Joint Image-Text Hashing for Fast Large-Scale Cross-Media Retrieval Using Self-Supervised Deep Learning

Gengshen Wu, Jungong Han, Zijia Lin, *Member, IEEE,* Guiguang Ding, Baochang Zhang, and Qiang Ni, *Senior Member, IEEE*

***Abstract*—Recent years have witnessed the promising future of hashing in the industrial applications for fast similarity retrieval. In this paper, we propose a novel supervised hashing method for large-scale cross-media search, termed Self-Supervised Deep Multimodal Hashing (SSDMH), which *learns* unified hash codes as well as deep hash functions for different modalities in a self-supervised manner. With the proposed regularized binary latent model, unified binary codes can be solved directly without relaxation strategy while retaining the neighborhood structures by the graph regularization term. Moreover, we propose a new discrete optimization solution, termed as *Binary Gradient Descent*, which aims at improving the optimization efficiency towards real-time operation. Extensive experiments on three benchmark datasets demonstrate the superiority of SSDMH over state-of-the-art cross-media hashing approaches.**

***Index Terms*—Cross-Media Retrieval, Deep learning, Regularized Binary Latent Model.**

I. Introduction

WITH the explosive growth of multimedia content, such as text, image/video, and audio, cross-media retrieval is becoming increasingly attractive, which allows users to get the results with various media types by submitting one query of any media type. In the context of big data, we need retrieval algorithms that are able to accurately search in large-scale datasets, and meanwhile, ensure the costs related to the processing overhead and storage requirements do not grow with the quantity of the data being produced.

Hashing, which represents the high-dimensional data with the compact binary codes, has drawn a considerable attention in the field of similarity retrieval for its low memory consumption (binary representation) and fast retrieval speed (bitwise XOR calculation). These properties make the hashing technique a widely used industrial solution for many applications [1]–[7], where the systems that have been commercialized include Shazam hashing/fingerprinting for music identification [8], Philips video hashing for broadcast monitoring [9] and Civolution SyncNow for cross-media search [10].

Regarding cross-media hashing, one of the main challenges is how to tackle the semantic gaps within different modalities. Most existing methods, both in unsupervised [11]–[13] and supervised [14]–[17] manners, concentrate on learning a common latent space for the multimodal data during the training process such that the heterogeneity among modalities can be minimized [11], [15]. Specifically for unsupervised methods, Ding et al. [11] adopt collective matrix factorization in modelling relations among different modalities, where unified binary codes are being learned via quantizing real-valued unified latent space. To deal with large quantization errors in [11], Wang et al. [13] improve the objective function to solve unified binary codes directly. However, they both fail to preserve the neighborhood structures of the original data, thus compromising the retrieval performance. Although some promising results have been achieved by the previous unsupervised methods, the overall performance is still far below satisfactory from the view of the real-world applications. It is commonly believed that a considerable performance gain can be obtained in supervised methods with aid of dedicated supervision information (e.g, semantic labels, affinity matrix) [1]. A semantic pooling approach is proposed by Chang et al. [18], where the relevance of each segment in untrimmed videos is evaluated by semantic saliency and then those shots are prioritized accordingly based on their saliency scores to make the final analysis with the ordering information exploited by an isotonic regularizer. Generally, the correlations among different modalities can be enhanced from the label information for unified hash codes in the Hamming space. For example, Lin et al. [15] introduce the probability distribution to learn unified hash codes with the semantic affinities. While Xu et al. [16] improve the quality of hash codes by means of the label information in shallow linear classifier. However, all the above methods employ a two-step like scheme in learning hash code, which inevitably yields suboptimal results.

Recently, deep learning technology has been widely incorporated in cross-media hashing [19], where several representative works are discussed briefly in the paper. For instance, a stacked auto-encoder architecture is proposed by Cao et

Manuscript received June 30, 2018; revised August 15, 2018; accepted September 9, 2018. This research was supported by the Royal Society Newton Mobility Grant (IE150997).

Gengshen Wu, Jungong Han and Qiang Ni are with School of Computing and Communications, Lancaster University, Lancaster, LA1 4YW, UK (e-mail: {gengshen.wu, jungong.han, q.ni}@lancaster.ac.uk).

Zijia Lin is with Microsoft Research Asia, Beijing, 100080, China (e-mail: linzijia07@163.com).

Guiguang Ding is with School of Software, Tsinghua University, Beijing, 100084, China (e-mail: dinggg@tsinghua.edu.cn).

Baochang Zhang is with Beihang University, Beijing, 100085, China (e-mail: bczhang@buaa.edu.cn). The corresponding authors are Jungong Han and Baochang Zhang.

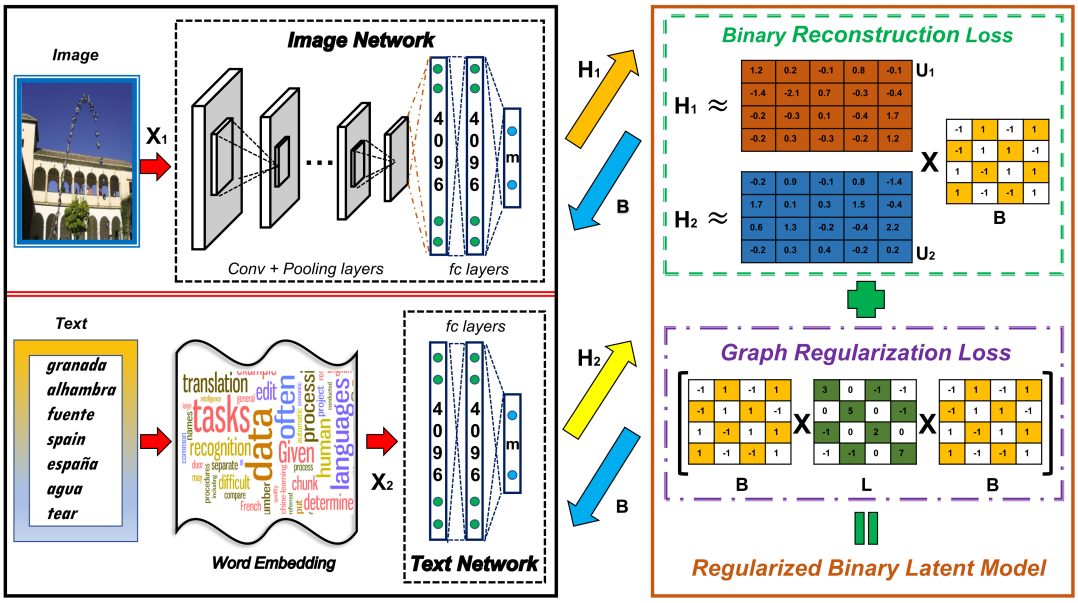


Fig. 1. The overview of our Self-Supervised Deep Multimodal Hashing. There are three subsections in the training process: deep feature learning (left), deep hash function learning (middle) and regularized binary latent representation learning (right). Specifically, the regularized binary latent model consists of two loss terms: binary reconstruction loss and graph regularization loss. The yellow arrows indicate the deep feature learning. The blue arrows show the iterative directions when learning deep hash functions with the guidance of the unified binary code $\mathbf{B}$. Better viewed in color.

al. [20], where the feature and semantic correlation across modalities are jointly maximized. While another work from Cao et al. [21] employs a metric-based approach to train the visual semantic fusion network with cosine hinge loss. However, the label information is not fully exploited and the performance compromises because of the noisy annotations. Subsequently, Li et al. [22] suggest another auto-encoder framework for unsupervised cross-modal hashing, which reconstructs the original features from the joint binary representation without considering the similarity relations. Jiang and Li [23] adopt a negative log likelihood criterion in an end-to-end deep framework, where the similarity structure between real-valued representations is retained. However, such similarity preservation is only performed on approximated hash codes with NO restrictions on binary codes in the training process.

As discussed above, we summarize three major limitations in the existing cross-modal hashing schemes as follows. Firstly, solving the discrete-constrained objective function usually undergoes a two-step procedure. At the relaxation step, supervised information is exploited to guide continuous hash codes learning, which are converted into binary codes by using rounding technology at the second step. Such a two-stage solution yields large quantization errors, which will be further magnified after the iterative code learning. Moreover, feature learning and bianrization are viewed as two independent steps in most previous methods, thus giving rise to suboptimal results. Last but no least, supervision knowledge cannot be fully explored in the code generation, as well as the hash function learning, which limits the improvement space of the hash code quality. The situation gets even worse when inaccurate or incomplete labels are provided [15], [16], [24]–[26]. Obviously, the retrieval performance would be heavily affected by those drawbacks, thus preventing the existing methods from mass deployment in the industrial applications.

To address the above issues, we propose a novel supervised cross-modal hashing method, termed as **S**elf-**S**upervised **D**eep **M**ultimodal **H**ashing (SSDMH), which integrates deep learning and regularized *binary* latent representation model jointly in a unified framework. Specifically, the discrete-constrained objective function is optimized directly without relaxation, and the deep hash functions are built via engaging deep feature learning with code learning in a self-supervised manner. The framework of SSDMH is illustrated in Fig. 1 and the corresponding contributions are summarized as follows:

1) The matrix factorization based supervised cross-modal hashing method is proposed to incorporate the deep feature learning and binarization seamlessly into a unified deep learning framework, where the deep hash functions are being built in a self-supervised manner via projecting the original features from various modalities into a common binary space.
2) A novel regularized *binary* latent model is proposed during the code learning, where the discrete unified binary codes can be solved without relaxation and the weights of different modalities are optimized dynamically. Particularly, to make the most advantage of supervision knowledge, we propose to minimize the graph regularization loss, which explicitly preserves the neighborhood structures of the original data and is prone to produce the discriminative hash codes.
3) An alternating optimization strategy is adopted in solving the discrete-constrained objective function, where deep parameters and unified binary codes are optimized jointly. Particularly, a novel discrete optimization method, termed as *Binary Gradient Descent*, is proposed to accelerate

TABLE I
THE NETWORK CONFIGURATIONS FOR TWO MODALITIES.

Modality	Layer	Description
Image	$conv1-conv5$	Follow the same configuration as AlexNet [27]
	$fc6\_I$, $fc7\_I$, fcb_I	4096-d, 4096-d, m-d
Text	$fc6\_T$, $fc7\_T$, fcb_T	4096-d, 4096-d, m-d

the optimization speed dramatically, in contrast to the traditional bit-by-bit fashions.

The reminder of the paper is organized as follows. We elaborate the proposed SSDMH in Section II. Experimental results along with data analysis are provided in Section III. Finally, the proposed method is concluded in Section IV.

II. PROPOSED METHOD

Fig. 1 illustrates the basic structure of the proposed SSDMH, where the sub procedures are described concisely as: we extract the deep features from the corresponding deep networks and then utilize those features to generate the unified binary representation via a novel regularized binary latent model. After that, the learnt binary code is adopted as 'supervision information' to re-train the previous deep networks, which exhibits the idea of the self-supervised manner. Those processes can be repeated iteratively to obtain the deep hash functions finally. In the next subsections, we will elaborate the proposed SSDMH in details.

A. Problem Definition

Without loss of generality, we use image and text to explain the proposed method. Assuming that the multimodal dataset contains n training instances, which is denoted as $\mathcal{O} = \{\mathbf{X}_i\}_{i=1}^2$, $i = \{1, 2\}$. Each instance has features from the image and text modality, which is represented by $\mathbf{X}_1 = \{x_1^j\}_{j=1}^n$ and $\mathbf{X}_2 = \{x_2^j\}_{j=1}^n$, respectively. Consequently, we use $x_1^j \in \mathbb{R}^{d_1}$ to denote the feature vector or the raw pixels of the j-th image and $x_2^j \in \mathbb{R}^{d_2}$ represents the feature vector of the j-th text, where d_1 and d_2 (usually $d_1 \neq d_2$) are the dimensionalities. The affinity matrix $\mathbf{S}_{n\times n} \in [0, 1]$ is also provided as the supervision information, which measures the similarity between data points. In the proposed SSDMH, the aim is to learn the deep hash functions $\mathcal{F}_i(\mathbf{X}_i; \Theta_i)$ that binarize the training data from two modalities into a set of unified binary codes $\mathbf{B} = \{b_i\}_{i=1}^n \in \{-1, +1\}^{m\times n}$, such that the similarities in the original spaces can be preserved. m denotes the code length, $\mathbf{X}_i$ are the input streams to those deep networks for two modalities. Here, the deep network parameters including weights and biases are uniformly defined as Θ_i.

B. Deep Architecture

Considering the favorable feature expressive ability and the deployment flexibility[1], we adopt AlexNet and Multi-Layer Perceptrons (MLP) as the feature modelers for the image and textual modalities, as shown in Fig. 1. For the purpose of making the networks compatible to the application, the last fully-connected (*fc*) layers of the original networks are replaced with the new bottleneck layers (fcb_I and fcb_T) comprising m hidden units to facilitate the network training afterwards. $Tanh$ function is added at the end of the last layers as the activation function to make the outputs fall into $[-1, 1]$. The network configurations are summarized in Table I. In this paper, the deep architectures not only provide the deep features (e.g. $\mathbf{H}_i \in \mathbb{R}^{4096\times n}$ from $fc7\_I$ and $fc7\_T$) in learning the unified binary representation, but also act as the deep hash functions $\mathcal{F}_i(\mathbf{X}_i; \Theta_i)$ to generate hash codes for new queries.

[1]The model sizes for AlexNet and MLP are < 240MB and < 90MB after training, which are affordable on most portable devices.

C. Regularized Binary Latent Model

In the hash code learning, we propose a novel regularized binary latent representation model to generate the unified binary code $\mathbf{B}$ for two modalities. Particularly, the proposed model consists of two loss terms: *binary reconstruction loss* and *graph regularization loss*. We formulate the objective function of the proposed model as below:

$$\min_{\mathbf{B},\mathbf{U}_i,\alpha_i} \sum_{i=1}^{2} \alpha_i^\gamma (\|\mathbf{H}_i - \mathbf{U}_i\mathbf{B}\|_F^2 + \beta Tr(\mathbf{B}\mathbf{L}\mathbf{B}^T)), \quad (1)$$

where β and γ are balance parameters. γ is a positive number controlling the weight of each modality while β estimates the impact of the loss term in (1). $\mathbf{H}_i \in \mathbb{R}^{4096\times n}$ are the deep features extracted from $fc7\_I$ and $fc7\_T$ layers, $\mathbf{U}_i \in \mathbb{R}^{4096\times m}$ are the latent factor matrices, $\mathbf{L} \in \mathbb{R}^{n\times n}$ is the Laplacian matrix. $\alpha_i (\alpha_i > 0)$ are the weight factors for two modalities separately and satisfy $\sum_{i=1}^2 \alpha_i = 1$. $Tr(.)$ is the trace norm. Those terms are elaborated in the next subsections.

1) Binary Reconstruction Loss: As shown in (1), the first term measures the reconstruction losses from their latent common binary representation $\mathbf{B}$ to the deep features $\mathbf{H}_i$, which shares similar idea with CMFH [11]. However, it differs from [11] in two aspects. Firstly, CMFH adopts a two-step approach in generating the unified binary representation, which solves the *real-valued* latent common space $\mathbf{V} \in \mathbb{R}^{m\times n}$ first and binarizes it afterwards, as shown in the top part of (2). This inevitably yields the large quantization errors, no matter which rounding schemes are used [13], [15]. However, this problem can be avoided in the proposed model by solving the binary code directly as the bottom of (2).

$$\begin{aligned} &\min_{\mathbf{U}_i,\mathbf{V}} \alpha\|\mathbf{H}_1 - \mathbf{U}_1\mathbf{V}\|_F^2 + (1-\alpha)\|\mathbf{H}_2 - \mathbf{U}_2\mathbf{V}\|_F^2 \\ &\Rightarrow \min_{\mathbf{U}_i,\mathbf{B},\alpha_i} \sum_{i=1}^{2} \alpha_i^\gamma \|\mathbf{H}_i - \mathbf{U}_i\mathbf{B}\|_F^2. \end{aligned} \quad (2)$$

Moreover, the modality weight is set empirically in CMFH (e.g. $\alpha = 0.5$), while the weights α_i are optimized dynamically in the proposed model. It is more sensible for the important modality to hold the dominant position in the optimization [13], [28].

2) Graph Regularization Loss: In the second term, we introduce graph regularization to preserve the semantic consistency of data points from multiple modalities, which aims at restricting the neighboring relationships in solving the unified

binary code [29]. Particularly, the spectral graph problem can be formulated as:

$$\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\|b_i - b_j\|_F^2\mathbf{S}_{ij} = Tr(\mathbf{B}\mathbf{L}\mathbf{B}^T), \tag{3}$$

where $\mathbf{S} \in \mathbb{R}^{n\times n}$ represents the semantic affinity matrix that can be derived from manual scoring [15], $s_{ij} = 1$ if x_1^i and x_2^j share the same label and otherwise 0. $\mathbf{D}$ is the diagonal matrix whose entries are the column sum of $\mathbf{S}$, i.e., $\mathbf{D}_{ii} = \sum_{j=1}^{n}\mathbf{S}_{ij}$. The Laplacian matrix $\mathbf{L}$ can be calculated as $\mathbf{L} = \mathbf{D} - \mathbf{S}$.

D. Deep Hash Function Learning

Having obtained the unified binary representation $\mathbf{B}$, the next step is to train the deep hash models [2] $\mathcal{F}_i(\mathbf{X}_i;\Theta_i)$ with Euclidean loss layers, which aims at projecting the original features from different views into the common binary space. This strategy integrates the learning processes of deep feature and hash function in a self-supervised manner, thus predicting the discriminative hash code for new query instance in the testing stage [5]. The problem is formulated as:

$$\min_{\Theta_i}\sum_{i=1}^{2}\|\mathcal{F}_i(\mathbf{X}_i;\Theta_i) - \mathbf{B}\|_F^2. \tag{4}$$

Particularly, Euclidean distances are minimized between the outputs of the deep networks and the unified binary code $\mathbf{B}$, while the network parameters Θ_i can be updated through the back-propagation with Stochastic Gradient Descent (SGD). It is worth noting that the original images or features (i.e, $\mathbf{X}_i$) are fixed and used as the input streams for the deep architecture to facilitate the network training.

E. Objective Function and Optimization

By combining (1) and (4), the overall objective function of SSDMH is written as below:

$$\begin{aligned}\min_{\Theta_i,\mathbf{B},\mathbf{U}_i,\alpha_i}&\sum_{i=1}^{2}\alpha_i^{\gamma}(\|\mathbf{H}_i - \mathbf{U}_i\mathbf{B}\|_F^2 + \beta Tr(\mathbf{B}\mathbf{L}\mathbf{B}^T))\\ &+\lambda\sum_{i=1}^{2}\|\mathcal{F}_i(\mathbf{X}_i;\Theta_i) - \mathbf{B}\|_F^2,\end{aligned} \tag{5}$$

where $\mathbf{B} \in \{-1,+1\}^{m\times n}$. The proposed objective function is a NP-hard problem and cannot be solved directly because of the binary constraints. Subsequently, we adopt an alternating strategy to solve (5), where the involved parameters are optimized iteratively by the following steps.

1) $\mathbf{U}_i$-Step: Firstly, by fixing all other variables except for $\mathbf{U}_i$, (5) is reduced as:

$$\min_{\mathbf{U}_i}\sum_{i=1}^{2}\alpha_i^{\gamma}\|\mathbf{H}_i - \mathbf{U}_i\mathbf{B}\|_F^2,\ \text{s.t. } \mathbf{B}\in\{-1,+1\}^{m\times n}. \tag{6}$$

Then we calculate the derivation of (6) with respect to $\mathbf{U}_i$ and the closed-form solution of $\mathbf{U}_i$ can be obtained by setting the derivation as 0:

$$\mathbf{U}_i = \mathbf{H}_i\mathbf{B}^T(\mathbf{B}\mathbf{B}^T)^{-1}. \tag{7}$$

2) $\mathbf{B}$-Step: Then, with all variables fixed but $\mathbf{B}$ as the only argument, (5) is re-written as:

$$\begin{aligned}&\min_{\mathbf{B}}\sum_{i=1}^{2}\alpha_i^{\gamma}(\|\mathbf{H}_i - \mathbf{U}_i\mathbf{B}\|_F^2 + \beta Tr(\mathbf{B}\mathbf{L}\mathbf{B}^T))\\ &+\lambda\sum_{i=1}^{2}\|\mathcal{F}_i(\mathbf{X}_i;\Theta_i) - \mathbf{B}\|_F^2,\ \text{s.t. } \mathbf{B}\in\{-1,+1\}^{m\times n}.\end{aligned} \tag{8}$$

Then we can expand (8) to:

$$\begin{aligned}&\min_{\mathbf{B}}\sum_{i=1}^{2}\alpha_i^{\gamma}(Tr(\mathbf{B}^T\mathbf{U}_i^T\mathbf{U}_i\mathbf{B} - 2\mathbf{B}^T\mathbf{U}_i^T\mathbf{H}_i) + \beta Tr(\mathbf{B}\mathbf{L}\mathbf{B}^T))\\ &\quad+\lambda\sum_{i=1}^{2}Tr(\mathbf{B}\mathbf{B}^T - 2\mathbf{B}^T\mathcal{F}_i(\mathbf{X}_i;\Theta_i))\\ &\quad=\min_{\mathbf{B}}\|\mathbf{G}^T\mathbf{B}\|_F^2 - 2Tr(\mathbf{B}^T\mathbf{Q}) + Tr(\mathbf{B}\mathbf{K}\mathbf{B}^T),\end{aligned} \tag{9}$$

where $\mathbf{G} = [\sqrt{\alpha_1^{\gamma}}\mathbf{U}_1;\sqrt{\alpha_2^{\gamma}}\mathbf{U}_2]^T$, $\mathbf{K} = \sum_{i=1}^{2}\alpha_i^{\gamma}\beta\mathbf{L} + 2\lambda\mathbf{I}_n$, $\mathbf{Q} = \sum_{i=1}^{2}(\alpha_i^{\gamma}\mathbf{U}_i^T\mathbf{H}_i + \lambda\mathcal{F}_i(\mathbf{X}_i;\Theta_i))$. Following discrete cyclic coordinate descent (DCC) [30], we denote b^T as the j-th row of $\mathbf{B}$, and $\mathbf{B}'$ the matrix of $\mathbf{B}$ excluding b. Similarly, let g^T be the j-th row of $\mathbf{G}$, $\mathbf{G}'$ be the matrix of $\mathbf{G}$ excluding g and q^T be the j-th row of $\mathbf{Q}$, then we have

$$\begin{aligned}&\min_b(g^T\mathbf{G}'^T\mathbf{B}' - q^T)b + b^T\mathbf{K}b = \min_b p^Tb + b^T\mathbf{K}b,\\ &\text{s.t. } p = (\mathbf{B}'^T\mathbf{G}'g - q)\in\mathbb{R}^n,\ \mathbf{B}'\in\{-1,+1\}^{m-1\times n}.\end{aligned} \tag{10}$$

Obviously, the above equation can be considered as the classical Binary Quadratic Programming (BQP) problem in most previous papers and it can be optimized via solving each entry of b sequentially (flip one entry per time) as described in some coordinate descent based methods [24], [30], [31]. However, those methods usually suffer from the slow convergence issues, especially for the cases with long code. In this paper, we propose a new solution called *Binary Gradient Descent* (BGD), which is detailed in the following paragraphs, to accelerate the convergence in optimizing (10).

Suppose that the current value of b is b_0 and the new value b_1 can be obtained by adding an offset Δ to b_0, namely $b_1 = b_0 + \Delta$. We substitute b_0 and b_1 into (10), the deviation $\mathcal{L}$ between the values of (10) is calculated as follows:

$$\begin{aligned}\mathcal{L} &= b_1^T\mathbf{K}b_1 + p^Tb_1 - b_0^T\mathbf{K}b_0 - p^Tb_0\\ &= (b_0+\Delta)^T\mathbf{K}(b_0+\Delta) + p^T(b_0+\Delta) - b_0^T\mathbf{K}b_0 - p^Tb_0\\ &= 2\Delta^T\mathbf{K}b_0 + \Delta^T\mathbf{K}\Delta + p^T\Delta\\ &= \Delta^T\mathbf{K}\Delta + (2\mathbf{K}b_0 + p)^T\Delta.\end{aligned} \tag{11}$$

Since there is only one entry with the value[3] of -2 or 2 in Δ, then we have $\Delta^T\mathbf{K}\Delta = 4\mathbf{K}_{j,j}$, where j is the index for the entry that is non-zero in Δ. Thus, (11) can be reformed as:

$$\mathcal{L} = 4diag(\mathbf{K}) + (2\mathbf{K}b_0 + p)^T\Delta, \tag{12}$$

[2]Other predictive models like linear classifier and kernel logistic regression can also be applied here [13], [15], we will leave those in the future research.

[3]The position of -2 or 2 in Δ is based on the corresponding entry in b_0 so as to change -1 to 1 with 2 or 1 to -1 with -2. All the rest entries are 0 in Δ.

where $diag(\mathbf{K})$ preserves the diagonal elements of $\mathbf{K}$. Therefore, the deviation $\mathcal{L}$ must be negative if we try to find b_1 to make the objective function descent and it can be obtained by calculating another vector h regarding each entry in b_0 as:

$$h = 4\delta + (2\mathbf{K}b_0 + p) \odot d, \tag{13}$$

where δ is the column vector of all diagonal elements of $\mathbf{K}$, $\odot$ denotes the entry-wise multiplication of the vector, and d satisfies: 1) if the j-th entry of b_0 is 1 , then $d_j = -2$; 2) if the j-th entry in b_0 is -1, then $d_j = 2$. The optimization process will be completed if the smallest value in h is non-negative, otherwise we only retain the value of the corresponding entry in d, and set other entries to 0 to obtain Δ. After getting b_1 with $\Delta + b_0$, we update b_0 above and recalculate the new Δ accordingly. Essentially, the proposed method flips all the entries by repeating the above computations and selects the entry that is most likely to make the objective function descend in a monotonic discrete manner. As observed from the experiments, it usually requires $n/2$ updates on the entries such that the objective function descends. The proposed BGD only needs 1 iteration to make (10) descent with faster converging speed compared to the later cases that require at least n iterations, thus obtaining the local optimal solution efficiently.

3) α_i-Step: With other parameters fixed except for α_i, we formulate (5) as below:

$$\min_{\alpha_i} \sum_{i=1}^{2} \alpha_i^{\gamma} \mathbf{E}_i, \text{ s.t. } \alpha_i > 0, \tag{14}$$

where $\mathbf{E}_i = \|\mathbf{H}_i - \mathbf{U}_i\mathbf{B}\|_F^2 + \beta Tr(\mathbf{B}\mathbf{L}\mathbf{B}^T)$. Subsequently, the Lagrange function of (14) can be formulated as:

$$\sum_{i=1}^{2} \alpha_i^{\gamma} \mathbf{E}_i - \mu(\sum_{i=1}^{2} \alpha_i^{\gamma} - 1), \tag{15}$$

where μ is the Lagrange multiplier. Taking $\sum_{i=1}^{2} \alpha_i = 1$ into consideration, the optimal solution of α_i is derived as:

$$\alpha_i = \frac{(1/\mathbf{E}_i)^{\frac{1}{\gamma-1}}}{\sum_{i=1}^{2}(1/\mathbf{E}_i)^{\frac{1}{\gamma-1}}}. \tag{16}$$

4) Θ_i-Step: When fixing all other parameters but Θ_i, the objective function (5) is reduced to

$$\min_{\Theta_i} \sum_{i=1}^{2} \|\mathcal{F}_i(\mathbf{X}_i; \Theta_i) - \mathbf{B}\|_F^2, \text{ s.t. } \mathbf{B} \in \{-1, +1\}^{m \times n}, \tag{17}$$

where the deep hash functions $\mathcal{F}_i(\mathbf{X}_i; \Theta_i)$ can be solved by optimizing the network parameters Θ_i under the guidance of the unified binary code via mini-batch back-propagation [5]. Repeating the above optimization processes until convergence, the deep hash functions can be learned and deployed for the large-scale multimodal retrieval application. When giving new query instances $\mathbf{X}_i^q \notin \mathcal{O}$, the new hash codes can be obtained by calculating $sign(\mathcal{F}_i(\mathbf{X}_i^q; \Theta_i))$. The proposed SSDMH is summarized in Algorithm 1.

Algorithm 1. Self-Supervised Deep Multimodal Hashing

Input: Original feature $\mathbf{X}_i$, code length m, parameters β and γ, affinity matrix $\mathbf{S}$. Randomly initialize binary code $\mathbf{B}$, latent matrices $\mathbf{U}_i$ and deep parameters Θ_i. Set weights $\alpha_i = [0.5, 0.5]$, $i = \{1, 2\}$.

Output: Deep hash functions $\mathcal{F}_i(\mathbf{X}_i; \Theta_i)$;

```
for T = 1 to 5 do
    Extract the feature matrices H_i from fc7_I and fc7_T layers of two deep networks, respectively;
    for t = 1 to 5 do
        Update the latent factor matrices U_i by (7);
        Update the unified hash code B by (8)∼(13);
        Update the weight factors α_i by (16);
    end for
    Update the network parameters Θ_i by (17);
end for
return F_i(X_i; Θ_i);
```

F. Computational Complexity

The computational complexity of SSDMH is composed of two parts: learning binary code and deep hash function. However, it is not straightforward to calculate the complexity for network training, which depends on many external conditions. Regarding the regularized binary latent model, the computational complexity is $O(d^2n + dn)$ during each optimization iteration and $d = max\{d_1, d_2, m\}$. In total, the training complexity is $O((d^2n + dn)t)$, where t is the maximum iteration (less than 5) in updating the binary code.

III. Experiment

In this section, extensive experiments are conducted on three datasets to evaluate the performance of the proposed SSDMH.

A. Dataset Descriptions

The Wiki [32] dataset is made up of 2,866 image-text pairs collected from Wikipedia. Each image is represented by a 128-dimensional SIFT feature vector and a 10-dimensional topic vector is given to describe the text. These pairs contain 10 semantic categories and each pair is manually assigned to one of them. All data pairs are split into the training (2,173) and query (693) sets. The MIRFlickr [33] dataset collects $25,000$ instances from Flickr, which are annotated by at least one of 24 provided labels. A 100-dimensional SIFT descriptor is provided to represent each image, while the text is expressed as a 500-dimensional tagging vector. We randomly select $2,000$ image-text pairs as the queries and use the remaining pairs for training. The NUS-WIDE [34] dataset contains 269,648 images and each image is associated with a textual tag. Those instances are manually labeled with 81 different concepts. Following [11], [15], we only retain the instances annotated with the 10 most frequent concepts, thus preserving 186,577 image-tag pairs for the experiment. Each image is represented by a 500-dimensional SIFT feature vector and an index vector of the most frequent 1,000 tags is provided for each text.

TABLE II
MAP RESULTS FOR 'IMAGE→TEXT' AND 'TEXT→IMAGE' TASKS ON THREE DATASETS AT VARIOUS CODE LENGTHS (BITS) WHEN USING DIFFERENT METHODS. THE BEST PERFORMANCE IS SHOWN IN BOLDFACE.

Task	Method	Wiki				MIRFlickr				NUS-WIDE			
		16	32	64	128	16	32	64	128	16	32	64	128
Image→Text	IMH [12]	0.219	0.222	0.224	0.213	0.591	0.593	0.588	0.601	0.616	0.612	0.603	0.576
	DBRC [22]	0.278	0.283	0.291	0.302	0.554	0.59	0.597	0.607	0.621	0.629	0.632	0.639
	RFDH [13]	0.369	0.373	0.377	0.388	0.674	0.718	0.742	0.766	0.659	0.707	0.742	0.764
	DCMH [23]	0.264	0.269	0.279	0.284	0.732	0.747	0.748	0.752	0.584	0.603	0.612	0.623
	CAH [20]	0.242	0.248	0.253	0.261	0.688	0.705	0.708	0.715	0.509	0.542	0.567	0.582
	SCMFH [11]	0.284	0.294	0.299	0.305	0.651	0.654	0.655	0.664	0.495	0.499	0.506	0.624
	DisCMH [16]	0.375	0.394	0.395	0.392	0.72	0.727	0.721	0.732	0.683	0.758	0.775	0.764
	SePH$_{km}$ [15]	0.399	0.405	0.408	0.412	0.763	0.769	0.773	0.776	0.739	0.75	0.761	0.767
	SSDMH	**0.421**	**0.436**	**0.446**	**0.451**	**0.797**	**0.801**	**0.808**	**0.823**	**0.803**	**0.809**	**0.821**	**0.834**
Text→Image	IMH [12]	0.489	0.495	0.472	0.473	0.553	0.583	0.592	0.603	0.608	0.615	0.604	0.578
	DBRC [22]	0.594	0.608	0.613	0.621	0.596	0.599	0.605	0.613	0.634	0.648	0.646	0.658
	RFDH [13]	0.619	0.626	0.646	0.65	0.655	0.692	0.701	0.711	0.616	0.645	0.677	0.704
	DCMH [23]	0.621	0.628	0.648	0.658	0.733	0.745	0.749	0.753	0.639	0.656	0.661	0.678
	CAH [20]	0.373	0.386	0.393	0.402	0.661	0.674	0.694	0.722	0.514	0.545	0.584	0.608
	SCMFH [11]	0.635	0.641	0.656	0.664	0.682	0.703	0.716	0.726	0.569	0.612	0.657	0.684
	DisCMH [16]	0.676	0.662	0.663	0.654	0.747	0.758	0.75	0.759	0.652	0.736	0.75	0.749
	SePH$_{km}$ [15]	0.664	0.696	0.695	0.702	0.727	0.731	0.748	0.743	0.686	0.695	0.709	0.711
	SSDMH	**0.716**	**0.735**	**0.737**	**0.745**	**0.833**	**0.836**	**0.843**	**0.852**	**0.815**	**0.821**	**0.833**	**0.836**

Finally, 2,000 image-tag pairs are randomly picked up as the queries and the rest pairs are used for training. Each pair is labeled with at least one of the 10 concepts and two image-tag pairs are considered to be similar if one of labels matched.

B. Experiment Settings

We compare the proposed SSDMH with some extremely competitive works published previously, including IMH [12], RFDH [13], DBRC [22], DCMH [23], CAH [20], SCMFH[4] [11], DisCMH [16] and SePH$_{km}$ [15] in the experiments. For the fair comparison, the identical training and query sets are utilized in the performance evaluation and the best results are reported by adopting and tuning the suggested parameters in their papers. Regarding the evaluation metrics, we generally adopt two widely-used criteria in the multimodal retrieval: Mean Average Precision (MAP) and Precision-Recall (PR) curve, as the main metrics in the following experiments [11], [15]. The number of top returned instances is set to 50 when calculating MAP. In this paper, we focus on two cross-media retrieval tasks: 'Image Query versus Text database' (Image→Text) and 'Text Query versus Image database' (Text→Image).

Following the settings in [11], [12], [15], the whole training set for the Wiki dataset is utilized in generating the unified binary representation. While for the other two benchmarks, $5,000$ instances are randomly sampled from their training sets to produce the binary code. For fair comparison, instead of using the original image features (e.g. SIFT), the 4096-d deep feature vectors are extracted from the *fc*7 layer of the pre-trained AlexNet for those non-deep methods (e.g. RFDH [13], SePH$_{km}$ [15]) during the code learning. For the parameter settings in the semantic binary latent model, γ, β and λ are set to 5, 1 and 1, respectively. The maximum iteration t is set to 5 in updating the binary code. When building the deep hash functions, the original image pixels and their tagging vectors are kept fixed and employed as the inputs to those deep networks for two modalities, respectively. We adopt SGD optimizer in the network training with the basic learning rates 0.0001 and 0.01 for the image and text modality, respectively. The batch sizes are fixed as 512 and they take 10 epochs at most until the networks converge. In this paper, we construct the deep architectures using *Caffe* [35]. The codes for the above prior arts are implemented by MATLAB 2014a on an Ubuntu 14.04 LTS machine, which is configured with Intel Core i7-6700k CPU, 64GB RAM and NVIDIA 1080i GPU.

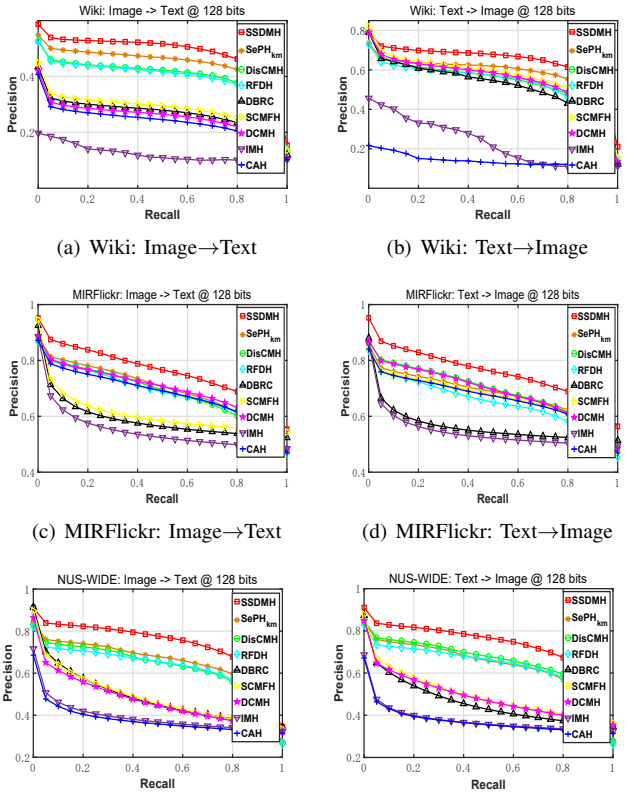
[4]We adopt the supervised version of CMFH in the comparison.

C. Results and Analysis

1) Architecture Investigation: In this section, we first investigate the impact of each loss term in the regularized binary latent model on the multimodal retrieval performance. Particularly, two different cases are analyzed: SSDMH$_{brl}$ and SSDMH$_{brl+grl}$, where brl and grl are shorts for *binary reconstruction loss* and *graph regularization loss* respectively. Here, SSDMH$_{brl}$ is realized by setting β as 0 during the optimization. We report the MAP results on three datasets at 128 bits

TABLE III
MAP RESULTS AT THE CODE LENGTH OF 128 WHEN INVOLVING VARIOUS LOSS TERMS DEPLOYED IN THE PROPOSED METHOD: SSDMH$_{brl}$ AND SSDMH$_{brl+grl}$.

Method	Task	Dataset		
		Wiki	MIRFlickr	NUS-WIDE
SSDMH$_{brl}$	Image→Text	0.408	0.745	0.774
	Text→Image	0.703	0.767	0.802
SSDMH$_{brl+grl}$	Image→Text	0.451	0.823	0.834
	Text→Image	0.745	0.852	0.836



(a) Wiki: Image→Text (b) Wiki: Text→Image

(c) MIRFlickr: Image→Text (d) MIRFlickr: Text→Image

(e) NUS-WIDE: Image→Text (f) NUS-WIDE: Text→Image

Fig. 2. The Precision-Recall curves at 128 bits on three datasets.

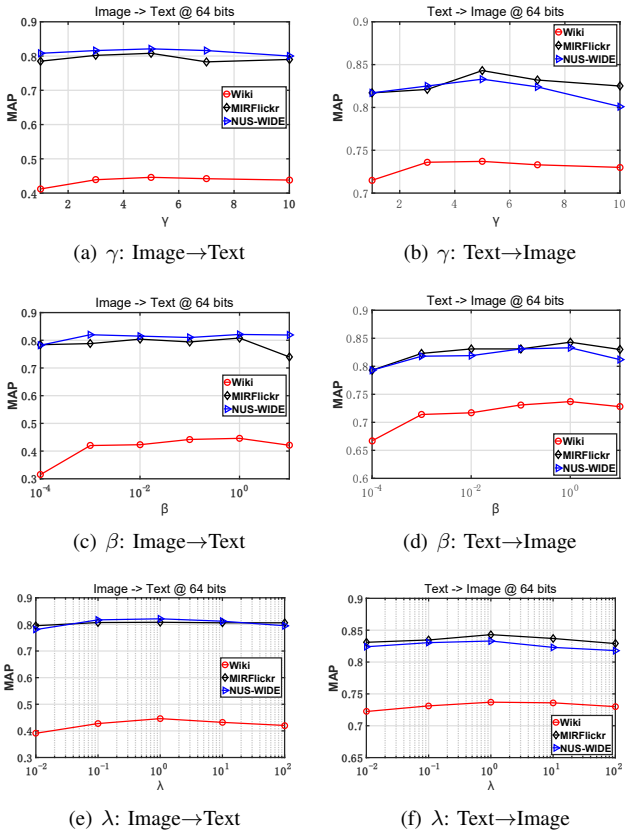
in Table III. As can be seen, the worst performance has been achieved by SSDMH$_{brl}$ without any supervision information. With the graph regularization loss involved, SSDMH$_{brl+grl}$ improves the MAP values by approximately $3.4\% \sim 8.5\%$ on two retrieval tasks, implying the importance of preserving the neighborhood structure within the original data in the hash code learning.

2) Overall Comparisons with Baselines: To validate the superiority of the proposed SSDMH, we compare it with the state-of-the-arts and report the MAP results at various code lengths on three datasets, as shown in Table II. Generally, the proposed SSDMH outperforms all baselines in terms of MAP on two retrieval tasks. Specifically, regarding 'Image→Text' tasks, the MAP values from SSDMH are 3.9%, 4.7% and 6.7% higher than those achieved by the most competitive baselines at the code length of 128 on Wiki, MIRFlickr and NUS-WIDE, respectively. While for 'Text→Image' task, the gaps have increased to 4.3%, 9.3% and 8.7%. When dealing with the short codes (e.g. 16, 32 bits), SSDMH still achieves the best performance showing the great potential of SSDMH on wide deployment in the industrial applications. Moreover, we also plot the PR curves at 128 bits when using those methods on three datasets, as shown in Fig. 2. It can be seen the curves of the proposed SSDMH are always at the top of the figures, which comply with the results in Table II.

3) Effect of Training Size: Moreover, the variations on the MAP results are evaluated when using different amount of data points in the code learning, as shown in Table IV. Specifically, we report the results on MIRFlickr and NUS-WIDE at 64 bits

TABLE IV
EFFECT OF TRAINING SIZE ON MIRFLICKR AND NUS-WIDE AT THE CODE LENGTH OF 64.

Dataset	Task	Training Size				
		1k	2k	5k	10k	15k
MIRFlickr	Image→Text	0.761	0.769	0.808	0.815	0.821
	Text→Image	0.793	0.811	0.843	0.854	0.861
NUS-WIDE	Image→Text	0.761	0.783	0.821	0.834	0.837
	Text→Image	0.781	0.803	0.833	0.842	0.85



(a) γ: Image→Text (b) γ: Text→Image

(c) β: Image→Text (d) β: Text→Image

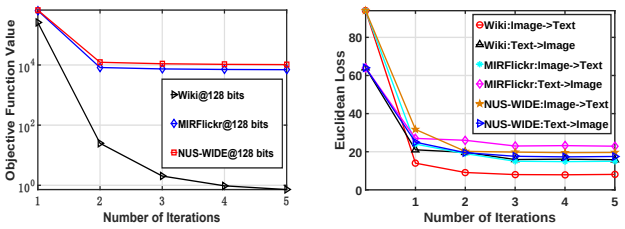
(e) λ: Image→Text (f) λ: Text→Image

Fig. 3. MAP versus γ, β and λ at 64 bits on three datasets.

on two retrieval tasks. As can be seen, the MAP values keep increasing with more data points employed in the initial stage and tend to converge after the size of $10,000$. It is worth pointing out that SSDMH still achieves competitive results when limited data points (e.g., $5,000$) available.

4) Parameter Sensitivity Analysis: We further analyze the retrieval performance variations from adjusting the hyper-parameters in the code learning. By fixing the code length to 64 bits in the experiments, we plot the MAP variations in Fig. 3 when altering γ, β and λ. As can be seen, the MAP results have minor changes when varying the parameters, which indicates that SSDMH is not very sensitive to the hyper-parameters. Moreover, the empirical values of γ, β and λ are close to the optimal settings in the figures and they can make great contribution in yielding superior retrieval performance.

5) Convergence Study: Fig. 4 is plotted to estimate the convergence rates in solving the unified binary code and learning the deep hash functions at 128 bits. As can be seen, the objective function values converge very fast within 5 iterations



(a) (b)

Fig. 4. (a) Objective function values after each iteration (t) when solving the unified binary code at 128 bits; (b) Euclidean losses after every iteration (T) when learning the deep hash functions at 128 bits.

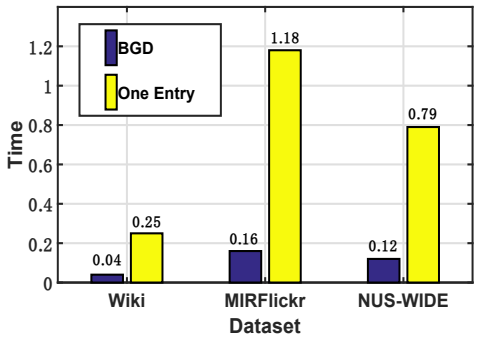


Fig. 5. Time costs (in seconds) in optimizing one row of the unified binary code at 128 bits on three datasets when using BGD and One Entry [30] separately.

in the code learning, while the deep hash functions for two modalities can be built efficiently after $3 \sim 5$ iterations.

6) BGD versus One Entry: We further compare the efficiency of the proposed BGD and One Entry (namely flipping one entry per time) [30], where the latter one denotes the most representative method in the discrete optimization [24], [31]. As can be observed from Fig. 5, the proposed BGD costs much shorter time, averagely over 80%, against One Entry in solving one row of the unified binary code at 128 bits, thus accelerating the code optimization dramatically. Although some recent papers [24], [36] make minor changes during the discrete optimization, they all utilize the same entry flipping strategy as One Entry. There is no evidence showing that such efficiency issue could be alleviated in their methods.

7) Training Efficiency: Finally, the training costs of the proposed SSDMH at 128 bits on three datasets are reported in Table V. There are two main sub processes: code learning and network training, during the optimization in each loop. As can be seen, the optimization for each loop can be done within 18 minutes for most cases. Considering the proposed SSDMH usually converges within $T = 5$ loops for one code length, the total optimization costs less than 1.5 hours while the values for other cases of short codes are far below.

IV. Conclusion

This paper has provided an industrial solution for fast large-scale cross-media retrieval. Specifically, a novel self-supervised deep multimodal hashing method, SSDMH, is presented, where the deep feature learning and the semantic binary code learning are integrated in a unified framework.

TABLE V
Time Costs (in Seconds) in The Training Processes of SSDMH on Three Datasets at 128 Bits For One Loop (T).

Dataset	Code Learning	Network Training	
		Image	Text
Wiki	117.3	123.2	15.7
MIRFlickr	614.4	362.3	27.3
NUS-WIDE	582.1	413.1	38.3

Particularly, by solving the discrete constrained objective function in an alternating manner, the unified binary code can be generated directly without relaxation. Moreover, the semantic affinity matrix is utilized in the code learning with the neighborhood structure of original data preserved. Besides, Binary Gradient Descent is proposed to *accelerate* the discrete optimization. Extensive experiments on three datasets demonstrate the superiority over several state-of-the-art baselines.

References

[1] K. Wang, Q. Yin, W. Wang, S. Wu, and L. Wang, "A comprehensive survey on cross-modal retrieval," *arXiv preprint arXiv:1607.06215*, 2016.

[2] Y. Gao, M. Wang, R. Ji, X. Wu, and Q. Dai, "3-d object retrieval with hausdorff distance learning," *IEEE Transactions on industrial electronics*, vol. 61, no. 4, pp. 2088–2098, 2014.

[3] W. Hu, G.-P. Liu, and H. Zhou, "Web-based 3-d control laboratory for remote real-time experimentation," *IEEE Transactions on Industrial Electronics*, vol. 60, no. 10, pp. 4673–4682, 2013.

[4] Z. Stejic, Y. Takama, and K. Hirota, "Relevance feedback-based image retrieval interface incorporating region and feature saliency patterns as visualizable image similarity criteria," *IEEE Transactions on Industrial Electronics*, vol. 50, no. 5, pp. 839–852, 2003.

[5] F. Shen, Y. Xu, L. Liu, Y. Yang, Z. Huang, and H. T. Shen, "Unsupervised deep hashing with similarity-adaptive and discrete optimization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[6] Q. Wang, J. Wan, and Y. Yuan, "Deep metric learning for crowdedness regression," *IEEE Transactions on Circuits and Systems for Video Technology*, 2017.

[7] Q. Wang, Z. Yuan, Q. Du, and X. Li, "Getnet: A general end-to-end 2-d cnn framework for hyperspectral image change detection," *IEEE Transactions on Geoscience and Remote Sensing*, DOI 10.1109/TGRS.2018.2849692, pp. 1–11, 2018.

[8] A. Wang *et al.*, "An industrial strength audio search algorithm." in *Ismir*, vol. 2003, pp. 7–13. Washington, DC, 2003.

[9] J. Oostveen, T. Kalker, and J. Haitsma, "Feature extraction and a database strategy for video fingerprinting," in *International Conference on Advances in Visual Information Systems*, pp. 117–128. Springer, 2002.

[10] J. Han and G. C. Langelaar, "Method and device for generating fingerprints of information signals," Jan. 18 2018, uS Patent App. 15/301,554.

[11] G. Ding, Y. Guo, J. Zhou, and Y. Gao, "Large-scale cross-modality search via collective matrix factorization hashing," *IEEE Transactions on Image Processing*, vol. 25, no. 11, pp. 5427–5440, 2016.

[12] J. Song, Y. Yang, Y. Yang, Z. Huang, and H. T. Shen, "Inter-media hashing for large-scale retrieval from heterogeneous data sources," in *ACM SIGMOD ICMD*, pp. 785–796. ACM, 2013.

[13] D. Wang, Q. Wang, and X. Gao, "Robust and flexible discrete hashing for cross-modal similarity search," *TCSVT*, 2017.

[14] M. M. Bronstein, A. M. Bronstein, F. Michel, and N. Paragios, "Data fusion through cross-modality metric learning using similarity-sensitive hashing," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 3594–3601. IEEE, 2010.

[15] Z. Lin, G. Ding, M. Hu, and J. Wang, "Semantics-preserving hashing for cross-view retrieval," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3864–3872, 2015.

[16] X. Xu, F. Shen, Y. Yang, and H. T. Shen, "Discriminant cross-modal hashing," in *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval*, pp. 305–308. ACM, 2016.

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

[18] X. Chang, Y.-L. Yu, Y. Yang, and E. P. Xing, "Semantic pooling for complex event analysis in untrimmed videos," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 8, pp. 1617–1632, 2017.

[19] Y. Wei, Y. Zhao, C. Lu, S. Wei, L. Liu, Z. Zhu, and S. Yan, "Cross-modal retrieval with cnn visual features: A new baseline," *IEEE transactions on cybernetics*, vol. 47, no. 2, pp. 449–460, 2017.

[20] Y. Cao, M. Long, J. Wang, and H. Zhu, "Correlation autoencoder hashing for supervised cross-modal search," in *Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval*, pp. 197–204. ACM, 2016.

[21] Y. Cao, M. Long, J. Wang, Q. Yang, and 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*, pp. 1445–1454. ACM, 2016.

[22] X. Li, D. Hu, and F. Nie, "Deep binary reconstruction for cross-modal hashing," *arXiv preprint arXiv:1708.05127*, 2017.

[23] Q.-Y. Jiang and W.-J. Li, "Deep cross-modal hashing," *arXiv preprint arXiv:1602.02255*, 2016.

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

[25] Q. Wang, G. Zhu, and Y. Yuan, "Statistical quantization for similarity search," *Computer Vision and Image Understanding*, vol. 124, pp. 22–30, 2014.

[26] Q. Wang, J. Gao, and Y. Yuan, "A joint convolutional neural networks and context transfer for street scenes labeling," *IEEE Transactions on Intelligent Transportation Systems*, 2017.

[27] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[28] M. Luo, X. Chang, Z. Li, L. Nie, A. G. Hauptmann, and Q. Zheng, "Simple to complex cross-modal learning to rank," *Computer Vision and Image Understanding*, vol. 163, pp. 67–77, 2017.

[29] W. Liu, J. Wang, S. Kumar, and S.-F. Chang, "Hashing with graphs," in *ICML*, pp. 1–8, 2011.

[30] F. Shen, C. Shen, W. Liu, and H. Tao Shen, "Supervised discrete hashing," in *CVPR*, pp. 37–45, 2015.

[31] F. Shen, X. Zhou, Y. Yang, J. Song, H. T. Shen, and D. Tao, "A fast optimization method for general binary code learning," *IEEE Transactions on Image Processing*, vol. 25, no. 12, pp. 5610–5621, 2016.

[32] N. Rasiwasia, J. Costa Pereira, E. Coviello, G. Doyle, G. R. Lanckriet, R. Levy, and N. Vasconcelos, "A new approach to cross-modal multimedia retrieval," in *Proceedings of the 18th ACM international conference on Multimedia*, pp. 251–260. ACM, 2010.

[33] M. J. Huiskes and M. S. Lew, "The mir flickr retrieval evaluation," in *ACM Multimedia*, pp. 39–43. ACM, 2008.

[34] 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 *Proceedings of the ACM international conference on image and video retrieval*, p. 48. ACM, 2009.

[35] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *ACM Multimedia*, pp. 675–678. ACM, 2014.

[36] J. Gui and P. Li, "R 2 sdh: Robust rotated supervised discrete hashing," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1485–1493. ACM, 2018.

Gengshen Wu is currently a Ph. D. candidate with School of Computing and Communications at Lancaster University, Lancaster, UK. Previously, he obtained his M.Sc. degree from the University of Sheffield, Sheffield, UK.

Jungong Han is a senior lecturer with the School of Computing and Communications at Lancaster University, Lancaster, U.K and was with the Department of Computer Science at Northumbria University, UK. Previously, he was a senior scientist (2012-2015) with Civolution Technology (a combining synergy of Philips CI and Thomson STS), a research staff (2010-2012) with the Centre for Mathematics and Computer Science, and a researcher (2005-2010) with the Technical University of Eindhoven in Netherlands.

Zijia Lin received his B.Sc. degree (2011) from School of Software, Tsinghua University, and Ph.D. degree (2016) from Department of Computer Science and Technology in the same campus. He is currently associate researcher in Big Data Mining Group of Microsoft Research Asia (MSRA). His research interests include multimedia information retrieval and machine learning.

Guiguang Ding received his Ph.D degree in electronic engineering from Xidian University, China, in 2014. He is currently an associate professor of School of Software, Tsinghua University. He has published 80 papers in major journals and conferences, including the IEEE TIP, TMM, TKDE, SIG IR, AAAI, ICML, IJCAI, CVPR, and ICCV. His current research centers on the area of multimedia information retrieval, computer vision and machine learning.

Baochang Zhang received the B.S., M.S., and Ph.D. degrees in computer science from the Harbin Institute of Technology, Harbin, China, in 1999, 2001, and 2006, respectively. He is currently an Associate Professor with the Science and Technology on Aircraft Control Laboratory, School of Automation Science and Electrical Engineering, Beihang University, Beijing, China.

Qiang Ni received the B.Sc., M.Sc., and Ph.D. degrees from the Huazhong University of Science and Technology, China, all in engineering. He is currently a Professor and the Head of Communication Systems Research Group, School of Computing and Communications, Lancaster University, InfoLab21, Lancaster, U.K. His research interests include future generation communications and networking systems, including green communications and networking, cloud systems, cognitive radio network systems, heterogeneous networks, 5G, SDN, IoTs, big data analytics and vehicular networks. He is a Voting Member of the IEEE 1932.1 standard. He was an IEEE 802.11 Wireless Standard Working Group Voting member and a Contributor to the IEEE Wireless Standard.