Intelligence, 2017, pp. 2415–2421.
[39] K. Chatfield, K. Simonyan, A. Vedaldi, A. Zisserman, Return of the devil in the details: Delving deep into convolutional nets, 2014, arXiv preprint arXiv:1405.3531.
[40] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, Ieee, 2009, pp. 248–255.
[41] M.A. Kramer, Nonlinear principal component analysis using autoassociative neural networks, AIChE J. 37 (2) (1991) 233–243.
[42] X. Peng, J. Feng, S. Xiao, W.-Y. Yau, J.T. Zhou, S. Yang, Structured autoencoders for subspace clustering, IEEE Trans. Image Process. 27 (10) (2018) 5076–5086.
[43] Y. Wu, S. Wang, Q. Huang, Multi-modal semantic autoencoder for cross-modal retrieval, Neurocomputing 331 (2019) 165–175.
[44] F. Shen, C. Shen, W. Liu, H. Tao Shen, Supervised discrete hashing, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 37–45.
[45] M.J. Huiskes, M.S. Lew, The MIR flickr retrieval evaluation, in: Proceedings of the 1st ACM International Conference on Multimedia Information Retrieval, ACM, 2008, pp. 39–43.
[46] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, Y. Zheng, NUS-WIDE: a real-world web image database from National University of Singapore, in: Proceedings of the ACM International Conference on Image and Video Retrieval, ACM, 2009, p. 48.
[47] N. Rasiwasia, J. Costa Pereira, E. Coviello, G. Doyle, G. Lanckriet, R. Levy, N. Vasconcelos, A New Approach to Cross-Modal Multimedia Retrieval, in: ACM International Conference on Multimedia, 2010, pp. 251–260.
[48] W. Liu, C. Mu, S. Kumar, S.-F. Chang, Discrete graph hashing, in: Advances in Neural Information Processing Systems, 2014, pp. 3419–3427.
[49] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., PyTorch: An imperative style, high-performance deep learning library, in: Advances in Neural Information Processing Systems, 2019, pp. 8024–8035.

Haopeng Qiang received a B.Sc. degree from the Wuhan University of Technology, Hubei, China, in 2018, where he is currently pursuing an M.Sc. degree in applied mathematics. His research interests focus on machine learning, deep learning and large-scale multimedia retrieval.

Yuan Wan is the corresponding author. She received the B.S. degree from Ocean University of China, Qingdao, China, the M.S. and the Ph.D. degrees in Computer Science from Wuhan University of Technology, Wuhan, China, in 2004 and 2012.

She is currently a Professor with the faculty of the Mathematical Department of Wuhan University of Technology. Her research interests include machine learning, feature selection, manifold learning and pattern recognition.

Ziyi Liu is currently an undergraduate of the Mathematical Department of Wuhan University of Technology. His research interests include Machine learning, computer vision and reinforcement learning.



Lun Xiang is currently a Master's graduate student of the Statistics Department of Wuhan University of Technology. His research interests include machine learning and image processing.

Xiaojing Meng is currently a Master's graduate student of the Mathematical Department of Wuhan University of Technology. Her research interests include machine learning and data knowledge.