MESH: A Flexible Manifold-Embedded Semantic Hashing for Cross-Modal Retrieval

FANGMING ZHONG\textsuperscript{1,}, (Member, IEEE), GUANGZE WANG\textsuperscript{1,}, ZHIKUI CHEN\textsuperscript{1,2,} (Senior Member, IEEE), AND FENG XIA\textsuperscript{1,2,} (Senior Member, IEEE)

\textsuperscript{1}School of Software Technology, Dalian University of Technology, Dalian 116620, China
\textsuperscript{2}Key Laboratory for Ubiquitous Network and Service Software of Liaoning Province, Dalian University of Technology, Dalian 116620, China

Corresponding author: Fangming Zhong (fmzhong@dlut.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFC0831305.

ABSTRACT Hashing based methods for cross-modal retrieval has been widely explored in recent years. However, most of them mainly focus on the preservation of neighborhood relationship and label consistency, while ignore the proximity of neighbors and proximity of classes, which degrades the discrimination of hash codes. And most of them learn hash codes and hashing functions simultaneously, which limits the flexibility of algorithms. To address these issues, in this article, we propose a two-step cross-modal retrieval method named Manifold-Embedded Semantic Hashing (MESH). It exploits Local Linear Embedding to model the neighborhood proximity and uses class semantic embeddings to consider the proximity of classes. By so doing, MESH can not only extract the manifold structure in different modalities, but also can embed the class semantic information into hash codes to further improve the discrimination of learned hash codes. Moreover, the two-step scheme makes MESH flexible to various hashing functions. Extensive experimental results on three datasets show that MESH is superior to 10 state-of-the-art cross-modal hashing methods. Moreover, MESH also demonstrates superiority on deep features compared with the deep cross-modal hashing method.

INDEX TERMS Cross-modal hashing, manifold embedding, semantic, discrete optimization.

I. INTRODUCTION

Due to its significant role in many real-world applications, cross-modal retrieval has been widely studied in recent years [1]–[4]. Among the existing methods for cross-modal retrieval, the hashing based methods (cross-modal hashing) are the most representative ones and have been investigated intensively. Therefore, cross-modal hashing is a class of methods for implementing cross-modal retrieval. Cross-modal hashing maps data from different modalities onto a common Hamming space, where cross-modal retrieval can be performed by using fast XOR operation [5], [6].

A large number of cross-modal hashing methods have been proposed recently, which consist of shallow learning [7]–[10] and deep learning methods [11]–[13]. All these methods have made significant efforts on improving the performance of cross-modal retrieval. Although the deep learning based methods usually obtain superior performance, they are sample-intensive and require large amounts of labelled training data. Therefore, many state-of-the-art methods are still shallow learning models, which are the main concerns of this article.

Most of the previous shallow learning methods can be categorized into two groups according to the utilization of class labels, i.e., unsupervised and supervised methods. The unsupervised cross-modal hashing methods discard the label information and construct hashing functions by preserving the intrinsic structure of data [14]–[16]. For example, Collective Matrix Factorization Hashing (CMFH) [17] is proposed based on matrix factorization to learn the unified hash codes for different modalities. Fusion Similarity Hashing [18] explicitly embeds the graph-based fusion similarity across modalities into a common Hamming space, in which an undirected asymmetric graph is constructed to model the fusion similarity, and binary codes are learned with a graph hashing scheme.

The supervised ones exploit the explicit semantic labels of training data to enhance the correlation across different modalities for boosting cross-modal retrieval. For example, several supervised extensions based on matrix factorization have been proposed, such as supervised matrix factorization hashing [9], [19]. In particular, Intra- and Inter-modality
Similarity Preserving Hashing [9] combines matrix factorization and maintains similarity preserving under the low-dimensional Hamming space. It is worth noting that the above methods generate binary codes by relaxing the discrete constraints, which leads to a large quantization error. To address this issue, many studies propose to learn binary codes with discrete optimization [7], [20], [21]. For instance, Discrete cross-modal hashing (DCH) [5] was proposed where binary codes are directly learned without relaxation, and label information is used to enhance the discrimination of binary codes through linear classifiers. Cross-modal Discrete Hashing (CMDH) [10] learns a shared set of binary codes for samples from different modalities based on their semantic affinities.

As mentioned above, although tremendous efforts have been made, there still remains some issues that need to be further investigated. Most of them mainly focus on the preservation of neighborhood relationship while ignore the proximity across neighbors. Usually, the $k$-nearest neighbors are selected to preserve intra-modality relationship i.e. neighborhood relationship in the previous methods. However, they treat the $k$-nearest neighbors as the same, in other words, they ignore the proximity among these $k$-nearest neighbors. It cannot be distinguished which neighbor is closer or which one is farther away. In such case, sufficient discrimination cannot be guaranteed for cross-modal research in hash codes. Several works [22], [23] have noticed the importance of proximity across neighbors and employed manifold structure information to preserve it. Although they have addressed the issue of relationship preserving to some degree, they neglect the proximity across classes. This is because they preserve the inter-modality relationship via one-hot label matrix, which neglects the distribution of class semantic embeddings. In the Hamming space, samples belonging to different classes should be mapped further apart and the same class should be clustered together. However, such scheme is not strong enough to describe the semantic distribution of different classes. For those similar classes, they should be closer than dissimilar classes. Therefore, neglecting the proximity across classes will limit the semantic information embedded in hash codes.

To address these two issues, in this article, we propose a two-step cross-modal retrieval method named Manifold-Embedded Semantic Hashing (MESH). The framework of MESH is shown in Fig. 1, in which the solid lines indicate the training procedure, and the dash lines depict the testing procedure. In the first step, the unified hash codes of training data are learned. Specifically, MESH exploits Local Linear Embedding to model the neighborhood proximity and uses class semantic embeddings to compute inter-modality similarity to formulate the proximity across classes. In addition, we inversely regress the class semantic embeddings to correspond to hash codes to further improve the discrimination. In the second step, hashing functions using linear regression are learned based on unified hash codes. Therefore, MESH can not only extract the manifold structure in different modalities, but also can embed the class semantic information into hash codes to further improve the discrimination of learned hash codes. Extensive experimental results on three datasets show that MESH outperforms 10 state-of-the-art cross-modal hashing methods, especially on the task of Image-to-Text.

The main contributions of this work are summarized as follows:

- A two-step cross-modal hashing method is proposed by leveraging manifold structure to capture the neighborhood proximity, which can improve the discrimination of hash codes. Moreover, the two-step scheme makes MESH flexible to various hashing functions.
- Class semantic embeddings are introduced to extract the proximity of classes to further augment the semantic information of hash codes, which is beneficial to improve the performance on Image-to-Text task in cross-modal retrieval.
Different from most of the existing methods, we regress the class semantic embeddings to hash codes, which can enforce the generated hash codes of image and text modality to be same with that of class semantic embeddings.

The rest of this article is organized as follows. Section 2 reviews the related work. Section 3 introduces the proposed MESH approach. Section 4 presents the experimental settings and results analysis. Finally, we conclude our work in Section 5.

II. RELATED WORK

This section reviews the most recent cross-modal hashing approaches, which can be categorized into two groups according to the utilization of class labels, i.e., unsupervised and supervised methods.

The unsupervised cross-modal hashing methods discard the label information and construct hashing functions by preserving the intrinsic structure of data [14]–[16]. For example, Xie et al. [24] proposed an unsupervised method Multi-graph Cross-modal Hashing (MGC-MH), which integrates multi-graph learning and hash function learning into a joint framework. In [25], Fang et al. proposed Multi-modal graph regularized Smooth matrix Factorization Hashing (MSFH) to reinforce the spar-sity of extracted common semantic features and the spar-sity of dictionaries learned. There are also many deep neural network based approaches have been presented. Wu et al. [26] proposed an Unsupervised Deep Cross-Modal Hashing via integrating deep learning and matrix factorization with binary latent factor models. Extending the Generative Adversarial Nets (GAN) to cross-modal retrieval, Zhang et al. [27] developed an Unsupervised Generative Adversarial Cross-Modal Hashing which makes full use of GAN’s ability for unsupervised representation learning to explore the underlying manifold structure of cross-modal data. Although these unsupervised methods can achieve impressive performance and improve the flexibility, the main concern is that the semantic limitation further reduces the discrimination of hashing functions and codes.

The supervised cross-modal hashing methods utilize the label information of data to improve the performance of cross-modal retrieval. For instance, many matrix factorization based methods are proposed to learn latent semantic representation [9], [17], [19]. However, one common limitation of them is that they relax the discrete constraints within the optimization resulting in suboptimal binary codes. Therefore, a number of cross-modal hashing methods based on discrete optimization are studied [7], [8], [21]. In [22], Locally Linear Embedding is used to extract the manifold information as similarity matrix for learning unified hash codes where the binary codes are learned directly without relaxation. Lu et al. [8] proposed Discrete Latent Semantic Hashing (DLSH) that first learns the latent semantic representations by matrix factorization, and then projects them into a shared Hamming space. Mandal et al. [28] generalized cross-modal hashing to different scenario cases with unpaired and multi-label data, which enhances the usefulness of cross-modal retrieval in real-word applications. In [3], Fang and Ren proposed a supervised discrete cross-modal hashing framework to directly learn the unified discrete binary codes with a close-form, rather than bit by bit which is very time-consuming.

In recent years, many approaches are proposed to learn the nonlinear hashing functions, such as deep learning based methods [2], [29]–[31]. Jin et al. [32] proposed a Deep Ordinal Hashing (DOH) method to learn hashing functions by leveraging the ranking structure of feature space from both local and global views. In [33], a deep architecture that learns instance-aware representations for multi-label image data is proposed, where an image is represented by multiple pieces of hash codes, which is different from the semantic hashing. Deep cross-modal hashing (DCMH) [11] uses deep neural network as hashing functions, one for each modality. Deep Discrete Cross-Modal Hashing (DDCMH) [12] is similar to DCMH but learns hashing functions with discrete optimization. In [34], a Dense Auto-encoder Hashing (DAH) is proposed which can explicitly impute the missed modality and produce robust binary codes. Due to the effectiveness of GAN, several cross-modal hashing methods based on GAN have been proposed [13], [35], [36] which transform the problem of traditional cross-modal retrieval to generative issues. In addition, the cross-modal semantic embeddings learned by deep network structure were also explored in [37] and [38], in which the semantic information is transferred from texts to images for image classification task. Different from [37] and [38], our method intends to learn common semantic representation of text and image to conduct cross-modal retrieval task. Moreover, we employ the class semantic embeddings to improve the discriminative property of learned common representation, which will be introduced in detail in Section III.

Different from the existing methods, we consider the proximity across neighbors in cross-modal hashing and attempt to investigate the preservation of semantic information via proximity across classes to enhance the semantic information embedded in hash codes.

III. APPROACH

A. PROBLEM DEFINITION

In cross-modal retrieval, we take image and text as an example. Given image data $X^{(1)} = \{x^{(1)}_1, \cdots, x^{(1)}_n\}$ and text data $X^{(2)} = \{x^{(2)}_1, \cdots, x^{(2)}_n\}$, $x^{(1)}_i \in \mathbb{R}^{d_1 \times n}$, $x^{(2)}_i \in \mathbb{R}^{d_2 \times n}$, where $d_1$ is the dimensionality of image features, $d_2$ denotes the dimensionality of text features (usually $d_1 \neq d_2$), and $n$ is the number of paired samples, the main goal is to learn unified binary codes $B \in \{-1, +1\}^{k \times n}$ as well as hashing functions for image and text modality respectively, where $k$ is code length. The semantic label of the given data is represented as $Y \in \{0, 1\}^{c \times n}$, where $c$ is the number of classes. Without loss of generality, we assume that the data are zero-centred, i.e., $\sum_{i=1}^{n} x^{(1)}_i = 0$ and $\sum_{i=1}^{n} x^{(2)}_i = 0$. 

![Image](image-url)
B. MANIFOLD STRUCTURE LEARNING

In order to capture the manifold structure of each modality, we first employ Locally Linear Embedding (LLE) [39] to compute the weighted representation of each sample. Specifically, each sample in the data set can be represented as a linear combination of its K-nearest neighbors, which can be formulated as follows,

$$\min_{W} \sum_{i=1}^{n} \left| X_i - \sum_{j}^{K} W_{ij} X_j \right|^2_F, \text{s.t.} \sum_{j} W_{ij} = 1. \tag{1}$$

where \( W \) is the weight matrix which is used as the similarity matrix of \( X \). Equation (1) can be solved using the method proposed in [39].

Then, we can construct the similarity matrix \( S \) by setting the similarities as the weights calculated via solving LLE, i.e., \( S = |W| \). Here, we denote the similarity matrix for preserving manifold structure of \( X^{(1)} \) and \( X^{(2)} \) modality as \( S^{(1)} \) and \( S^{(2)} \), respectively. We further define update them as follows to ensure symmetry,

$$S^{(1)} = S^{(1)T}, \tag{2}$$

and

$$S^{(2)} = S^{(2)T}. \tag{3}$$

C. CLASS PROXIMITY LEARNING

Our approach aims to augment the semantic information of hash codes to improve their discrimination. Thus, we propose to preserve the class proximity by employing the class semantic embeddings. Different from most of the previous methods that preserve the inter-modality similarity by using one-hot label matrix \( Y \), we employ class semantic embeddings \( A \in \mathbb{R}^{a \times n} \) to compute the similarity matrix \( S^{(12)} \) of samples from different modalities. By so doing, the samples with the same label should have similar hash codes in the Hamming space, and the similar classes, such as dog and cat should be close.

The class semantic embeddings can be the class attributes or the distribute word embeddings trained from a large-scale linguistic text corpus, which can be easily extracted from pre-trained models such as Glove [40]. In this article, we obtain the embeddings from Glove by searching the word in Glove. For single word class such as cat, we regard its word vector as the class semantic embeddings. For two-word class or more, such as dining table, we obtain its class semantic embeddings by averaging the word vectors of its compositions. According to the class label matrix \( Y \in \{0, 1\}^{c \times n} \) and the \( a \)-dimensional distribute word embeddings of class names \( C \in \mathbb{R}^{a \times c} \), the representations corresponding to the \( j \)-th sample of \( A \) can be computed as follows,

$$A_j = CY_j, \quad j = 1, \ldots, n. \tag{4}$$

where \( Y_j \) is the \( j \)-th column of \( Y \). Then, we use the class semantic embeddings \( A \) to compute the similarity matrix \( S^{(12)} \) of between each samples from different modalities.

D. OVERALL OBJECTIVE FUNCTION

First of all, the latent semantic representations \( V \in \mathbb{R}^{k \times n} \) of unified hash codes are learned according to the manifold structure information and class proximity. We formulate the problem by Laplacian Eigenmaps (LE) [41] as follows,

$$\min_{V} Tr(VLV^T), \tag{5}$$

where \( L \) is the hybrid Laplacian matrix of \( W = S^{(1)} + S^{(2)} + S^{(12)}. L = D - W \), where \( D \in \mathbb{R}^{n \times n} \) is a diagonal matrix whose diagonal entries are the column sum of \( W \), i.e. \( D(i, i) = \sum_{j} W_{ij}. V \) is the to be learned latent semantic representations.

Then, we define a orthogonal rotation matrix \( R \in \mathbb{R}^{k \times k} \) to generate unified hash codes according to the latent semantic representations, which can be stated as follows,

$$\min_{B, R} \|B - RV\|^2_F, \quad s.t. \quad R \in \{-1, +1\}^{k \times k}, \quad RR^T = I. \tag{6}$$

Through Eq. (6) we can learn the unified binary codes of training data directly avoiding the large quantization error.

Different from [7], we further regress the class semantic embeddings \( A \) to enhance the semantics of hash codes. In contrast to most of existing methods that regress the hash codes to class labels, we inversly regress the class semantic embeddings to Hamming space to re-align the hash codes and improve discrimination, which has been proved more stable than the former [42]. Thus, we enforce the generated hash codes of image and text modality should be the same as the semantic labels. Then we have,

$$\min_{M, B} \|B - MT_A\|^2_F + \lambda \|M\|^2_F, \quad s.t. \quad B \in \{-1, +1\}^{k \times n}, \tag{7}$$

where \( M \in \mathbb{R}^{c \times k} \) is a projection matrix.

Combining Eqs. (5), (6), and (7), we come to the following overall objective function,

$$\min_{V, B, R, M} Tr(VLV^T) + \alpha \|B - RV\|^2_F + \beta \|B - MT_A\|^2_F + \lambda \|M\|^2_F, \quad s.t. \quad B \in \{-1, +1\}^{k \times n}, \tag{8}$$

where \( \alpha, \beta, \lambda \) are trade-off parameters.

In addition, we apply the kernel features to replace the original features to capture the nonlinear relationship of different modalities. Following [10], we adopt a RBF kernel mapping \( \phi (x) \) to obtain the kernel features of image and text modality respectively, \( \phi_i (x) = \exp (-D(x, a_i)/\sigma) \), where \( D(\cdot) \) is the distance function, \( \sigma \) is the kernel bandwidth parameter, and \( \{a_i\}_{i=1}^{q} \) are the randomly selected \( q \) anchors from training dataset.
E. OPTIMIZATION

It is intractable to directly minimize the objective in Eq. (8) because of the non-convexity with four matrix variables V, M, R, and B. Fortunately, it is convex with respect to any of the four variables in the case that the others are fixed. Therefore, we propose an alternative optimization in an iterative manner to address the optimization problem. The detailed optimization steps are listed as follows:

**Step 1. Update M.** The objective function shown in Eq. (8) can be simplified to the following with the other variables fixed but M,

\[
\min_M \beta \left\| B - M^T A \right\|_F^2 + \lambda \left\| M \right\|_F^2. \tag{9}
\]

By setting its derivative w.r.t M to 0, we have,

\[
M = \beta (\beta A A^T + \lambda I)^{-1} A B^T. \tag{10}
\]

**Step 2. Update V.** When the other three variables are fixed but V, the overall objective can be transformed to,

\[
\min_V \text{Tr}(V L V^T) + \alpha \left\| B - RV \right\|_F^2. \tag{11}
\]

By setting its derivative w.r.t V to 0, we can arrive at a Sylvester equation [43] as shown in Eq. (12).

\[
\tilde{A} V + V \tilde{B} = \tilde{C}, \tag{12}
\]

where

\[
\tilde{A} = \alpha R^T R, \quad \tilde{B} = L, \quad \tilde{C} = \alpha R^T B. \tag{13}
\]

V can be easily achieved through addressing Eq. (12) via the sylvestre function in Matlab.

**Step 3. Update R.** When the other three variables are fixed but R, the overall objective can be transformed to,

\[
\min_R \left\| B - RV \right\|_F^2, \text{ s.t. } RR^T = I. \tag{14}
\]

Apparently, it can be solved by Singular Value Decomposition (SVD). Let Q = BV^T, we can have Q = GΣH^T after SVD operation. Then, R can be updated as follows,

\[
R = GH^T. \tag{15}
\]

**Step 4. Update B.** When the other three variables are fixed but B, the overall objective can be simplified to,

\[
\min_B \alpha \left\| B - RV \right\|_F^2 + \beta \left\| B - M^T A \right\|_F^2, \quad \text{s.t. } B \in \{-1, +1\}^{k \times n}. \tag{16}
\]

It can be transformed to,

\[
\min_B \alpha \left( \left\| B \right\|_F^2 - 2 \text{Tr}(B^T RV) + \left\| RV \right\|_F^2 \right) + \beta \left( \left\| B \right\|_F^2 - 2 \text{Tr}(B^T M^T A) + \left\| M^T A \right\|_F^2 \right). \tag{17}
\]

**Algorithm 1 Manifold-Embedded Semantic Hashing**

**Input:** Training data \(X^{(1)}\) and \(X^{(2)}\), label \(Y\), distributed word embeddings of class names \(C\), parameters \(\alpha, \beta, \lambda\), and the iteration number \(T\).

**Output:** Unified hash codes \(B\).

Construct class semantic embeddings \(A\): Compute kernel features \(\phi(X^{(1)})\) and \(\phi(X^{(2)})\);

Initialize \(V, M, B, R\) randomly;

for iter = 1 → T do

  Update \(M\) via Eq. (10); Update \(V\) via Eq. (12); Update \(R\) via Eq. (15); Update \(B\) via Eq. (18);

end

Apparently, the solution can be easily obtained as followings, where \(sgn\) is the sign function.

\[
B = sgn(\alpha RV + \beta M^T A). \tag{18}
\]

The description of the proposed method is summarized in Algorithm 1.

F. HASHING FUNCTIONS LEARNING

As MESH is a two-step cross-modal hashing method, in the first step, the unified hash codes of training data are learned. In the second step, we need to learn hashing functions based on the unified hash codes. Since hashing functions learning can be formulated as a regression problem, we can use various types of classifiers as hashing functions, which shows the flexibility of MESH.

Here, we adopt linear regression as the hashing functions. For example, given the unified hash codes \(B\), the hashing functions for image and text modality are learned by solving the following problems respectively.

\[
\min_{P_i} \left\| B - P_i X^{(i)} \right\|_F^2 + \theta \left\| P_i \right\|_F^2, \quad i = 1, 2, \tag{19}
\]

where \(\theta\) is a balance parameter. \(P_i\) is the to be learned hashing function for the \(t\)-th modality. The solution is computed as,

\[
P_i = (X^{(i)}(X^{(i)})^T + \theta I)^{-1} X^{(i)} B^T. \tag{20}
\]

Therefore, given a new query sample \(x_q^{(i)}\) from the \(t\)-th modality, the binary codes can be computed by,

\[
b_q^{(i)} = sgn(P_i^T x_q^{(i)}). \tag{21}
\]

G. COMPLEXITY ANALYSIS

Here, we discuss the computational complexity of our iterative optimization. For updating \(M\), the time complexity is \(O(na^2 + nak + na^3)\). Updating \(V\) costs \(O(n^2 + nk^2 + k^3)\), and updating \(R\) costs \(O(nk^2 + k^3)\). As for computing the hash codes, the time complexity scales \(O(nk^2 + nak)\). Therefore, the time complexity is \(O(n^2 + n(a^3 + a^2 + k^2 + ak) + k^3)\) in each iteration.
IV. EXPERIMENTS AND RESULTS ANALYSIS

In this section, we conduct experiments on three datasets to evaluate the performance of our method for cross-modal retrieval. We design two representative cross-modal retrieval tasks, i.e., Text-to-Image utilizing text as a query to search relevant images and Image-to-Text using image as a query to search relevant text. In the experiments, images and texts are considered to be relevant if they share at least one common semantic label.

A. DATASETS

Wiki [44] dataset contains 2866 pairs of image and text. All the image-text pairs can be divided into 10 semantic classes. In Wiki, image features are the 128-dimensional bag-of-visual words SIFT feature vectors, and text features are 10-dimensional topic vectors which are generated by the latent Dirichlet allocation (LDA) model. In our experiments, the initial split i.e., 2173 for training and 693 for testing is employed.

LabelMe [45] dataset consists of 2688 outdoor scenes which can be grouped into 8 classes. Each image is annotated with tags which are from a 781 words dictionary. Following [12], we discard the words that occur in less than 3 times. Therefore, the representation of text modality is a 366-dimensional word frequency. While image features are represented by 512-dimensional GIST features. As [12], the samples without tags are deleted, which leads a 2686 image-text pairs new dataset. In our experiments, 75% of the data are randomly selected for training, and the rest are chose as the query set for testing.

Pascal VOC [46] dataset contains 9963 image-tag pairs, which can be divided into 20 categories. Following [12], we select the pairs with only one label, which results in 2808 training and 2841 testing image-tag pairs. The image modality is represented by 512-dimensional GIST features, and the representations of text modality are 399-dimensional word frequency features.

B. BASELINES

In order to evaluate the effectiveness of our proposed MESH, we choose 10 of the state-of-the-art cross-modal hashing methods as baselines, including Cross-View Hashing (CVH) [47], Collective Matrix Factorization Hashing (CMFH) [17], Latent Semantic Sparse Hashing (LSSH) [16], Supervised Matrix Factorization Hashing (SMFH) [6], Fusion Similarity Hashing (FSH) [18], Semantic Correlation Maximization (SCM-seq [48], SCM-orth [48]), Intra-and Inter-Modality Similarity Preserving Hashing (IISPH) [9], Fast Discrete Cross-modal Hashing (FDCH) [7], and Discrete Cross-modal Hashing (DCH) [5]. The baselines contain both supervised and unsupervised ones. In particular, DCH and FDCH are the most recent representative methods.

For all baselines, we carefully tune their parameters and report the average results of 10 repeats. As for the proposed MESH, we empirically set $\alpha = 1e^{-4}$, $\beta = 1$, $\lambda = 1e^{-3}$, $\theta = 1$, and the iteration number is restricted to 10.

Table 1. MAP results of various methods on Wiki.

| Task     | Method   | 16 bits | 32 bits | 64 bits | 96 bits |
|----------|----------|---------|---------|---------|---------|
| Text to Image | CVH   | 0.1519  | 0.1322  | 0.1243  | 0.1194  |
|          | CMFH    | 0.4814  | 0.5148  | 0.5319  | 0.5382  |
|          | LSSH    | 0.5011  | 0.5207  | 0.5311  | 0.5332  |
|          | SMFH    | 0.6160  | 0.6230  | 0.6385  | 0.6405  |
|          | FSH     | 0.5582  | 0.5812  | 0.5903  | 0.5989  |
|          | SCM-seq | 0.2257  | 0.2459  | 0.2474  | 0.2487  |
|          | SCM-orth| 0.1560  | 0.1362  | 0.1257  | 0.1220  |
|          | IISPH   | 0.5097  | 0.5352  | 0.5452  | 0.5565  |
|          | FDCH    | 0.7021  | 0.7081  | 0.7140  | 0.7208  |
|          | DCH     | 0.7232  | 0.6965  | 0.7140  | 0.7211  |
|          | MESH    | 0.7478  | 0.7473  | 0.7404  | 0.7496  |

TABLE 1. MAP results of various methods on Wiki.

C. EVALUATION METRICS

In order to evaluate the performance of the proposed MESH and competitors for cross-modal retrieval, we employ the widely used Mean Average Precision (MAP) as a measurement metric. In addition, we also plot the precision-recall and top-N precision curves when hash codes are 16 and 64 bits to further compare the performance. It is worth noting that, for the three metrics, the larger the values are, the better the performance is.

D. RESULTS AND ANALYSIS

1) RESULTS ON WIKI

The MAP results of MESH and all baselines on Wiki with varied code lengths are reported in Table 1, and the precision-recall curves and top-N precision curves of Text-to-Image and Image-to-Text with 16 and 64 bits are shown in Fig. 2. We have the following observations from the results in Table 1 and Figure 1. 1) MESH slightly outperforms the baselines on both cross-modal retrieval tasks. 2) In case of top-N precision, the results of MESH are consistently better than other baselines. In particular, the precision at top-200 are much better than baselines, which is significant to search task where users mainly focus on results on the first few pages. 3) Most supervised methods, such as MESH, DCH, FDCH, and SMFH achieve superior performance to unsupervised methods, such as CMFH and FSH, which illustrates the significance of employing supervised label information. 4) The results of DCH and FDCH are similar but much better than other baselines. This suggests that discrete optimization can boost the cross-modal retrieval. However, they are still inferior to our MESH. 5) It can be easily seen that the Text-to-Image MAPs of most methods are higher than Image-to-Text.
with a large margin. The reason may be that text feature is closer than image visual feature to the semantic information, and can better describe the object.

2) RESULTS ON LabelMe

The MAP results of MESH and all baselines on LabelMe are shown in Table 2, and we also plot precision-recall curves and top-N precision curves of various methods for both two tasks with 16 and 64 bits in Fig. 3. From these results, we can observe that: 1) Our proposed MESH consistently outperforms the other competitors. 2) On Image-to-Text task, the MAPs of MESH are substantially better than that of baselines. Specifically, compared with the best baseline, MESH achieves performance improvement at least by 19% and 10% at 16 and 32 bits, respectively. This may because of the consideration of class proximity in MESH, which can enhance the semantic information of visual feature. 3) Compared to the baselines, the curves in Figure 2 of MESH are consistently higher, especially for Image-to-Text task, which is similar to MAP. 4) Most of methods perform better on Text-to-Image task than on Image-to-Text task. However, the margins are smaller than on Wiki. One possible reason is that the semantic gap between text and image features in LabelMe is smaller than in Wiki.
3) RESULTS ON PASCAL
The MAP results of MESH and all baselines on Pascal VOC are reported in Table 3, and the precision-recall and top-N precision curves of all methods for both tasks with 16 and 64 bits are plotted in Fig. 4. From the above results, we obtain the following observations that are similar to Wiki and LabelMe. 1) MESH is superior to other baselines with varied code lengths on both tasks. 2) Similar to Wiki, the Text-to-Image results of most methods are higher than Image-to-Text with a large margin. 3) On Image-to-Text task, the curves of MESH are significantly higher than that of other baselines, which further validates the effectiveness of our method. 4) MESH, DCH, and FDCH are in the first group, performing much better than other baselines, which further demonstrates the effectiveness of learning hash codes discretely in reducing quantization error.

In summary, MESH yields state-of-the-art performance on three datasets, which demonstrates the effectiveness of our method in considering the preservation of neighborhood proximity and class proximity. In addition, the class proximity preserving can boost the semantic information of image visual feature resulting in a significant performance improvement on Image-to-Text task, especially on LabelMe and Pascal datasets. Moreover, the two-step hashing scheme can make MESH flexible to various hashing functions.

![FIGURE 4. Top-N precision and precision-recall curves of different methods on Pascal VOC at 16 and 64 bits.]

### TABLE 2. MAP results of various methods on LabelMe.

| Task | Method      | 16 bits  | 32 bits  | 64 bits  | 96 bits  |
|------|-------------|----------|----------|----------|----------|
| Text | CVH         | 0.2944   | 0.2665   | 0.2477   | 0.2369   |
|      | CMFH        | 0.4973   | 0.4742   | 0.4512   | 0.4533   |
|      | LSSH        | 0.6093   | 0.6265   | 0.6464   | 0.6582   |
|      | SMFH        | 0.8447   | 0.8852   | 0.9024   | 0.9048   |
|      | FSH         | 0.5098   | 0.6166   | 0.6888   | 0.7223   |
|      | SCM-orth    | 0.4608   | 0.8135   | 0.8245   | 0.7031   |
|      | IISPH       | 0.5809   | 0.7021   | 0.7353   | 0.7551   |
|      | PDCCH       | 0.9009   | 0.9096   | 0.9198   | 0.9198   |
|      | DCH         | 0.9064   | 0.9212   | 0.9267   | 0.9247   |
|      | MESH        | 0.9182   | 0.9259   | 0.9282   | 0.9288   |

| Task | Method  | 16 bits  | 32 bits  | 64 bits  | 96 bits  |
|------|---------|----------|----------|----------|----------|
| Text | CVH     | 0.2509   | 0.2229   | 0.2031   | 0.1945   |
|      | CMFH    | 0.3853   | 0.3750   | 0.3654   | 0.3680   |
|      | LSSH    | 0.6008   | 0.6371   | 0.6613   | 0.6610   |
|      | SMFH    | 0.6319   | 0.6948   | 0.7336   | 0.7401   |
|      | FSH     | 0.3865   | 0.4820   | 0.5517   | 0.5823   |
|      | SCM-seq | 0.3938   | 0.6727   | 0.6854   | 0.5795   |
|      | IISPH   | 0.3470   | 0.3034   | 0.5695   | 0.5787   |
|      | PDCCH   | 0.6817   | 0.7507   | 0.7169   | 0.7603   |
|      | DCH     | 0.6899   | 0.7618   | 0.7573   | 0.7846   |
|      | MESH    | 0.8256   | 0.8431   | 0.8458   | 0.8584   |

### TABLE 3. MAP results of various methods on Pascal VOC.

| Task | Method      | 16 bits  | 32 bits  | 64 bits  | 96 bits  |
|------|-------------|----------|----------|----------|----------|
| Text | CVH         | 0.1298   | 0.1364   | 0.1482   | 0.1514   |
|      | CMFH        | 0.4851   | 0.5447   | 0.5180   | 0.5040   |
|      | LSSH        | 0.5391   | 0.6015   | 0.6189   | 0.6289   |
|      | SMFH        | 0.5907   | 0.6635   | 0.6334   | 0.6329   |
|      | FSH         | 0.3578   | 0.5091   | 0.5380   | 0.6369   |
|      | SCM-orth    | 0.2597   | 0.4082   | 0.4345   | 0.2546   |
|      | IISPH       | 0.4352   | 0.4978   | 0.5906   | 0.6807   |
|      | PDCCH       | 0.8655   | 0.8967   | 0.9007   | 0.9057   |
|      | DCH         | 0.8794   | 0.9002   | 0.9044   | 0.9076   |
|      | MESH        | 0.9134   | 0.9236   | 0.9267   | 0.9222   |

![FIGURE 4. Top-N precision and precision-recall curves of different methods on Pascal VOC at 16 and 64 bits.]

3) RESULTS ON PASCAL
The MAP results of MESH and all baselines on Pascal VOC are reported in Table 3, and the precision-recall and top-N precision curves of all methods for both tasks with 16 and 64 bits are plotted in Fig. 4. From the above results, we obtain the following observations that are similar to Wiki and LabelMe. 1) MESH is superior to other baselines with varied code lengths on both tasks. 2) Similar to Wiki, the Text-to-Image results of most methods are higher than Image-to-Text with a large margin. 3) On Image-to-Text task, the curves of MESH are significantly higher than that of other baselines, which further validates the effectiveness of our method. 4) MESH, DCH, and FDCH are in the first group, performing much better than other baselines, which further demonstrates the effectiveness of learning hash codes discretely in reducing quantization error.

In summary, MESH yields state-of-the-art performance on three datasets, which demonstrates the effectiveness of our method in considering the preservation of neighborhood proximity and class proximity. In addition, the class proximity preserving can boost the semantic information of image visual feature resulting in a significant performance improvement on Image-to-Text task, especially on LabelMe and Pascal datasets. Moreover, the two-step hashing scheme can make MESH flexible to various hashing functions.
We further compare our method with a state-of-the-art deep cross-modal hashing method, i.e., DCMH [11] on MIRFlickr-25K [49]. In this experiment, deep features of images that are used in our proposed method are provided by [11]. The MAP results of cross-modal retrieval are presented in Table 4, in which the results of DCMH are cited from SCRATCH [20]. As it can be seen, the proposed MESH outperforms DCMH consistently with a large margin. This demonstrates that the proposed MESH can achieve much better cross-modal retrieval performance compared to the deep cross-modal hashing method DCMH on deep features, which further validates the effectiveness of our method. In addition, the MAPs of our method are superior to SCRATCH in most cases, especially on the task of Image to Text.

4) COMPARISON WITH DEEP FEATURES

We design two diverse variations for comparison. MESH-1 is built without the Laplacian graph regularization term in Eq. (5). MESH-2 is built using one-hot label information $Y$ by replacing the class semantic embeddings. Table 5 reports the results on Wiki dataset. As can be seen, the proposed MESH outperforms MESH-1 and MESH-2. We also can conclude that the semantic regress term in Eq. (7) plays the key role in MESH, and the manifold embedding is icing on the cake.

E. ABLATION STUDY

We also verified the impact of different part of the objective functions on the performance of the proposed MESH. We design two diverse variations for comparison. MESH-1 is built without the Laplacian graph regularization term in Eq. (5). MESH-2 is built using one-hot label information $Y$ by replacing the class semantic embeddings. Table 5 reports the results on Wiki dataset. As can be seen, the proposed MESH outperforms MESH-1 and MESH-2. We also can conclude that the semantic regress term in Eq. (7) plays the key role in MESH, and the manifold embedding is icing on the cake.

F. PARAMETER SENSITIVITY

Fig. 5 shows the effect of the trade-off parameters $\alpha$, $\beta$, and $\lambda$ on LabelMe dataset. As shown in this figure, we can see that the cross-modal results are influenced by $\alpha$ and $\beta$. Generally, the proposed method can achieve a better performance with a wide range of parameter values.

G. CONVERGENCE

Since the alternative optimization in an iterative manner is used to optimize the objective function, we further investigate
the convergence of our algorithm. We conduct experiments on three datasets to record the objective function value in each iteration, resulting to the convergence curves in Fig. 6. As can be seen, MESH can often converge within 10 iterations which further demonstrates the efficiency of the optimization algorithm. Moreover, we report the training time of each iteration in Algorithm 1 over different datasets in Table 6. The results are obtained with Matlab R2015b on a PC with 2.7 GHz Intel CPU and 16 GB memory.

**V. CONCLUSION**

In this article, we proposed a flexible cross-modal hashing method, named Manifold-Embedded Semantic Hashing, MESH for short. MESH preserves the neighborhood proximity and class proximity in the hash codes to augment the semantic information and improve the discrimination of hash codes. In addition, the two-scheme cross-modal hashing makes MESH flexible to various types of hashing functions. Extensive experiments on three datasets demonstrated the superiority of MESH against ten state-of-the-art cross-modal hashing approaches.

Recently, zero-shot cross-modal hashing has drawn considerable interests, our model cannot handle the zero-shot problem currently. We plan to extend MESH to solve the cross-modal retrieval for new emerging concepts combining deep representations.

**REFERENCES**

[1] T. Yao, G. Wang, L. Yan, X. Kong, Q. Su, C. Zhang, and Q. Tian, “Online latent semantic hashing for cross-media retrieval,” Pattern Recognit., vol. 89, pp. 1–11, May 2019.

[2] L. Jin, K. Li, Z. Li, F. Xiao, G.-J. Qi, and J. Tang, “Deep semantic-preserving ordinal hashing for cross-modal similarity search,” IEEE Trans. Neural Netw. Learn. Syst., vol. 30, no. 5, pp. 1429–1440, May 2019.

[3] Y. Fang and Y. Ren, “Supervised discrete cross-modal hashing based on kernel discriminant analysis,” Pattern Recognit., vol. 98, Feb. 2020, Art. no. 107062.

[4] Y. Cao, M. Long, J. Wang, and S. Liu, “Collective deep quantization for efficient cross-modal retrieval,” in Proc. AAAI, 2017, pp. 3974–3980.

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

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

[7] X. Liu, X. Nie, W. Zeng, C. Cui, L. Zhu, and Y. Yin, “Fast discrete cross-modal hashing with regressing from semantic labels,” in Proc. ACM Multimedia Conf. Multimedia Conf. (MM), 2018, pp. 1662–1669.

[8] X. Lu, L. Zhu, Z. Cheng, X. Song, and H. Zhang, “Efficient discrete latent semantic hashing for scalable cross-modal retrieval,” Signal Process., vol. 154, pp. 217–231, Jan. 2019.

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

[10] V. E. Liong, J. Lu, and Y.-P. Tan, “Cross-modal discrete hashing,” Pattern Recognit., vol. 79, pp. 114–129, Jul. 2018.

[11] Q.-Y. Jiang and W.-J. Li, “Deep cross-modal hashing,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 3270–3278.

[12] Z. Chen, F. Zhong, G. Min, Y. Leng, and Y. Ying, “Supervised intra-and inter-modality similarity preserving hashing for cross-modal retrieval,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 3270–3278.

[13] X. Liu, Y. Guo, L. Liu, E. M. Bakker, and M. S. Lew, “CycleMatch: A cycle-consistent embedding network for image-text matching,” Pattern Recognit., vol. 95, pp. 365–379, Sep. 2019.

[14] X. Xu, L. He, A. Shimada, R.-I. Taniguchi, and H. Lu, “Learning unified binary codes for cross-modal retrieval via latent semantic hashing,” Neurocomputing, vol. 213, pp. 191–203, Nov. 2016.

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

[16] G. Ding, Y. Guo, and J. Zhou, “Collective matrix factorization hashing for multimodal data,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2014, pp. 2075–2082.

[17] H. Liu, R. J. Wu, F. Huang, and B. Zhang, “Cross-modality binary code learning via fusion similarity hashing,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 7380–7388.

[18] H. Liu, R. Ji, Y. Wu, and G. Hua, “Supervised matrix factorization for cross-modality hashing,” in Proc. IJCAI, 2016, pp. 1767–1773.

[19] C.-X. Li, Z.-D. Chen, P.-F. Zhang, X. Luo, L. Nie, W. Zhang, and X.-S. Xu, “SCRATCH: A scalable discrete matrix factorization hashing for cross-modal retrieval,” in Proc. 26th ACM Int. Conf. Multimedia, 2018, pp. 1–9.

[20] L. Liu, Y. Yang, M. Hu, X. Xu, F. Shen, N. Xie, and Z. Huang, “Index and retrieve multimedia data: Cross-modal hashing by learning subspace relation,” in Proc. DASFAA, 2018, pp. 606–621.

[21] X. Luo, X.-Y. Yin, L. Nie, X. Song, Y. Wang, and X.-S. Xu, “SDMCH: Supervised discrete manifold-embedded cross-modal hashing,” in Proc. 27th Int. Joint Conf. Artif. Intell., Jul. 2018, pp. 2518–2524.

[22] P. Zhou, L. Du, M. Fan, and Y.-D. Shen, “An LLE based heterogeneous metric learning for cross-media retrieval,” in Proc. SIAM Int. Conf. Data Mining, Jun. 2015, pp. 64–72.

[23] L. Xie, L. Zhu, and G. Chen, “Unsupervised multi-graph cross-modal hashing for large-scale multimedia retrieval,” Multimedia Tools Appl., vol. 75, no. 15, pp. 9185–9204, Aug. 2016.

[24] Y. Fang, H. Zhang, and Y. Ren, “Unsupervised cross-modal retrieval via multi-modal graph regularized smooth matrix factorization hashing,” Knowl.-Based Syst., vol. 171, pp. 69–80, May 2019.

[25] G. Wu, Z. Lin, J. Han, L. Liu, G. Ding, B. Zhang, and J. Shen, “Unsupervised deep hashing via binary latent factor models for large-scale cross-modal retrieval,” in Proc. 27th Int. Joint Conf. Artif. Intell., Jul. 2018, pp. 2854–2860.

[26] J. Zhang, Y. Peng, and M. Yuan, “Unsupervised generative adversarial cross-modal hashing,” in Proc. AAAI, 2018, pp. 539–546.

[27] D. Mandal, K. N. Chaudhury, and S. Biswas, “Generalized semantic preserving hashing for cross-modal retrieval,” IEEE Trans. Image Process., vol. 28, no. 1, pp. 102–112, Jan. 2019.

[28] Y. Cao, M. Long, J. Wang, Q. Yang, and P. S. Yu, “Deep visual-semantic hashing for cross-modal retrieval,” in Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining, Aug. 2016, pp. 1445–1454.

[29] C. Li, C. Deng, N. Li, W. Liu, X. Gao, and D. Tao, “Self-supervised adversarial hashing networks for cross-modal retrieval,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 4242–4251.

[30] Z.-D. Chen, W.-J. Yu, C.-X. Li, L. Nie, and X.-S. Xu, “Dual deep neural networks cross-modal hashing,” in Proc. AAAI, 2018, pp. 274–281.

[31] B. Jin, X. Shu, K. Li, Z. Li, G.-J. Qi, and J. Tang, “Deep ordinal hashing with spatial attention,” IEEE Trans. Image Process., vol. 27, no. 5, pp. 2173–2186, May 2019.

[32] H. Lai, P. Yan, X. Shu, Y. Wei, and S. Yan, “Instance-aware hashing for multi-label image retrieval,” IEEE Trans. Image Process., vol. 25, no. 6, pp. 2469–2479, Jun. 2016.

[33] H. Liu, M. Lin, S. Zhang, W. Fu, H. Huang, and R. Ji, “Dense auto-encoder hashing for robust cross-modality retrieval,” in Proc. ACM Multimedia Conf. Multimedia Conf. (MM), 2018, pp. 1589–1597.

[34] Y. Peng and J. Qi, “CM-GANs: Cross-modal generative adversarial networks for common representation learning,” ACM Trans. Multimedia Comput., Commun., Appl., vol. 15, no. 1, p. 22, Feb. 2019.
[36] L. Wu, Y. Wang, and L. Shao, “Cycle-consistent deep generative hashing for cross-modal retrieval,” *IEEE Trans. Image Process.*, vol. 28, no. 4, pp. 1602–1612, Apr. 2019.

[37] X. Shu, G.-J. Qi, J. Tang, and J. Wang, “Weakly-shared deep transfer networks for heterogeneous-domain knowledge propagation,” in *Proc. 23rd ACM Int. Conf. Multimedia (MM)*, 2015, pp. 35–44.

[38] J. Tang, X. Shu, Z. Li, G.-J. Qi, and J. Wang, “Generalized deep transfer networks for knowledge propagation in heterogeneous domains,” *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 12, no. 4s, pp. 1–22, Nov. 2016.

[39] S. T. Roweis, “Nonlinear dimensionality reduction by locally linear embedding,” *Science*, vol. 290, no. 5500, pp. 2323–2326, Dec. 2000.

[40] J. Pennington, R. Socher, and C. Manning, “Glove: Global vectors for word representation,” in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1532–1543.

[41] M. Belkin and P. Niyogi, “Laplacian eigenmaps for dimensionality reduction and data representation,” *Neural Comput.*, vol. 15, no. 6, pp. 1373–1396, Jun. 2003.

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

[43] S.-G. Lee and Q.-P. Vu, “Simultaneous solutions of Sylvester equations and idempotent matrices separating the joint spectrum,” *Linear Algebra Appl.*, vol. 435, no. 9, pp. 2097–2109, Nov. 2011.

[44] N. Rasiwasia, J. Costa Pereira, E. Covello, G. Doyle, G. R. G. Lanckriet, R. Levy, and N. Vasconcelos, “A new approach to cross-modal multimedia retrieval,” in *Proc. Int. Conf. Multimedia (MM)*, 2010, pp. 251–260.

[45] A. Oliva and A. Torralba, “Modeling the shape of the scene: A holistic representation of the spatial envelope,” *Int. J. Comput. Vis.*, vol. 42, no. 3, pp. 145–175, 2001.

[46] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The Pascal visual object classes (VOC) challenge,” *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010.

[47] S. Kumar and R. Udupa, “Learning hash functions for cross-view similarity search,” in *Proc. 22nd Int. Joint Conf. Artif. Intell.*, 2011, pp. 1360–1365.

[48] D. Zhang and W.-J. Li, “Large-scale supervised multimodal hashing with semantic correlation maximization,” in *Proc. 28th AAAI Conf. Artif. Intell.*, 2014, pp. 2177–2183.

[49] M. J. Huiskes and M. S. Lew, “The MIR flickr retrieval evaluation,” in *Proc. 28th AAAI Conf. Artif. Intell.*, 2014, pp. 35–44.

[50] J. Pennington, R. Socher, and C. Manning, “Glove: Global vectors for word representation,” in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1532–1543.

[51] M. Belkin and P. Niyogi, “Laplacian eigenmaps for dimensionality reduction and data representation,” *Neural Comput.*, vol. 15, no. 6, pp. 1373–1396, Jun. 2003.

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

[53] S.-G. Lee and Q.-P. Vu, “Simultaneous solutions of Sylvester equations and idempotent matrices separating the joint spectrum,” *Linear Algebra Appl.*, vol. 435, no. 9, pp. 2097–2109, Nov. 2011.

[54] N. Rasiwasia, J. Costa Pereira, E. Covello, G. Doyle, G. R. G. Lanckriet, R. Levy, and N. Vasconcelos, “A new approach to cross-modal multimedia retrieval,” in *Proc. Int. Conf. Multimedia (MM)*, 2010, pp. 251–260.

[55] A. Oliva and A. Torralba, “Modeling the shape of the scene: A holistic representation of the spatial envelope,” *Int. J. Comput. Vis.*, vol. 42, no. 3, pp. 145–175, 2001.

[56] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The Pascal visual object classes (VOC) challenge,” *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Jun. 2010.

[57] S. Kumar and R. Udupa, “Learning hash functions for cross-view similarity search,” in *Proc. 22nd Int. Joint Conf. Artif. Intell.*, 2011, pp. 1360–1365.

[58] D. Zhang and W.-J. Li, “Large-scale supervised multimodal hashing with semantic correlation maximization,” in *Proc. 28th AAAI Conf. Artif. Intell.*, 2014, pp. 2177–2183.

[59] M. J. Huiskes and M. S. Lew, “The MIR flickr retrieval evaluation,” in *Proc. 28th AAAI Conf. Artif. Intell.*, 2014, pp. 35–44.

[60] J. Pennington, R. Socher, and C. Manning, “Glove: Global vectors for word representation,” in *Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP)*, 2014, pp. 1532–1543.

[61] M. Belkin and P. Niyogi, “Laplacian eigenmaps for dimensionality reduction and data representation,” *Neural Comput.*, vol. 15, no. 6, pp. 1373–1396, Jun. 2003.