Multimedia Tools and Applications (2022) 81:15653–15670
https://doi.org/10.1007/s11042-022-12187-6

Deep fused two-step cross-modal hashing with multiple semantic supervision

Peipei Kang[1,2] · Zehang Lin[3] · Zhenguo Yang[1] · Alexander M. Bronstein[2] · Qing Li[3] · Wenyin Liu[1]

Received: 21 December 2020 / Revised: 11 July 2021 / Accepted: 10 January 2022 /
Published online: 28 February 2022
© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2022

Abstract
Existing cross-modal hashing methods ignore the informative multimodal joint information and cannot fully exploit the semantic labels. In this paper, we propose a deep fused two-step cross-modal hashing (DFTH) framework with multiple semantic supervision. In the first step, DFTH learns unified hash codes for instances by a fusion network. Semantic label and similarity reconstruction have been introduced to acquire binary codes that are informative, discriminative and semantic similarity preserving. In the second step, two modality-specific hash networks are learned under the supervision of common hash codes reconstruction, label reconstruction, and intra-modal and inter-modal semantic similarity reconstruction. The modality-specific hash networks can generate semantic preserving binary codes for out-of-sample queries. To deal with the vanishing gradients of binarization, continuous differentiable tanh is introduced to approximate the discrete sign function, making the networks able to back-propagate by automatic gradient computation. Extensive experiments on MIRFlickr25K and NUS-WIDE show the superiority of DFTH over state-of-the-art methods.

Keywords Cross-modal hashing · Deep fusion network · Semantic reconstruction · Two-step learning · Supervised learning

1 Introduction

The development of multimedia arouses people's demand for interactive multimodal tasks. For example, cross-modal retrieval (CMR) [20, 25] gives one modality of query data such as image, and retrieves semantically related data of other modalities such as text. Similar applications can be found in image captioning [27] that generates descriptive text for the given images. Cross-modal hashing (CMH) aims to learn short hash codes for cross-modal retrieval, which can be used to measure the distances among the multimodal data in the Hamming space [21]. Modality-specific hashing (MSH) and unified hashing (UH) are two

✉ Zhenguo Yang
zhengyang5-c@my.cityu.edu.hk

Extended author information available on the last page of the article.

typical strategies. MSH generates hash codes for paired image and text respectively, and aligns them by using correlation maximization or preserving semantic similarity. In contrast, UH learns common hash codes for image and text pairs jointly, aiming to model strong relationship between multimodal data.

In terms of UH, there exist one-step and two-step methods. One-step methods learn common hash codes and modality-specific hash functions simultaneously, i.e., integrating the task in one objective for optimizations. In contrast, two-step methods usually learn common hash codes and hash functions separately, and is flexible to adopt different hash functions. For instance, [24] constructed a similarity matrix between raw samples to guide the learning of hash codes, and modality-specific hash functions can be learned by reconstructing common hash codes. Chen et al. [3] formulated hash codes learning into a linear classification problem with label correlation preserved, and leaned hash functions with semantic similarity preserved in the second step. To deal with vanishing gradients during binarization in deep networks-based models, Zhang et al. [37] directly discarded the binary constraint during optimization, causing large quantization loss. Other researchers [31, 32] proposed bit-by-bit optimization, which is nevertheless time-consuming. Indeed, there are still quite some challenging issues associated with the two-step methods. (1) In the first step, rich information of an instance is expected to be exploited hence hash codes can be more informative. (2) The semantic labels have not been fully exploited during the two-step training. (3) Linear hash functions cannot capture nonlinear relationships. (4) The optimization for discrete binarization is not flexible in deep networks.

In this paper, we propose a novel supervised two-step framework, termed as deep fused two-step cross-modal hashing (DFTH) with multiple semantic supervision. In the first step, DFTH learns unified hash codes by reconstructing semantic label and semantic similarity. More specifically, paired samples from different modalities are fused through a fusion network, and are encoded as common hash codes that are constrained by reconstructing semantic similarity matrix in a batch manner. Meanwhile, these codes are transformed to align the semantic labels through a transformation layer. As a result, unified codes are obtained which are informative and discriminative, and preserve the semantic similarities simultaneously. In the second step, DFTH learns modality-specific hash networks, which are regularized to reconstruct label matrix, as well as the intra-modal and inter-modal semantic similarity matrix. Therefore, the learned hash networks can achieve semantic-preserving binary codes for out-of-sample queries. Furthermore, to tackle the vanishing gradients of the discrete binarization and make the optimizations more flexible, we replace the discrete sign function by an approximate and continuous differentiable function, i.e., scaled hyperbolic tangent (tanh) [2]. Consequently, the networks can be back-propagated by automatic gradient computation. In summary, the main contributions of this paper are listed as follows.

- We propose a novel end-to-end deep CMH framework (denoted as DFTH), which individually learns common hash codes through a fusion network and modality-specific hash networks by preserving multiple semantics.
- We introduce reconstruction constraints of semantic similarity and semantic label to jointly learn common hash codes that are informative, discriminative and semantic similarity preserving for multimodal data.
- We adopt a continuous function (scaled tanh) instead of the discrete sign function, which makes the binarization optimization automatic and flexible in neural networks.
- We conduct extensive experiments on two public datasets, which demonstrates that DFTH achieves significant performance even with small amounts of training data and hash codes in short bits.

The paper is organized as follows. Section 1 introduces the background of cross-modal hashing, especially the unified hashing. Section 2 analyzes the related techniques of unified hashing, including the one-step learning and two-step learning methods. Section 3 describes the proposed approach, including the framework, symbol definition, common hash codes learning, and modality-specific hash networks learning. Experiments such as the mAP comparison, precision-recall curves, ablation analysis, etc., are presented in Section 4, followed by the conclusion in Section 5.

2 Related work

The explosive increase of multimedia data motivates researchers to focus on cross-modal hashing techniques (CMH) [1, 12, 22] instead of retrieval by real-value representations [11, 35, 38], in order to save computation and storage costs. CMH can be categorized into modality-specific hashing (MSH) and unified hashing (UH). MSH learns modality-specific hash codes and align them via specific consideration [37], but usually weak connections between cross-modal data can be established due to the heterogeneous gap. In contrast, UH models strong connections by learning common hash codes for paired data from different modalities, which tends to be more advantageous [7, 10]. Furthermore, UH can be further divided into one-step learning and two-step learning, according to whether learning common hash codes and modality-specific hash functions simultaneously or separately. The former implements the two objectives simultaneously [6, 13, 23, 32, 34], causing suboptimal hash codes or hash functions, and the adoption of different hash functions is not flexible in this branch. Therefore, we focus on the two-step learning in this paper.

In terms of the two-step methods, they achieve hash codes firstly, and learn hash functions subsequently, which usually obtain high performance [3]. In addition, two-step methods can flexibly adopt different hash functions in the second step. For instance, semantic preserving hashing (SePH) [14] transforms the semantic affinities into a probability distribution, and approximates it with to-be-leaned hash codes in Hamming space. Supervised discrete manifold-embedded cross-modal hashing (SDMCH) [19] preserves the modality-specific manifold similarities by common latent representations in a Laplacian manner, and regresses common hash codes to the class labels. Semi-supervised semantic-preserving hashing (S3PH) [29] learns common hash codes using collective matrix factorization (CMF) [6] and semantic similarity preserving in the first step, and reconstructs the common hash codes in the next step to learn modality-specific hash functions. Similarly, scalable asymmetric discrete cross-modal hashing (BATCH) [30] also adopts CMF with treating label as a special modality. Two-step cross-modal hashing (TECH) [3] exploits the label correlation when mapping common binary codes to label space in the first step, and constrains hash codes in the second step to preserve semantic similarity. Nonlinear robust discrete hashing (NRDH) [36] reconstructs each modality of data from the common latent representation by nonlinear descriptors, and requires binary codes to preserve semantic similarity at the same time. To fully capture the heterogeneous multi-modal correlation, fusion-supervised deep cross-modal hashing (FDCH) [26] is the seminal deep cross-modal hashing work, which inputs paired cross-modal data into a fusion network, and learns unified hash codes for the instance in different modalities. Based on the unified hash codes, it trains modality-specific networks for out-of-sample queries. Dual deep neural networks cross-modal hashing (DDCMH) [4] is a tri-stage training method. Firstly, it adopts a single-modal hashing method to obtain textual binary codes, which are used as supervision for image

network training in the second stage. Finally, a textual network is trained to reconstruct the visual binary codes.

Despite great progress has been made in cross-modal retrieval, there are some challenging issues. Linear transformations are commonly adopted [3, 17, 19], which cannot capture the complex nonlinear relationship between intra-modal and inter-modal data. Therefore, deep neural networks (DNN) become popular. However, most of DNN-based CMH methods are not end-to-end, i.e., hand-crafted features such as SIFT or GIST are used as inputs [36]. The hand-crafted features are extracted without considering the specific task of CMR, e.g., the interaction and relation of different modalities, etc. In addition, the common hash codes are often extracted from separate modalities, ignoring the informative multimodal representations of instances. Furthermore, the semantic labels are not fully exploited, leading to the achieved hash codes with weak ability of semantic preserving. Last but not least, as for the vanishing gradients problem in discrete constraint, directly discarding the constraint [37] causes large quantization loss; bit-by-bit optimization [31, 32] brings in extra calculation time; and the carefully designed ADMM optimization [3, 10, 13, 15, 16, 18, 19, 23, 24, 29–31, 36] is not flexible in neural networks training, because the binary codes need specific optimization instead of the automatic gradient in back-propagation.

3 The proposed approach

The framework of the proposed DFTH is illustrated in Fig. 1. The training stage is divided into two steps, i.e., common hash codes learning (red stream) and modality-specific hash

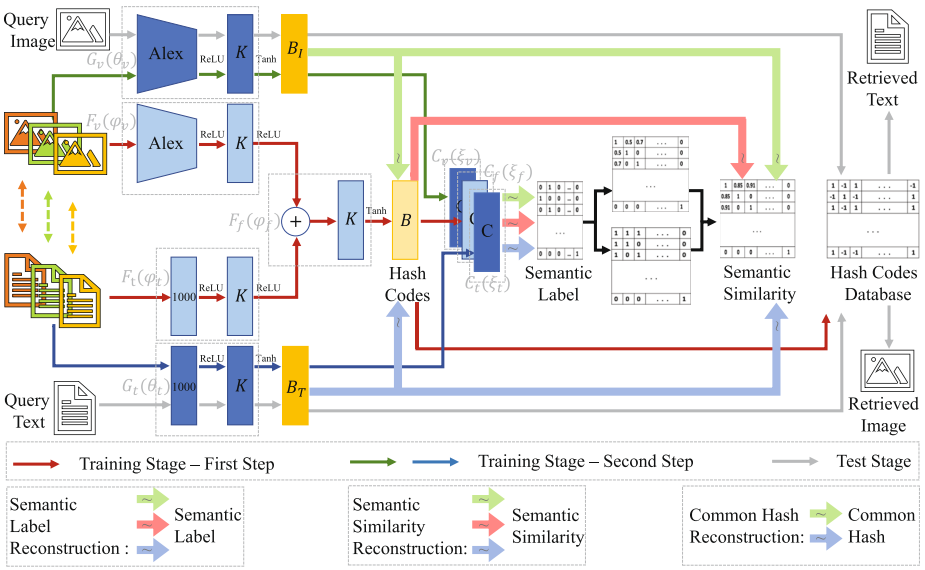


Fig. 1 The framework of DFTH. Red stream denotes the first training step to learn common hash codes, green and blue streams denote the second training step to learn modality-specific hash networks, and grey streams denote retrieval in the test stage. '⊕' represents the row-wise concatenation of vectors. Thick lines with the symbol '∼' represent different constraints. For example, the thick line with the symbol '∼' to Semantic Similarity represents the semantic similarity reconstruction constraint

networks learning (green and blue streams). We give the notations first, and then introduce each step in details.

3.1 Preliminaries

As DFTH is trained batch-wisely, the notations below are defined in a batch manner. Assume that we have n cross-modal instances $O = \{o_i\}_{i=1}^{n}$ in a batch for training, where $o_i = (v_i, t_i, y_i)$ denotes the i-th instance. v_i is the raw image feature, $t_i \in R^{1\times d_t}$ is the bag-of-words text feature where d_t equals to the number of words. $y_i \in R^{1\times c}$ is the semantic label where c is the number of classes, and $y_{ij} = 1$ if o_i belongs to the j-th class, otherwise, $y_{ij} = 0$. Similarly, $V = [v_1, ..., v_n] \in R^{n\times 224\times 224\times 3}$, $T = [t_1, ..., t_n] \in R^{n\times d_t}$, and $Y = [y_1, ..., y_n] \in R^{n\times c}$ denote the image feature matrix, text feature matrix, and the semantic label matrix, respectively. In the first step, DFTH learns common hash codes $B = \{b_i\}_{i=1}^{n} \in \{-1, +1\}^{n\times K}$ through a fusion network $F(V; T; \varphi)$ where K is the code length, and φ are the parameters of F. In the second step, DFTH aims to learn two modality-specific hash networks $G_v(V; \theta_v)$ and $G_t(T; \theta_t)$ for out-of-sample queries, where θ_v and θ_t represent the parameters of G_v and G_t respectively. Then we can get modality-specific hash codes $B_I \in \{-1, +1\}^{n\times K}$ and $B_T \in \{-1, +1\}^{n\times K}$ for image and text, respectively. Besides, we also use three linear layers, i.e., $C_f(B, \xi_f)$, $C_v(B_I, \xi_v)$, and $C_t(B_T, \xi_t)$, as separate transformations for class label reconstruction.

3.2 Common hash codes learning

As shown in Fig. 1, DFTH firstly learns unified hash codes for instances by inputting raw image and text features into a fusion network F. Specifically, images are put into a sub-network $F_v(V; \varphi_v)$ to get the latent representations $H_v \in R^{n\times K}$. Similarly, latent text representations $H_t \in R^{n\times K}$ are obtained by the subnetwork $F_t(T; \varphi_t)$. As different modalities describe an instance in different views, existing works that learn hash codes from single modality is unilateral. Therefore, we make full use of the multimodal information of instances, and get comprehensive representations $H \in R^{n\times 2K}$ by concatenating image and text latent representations. As a result, unified hash codes B are obtained through a sub-fusion network $F_f(H; \varphi_f)$, which can be formalized as follows:

$$U = F_f(F_v(V; \varphi_v) \oplus F_t(T; \varphi_t); \varphi_f), \tag{1}$$

and

$$B = sgn(U), \tag{2}$$

where $\oplus$ denotes the row-wise concatenation operation, U is the output representation matrix from F_f, and $sgn(\cdot)$ is the sign activation function.

As the discrete sign activation function may cause ill-posed gradient problem, i.e., the gradients of the sign function are zeros for all nonzero inputs, we adopt a scaled tanh activation function [8, 22, 33] instead:

$$B = tanh(\lambda U), \lambda \in R^{+}. \tag{3}$$

With the increase of λ during training, the scaled tanh function will converge to the original sign function, i.e., $lim_{\lambda\to\infty} tanh(\lambda x) = sgn(x)$. Therefore, the discrete binarization is replaced by a continuous function, and the networks can be back-propagated by automatic gradient. It is noted that a sign activation function is added at the end of the networks after training to obtain rigorous binary hash codes.

The quality of common hash codes will directly affect the hash networks learning and the retrieval performance. Therefore, multiple semantic supervision are imposed on the common hash codes to make them semantically aligned.

Semantic similarity reconstruction Cross-modal retrieval is based on similarity search that needs to quantify the degree of similarity between vectorized objects, and how to define the similarity quantizer is important and challenging. For example, Wang et al. [28] proposed a novel quantizer for similarity search by formulating a generalized likelihood ratio analysis. In the Hamming space, the retrieval for CMH is achieved based on comparisons in the Hamming distance, i.e., if two hash codes are close, they are deemed to be semantically similar. The Hamming distance between two hash codes is calculated by

$$D_H(b_i, b_j) = \frac{K - < b_i, b_j >}{2}, \tag{4}$$

where $< \cdot, \cdot >$ denotes the inner product, and higher inner product corresponds to smaller Hamming distance, indicating higher semantic similarity. In addition, the cosine similarity between two hash codes can be defined as

$$sim(b_i, b_j) = \frac{< b_i, b_j >}{||b_i||_2 \cdot ||b_j||_2}. \tag{5}$$

The matrix form $Sim(B, B) \in R^{n \times n}$ is defined as follows,

$$Sim(B, B)_{ij} = sim(b_i, b_j). \tag{6}$$

Comparing (4) with (5), it is not difficult to find that after normalizing the hash codes, the cosine similarity indicates the semantic similarity. This phenomenon inspired us to train hash codes that can reconstruct semantic similarity, so that the learned hash will achieve better retrieval performance. We construct the semantic similarity considering two aspects.

On one hand, we take into account the retrieval label, or semantically similar label. Given a query in the context of CMR, semantically similar samples are expected to be returned. Furthermore, two samples are regarded as semantically similar when they belong to at least one class. The semantically similar label matrix $S \in \{0, 1\}^{n \times n}$ is defined as below.

$$S_{ij} = \begin{cases} 1, & \textit{if } o_i \textit{ and } o_j \textit{ share at least one class}, \\ 0, & \textit{otherwise}. \end{cases} \tag{7}$$

On the other hand, although samples sharing at least one common class are seemed as semantically similar, their semantic similarities can be different. Intuitively, samples that have 5 common classes should be more similar than those have 1 common class. Therefore, we treat the normalized semantic labels as data representations, and model the semantically similar structure $Z \in [0, 1]^{n \times n}$ by the cosine similarity between labels, which is defined as follows:

$$Z = Sim(Y, Y). \tag{8}$$

Combining these two aspects through a weighting scheme with a hyper-parameter γ, we obtain the desired semantic similarity $\hat{S}$, which is reconstructed by common hash codes, formulated as,

$$Q_1 = \frac{1}{n^2} ||Sim(B, B) - \hat{S}||_F, \tag{9}$$

where $\hat{S} = \gamma S + (1-\gamma)Z$, and $||\cdot||_F$ denotes the Frobenius norm. As a result, semantically similar samples will be pulled close in the Hamming space, while semantically dissimilar samples will be pushed away.

Semantic label reconstruction To distinguish hash codes of different classes and enhance the discriminative ability, common hash codes are required to reconstruct semantic labels, which is defined as follows:

$$Q_2 = \frac{1}{n \times c}||C_f(B, \xi_f) - Y||_F. \tag{10}$$

Therefore, the overall objective for training in the first step is defined as follows.

$$Q = Q_1 + \alpha Q_2, \tag{11}$$

where α is a hyper-parameter. The fusion networks can be easily optimized by batch-wise SGD with automatic gradient computation. To obtain rigorous binary hash codes, a discrete sign activation function is added at the end of the networks after training as shown in Algorithm 1.

3.3 Modality-specific hash networks learning

For encoding out-of-sample queries in the test stage, DFTH learns modality-specific hash networks $G_v(V; \theta_v)$ and $G_t(T; \theta_t)$ in the second training step. As shown in Fig. 1, the modality-specific networks structure is the same as the sub-network structure in the first step before multimodal fusion. Besides, we adopt the same strategy to obtain hash codes during training, i.e., replacing original sign activation function with a scaled tanh activation function, so that modality-specific hash codes B_I and B_T can be obtained as follows:

$$\begin{cases} B_I = tanh(\lambda G_v(V; \theta_v)), \\ B_T = tanh(\lambda G_t(T; \theta_t)). \end{cases} \tag{12}$$

To make modality-specific hash networks achieve high-quality and cross-modal aligned hash codes, B_I and B_T are constrained to be close to the common hash codes B, known as the common hash reconstruction in Fig. 1:

$$J_1 = \frac{1}{n \times K}(||B_I - B||_F + ||B_T - B||_F). \tag{13}$$

Moreover, semantic similarity is expected to be preserved in both intra-modal and inter-modal hash codes, and the semantic similarity reconstruction in the second step is defined as:

$$\begin{aligned} J_2 = \frac{1}{n^2}(&||Sim(B_I, B_I) - \hat{S}||_F + ||Sim(B_T, B_T) - \hat{S}||_F \\ &+ ||Sim(B_I, B_T)) - \hat{S}||_F), \end{aligned} \tag{14}$$

where the first two terms in J_2 are imposed on intra-modal hash codes, while the last term is on inter-modal hash codes.

Similar to the first step, semantic label reconstruction is also constrained on modality-specific hash codes as in (15), in order to learn modality-specific networks that can generate discriminative hash codes.

$$J_3 = \frac{1}{n \times c}(||C_v(B_I, \xi_v) - Y||_F + ||C_t(B_T, \xi_t) - Y||_F). \tag{15}$$

The overall objective for learning modality-specific hash networks in the second step is,

$$J = J_1 + \beta_1 J_2 + \beta_2 J_3, \tag{16}$$

where β_1 and β_2 are two hyper-parameters. After training, the modality-specific hash networks can be used for encoding out-of-sample queries by adding a sign function.

For completeness, the whole procedure of DFTH is summarized in Algorithm 1.

Algorithm 1 DFTH.

Input: Training set O_{all} with N instances, parameters γ, α, β_1, and β_2, code length K, batch size n.

Output: Common hash codes B_{all}, modality-specific hash networks $sign(G_v(\theta_v))$ and $sign(G_t(\theta_t))$.

Initialize the fusion networks $F(\varphi)$ including $F_v(\varphi_v)$, $F_t(\varphi_t)$, and $F_f(\varphi_f)$, modality-specific hash networks $G_v(\theta_v)$ and $G_t(\theta_t)$, transformation layers $C_f(\xi_f)$, $C_v(\xi_v)$, and $C_t(\xi_t)$;

`/* First training step */`

while *not converge* **do**

  for $t = 1$ *to* $\lfloor \frac{N}{n} \rfloor$ **do**

    $\lambda = \sqrt{t}$;

    Construct a batch O by randomly selecting n instances from O_{all};

    Forward-propagate F and C_f, to obtain B by (3); calculate semantic similarity $\hat{S}$;

    Calculate loss by (11); back-propagate the gradients with the chain rule and update parameters φ and ξ_f for F and C_f;

  end

end

Fix φ of F; $B_{all} = sign(F(O_{all}))$;

`/* Second training step */`

while *not converge* **do**

  for $t = 1$ *to* $\lfloor \frac{N}{n} \rfloor$ **do**

    $\lambda = \sqrt{t}$;

    Construct a batch O by randomly selecting n instances from O_{all};

    Forward-propagate F, G_v, G_t, C_v and C_t, to obtain B by (3), B_I and B_T by (12); $B = sign(B)$; calculate semantic similarity $\hat{S}$;

    Calculate loss by (16); back-propagate the gradients with the chain rule and update parameters θ_v, θ_t, ξ_v, ξ_t for G_v, G_t, C_v and C_t, respectively;

  end

end

4 Experiments

4.1 Experimental setup

To evaluate the performance of DFTH, extensive experiments are conducted on two widely used public datasets: MirFlickr25K and NUS-WIDE.

Datasets MirFlickr25K [9] contains 25,000 images with associated textual tags, and is annotated with 24 semantic labels. Following [36], we select images with more than 20

tags, and finally get 20,015 image-text pairs for the experiments. 2,000 pairs are randomly selected as query set, and the remaining 18,015 pairs as training and retrieval set. Raw pixels and Bag-Of-Words features are used as image and text inputs, respectively.

NUS-WIDE [5] includes 269,648 images with associated textual tags. Following [36], samples from 10 most frequent labels are retained, i.e., 186,577 image-text pairs are preserved for the experiments. We randomly select 2,000 pairs for querying, and the remaining 184,577 pairs for training and retrieval.

Baselines SCM [37] learns modality-specific hash codes by linear transformations; CMFH [6], DCH [32], SCRATCH [13], DCMH [10], RDCMH [18], and SSAH [12] learn common hash codes and hash functions in one step, and SCRATCH is implemented on kernel space, DCMH, RDCMH and SSAH are deep methods; SePH [14], TECH [3], NRDH (and deep-NRDH) [36], and FDCH [26] are two-step methods that learn common hash codes and hash functions separately. Moreover, NRDH adopts nonlinear descriptors, while FDCH is an end-to-end deep method fusing multimodal information, which is the closest work to our proposed DFTH. Note that the performance results of the baselines are referenced from [26, 36].

Evaluation metric We conduct two cross-modal retrieval tasks: image as a query to retrieve texts (I2T) and text as a query to retrieve images (T2I). The popular used mean average precision (mAP) and precision-recall (PR) curve are adopted to evaluate the performance. Note that higher mAP value and larger area enclosed by PR curve indicate better retrieval performance.

Implementation details The network architecture is implemented as follows. Subnetworks G_v and F_v share the same architecture, i.e., the AlexNet followed with a fully connected layer ($1000 \rightarrow K$). Subnetworks G_t and F_t share the same architecture, i.e., two fully connected layers ($d_t \rightarrow 1000 \rightarrow K$), and the subnetwork F_f is a fully connected layer ($2K \rightarrow K$). The subnetworks C_f, C_v, and C_t share the same architecture, i.e., a fully connected layer ($K \rightarrow C$). The training images were resized to $224 \times 224 \times 3$ with random horizontal flip before feeding into the networks. We fixed batch size as 32, and exploited the SGD optimizer with 0.9 momentum and 0.0005 weight decay. The learning rate for the image and fusion networks was set as 0.001, and 0.01 for the text networks. We cross-validated the hyper-parameters, and finally set $\gamma = 0.5$, $\alpha = 0.01$, and $\beta_1 = \beta_2 = 1$ for MirFlickr25K, $\gamma = 0.3$, $\alpha = 0.01$, $\beta_1 = 0.1$, and $\beta_2 = 1$ for NUS-WIDE. The proposed DFTH is implemented via PyTorch and is run on a server with two NVIDIA GeForce RTX 2080 Ti GPUs, an Intel i9-9900X 3.59GHz CPU, and a RAM of 128G.

4.2 mAP comparison

Tables 1 and 2 show the mAP comparison results on the MirFlickr25K and NUS-WIDE datasets respectively. The results of the baselines are borrowed from [36]. For the MirFlickr25K dataset, image and text are represented as 500-dimensional Bag-Of-Visual-Words feature vectors and 1000-dimensional Bag-Of-Word feature vectors respectively; while for the NUS-WIDE dataset, 512-dimensional GIST feature vectors and 1386-dimensional Bag-Of-Words feature vectors are extracted as image and text representations respectively. For the deep methods including DCMH, RDCMH, SSAH, deep-NRDH, and

Table 1 mAP comparison on MirFlickr25K

Task	Methods	8bits	16bits	32bits	64bits	128bits
I2T	CMFH	0.5599	0.5687	0.5680	0.5685	0.5687
	SCM	0.6235	0.6373	0.6478	0.6537	0.6611
	SePH	0.6641	0.6685	0.6818	0.6830	0.6873
	DCH	0.6659	0.6738	0.6859	0.6897	0.7030
	SCRATCH	0.7092	0.7131	0.7222	0.7265	0.7346
	TECH	0.7192	0.7304	0.7478	0.7553	0.7615
	DCMH	–	0.7411	0.7465	0.7485	–
	RDCMH	–	0.7723	0.7735	0.7789	–
	SSAH	–	0.7970	0.8090	0.8100	–
	NRDH	0.7224	0.7616	0.7859	0.8073	0.8120
	deep-NRDH	–	0.8023	0.8219	0.8353	–
	DFTH	**0.8446**	**0.8829**	**0.8991**	**0.9117**	**0.9147**
T2I	CMFH	0.5615	0.5615	0.5606	0.5606	0.5608
	SCM	0.6103	0.6206	0.6298	0.6372	0.6427
	SePH	0.7033	0.7076	0.7212	0.7293	0.7348
	DCH	0.7256	0.7511	0.7585	0.7681	0.7909
	SCRATCH	0.7591	0.7762	0.7822	0.7978	0.8063
	TECH	0.7778	0.8000	0.8241	0.8361	0.8449
	DCMH	–	0.7827	0.7900	0.7932	–
	RDCMH	–	0.7931	0.7924	0.8001	–
	SSAH	–	0.7820	0.7970	0.7990	–
	NRDH	0.7917	0.8233	0.8353	0.8462	0.8528
	deep-NRDH	-	0.8105	0.8208	0.8290	–
	DFTH	**0.8147**	**0.8483**	**0.8551**	**0.8710**	**0.8726**

The values in bold indicate the best performance of the methods on each task with the same specific settings

DFTH, raw pixels or 4096-dimensional VGG feature vectors are used. From Table 1 we have the following observations.

- Compared with the linear methods, nonlinear and deep methods obtain noticeable retrieval improvement. This is reasonable, because linear transformations cannot extract complex nonlinear relationship between cross-modal samples, while deep methods exploit neural networks to model complex structure.
- Compared with the one-step methods, two-step methods are competitive for the retrieval tasks, because they learn optimal common hash codes and hash functions separately for cross-modal retrieval.
- DFTH achieves the best performance. Even short hash bits can also obtain excellent retrieval performance, which is comparable to or better than other methods with long hash bits.

The significant performance of DFTH can be explained from a few aspects. Firstly, the baselines adopt shallow or VGG features as inputs, which are extracted without taking into account the retrieval task; while DFTH is an end-to-end method, which can extract features that adapt for the task, and learn correlation between different modalities. Secondly,

Table 2 mAP comparison on NUS-WIDE

Task	Methods	8bits	16bits	32bits	64bits	128bits
I2T	CMFH	0.3406	0.3437	0.3399	0.3409	0.3440
	SCM	0.5013	0.5120	0.5422	0.5488	0.5483
	SePH	0.5256	0.5537	0.5627	0.5622	0.5698
	DCH	0.5840	0.5808	0.5907	0.5932	0.5843
	SCRATCH	0.6038	0.6207	0.6338	0.6459	0.6496
	TECH	0.6262	0.6521	0.6675	0.6750	0.6715
	NRDH	0.6372	0.6601	0.6712	0.6823	0.6881
	DFTH	**0.7459**	**0.7591**	**0.7965**	**0.8039**	**0.8162**
T2I	CMFH	0.3456	0.3498	0.3435	0.3486	0.3529
	SCM	0.4709	0.4836	0.5067	0.5141	0.5161
	SePH	0.6102	0.6407	0.6515	0.6608	0.6651
	DCH	0.7106	0.7103	0.7098	0.7260	0.7223
	SCRATCH	0.7210	0.7392	0.7549	0.7680	0.7755
	TECH	0.7638	0.7865	0.8058	0.8175	0.8150
	NRDH	**0.7717**	**0.8012**	**0.8161**	**0.8253**	**0.8314**
	DFTH	0.7041	0.7242	0.7591	0.7652	0.7701

The values in bold indicate the best performance of the methods on each task with the same specific settings

DFTH learns common hash codes with rich multimodal information through a fusion network, while others learn common hash codes from each single modality. Thirdly, multiple semantic supervision are imposed on hash codes, benefiting them for preserving semantic information. Finally, the scaled tanh can approximate sign during training with small quantization loss. These also contribute to the favorable mAP of short hash bits, which is more suitable for large-scale cross-modal retrieval.

Table 2 shows the mAP comparison results on the NUS-WIDE dataset. Similar to the findings in Table 1, deep methods and two-step methods are generally superior to the linear methods and one-step methods in terms of the retrieval precision. However, there are also some different observations on the NUS-WIDE dataset.

- DFTH outperforms other baselines with a tremendous gap on the task of I2T, while achieves moderate performance on the task of T2I.
- Compared with other baselines that gain higher mAP on T2I and lower mAP on I2T, DFTH contrastively obtains higher mAP on I2T and lower mAP on T2I.

The superb results of DFTH on the task of I2T can be attributed to its end-to-end network training and multiple semantic supervision. Complex cross-modal relationship can be captured, and deep features that adapt for the retrieval tasks can be obtained. However, DFTH cannot obtain the best T2I performance on NUS-WIDE, the reason may be that the network is sensitive to noise due to its capacity to fit noise label. Actually, more than 65% [5] of tags for most of the classes on this NUS-WIDE are noise.

mAP comparison with 5K training set In order to evaluate the learning ability, we also compare DFTH with other baselines with small training set. Specifically, we only use 5,000 pairs for training that are randomly selected from the original training set, while the retrieval

Table 3 mAP comparison on MirFlickr25K and NUS-WIDE datasets with 5K training set

Task	Methods	MirFlickr25K			NUS-WIDE		
		16bits	32bits	64bits	16bits	32bits	64bits
I2T	CMFH	0.5644	0.5648	0.5666	0.3384	0.3407	0.3452
	SCM	0.5821	0.5726	0.5672	0.3858	0.3744	0.3666
	SePH	0.6534	0.6665	0.7277	0.6056	0.6223	0.6674
	DCH	0.5991	0.6025	0.6009	0.4275	0.4402	0.4526
	DCMH	0.7050	0.7194	0.7366	0.6195	0.6472	0.6721
	FDCH	0.7762	0.7877	0.7897	0.6861	0.7003	0.7139
	DFTH	**0.8143**	**0.8301**	**0.8404**	**0.7129**	**0.7463**	**0.7616**
T2I	CMFH	0.5632	0.5631	0.5639	0.3377	0.3401	0.3424
	SCM	0.5976	0.5826	0.5724	0.3858	0.3708	0.3576
	SePH	0.6135	0.6238	0.6768	0.5876	0.5747	0.6314
	DCH	0.5804	0.5981	0.5968	0.4003	0.4179	0.4428
	DCMH	0.7038	0.7048	0.7291	0.6147	0.6432	0.6651
	FDCH	0.7504	0.7555	0.7552	0.6518	0.6712	0.6834
	DFTH	**0.7826**	**0.7935**	**0.8046**	**0.6747**	**0.7053**	**0.7162**

The values in bold indicate the best performance of the methods on each task with the same specific settings

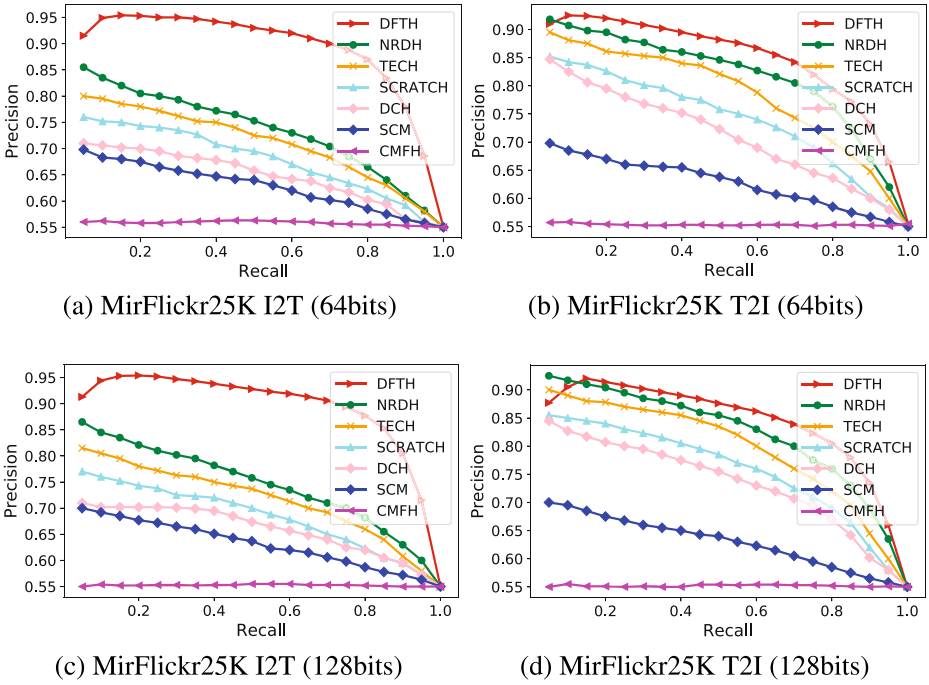
set and the query set keep the same. From Table 3 , we can find that DFTH can achieve comparable performance even with small amounts of training set, which reflects the strong learning ability of DFTH. Besides, although FDCH is also an end-to-end fusion framework, DFTH outperforms FDCH dramatically. The reasons may be in two aspects. Firstly, FDCH fuses multiple data by directly adding them together, which may lose their characteristic information, while DFTH concatenates them together without information loss. Secondly, DFTH adopts multiple semantic supervision, making hash codes to preserve more semantic similarity information.

4.3 Precision-recall curves

For evaluations, we also plot the precision-recall (PR) curves of DFTH and the baselines. Figure 2 shows the PR curves in respect of 64 bits and 128 bits on the MirFlickr25K dataset, and Fig. 3 shows the PR curves on the NUS-WIDE dataset. It is obvious that the areas enclosed by the DFTH curves are the largest on both tasks and datasets for most cases, which is consistent with the mAP comparison, and indicates the superiority of DFTH (Table 4).

4.4 Ablation analysis

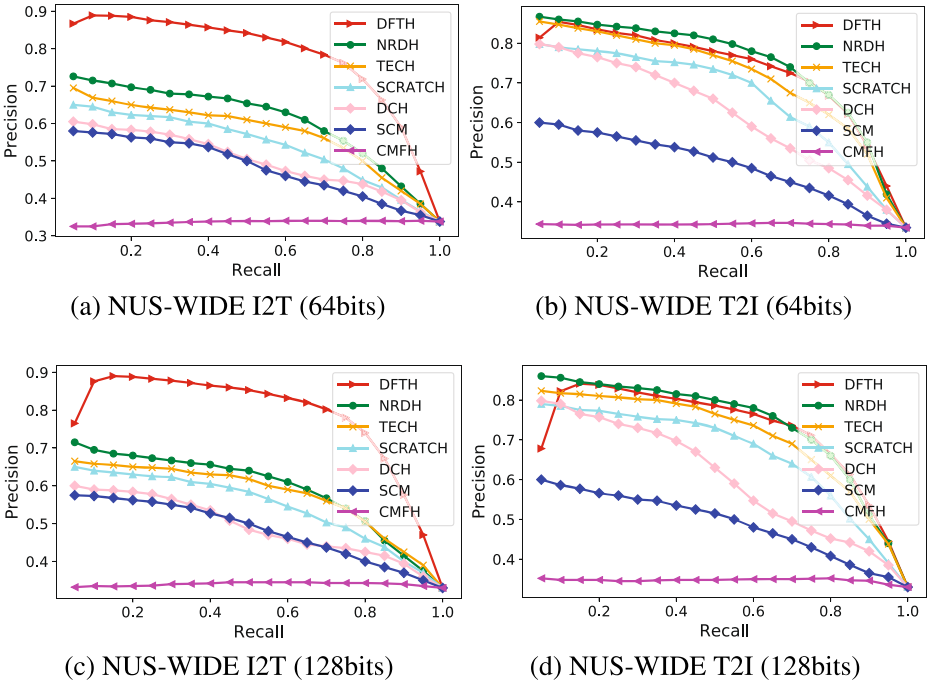
In order to evaluate the contributions of different semantic supervision, we report the mAP performance of the variations of DFTH on MirFlickr25K, including DFTH_RS, DFTH_RL, DFTH_RF, DFTH_RT, DFTH_S, DFTH_Z. In details, DFTH_RS denotes DFTH that removes the semantic similarity reconstruction constraint; DFTH_RL is the DFTH that removes the label reconstruction constraint; DFTH_RF removes the fusion networks as well as the common hash codes reconstruction from DFTH, and directly trains modality-specific



(a) MirFlickr25K I2T (64bits) (b) MirFlickr25K T2I (64bits)

(c) MirFlickr25K I2T (128bits) (d) MirFlickr25K T2I (128bits)

Fig. 2 Comparison of Precision-Recall curves generated by different methods on MirFlickr25K dataset



(a) NUS-WIDE I2T (64bits) (b) NUS-WIDE T2I (64bits)

(c) NUS-WIDE I2T (128bits) (d) NUS-WIDE T2I (128bits)

Fig. 3 Comparison of Precision-Recall curves generated by different methods on NUS-WIDE dataset

Table 4 mAP comparison between DFTH and its variations on MirFlickr25K

Task	Methods	8bits	16bits	32bits	64bits
I2T	DFTH_RS	0.7836	0.7927	0.7941	0.8065
	DFTH_RL	**0.8525**	0.8811	0.8931	0.9082
	DFTH_RF	0.8229	0.8432	0.8704	0.8832
	DFTH_RT	0.8077	0.8714	0.8864	**0.9119**
	DFTH_S	0.8480	0.8705	0.8912	0.8965
	DFTH_Z	0.8226	0.8121	0.8346	0.8524
	DFTH	0.8446	**0.8829**	**0.8991**	0.9117
T2I	DFTH_RS	0.7565	0.7579	0.7528	0.7374
	DFTH_RL	**0.8168**	0.8445	0.8542	0.8667
	DFTH_RF	0.7966	0.8249	0.8430	0.8523
	DFTH_RT	0.7715	0.8319	0.8458	**0.8721**
	DFTH_S	0.8134	0.8412	**0.8563**	0.8605
	DFTH_Z	0.7905	0.7832	0.8047	0.8136
	DFTH	0.8147	**0.8483**	0.8551	0.8710

The values in bold indicate the best performance of the methods on each task with the same specific settings

networks for each modality. The training is under the supervision of semantic similarity reconstruction and label reconstruction; To speculate the effect of the scaled tanh, we also design DFTH_RT that removes the scaled tanh function during training, and utilizes the mean-threshold binarization for test; DFTH_S and DFTH_Z utilize extreme semantic similarity during training, i.e., $\gamma = 1$ and $\gamma = 0$, respectively. We have the following observations and analyses.

- Semantic similarity reconstruction is the most useful supervision, because the objective corresponds closely to the retrieval task.
- Comparing DFTH_RF with DFTH, we find that common hash codes learning with multimodal fusion works better than the modality-specific hash codes learning. In addition, DFTH_RF also obtains excellent results, which indicates the effectiveness of the multiple semantic supervision.
- Comparing DFTH_S, DFTH_Z with DFTH, we observe that both kinds of semantic similarity reconstructions benefit the retrieval tasks, and combining them together obtains the best performance since both semantically similar label and similarity structure are taken into account.
- Label reconstruction contributes less compared with the semantic similarity reconstruction, and even decreases the performance of short hash codes, because the former focuses on enhancing the discriminative ability, which may be too strict for less informative short hash codes.
- During training, replacing the discrete sign function with the scaled tanh function can generate binary codes with less quantization. However, with the increase of hash bits, using mean-threshold binarization directly can achieve better results, which is exciting and enlightening. This phenomenon can be explained as longer hash bits can preserve richer information.

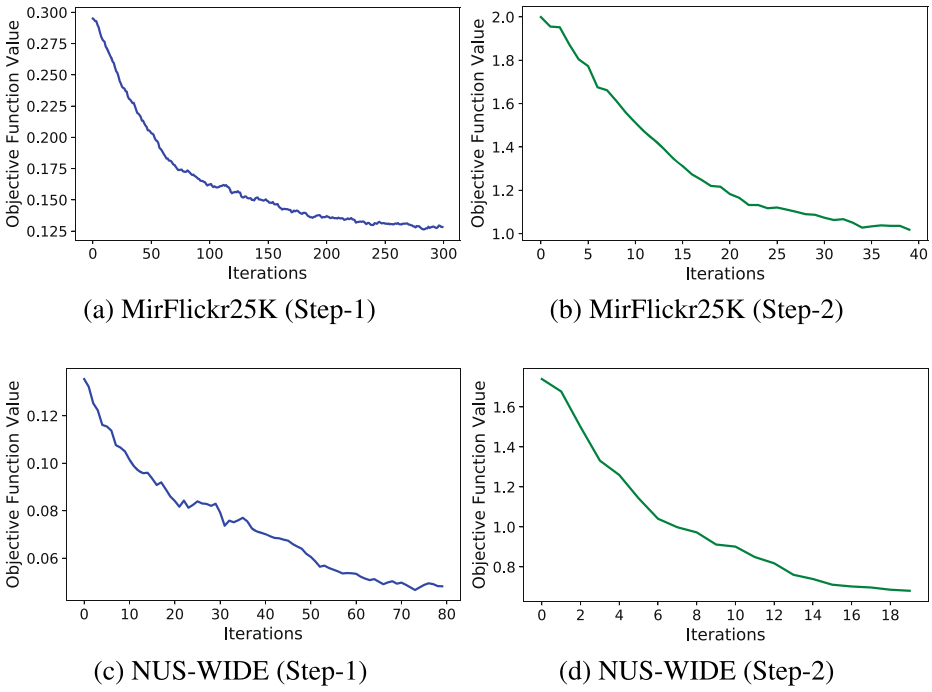


Fig. 4 Objective function values versus iterations on MirFlickr25K and NUS-WIDE datasets

4.5 Convergence analysis

Figure 4 shows the variation of objective function values along with the training iterations on the MirFlickr25K and NUS-WIDE datasets. DFTH consists of two-step training, and the objectives are as (11) and (16) respectively. It can be seen that the objectives converge fast on both datasets. Comparing with the step-1 training, the step-2 objective function values decrease quickly along with the increase of training iterations. Besides, the objective function values converge faster on the NUS-WIDE dataset than on the MirFlickr25K dataset.

4.6 Parameter sensitivity

There are four hyper-parameters in the objective of DFTH including γ, β_1, α, and β_2, which control the importance of semantically similar label matrix, semantic label reconstruction in step-1, semantic similarity reconstruction in step-2, and semantic label reconstruction in step-2, respectively. We choose the parameter γ first with the other three parameters fixed as 0, and then choose the parameter β_1, and fix the value of γ and the other two parameters as 0, and so on. Figure 5 shows the mAP variation in results of the variation of parameters on the MirFlickr25K dataset. We can observe that these curves are mostly flat, which illustrates the insensitivity to parameters. Moreover, parameter γ influences the results relatively more than the other three parameters, which demonstrates the importance of the semantically similar label matrix.

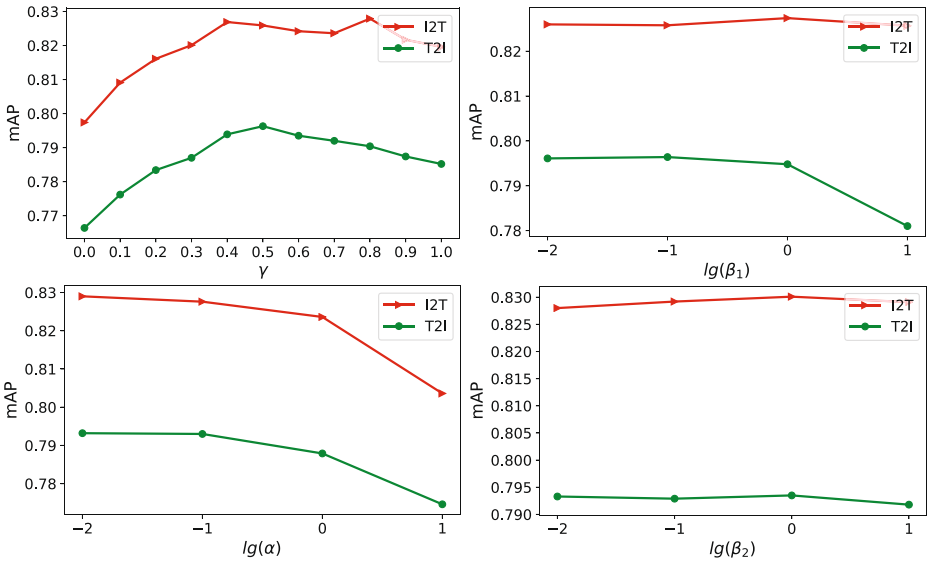


Fig. 5 mAP variation in results of the variation of parameters on the MirFlickr25K dataset

5 Conclusion

In this paper, we proposed a novel end-to-end deep fusion framework for cross-modal retrieval, named as DFTH. DFTH took into account rich multimodal information to jointly learn common hash codes through a fusion network, making the codes more informative. Furthermore, it constrained the common hash codes to reconstruct the semantic label and semantic similarity matrix, which enhanced the discriminative and semantic similarity preserving abilities of common hash codes. In the second step, modality-specific hash codes reconstructed the common hash codes, the semantic similarity matrix, and the semantic label simultaneously, which were beneficial for learning modality-specific networks. During training, DFTH adopted the scaled tanh function to replace the discrete sign function, which could approximate the binarization operation well and be more flexible to implement the networks. Extensive experiments showed that DFTH can achieve more informative and semantic preserving binary codes that are superior to state-of-the-art methods for cross-modal retrieval.

Acknowledgements This work is supported by the National Natural Science Foundation of China (No. 62076073, No.61902077, No.62006048, No.61772141, and No.61972102), the Guangdong Basic and Applied Basic Research Foundation (No. 2020A1515010616), Science and Technology Program of Guangzhou (No.202102020524, No.201903010107, No.201802010042), the Guangdong Innovative Research Team Program (No.2014ZT05G157), Special Funds for the Cultivation of Guangdong College Students' Scientific and Technological Innovation (pdjh2020a0173), and the Key-Area Research and Development Program of Guangdong Province (2019B010136001), and the Science and Technology Planning Project of Guangdong Province (No.LZC0023, No.2019B020208001 and No.2019B110210002).

Declarations

Conflict of Interests The authors declare that they have no conflict of interest.

References

1. Cao Y, Liu B, Long M, Wang J (2018) Cross-modal hamming hashing. In: ECCV, pp 202–218
2. Cao Z, Long M, Wang J, Yu PS (2017) Hashnet: Deep learning to hash by continuation. In: ICCV, pp 5608–5617
3. Chen ZD, Wang Y, Li H, Luo X, Nie L, Xu X (2019) A two-step cross-modal hashing by exploiting label correlations and preserving similarity in both steps. In: ACMMM, pp 1694–1702
4. Chen ZD, Yu WJ, Li C, Nie L, Xu X (2018) Dual deep neural networks cross-modal hashing. In: AAAI, pp 274–281
5. Chua TS, Tang J, Hong R, Li H, Luo Z, Zheng Y (2009) Nus-wide: a real-world web image database from national university of singapore. In: CIVR, pp 1–9
6. Ding G, Guo Y, Zhou J, Gao Y (2016) Large-scale cross-modality search via collective matrix factorization hashing. TIP 25(11):5427–5440
7. Erin Liong V, Lu J, Tan YP, Zhou J (2017) Cross-modal deep variational hashing. In: ICCV, pp 4077–4085
8. Hu D, Nie F, Li X (2018) Deep binary reconstruction for cross-modal hashing. TMM 21(4):973–985
9. Huiskes MJ, Lew MS (2008) The mir flickr retrieval evaluation. In: ACMMM, pp 39–43
10. Jiang QY, Li W (2017) Deep cross-modal hashing. In: CVPR, pp 3232–3240
11. Kang P, Lin Z, Yang Z, Fang X, Li Q, Liu W (2019) Deep semantic space with intra-class low-rank constraint for cross-modal retrieval. In: ICMR, pp 226–234
12. Li C, Deng C, Li N, Liu W, Gao X, Tao D (2018) Self-supervised adversarial hashing networks for cross-modal retrieval. In: CVPR, pp 4242–4251
13. Li C, Chen ZD, Zhang PF, Luo X, Nie L, Zhang W, Xu X (2018) Scratch: A scalable discrete matrix factorization hashing for cross-modal retrieval. In: ACMMM, pp 1–9
14. Lin Z, Ding G, Hu M, Wang J (2015) Semantics-preserving hashing for cross-view retrieval. In: CVPR, pp 3864–3872
15. Liu H, Ji R, Wu Y, Huang F, Zhang B (2017) Cross-modality binary code learning via fusion similarity hashing. In: CVPR, pp 7380–7388
16. Liu X, Hu Z, Ling H, Cheung Ym (2019) Mtfh: a matrix tri-factorization hashing framework for efficient cross-modal retrieval TPAMI
17. Liu X, Nie X, Zeng W, Cui C, Zhu L, Yin Y (2018) Fast discrete cross-modal hashing with regressing from semantic labels. In: ACMMM, pp 1662–1669
18. Liu X, Yu G, Domeniconi C, Wang J, Ren Y, Guo M (2019) Ranking-based deep cross-modal hashing. In: AAAI, pp 4400–4407
19. Luo X, Yin XY, Nie L, Song X, Wang Y, Xu X (2018) Sdmch: supervised discrete manifold-embedded cross-modal hashing. In: IJCAI, pp 2518–2524
20. Peng Y, Huang X, Zhao Y (2017) An overview of cross-media retrieval: concepts, methodologies, benchmarks, and challenges. TCSVT 28(9):2372–2385
21. Shen HT, Liu L, Yang Y, Xu X, Huang Z, Shen F, Hong R (2020) Exploiting subspace relation in semantic labels for cross-modal hashing TKDE
22. Su S, Zhong Z, Zhang C (2019) Deep joint-semantics reconstructing hashing for large-scale unsupervised cross-modal retrieval. In: ICCV, pp 3027–3035
23. Wang D, Wang Q, An Y, Gao X, Tian Y (2020) Online collective matrix factorization hashing for large-scale cross-media retrieval. In: SIGIR, pp 1409–1418
24. Wang H, Chen H, Meng M, Wu J (2019) Robust multi-view hashing for cross-modal retrieval. In: ICME, pp 1012–1017
25. Wang K, Yin Q, Wang W, Wu S, Wang L (2016) A comprehensive survey on cross-modal retrieval. arXiv:1607.06215
26. Wang L, Zhu L, Yu E, Sun J, Zhang H (2019) Fusion-supervised deep cross-modal hashing. In: ICME, pp 37–42
27. Wang Q, Huang W, Zhang X, Li X (2020) Word-sentence framework for remote sensing image captioning. IEEE Transactions on Geoscience and Remote Sensing
28. Wang Q, Zhu G, Yuan Y (2014) Statistical quantization for similarity search. Comput Vis Image Underst 124:22–30
29. Wang X, Liu X, Hu Z, Wang N, Fan W, Du JX (2019) Semi-supervised semantic-preserving hashing for efficient cross-modal retrieval. In: ICME, pp 1006–1011
30. Wang Y, Luo X, Nie L, Song J, Zhang W, Xu X (2020) Batch: a scalable asymmetric discrete cross-modal hashing TKDE
31. Wu G, Lin Z, Han J, Liu L, Ding G, Zhang B, Shen J (2018) Unsupervised deep hashing via binary latent factor models for large-scale cross-modal retrieval. In: IJCAI, pp 2854–2860

32. Xu X, Shen F, Yang Y, Shen HT, Li X (2017) Learning discriminative binary codes for large-scale cross-modal retrieval. TIP 26(5):2494–2507
33. Yang D, Wu D, Zhang W, Zhang H, Li B, Wang W (2020) Deep semantic-alignment hashing for unsupervised cross-modal retrieval. In: ICMR, pp 44–52
34. Yang E, Deng C, Liu W, Liu X, Tao D, Gao X (2017) Pairwise relationship guided deep hashing for cross-modal retrieval. In: AAAI, pp 1618–1625
35. Yang Z, Lin Z, Kang P, Lv J, Li Q, Liu W (2020) Learning shared semantic space with correlation alignment for cross-modal event retrieval. TOMM 16(1):1–22
36. Yang Z, Long J, Zhu L, Huang W (2020) Nonlinear robust discrete hashing for cross-modal retrieval. In: SIGIR, pp 1349–1358
37. Zhang D, Li W (2014) Large-scale supervised multimodal hashing with semantic correlation maximization. In: AAAI, pp 2177–2183
38. Zhen L, Hu P, Wang X, Peng D (2019) Deep supervised cross-modal retrieval. In: CVPR, pp 10,394–10,403

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Affiliations

Peipei Kang[1,2] · Zehang Lin[3] · Zhenguo Yang[1] · Alexander M. Bronstein[2] · Qing Li[3] · Wenyin Liu[1]

✉ Wenyin Liu
liuwy@gdut.edu.cn

Peipei Kang
ppkanggdut@126.com

Zehang Lin
cszhlin@outlook.com

Alexander M. Bronstein
bron@cs.technion.ac.il

Qing Li
csqli@comp.polyu.edu.hk

[1] School of Computer Science and Technology, Guangdong University of Technology, Guangzhou, China

[2] Computer Science Department, Technion - Israel Institute of Technology, Haifa, Israel

[3] Department of Computing, Hong Kong Polytechnic University, Hong Kong, China