IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 2, 2020, accepted January 17, 2020, date of publication January 20, 2020, date of current version January 27, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2968154

A Review of Hashing Methods for Multimodal Retrieval

WENMING CAO[1,2], (Member, IEEE), WENSHUO FENG[1], QIUBIN LIN[1], GUITAO CAO[3,4], AND ZHIHAI HE[2], (Fellow, IEEE)

[1]Guangdong Multimedia Information Service Engineering Technology Research Center, Shenzhen University, Shenzhen 518060, China
[2]Video Processing and Communication Laboratory, Department of Electrical and Computer Engineering, University of Missouri, Columbia, MO 65211, USA
[3]Shanghai Key Laboratory of Trustworthy Computing, East China Normal University, Shanghai 200062, China
[4]MOE Research Center for Software/Hardware Co-Design Engineering, East China Normal University, Shanghai 200062, China

Corresponding author: Zhihai He (HeZhi@missouri.edu)

This work was supported in part by the National Natural Science Foundation of China under Grant 61771322 and Grant 61375015, and in part by the Fundamental Research Foundation of Shenzhen under Grant JCYJ20160307154630057.

ABSTRACT With the advent of the information age, the amount of multimedia data has exploded. That makes fast and efficient retrieval in multimodal data become an urgent requirement. Among many retrieval methods, the hashing method is widely used in multimodal data retrieval due to its low storage cost, fast and effective characteristics. This review clarifies the definition of multimodal retrieval requirements and some related concepts, then introduces some representative hashing methods, mainly supervised methods that make full use of label information, especially the latest deep hashing methods. The principle and performance of these methods are compared and analyzed. At the same time, some remaining problems and improvement space would be discussed. This review will help researchers better understand the research status and future research directions in this field.

INDEX TERMS Multimedia, multimodal retrieval, hashing method, deep learning, reviews.

I. INTRODUCTION

With the advent of the information age and the rapid development of the Internet, multimedia data has explosive growth in various modalities such as text, image, audio, and video. Traditional single-modal data retrieval, such as image retrieval and text retrieval, has been unable to adapt to the reality of the gradual diversification of multimedia data. Multimodal data has the characteristics of low-level expressive heterogeneity and high-level semantic homogeneity, that is, the same thing has different expressions. A more diverse form of expression can help people understand the things themselves better. When searching for something, people often want to accurately find more search results with different expressions. Based on this, the fast and efficient retrieval of data in different modalities is particularly necessary. Multimodal retrieval is essentially a data similarity retrieval. In a certain sense, it belongs to the Nearest Neighbor (NN) [1] retrieval problem, that is, given a query data and a database, after the operation, it returns the data most similar to the query data. Because multimedia data is massive and high-dimensional, multimodal retrieval requires huge storage space and a long time. There is a semantic gap in the information representation of different modalities, which makes accurate retrieval difficult. To achieve more efficient retrieval, it is necessary to sacrifice a certain accuracy. In real-life scenarios, if the similarity is high enough to satisfy the retrieval requirements, then the most similar results returned from the dataset are sometimes unnecessary. The hashing method based on Approximate Nearest Neighbor (ANN) [2] is widely used because it is fast and efficient while costing low storage.

The hashing method saves storage and speeds up retrieval by mapping raw features to binary encoding (Hamming) space [3]. At the same time, the similarity of the data should be maintained in the mapping process (the data with high similarity in the original space is mapped to the Hamming space, and the distance between the hash codes is small, and vice versa). Hashing methods are divided into data-independent methods and data-dependent methods. Considering the retrieval effect and ubiquity, this article mainly introduces the latter. The key to this type of method is to use training data to learn the most suitable hash function. The learning of the hash function is mainly divided into two steps: dimensionality reduction and quantization.

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

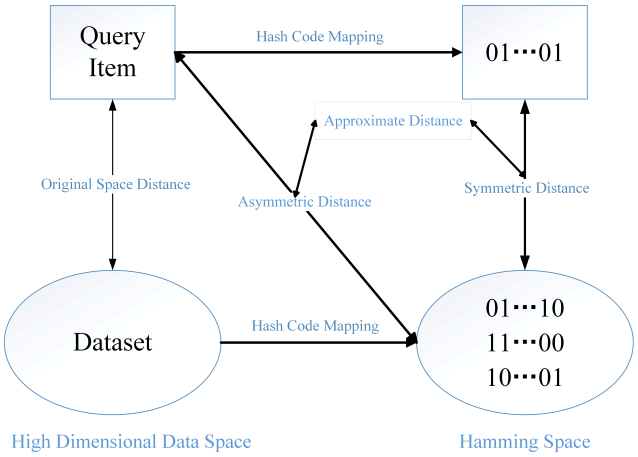

FIGURE 1. Principle of hashing methods for multimodal retrieval.

Dimensionality reduction refers to mapping information in the original space to a low-dimensional spatial representation. Quantification refers to the linear or non-linear transformation of the original features to binary segment the feature space to obtain hash codes. The information of different modalities indicates that there is a certain degree of the semantic gap, and the main problem that hashing methods for multimodal retrieval need to solve is to minimize it as much as possible. One common solution is to learn a uniform hash code to make it more consistent. The other is to minimize the coding distance and increase its compactness.

Hash retrieval is a research hotspot in recent years, and various excellent methods and improvements are constantly being proposed. In order to better demonstrate the vertical development of research, we will briefly describe the early data-independent methods and single-modal methods. Later we will focus on some representative supervised data-dependent multimodal methods. Besides, some of the latest deep hashing methods using deep learning [4], attention-aware [5] mechanisms, and the use of adversarial networks [6] will be introduced. We will compare and analyze the advantages and disadvantages of various methods, sort out the development process, point out the existing problems, and make some conjectures and predictions about future research trends.

II. RELATED CONCEPTS

A. HAMMING DISTANCE SORTING

Hamming distance refers to the number of characters in different equal length strings [7]. For any two binary vectors $\boldsymbol{a}, \boldsymbol{b} \in (0,1)^j$ of the same length, the Hamming distance between them can be calculated by an exclusive OR operation, namely:

$$d_h(\boldsymbol{a}, \boldsymbol{b}) = \sum_{i=1}^{j} xor(\boldsymbol{a}_i, \boldsymbol{b}_i) \tag{1}$$

The Hamming distance sorting method can be used to approximate nearest neighbor searching task. Firstly, the Hamming distance between the query item code and the dataset code is calculated; then the query results are sorted in an incremental form, and the first k data with the smallest Hamming distance from the query item is obtained as the search result. Since the Hamming distance sorting operation is very fast, it can meet the needs of the fast retrieval of the hashing method.

B. HASH TABLE RETRIEVAL

The hash code is used as the key value to establish a hash table [8], and then the approximate nearest neighbor search of the data is performed according to the hash code of the query item data, and the search time is constant level. However, in the process of hash table retrieval, due to the diversity of data, once the hash code of the query item does not match any hash table of the data to be retrieved, it could cause the retrieval task failed. Therefore, we need to calculate the Hamming distance between the query item hash code and the key value of the data hash table and use the hash table with a smaller Hamming distance as the candidate searching range to improve the success rate of the retrieval.

C. SIMILARITY MEASURES

In addition to the Hamming distance mentioned above, some other similarity measures are used to measure the similarity of data in the process of constructing the hashing algorithm model. Commonly used are Minkowski distance [9] and cosine distance [10].

1) MINKOWSKI DISTANCE

The Minkowski distance is used to measure the similarity between two real vectors, which can be calculated by the L_p norm [11]. For any two real-dimensional vectors $\boldsymbol{a} = (\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_j)$ and $\boldsymbol{b} = (\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_j)$ of the same dimension, their Minkowski distance is defined as:

$$L_p(\boldsymbol{a}, \boldsymbol{b}) = \left(\sum_{i=1}^{j} |\boldsymbol{a}_i - \boldsymbol{b}_i|^p \right)^{1/p} \tag{2}$$

The value of p is not unique. When p is 1, the formula 2 becomes

$$L_1(\boldsymbol{a}, \boldsymbol{b}) = \sum_{i=1}^{j} |\boldsymbol{a}_i - \boldsymbol{b}_i| \tag{3}$$

At this time, the absolute distance of the two vectors is calculated, that is, the Manhattan distance [12]. When p is 2, the formula 2 becomes

$$L_2(\boldsymbol{a}, \boldsymbol{b}) = \left(\sum_{i=1}^{j} |\boldsymbol{a}_i - \boldsymbol{b}_i|^2 \right)^{1/2} \tag{4}$$

At this time is calculated from the Euclidean distance of the two vectors.When p is taken as ∞, the formula 2 becomes

$$L_\infty(\boldsymbol{a}, \boldsymbol{b}) = \max \sum_{i=1}^{\infty} |\boldsymbol{a}_i - \boldsymbol{b}_i| \tag{5}$$

At this time is calculated from the Chebyshev distance [13] of the two vectors. The Minkowski distance of two vectors is positively correlated to their similarity.

2) COSINE DISTANCE

The cosine distance measures the similarity between two vectors by calculating the magnitude of the cosine of the angle between the two real vectors. For any two real-numbered vectors $\boldsymbol{a} = (\boldsymbol{a}_1, \boldsymbol{a}_2, \ldots, \boldsymbol{a}_j)$ and $\boldsymbol{b} = (\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_j)$ of the same dimension, their cosine distance can be calculated as follows:

$$d_{\cos}(\boldsymbol{a}, \boldsymbol{b}) = 1 - \frac{\boldsymbol{a}^T \boldsymbol{b}}{|\boldsymbol{a}| \cdot |\boldsymbol{b}|} \tag{6}$$

The cosine distance of two vectors is positively correlated to their similarity.

D. PERFORMANCE EVALUATION CRITERIA

There are generally four performance evaluation criteria for retrieval performance in hashing methods:

1) PRECISION (P)

It refers to the ratio of the number of nearest neighbor samples in the returned query results to the total number of returned samples, reflecting the retrieval signal-to-noise ratio of the method. The formula for calculating the accuracy rate is:

$$precision = \frac{T}{T + N} \tag{7}$$

where T is the number of nearest neighbor samples related to the query item data in the returned query result, and N is the number of samples in the returned query result that are not related to the query item data. The precision rate is positively correlated to the retrieval performance of a method.

2) RECALL (R)

It refers to the ratio of the number of nearest neighbors in the returned query to the number of samples related to the query data in the dataset, reflecting the success rate of the retrieval process. The formula for calculating the recall rate is:

$$recall = \frac{T}{T + F} \tag{8}$$

where F is the number of data samples associated with the query item data in the dataset but not retrieved. The recall rate is positively correlated to the retrieval performance of a method.

3) PRECISION-RECALL CURVE

In a hashing method, the precision rate and recall rate are mutually constrained. The precision rate and recall rate of the same method are negatively correlated [14]. Therefore, we can use the precision rate and recall rate as the horizontal and vertical coordinates to draw the precision-recall curve to further measure the performance of the retrieval method.

4) AVERAGE PRECISION (AP)

It is obtained by calculating the integral of the precision-recall curve on the abscissa. In the precision-recall curve, the precision rate is a function of the recall rate, recorded as $P = f(R)$. The average accuracy is calculated by integrating the precision rate against the x-axis when the recall rate changes from 0 to 1:

$$AP = \int_0^1 P dR = \int_0^1 f(R) dR \tag{9}$$

In practical applications, the data points of the precision-recall curve are often discrete, so we often use the sequence summation method to calculate the average precision:

$$AP = \frac{1}{L_q} \sum_{k=1}^{n} P(k) \Delta R(k) \tag{10}$$

where n is the number of all samples in the dataset, k is the number of samples returned during the retrieval process, L_q is the number of samples in the dataset related to the query item data, which is the total number of data samples. $P(k)$ is the precision rate of the first k samples returned by the retrieval process. $\Delta R(k)$ is the value of the recall rate when the number of samples varies from $k - 1$ to k. The average precision is equivalent to the average of the query accuracy in a single query data.

5) MEAN AVERAGE PRECISION (MAP)

It is the mean of the average precision of all query data. The calculation method is as follows:

$$mAP = \frac{\sum_{i=1}^{M} AP(q_i)}{M} \tag{11}$$

where q_i is the query sample and M is the total number of query data.

E. ELEMENT-WISE SIGN FUNCTION

When mapping raw multimedia data into a common space, an element-wise sign function:

$$sign(x) = \begin{cases} 1, & x > 0 \\ 0, & x = 0 \\ -1, & x < 0 \end{cases} \tag{12}$$

is often used. It can well normalize the characteristic information into an uniformed hash code, and improve retrieval efficiency. However, after this process, the information contained in the hash code will become sparse, and the loss of some related information is inevitable.

III. METHODOLOGIES

A. DATA-INDEPENDENT METHODS

The hash function design of the data-independent hashing method is independent of the data, and the hash function is generally generated by means of random mapping. The most typical representative method is the Locality-Sensitive

TABLE 1. A brief catalogue of representative hashing methods. For data-dependent methods, *U* means unsupervised methods, *S* means semi-supervised methods, *F* means fully supervised methods.

Data-independent			LSH [15], pLSH [16], SIKH [17], LSH forest [18]
Data-dependent	Single-modal	Spectral-based	SH [19](U), SSH [20](S), S3PLH [21](S), SPEC [22](U), MLSH [23](F), ITQ [24](U)
		Graph-based	AGH [25](U)
		PCA-based	IsoHash [26](U)
		Linear	MLH [27](F), LDAhash [28](F), SDH [29](F)
		kernel-base	BRE [30](F), KSH [31](F), KHLSSH [32](S), SCSSH [33](S)
	Multi/Cross-modal	Spectral-based	CVH [34](U), IMH [35](U)
		Linear	LCMH [36](U)
		Similarity-sensitive	CMSSH [37](F), CRH [38](F)
		Semantic-base	SCM [39](F), STMH [40](F), SePH [41](F)
		Common space	CMFH [42](U), LSSH [43](U)
		Multi-graph	MGH [44](S), S3FH [45](S)
		Deep hash	DCMH [46](F), PRDH [47](F), SSAH [48](F), ADAH [49](F), IISPH [50](F), DMSH [51](F), DRSH [52](F), AGAH [53](F), HRL [54](F), RoPH [55](F), DBRC [56](F)
		Others	HMM [57](F), SODA [58](F)

Hashing (LSH) [15]. The principle of locality-sensitive hashing is to map samples with high similarity in the original space to the same hash bucket with higher probability, which ensures that the hash codes of the neighbor samples in the original space can be as close as possible after hash mapping. The probability that any two samples fall into the same hash bucket is determined by the similarity measure of the two samples to satisfy the locality-sensitive in the mapping process.

The random mapping matrix is independent of the data and is determined by the probability distribution. Since the similarity of neighbor samples is preserved during the mapping process, locality-sensitive hashing is somehow reliable. However, in large-scale data retrieval, because of the hash collision problem, a longer hash code is needed to ensure the precision of the retrieval, which brings additional time and computational overhead, and leads to a decrease in the recall rate. There are many improvements, such as p-stable Locality-Sensitive Hashing (pLSH) [16], which complements the hash function family of LSH; Shift Invariant Kernel Hashing (SIKH) [17], the principle is to use the translation-invariant kernel to project the data from original space into the kernel space to maintain data similarity; LSH Forest [18] optimizes the need to construct multiple hash tables to reduce the hash collision problem in the random mapping process.

B. SINGLE-MODAL METHODS

Since data-independent methods cannot fully utilize the information of given data, it is difficult to obtain good retrieval performance. Hence, many data-dependent methods have been proposed. Earlier methods focused on a single modality, but this is the basis for the development of subsequent research. We will briefly introduce some typical single-modal methods.

1) SPECTRAL HASHING (SH) [19]

This method is an important starting point, which clarifies many basic requirements in this field. It requires data that is neighbors in the original space to remain neighbors after hash mapping, and the hash code must satisfy balance and independence which ensures that the hash code obtained by the mapping is compact and rich in information. The entire hashing process is an NP-hard problem [59]. The author first discretizes the data and then uses the spectral analysis method to optimize the data under the assumption that the data is evenly distributed. Finally, the angular frequency is calculated in the principal component analysis (PCA) [60] direction using the sinusoidal function to divide the data. The spectral hashing method belongs to the orthogonal mapping method, so the quality of the hash code is not high when the variance of the mapping is low. Moreover, in practical applications, the uniform distribution of data is still too idealistic.

2) ANCHOR GRAPH HASHING (AGH) [25]

It is a graph-based hashing method whose principle is similar to the spectral hashing, but it does not require the precondition of uniform data distribution. The method first selects the points in the dataset (generally using the K-means clustering [61]) as an anchor point to approximate the similarity matrix, and converts the similarity between any two data sample points into a sample-anchor relationship. The anchor graph hashing method is based on graph analysis and has strong scalability. It also uses a double hash function to generate a multidimensional hash code to solve the problem of uneven information content in the feature vector generated by the original data.

3) ISOTROPIC HASHING (ISOHASH) [26]

It performs a selection operation on the data space such that the data variance of each dimension is the same. First, use the principal component analysis(PCA) to reduce the dimension

of the data, then calculate a rotation matrix. Through these operations, the variance of each dimension of the data is the same, and the amount of information corresponding to the hash code is the same, which solves the problem of code quality in SH and AGH. Besides, the iterative quantization method can be used to optimize the Gaussian distribution data that satisfies isotropic [62].

4) SEMI-SUPERVISED SPECTRAL HASHING (SSH) [20] & Semi-supervised Sequential Projection Hashing (S3PLH) [21]

The semi-supervised hashing method don't rely on external interactions, instead of using the model assumptions on the data distribution to mark unlabeled samples to improve overall performance.

SSH and S3PLH are both special cases of extending Spectral Hashing (SH) to semi-supervised. They use partial data labels as supervisory information to know if some samples are neighbors (neighbor samples have the same label). In the construction of the hash function, the former mainly uses the similarity matrix, and the latter also adds linear projection and mean thresholding. They all use relaxation processing to ensure the independence and balance of the hash bits. Of course, the retrieval performance of the latter is better than the former.

5) OTHER SINGLE-MODAL METHODS

There are some methods similar to Spectral Hashing (SH), e.g. Similarity Preserving Entropy-based Coding (SPEC) [22], Multi-label Least-Squares Hashing(MLSH) [23], and Iterative Quantization (ITQ) [24].

There are quite a few single-modal methods that apply the idea of linear classification. For example, Minimal Loss Hashing (MLH) [27] uses a hinge-like loss [63] function as a penalty term to process sample points that are particularly close in distance (called positive samples) and particularly far sample points (called negative samples). It is based on the basic principle of structured SVM [64], and the retrieval performance is still good, but it has the disadvantage of the high complexity of of model training, and it is difficult to apply to large-scale datasets. In addition, Linear Discriminant Analysis Hashing (LDAhash) [28] and Supervised Discrete Hashing (SDH) [29] are also based on similar ideas which convert the hash learning problem into a linear classification problem.

Another common single-modal methods trend is kernel-based. Representatives are Binary Reconstructive Embedding (BRE) [30], Supervised Hashing with Kernels (KSH) [31], Kernel Hyper-plane Learning Semi-supervised Hashing (KHLSSH) [32], and Semantic Confidence Semi-supervised Hashing (SCSSH) [33].

C. MULTI/CROSS-MODAL METHODS

1) CROSS-VIEW HASHING (CVH) [34] & INTER-MEDIA HASHING (IMH) [35]

The Cross-View Hashing is an extension of the spectral hashing. The basic idea is to learn the hash function by minimizing the weighted average Hamming distance of different modalities and use the generalized eigenvalue solution method to obtain the minimum value. CVH can be applied to the data retrieval of multiple modalities, but the differences between modalities are not fully considered, hence the retrieval performance is limited. CVH is an unsupervised method. It uses graphs to describe the similarity between modalities. It can also use the label information to solve the similarity matrix and convert it into a supervised method.

The basic idea of the Inter-Media Hashing is similar to CVH, but it fully considers the associations and differences between modalities and emphasizes maintaining the inter-modal and intra-modal similarity of samples that are nearest neighbors to each other. However, IMH needs to calculate the similarity map of a large number of samples, hence its retrieval effect is guaranteed at the expense of time complexity, which is not suitable for application to large-scale datasets.

2) LINEAR CROSS-MODAL HASHING (LCMH) [36]

As a typical representative of the linear method, LCMH inherits some of the ideas of AGH [25]. By using the scalable k-means algorithm, the distance between the data point and the center point is calculated to maintain the similarity inside the data modalities; by minimizing the distance to the same object from different modalities in public space to ensure the similarity between the modalities. The advantage of LCMH is that time complexity is linear, which can increase efficiency for large-scale data retrieval.

3) COLLECTIVE MATRIX FACTORIZATION HASHING (CMFH) [42] & LATENT SEMANTIC SPARSE HASHING (LSSH) [43]

CMFH excels in the unsupervised hashing methods. It assumes that the hash codes of all modalities data are consistent when mapped to the common Hamming space, and the collective matrix factorization [65] method is used to help construct the hash function model. This method is optimized using a loop iteration. This method integrates the data of different modalities and seeks the common representation of the consistency of each modality, which can improve the retrieval effect.

The Latent Semantic Sparse Hashing (LSSH) is an extension of CMFH. It also narrows the semantic gap between modalities by learning uniform hash codes for semantically similar data. In particular, It uses the sparse coding [66] method to obtain high-level significant feature information, and uses the matrix decomposition to learn the latent semantic information. The resulting information is then mapped into a federated public space. By combining the potential semantic information of different modalities, the retrieval performance is greatly improved.

4) MULTI-GRAPH HASHING (MGH) [44] & SEMI-SUPERVISED SEMANTIC FACTORIZATION HASHING (S3FH) [45]

MGH is a semi-supervised method, which is based on the hashing with the graph structure by constructing the

neighborhood graph on the training data and can be extended to multiple modalities. Construct the semantic similarity matrix by using the label, and combine them to make complementary hash function learning by sequential.

The MGH simply combines the multi-modal graph and the semantic similarity matrix, resulting in a large amount of noise in the hash code, which does not preserve the semantic association between the modalities well. The S3FH improves the semantic labels and constitutes a joint framework consisting of three parts: Semantic factorization, Multi-graph semi-supervised learning, and Multi-modal correlation.

- Semantic factorization refers to a given prediction label matrix, which can be decomposed into a hash code by matrix decomposition. This part effectively preserves the semantic relevance of labels. In the S3FH, there is no orthogonal constraint in the semantic factor decomposition process, so the quantization loss of the hash code is very low, and the semantic information in all dimensions is balanced.
- Multi-graph semi-supervised learning uses the anchor graph method [67] to calculate the multi-graph matrix of each modality more efficiently. By properly merging the modalities, the more accurate prediction of the label matrix, the more semantic-related information of hash code is retained.
- Multi-modal correlation refers to learning a hash function for each modality and mapping them into a unified Hamming space.

The S3FH uses a joint framework, and each part can interact, making the method perform better than the previous works.

5) CROSS-MODAL SIMILARITY-SENSITIVE HASHING (CMSSH) [37] & CO-REGULARIZED HASHING (CRH) [38]

CMSSH was proposed in 2010 and is almost the first supervised cross-modal method in recent years. It first generates some positive and negative data pairs according to the similarity of the samples, then constructs two sets of linear hash functions as weak classifiers. Each hash code training is a binary classification process. The hash code learning is performed by the Boosting method [68], and the weak classifiers are combined into strong classifiers. The hash learning processing corresponds to a non-convex problem, which needs to be subjected to relaxation processing and then obtained by eigenvalue decomposition to obtain the hash codes.

CMSSH does not consider the similarity within the data modalities. Co-Regularized Hashing (CRH) has improved the CMSSH and added a loss function within the modalities. This method learns a single-bit hash function by solving the difference of convex functions and then learns multiple bits by sequential learning. This allows the deviation introduced by the hash function to be minimized sequentially. It uses a smooth clip inverse variance bias function to connect the similarities between the inter-modal relationships and the projections that form the hash code. In addition, the CRH method defines a loss term between modalities for a large edge hash function, projects the data away from zero to implement generalization, and effectively maintains the differences between modalities.

6) SEMANTIC CORRELATION MAXIMIZATION (SCM) [39]

SCM is a supervised method, which means that it makes full use of category tag information. This method learns the representation of the public space based on the label information, that is, the labels are used to map samples of different categories far away, and the mapping of samples of the same category is as close as possible. It uses semantic tag vectors to calculate the semantic similarity between data samples. A linear transformation is performed on the similarity matrix to facilitate calculation, and a new semantic similarity matrix is obtained. The SCM uses the Spectral Relaxation [19] method to construct the hash code, discarding the $sign(\cdot)$ term in the objective function, and by adding orthogonal constraints, each bit of the hash code satisfies the balance and is uncorrelated. SCM uses the orthogonal projection method based on eigenvalue decomposition to perform hash function learning, which has low time complexity, but at the same time causes large quantization loss and affects the retrieval performance.

7) SEMANTIC TOPIC MULTIMODAL HASHING (STMH) [40]

This method explicitly uses the implicit information of each modality. Specifically, the hidden text topic is explored by clustering the text data to better generate the hash code. The semantic information of image data is explored by matrix decomposition, and a norm is introduced to enhance the robustness of matrix decomposition. If different modal data have the same semantics, then a common semantic space, such as a text topic, can be described using corresponding image semantic information. STMH well maintains the discrete nature of hash codes. Each bit of the hash code indicates whether the text or picture contains the corresponding subject or concept. By maintaining the discrete nature of the hash code, It is more suitable for the hash learning mode, and also achieves better retrieval performance.

8) SEMANTICS-PRESERVING HASHING (SEPH) [41]

SePH uses the semantic correlation matrix of the sample data as the supervised information, converts it to the learned binary code into a probability distribution, and learns the hash code by minimizing the KL-divergence of the two probability distributions. In the process of hash code learning, kernel logistic regression [69] is used as a nonlinear projection method to map data features into binary codes. For any test set sample data of different modalities, SePH ensures that a uniform hash code can be obtained by predicting the probability of each modal hash code. The model of this method is complicated and requires a longer training time, but at the same time, it can obtain higher retrieval accuracy.

TABLE 2. Notations.

c	Hash code length
n	Number of data points
$\boldsymbol{X} = \{\boldsymbol{x}_i\}_{i=1}^n$	Image data
$\boldsymbol{Y} = \{\boldsymbol{y}_j\}_{j=1}^n$	Text data
θ	Parameters of neural networks
$f(\boldsymbol{x}_i;\theta_x) \in \mathbb{R}^c$	Image features of data ponit i
$g(\boldsymbol{y}_j;\theta_y) \in \mathbb{R}^c$	Text features of data ponit j
l	Category labels
$\boldsymbol{F} \in \mathbb{R}^{c\times n}$	Image features of n data ponits
$\boldsymbol{G} \in \mathbb{R}^{c\times n}$	Text features of n data ponits
$\boldsymbol{L}$	Category labels features
$\boldsymbol{H}$	Predicted hash code
$\boldsymbol{B}$	Hash code (for image/text)
$\boldsymbol{S}$	Similarity matrix
Θ	Inner matrix product
$\alpha, \beta, \gamma, \eta, \lambda$	Hyper-parameters
J	Loss items
$tr(\cdot)$	Trace of a matrix
$\|\cdot\|_F$	Frobenius norm of a matrix
$\sigma(\cdot)$	Sigmoid function

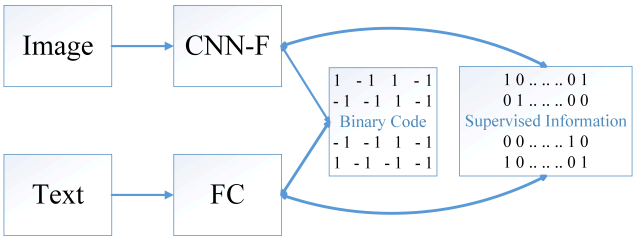

FIGURE 2. Framework of DCMH.

D. DEEP HASHING METHODS

In recent years, deep learning has performed well in many fields. The deep features extracted by deep learning method contain richer semantic information and have a stronger ability to express the original data. Therefore, the combination of deep learning and hashing methods applied to multimodal retrieval can significantly improve the retrieval efficiency. Based on this idea, many excellent methods have been proposed in recent years. Here we will introduce a few recent representative deep hashing methods. Besides, many new methods [50]–[53] are not described in detail, but also worthy of reference.

Since the following narration will involve some complicated formula expressions, we first make some notations of the symbols and expressions that will appear frequently (but not all), see Table 2.

1) DEEP CROSS-MODAL HASHING (DCMH) [46]

This method integrates feature learning and hash learning into an end-to-end framework involving two modal data of images and texts with excellent retrieval performance. The basic framework of DCMH is shown in Figure 2.

The DCMH framework is an end-to-end design, which means that each part can provide feedback to another part during the learning process. For the extraction of image modal data features, the network structure of CNN-F [70] is adopted, which consists of 5 convolutional layers and 3 fully connected layers. The first seven layers use Rectified Linear Unit (ReLU) [71] as the activation function, and the last fully connected layer used identity function as the activation function. For text modal data, it is vectorized using the bag-of-word (BoW) and then passed as input to a deep neural network with three fully connected layers. The activation function of the first two layers is ReLU. The last fully-connected layer uses the identity function.

The objective function of DCMH is:

$$\begin{aligned}\min_{\boldsymbol{B},\boldsymbol{B}^{(x)},\boldsymbol{B}^{(y)},\theta_x,\theta_y} J = &-\sum_{i,j=1}^{n}(S_{ij}\Theta_{ij}^{xy} - \log(1+e^{\Theta_{ij}^{xy}})) \\ &+\gamma(\left\|\boldsymbol{B}^x - \boldsymbol{F}\right\|_F^2 + \left\|\boldsymbol{B}^y - \boldsymbol{G}\right\|_F^2) \\ &+\eta(\|\boldsymbol{F}\cdot\boldsymbol{1}\|_F^2 + \|\boldsymbol{G}\cdot\boldsymbol{1}\|_F^2) \\ s.t.\ \ &\boldsymbol{B}^x = \boldsymbol{B}^y = \boldsymbol{B} \in \{-1,+1\}^{c\times n}\end{aligned} \tag{13}$$

where $\Theta_{ij}^{xy} = \frac{1}{2}\boldsymbol{F}_{*i}^T\boldsymbol{G}_{*j}$.

- The first term in formula 13 is the negative log likelihood of the cross-modal similarities. By optimizing this term, the similarity between image and text features under the supervisory information can be preserved. The likelihood function is defined as:

$$p(S_{ij}|\boldsymbol{F}_{*i},\boldsymbol{G}_{*j}) = \begin{cases}\sigma(\Theta_{ij}^{xy}), & S_{ij}=1\\ 1-\sigma(\Theta_{ij}^{xy}), & S_{ij}=0\end{cases} \tag{14}$$

  where $\Theta_{ij}^{xy} = \frac{1}{2}\boldsymbol{F}_{*i}^T\boldsymbol{G}_{*j}$, $\sigma(\Theta_{ij}^{xy}) = \dfrac{1}{1+e^{-\Theta_{ij}^{xy}}}$.
- The second item ensures that when the image and text data are converted into hash codes, the original cross-modal similarity is preserved as much as possible, and corresponding to the correct supervision information which can reduce the quantization loss.
- The third item ensures the balance of the hash code and maximizes the valid information contained in each binary code.

The restriction $\boldsymbol{B}^x = \boldsymbol{B}^y = \boldsymbol{B}$ in formula 13 is applicable to training data, and the meaning is to share a common hash representation of different modal data to improve training efficiency. DCMH uses alternating learning strategies to optimize θ_x, θ_y, and $\boldsymbol{B}$ during the training process (fixing two and optimizing the other). For any sample point that is not in the training set, DCMH only needs to obtain data in a single modality, then it can retrieve its other modal data through hash codes.

However, although DCMH was a groundbreaking deep hashing method with excellent performance, it didn't deal well with the intra-modal correlation of data and the further association between hash codes and features of different modalities.

2) PAIRWISE RELATIONSHIP DEEP HASHING (PRDH) [47]

The framework of PRDH is very similar to DCMH, which is an improvement of the latter. Its main innovation is integrating different types of pairwise constraints to better reflect the hash code similarity from inter-modal and intra-modal data. In addition, the method introduces additional decorrelation constraints, which enhanced the independence between the bits of the hash code.

The loss function during the training process consists of the following four items:

- inter-modal pairwise embedding loss:

$$\begin{aligned} J_1 &= -\log p(S_{ij}|\boldsymbol{F}_{*i}, \boldsymbol{G}_{*j}) \\ &= -\sum_{S_{ij}\in \boldsymbol{S}} \log p(S_{ij}|\boldsymbol{F}_{*i}, \boldsymbol{G}_{*j}) \\ &= -\sum_{S_{ij}\in \boldsymbol{S}} \left(S_{ij}\Theta_{ij}^{xy} - \log(1+e^{\Theta_{ij}^{xy}})\right) \end{aligned} \tag{15}$$

  where $\Theta_{ij}^{xy} = \frac{1}{2}\boldsymbol{F}_{*i}^T\boldsymbol{G}_{*j}$. By optimizing this loss item, the Hamming distance between similar instances would be reduced, and the Hamming distance between dissimilar instances would be expanded.
- intra-modal pairwise embedding loss:
  For image modality:

$$J_2 = -\sum_{S_{ij}\in \boldsymbol{S}} \left(S_{ij}\Theta_{ij}^{x} - \log(1+e^{\Theta_{ij}^{x}})\right) \tag{16}$$

  where $\Theta_{ij}^{x} = \frac{1}{2}\boldsymbol{F}_{*i}^T\boldsymbol{F}_{*j}$.
  For text modality:

$$J_3 = -\sum_{S_{ij}\in \boldsymbol{S}} \left(S_{ij}\Theta_{ij}^{y} - \log(1+e^{\Theta_{ij}^{y}})\right) \tag{17}$$

  where $\Theta_{ij}^{y} = \frac{1}{2}\boldsymbol{G}_{*i}^T\boldsymbol{G}_{*j}$. By optimizing this loss item can improve the validity of the hash code information and provide its own instance identification capability within the modality, thereby improving the cross-modal retrieval performance.
- decorrelation loss:

$$\begin{aligned} J_4 = &\frac{1}{2}\left(\left\|\boldsymbol{C}^x\right\|_F^2 - \left\|diag\left(\boldsymbol{C}^x\right)\right\|_F^2\right) \\ &+\frac{1}{2}\left(\left\|\boldsymbol{C}^y\right\|_F^2 - \left\|diag\left(\boldsymbol{C}^y\right)\right\|_F^2\right) \end{aligned} \tag{18}$$

  where $\boldsymbol{C}^x = \frac{1}{T}\sum_{n=1}^{T}(\boldsymbol{F}_{in} - \mu_i)(\boldsymbol{F}_{jn} - \mu_j)$ and $\boldsymbol{C}^y = \frac{1}{T}\sum_{n=1}^{T}(\boldsymbol{G}_{in} - \mu_i)(\boldsymbol{G}_{jn} - \mu_j)$ is the covariance matrix between two different hash code bits from image/text modalities, $i, j \in \{1, 2\ldots c\}$, $\mu_* = \frac{1}{T}\sum_{n=1}^{T}F_{*n}/\frac{1}{T}\sum_{n=1}^{T}G_{*n}$ is the instance mean of feature over the batch, and T is the batch size. By optimizing this loss item can reduces the redundancy-related information between the hash code bits and enhances the hash code independence to maximize its information representation efficiency.
- regularization loss:

$$R = \|\boldsymbol{B} - \boldsymbol{F}\|_F^2 + \|\boldsymbol{B} - \boldsymbol{G}\|_F^2 + \|\boldsymbol{F}\cdot\boldsymbol{1}\|_F^2 + \|\boldsymbol{G}\cdot\boldsymbol{1}\|_F^2 \tag{19}$$

  where B is the unified hash code for the two modalities. Optimizing this item can reduce the quantization loss and ensure the balance of the hash code.

The overall loss function is:

$$J = J_1 + J_2 + J_3 + \lambda J_4 + \gamma R \quad s.t.\, \boldsymbol{B} \in \{-1, +1\}^{c\times n} \tag{20}$$

Compared with DCMH, PRDH mainly increases the loss optimization within the modal of the training process and the decorrelation loss optimization of the hash code itself, which greatly improves the data utilization and achieves better retrieval performance.

3) SELF-SUPERVISED ADVERSARIAL HASHING (SSAH) [48]

This method introduces mechanisms such as self-supervised semantic generation and adversarial learning, and has made breakthrough progress in retrieval performance.The framework of SSAH is shown in Figure 3.There are two innovations for SSAH:

a: SELF-SUPERVISED SEMANTIC GENERATION

SSAH uses multi-label annotation to better bridge the fine-grained semantic similarity between different modalities. A fully connected deep neural network called LabNet is designed to extract features from multi-label information, thus transforming label information into self-supervised semantic information. The final objective function of LabNet is:

$$\begin{aligned} \min_{\boldsymbol{B}^l,\theta^l,\widehat{\boldsymbol{L}}} J^l &= \alpha J_1 + \gamma J_2 + \eta J_3 + \beta J_4 \\ &= -\alpha\sum\nolimits_{i,j=1}^{n}(S_{ij}\Theta_{ij}^l - \log(1+e^{\Theta_{ij}^l})) \\ &\quad -\gamma\sum\nolimits_{i,j=1}^{n}(S_{ij}\Theta_{ij}^h - \log(1+e^{\Theta_{ij}^h})) \\ &\quad +\eta\left\|\boldsymbol{H}^l - \boldsymbol{B}^l\right\|_F^2 + \beta\left\|\widehat{\boldsymbol{L}} - \boldsymbol{L}\right\|_F^2 \\ s.t. &\boldsymbol{B}^l \in \{-1, 1\}^{c\times n} \end{aligned} \tag{21}$$

where $\Theta_{ij}^l = \frac{1}{2}\boldsymbol{L}_{*i}{}^T\boldsymbol{L}_{*j}$, $\Theta_{ij}^h = \frac{1}{2}\boldsymbol{H}_{*i}^l{}^T\boldsymbol{H}_{*j}^l$, $\boldsymbol{H}^l$ is the predicted hash code of category labels, $\widehat{\boldsymbol{L}}$ is the predicted category labels. J_1 is used to maintain the similarity of semantic features. J_2 ensure that instances with similar label have similar hash codes. J_3 is the approximate loss for the binarization of the learned hash codes. J_4 is the classification loss between the original category labels and the predicted category labels.

b: ADVERSARIAL LEARNING

Under the influence of LabNet, semantic correlation can be maintained between different modalities. However, the inconsistent distribution of different modalities is not conducive

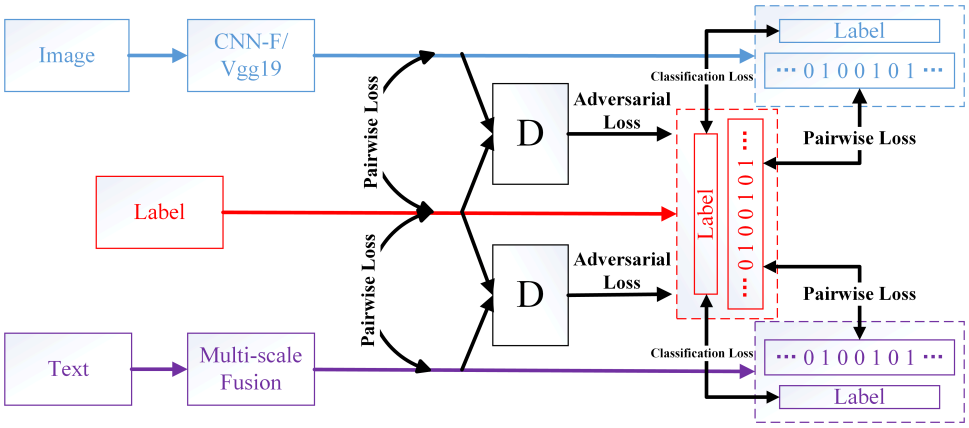

FIGURE 3. Framework of SSAH.

to the construction of the unified hash code. In order to eliminate the modality gap to achieve more efficient retrieval, SSAH designed two discriminators of image and text for adversarial learning. The discriminators have classified the input semantic features into class 0 or 1.

The introduction of adversarial learning makes the hash codes of different modalities more closely related to the original data. However, the instability in adversarial network training may also have a greater impact on the overall hash learning process. Operations such as gradient descent and normalization that are commonly used in deep hashing will bring additional noise, and the adversarial network is very sensitive to noise. In addition, SSAH uses adversarial learning to train the 0-1 discriminator. For the sparse modal data features (e.g. bag-of-words vectors), it is difficult to ensure that the optimization effect is globally effective and continuously effective.

In general, compared with the previous method, SSAH has a breakthrough innovation and also provides a possible direction for future research. However, the stability of the training process needs to be improved.

4) ATTENTION-AWARE DEEP ADVERSARIAL HASHING (ADAH) [49]

ADAH has a similar framework to SSAH, and it uses adversarial learning. In addition, another important innovation of it is the introduction of attention-aware mechanisms as shown in figure 4. The implementation is to further process the extracted image/text features (convolution for image features, fully connected for text features), and the softmax is used for rough classification, then use a threshold function to generate a binary mask. The original feature is multiplied to the binary mask in elements-wise to divide into attention-aware and inattention-aware features (corresponding to image regions/text segments).

The introduction of the attention-aware mechanisms means that ADAH seeks internal correlation from different modal data itself. This is also an improvement direction that is worth continuing to explore, especially for the current situation that

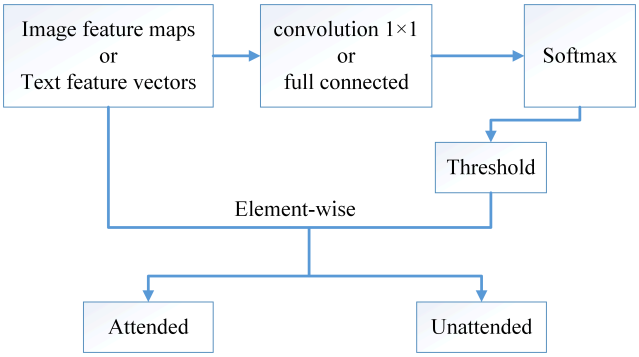

FIGURE 4. Attention-aware mechanisms.

multimedia data is becoming more and more abundant both number and content. It has great application potential.

However, excessively using the attention-aware mechanisms on the data features may ignore too much effective information and affect the retrieval authenticity, while the overly specific processing will reduce the generality of the method. In the future, the use of the attention-aware mechanisms in deep hashing will be further explored in terms of authenticity and generality.

IV. EVALUATION BENCHMARKS

A. DATASETS

There are many benchmark public datasets to evaulate the performance of hashing methods for multimodal retrieval. The common is the image-text dataset, e.g. Wikipedia [72], MIRFLICKR-25k [73], NUS-WIDE [74], IAPR-TC12 [75], MS COCO [76].

- **Wikipedia**'s data comes from articles and pictures crawled from the Wikipedia website. The commonly used version is 2,866 image-text pairs after sorting, which are divided into 10 categories. The text data is usually word segmentation from the articles.
- **MIRFLICKR-25k** contains 25,000 image-text pairs grabbed from Flickr website, which is divided into

TABLE 3. A commonly used statistics of different datasets.

Datasets	Total	Training	Test	Labels
Wikipedia	2,866	2,173	693	10
MIRFLICKR-25k	20,015	10,000	2,000	24
NUS-WIDE	195,834	10,500	2,100	21
IAPR-TC12	20,000	10,000	2,000	255
MS COCO	85,000	10,000	5,000	80

24 categories. Generally, instances with no less than 20 tags will be selected for a total of 20,015.

- **NUS-WIDE** contains 269,648 annotated web pictures. Each picture is associated with one or more text tags belonging to 81 semantic concepts. Selecting the most frequent 21 concepts can get 195,834 instances of image-text pairs. In some methods, the selection criteria are different. For example, PRDH [47] selects 10 concepts with a total of 186,577, and SSAH [48] got 190,421 instances by removing the pictures with empty tags after word frequency statistics processing.
- **IAPR-TC12** contains 20,000 image-text pairs annotated with 255 labels.
- **MS COCO** contains 82,783 image-text pairs for training, 40,504 for validation, and 40,775 for testing (2014 release). SSAH [48] selects 80,000 image-text pairs for training and 40,000 for validation. 5,000 image-text pairs are randomly selected from the test set, forming a total of 85,000 instances of image-text pairs with 80 labels.

Generally, a small number of instances of the dataset will be used for testing (queries), then 4-5 times the number of test instances will be used for training, and the part except the test part will be used for retrieval. A commonly used training/ test numbers statistics is shown in Table 3. Because the number of datasets is different, the same method perform differently on different datasets.

There are many ways to extract the features of the original data in the datasets mentioned above. For image data, traditional hand-crafted method e.g. GIST [77], SIFT [78], Bag-of-Visual-Words (BoVW) and deep learning methods e.g. CNN-F [70], VGG19 [79] are commonly used. For text data, Bag-of-Words (BoW), Word2Vec [80], Doc2Vec [81] are commonly used.

B. PUBLISHED EXPERIMENTAL RESULTS

The time span of hashing methods introduced in this paper is large. There are inconsistent environments and baselines in the implementation of the early and late methods. Therefore, some methods (mostly deep hashing methods) in the later stage are selected to compare the experimental performance. The mean Average Precision (mAP) is often used to evaluate the performance of hashing methods. Taking image and text query as an example, the experimental result is that the mAP of different query cases, also include the comparison of different hash code bit lengths.

The experimental results in Table 4-8 is derived from the published papers. It will help to consult the retrieval performance of some representative published methods in public datasets and facilitate the subsequent research as evaluation benchmarks. Since the experimental settings are not the same, there is no reference for superiority and inferiority. Different datasets have different performances in retrieval tasks due to their data volume and data complexity. In addition, it is also affected by experimental factors such as training hyper-parameters.

Normally, the performance evaluation of a hashing method also needs to include hyperparameter sensitivity experiments, key module ablation experiments and training time comparisons. Since the focus of this article is on the improvement of retrieval accuracy by different methods, it is not shown and discussed here which would be left in future work.

V. DISCUSSION

A. QUALITATIVE COMPARISON

1) MODALITY EXTENSION

Hashing methods for single-modal, cross-modal and multi-modal retrieval have their own characteristics, scope, advantages and disadvantages. In general, single-modal method is the basis for transition to cross-modal and multi-modal method. From the single image or text retrieval to the image and text pair retrieval has added more modalities into the task, and the versatility is gradually enhanced. See Table 9 for details. The expansion of modalities has brought about an increase in the differentiation of feature forms, the semantic gap has also widened, and the compatibility requirements of public spaces involved in retrieval have increased. This leads to an increase in the difficulty of optimizing the algorithm, and it takes more hardware and time to achieve the ideal retrieval performance. In future research, how to balance the common and unique parts of the algorithm to optimize the entire training process more efficiently should be the focus of consideration.

2) SUPERVISION MODE

The data-dependent method is divided into three categories: unsupervised, semi-supervised, and supervised according to the use of data supervision information. Each of the three types of methods has its scope of application, advantages, and disadvantages. In general, unsupervised methods are more suitable for small-scale, data-distributed retrieval tasks; semi-supervised method could perform well in absence of label information; supervised methods often achieve better search performance because of the full use of label information. Besides, the latest self-supervised (label information directly participates in hash code generation) methods are a hot trend in the future because of the flexible use of supervision information, and the organic participation of labels in data feature extraction and hash learning.See Table 10 for details. Hash retrieval is essentially a statistical task. Therefore, whether it is the early traditional hashing method or the

TABLE 4. mAP of different hashing methods on Wikipedia. The total number of instances is 2,866 with 2,173 for training and 693 for testing. Image data is represented as 128-D Bag-of-Visual-Words (BoVW) vectors, and text data as 10-D topic vectors. "i → t" denotes the case where the query is image and the database is text, and "t → i" denotes the case where the query is text and the database is image (the same below). The results are directly cited from SePH [41].

Dataset	WIkipedia					
Code length	16 Bits		32 Bits		64 Bits	
Query mode	i → t	t → i	i → t	t → i	i → t	t → i
CMSSH [37]	0.1877	0.1630	0.1771	0.1617	0.1646	0.1539
CVH [34]	0.1257	0.1185	0.1212	0.1034	0.1215	0.1024
IMH [35]	0.1573	0.1463	0.1575	0.1311	0.1568	0.1290
LSSH [43]	0.2141	0.5031	0.2216	0.5224	0.2218	0.5293
CMFH [42]	0.2132	0.4884	0.2259	0.5132	0.2362	0.5269
SCM [39]	0.2210	0.2134	0.2337	0.2366	0.2442	0.2479
SePH [41]	0.2787	0.6318	0.2956	0.6577	0.3049	0.6646

TABLE 5. mAP of different hashing methods on MIRFlickr-25k. The total number of instances is 20,015 with 10,000 for training and 2,000 for testing. Image data is represented as VGG19 features, and text data as 1,386-D Bag-of-Words (BoW) vectors. The results are directly cited from ADAH [49].

Dataset	MIRFlickr-25k					
Code length	16 Bits		32 Bits		64 Bits	
Query mode	i → t	t → i	i → t	t → i	i → t	t → i
CMFH [42]	0.6377	0.6365	0.6418	0.6399	0.6451	0.6429
SCM [39]	0.6851	0.6939	0.6921	0.7012	0.7003	0.7060
STMH [40]	0.6132	0.6074	0.6219	0.6153	0.6274	0.6217
SePH [41]	0.7123	0.7216	0.7194	0.7261	0.7232	0.7319
DCMH [46]	0.7410	0.7827	0.7465	0.7900	0.7485	0.7932
PRDH [47]	0.7499	0.7890	0.7546	0.7955	0.7612	0.7964
ADAH [49]	0.7563	0.7922	0.7719	0.8062	0.7720	0.8074

TABLE 6. mAP of different hashing methods on NUS-WIDE. The total number of instances is 195,834 with 10,500 for training and 2,100 for testing. Image data is represented as VGG19 features, and text data as 1,000-D Bag-of-Words (BoW) vectors. The results are directly cited from ADAH [49].

Dataset	NUS-WIDE					
Code length	16 Bits		32 Bits		64 Bits	
Query mode	i → t	t → i	i → t	t → i	i → t	t → i
CMFH [42]	0.4900	0.5031	0.5053	0.5187	0.5097	0.5225
SCM [39]	0.5409	0.5344	0.5485	0.5412	0.5553	0.5484
STMH [40]	0.4710	0.4471	0.4864	0.4677	0.4942	0.4780
SePH [41]	0.6037	0.5983	0.6136	0.6025	0.6211	0.6109
DCMH [46]	0.5903	0.6389	0.6031	0.6511	0.6093	0.6571
PRDH [47]	0.6107	0.6527	0.6302	0.6916	0.6276	0.6720
ADAH [49]	0.6403	0.6789	0.6294	0.6975	0.6520	0.7039

current deep hashing method, the full use of tag information has important significance. However, blindly pursuing retrieval performance under supervised conditions will lead to poor robustness of the algorithm in the face of incomplete data composition in reality. Therefore, in future research, we need to fully consider the performance of the algorithm in data with different labeling degrees.

3) DEEP LEARNING

The use of deep learning methods for data feature extraction and hash learning has a huge impact on retrieval performance. The general trend is that the deep learning method is significantly better than the traditional method in all respects, because the former is data-dependent, and its improvement in performance depends on a large increase in data scale. However, it also brings greater hardware costs in storage and calculation. See Table 11 for details. Under the retrieval task of a large data scale, the deep hashing method performs well. However, it doesn't mean that the traditional methods should be deprecated. In fact, the deep hashing method only combines part of the idea of deep learning. In terms of feature extraction, the black box processing characteristics of deep

TABLE 7. mAP of different hashing methods on IAPR TC-12. The total number of instances is 20,000 with 10,000 for training and 2,000 for testing. Image data is represented as VGG19 features, and text data as 2,912-D Bag-of-Words (BoW) vectors. The results are directly cited from ADAH [49].

Dataset	IAPR TC-12					
Code length	16 Bits		32 Bits		64 Bits	
Query mode	i → t	t → i	i → t	t → i	i → t	t → i
CMFH [42]	0.4189	0.4168	0.4234	0.4212	0.4251	0.4277
SCM [39]	0.3692	0.3453	0.3666	0.3410	0.3802	0.3470
STMH [40]	0.3775	0.3687	0.4002	0.3897	0.4130	0.4044
SePH [41]	0.4442	0.4423	0.4563	0.4562	0.4639	0.4648
DCMH [46]	0.4526	0.5185	0.4732	0.5378	0.4844	0.5468
PRDH [47]	0.5003	0.5244	0.4935	0.5434	0.5135	0.5548
ADAH [49]	0.5293	0.5358	0.5283	0.5565	0.5439	0.5648

TABLE 8. mAP of different hashing methods on MS COCO. The total number of instances is 85,000 with 10,000 for training and 5,000 for testing. Image data is represented as VGG19 features, and text data as 2,000-D Bag-of-Words (BoW) vectors. The results are directly cited from SSAH [48].

Dataset	MS COCO					
Code length	16 Bits		32 Bits		64 Bits	
Query mode	i → t	t → i	i → t	t → i	i → t	t → i
CVH [34]	0.4410	0.4130	0.4280	0.4020	0.4020	0.3880
STMH [40]	0.4450	0.4460	0.4820	0.4780	0.5020	0.5060
CMSSH [37]	0.5040	0.4170	0.4950	0.4200	0.4920	0.4160
SCM [39]	0.4980	0.4920	0.5560	0.5560	0.5650	0.5680
SePH [41]	0.4890	0.4850	0.5020	0.4950	0.4990	0.4850
DCMH [46]	0.4970	0.5070	0.5060	0.5200	0.5110	0.5270
SSAH [48]	0.5500	0.5520	0.5770	0.5780	0.5760	0.5780

TABLE 9. Comparison of hashing methods in different modal types (e.g. image and text query).

Model type	single-modal	cross-modal	multi-modal
Query item	image/text	image/text	image + text
Learned features	mage or text	image + text	Image + text
Learning space	Image/text	shared subspace	Image + text
Semantic enhancement	none	part	yes

TABLE 10. Comparison of hashing methods in different supervision mode.

Supervision mode	unsupervised	semi-supervised	supervised
Label use	none	part	yes
Data process	simple	simple	complicated
Hash learning	simple	complicated	complicated
Retrieval performance	fair	average	good
Performance in Large-scale data	poor	fair	good

TABLE 11. Comparison of traditional hashing and deep hashing methods.

Method	Parameter scale	Modeling complexity	Hardware cost	Generality	Retrieval performance
traditional	small	complicated	small	poor	fair
deep	large	simple	large	good	good

learning may lead to the omission of key information of some raw data. The optimization process of deep learning methods also depends on a lot of manual fine-tuning. In future research, when improving the deep hashing method, it is necessary to consider the refinement and effectiveness of the feature extraction process. An automatic machine learning

mechanism can also be introduced to participate in the optimization.

B. POSSIBLE DEVELOPMENT TRENDS

The existing hashing methods had excellent performance, but due to the continuous development of technology and the growing demand for reality, it still highlights their shortcomings and reveals the possibility of improvement. Future research deserves attention in the following directions:

- The combination of deep learning and hashing method can be better than previous works, such as combining the adversarial networks [48], [49] and other modules [54], which is worth further exploration.
- There are huge opportunities for improvement in the optimization process of hash learning. Some existing methods [55], [56] can provide some reference.
- Making full use of the supervised information can improve the retrieval performance, but since the data supervised information in the real world is often missing or incorrect, how to ensure a certain retrieval performance in this case will be a hot research direction.
- In the existing methods, the scope of application is often in images and texts query. Only a few research works [57], [58] has focused on audio and video modalities. In the future, we can consider extending generality into a larger range of multimodal data.

VI. CONCLUSION

This paper gives a review of hashing methods for multimodal retrieval, introduces many representative methods and compares them. Some of them are early methods, but still instructive in this research field. Their potential for improvement is worthy of attention. Some are relatively recent methods, have outstanding performance, and lead the current research hotspots, but at the same time, there are still many unresolved problems.

It is noteworthy that the authors focus on clarifying the development process, identifying problems to explore the future direction of improvement rather than listing methods, so there is no detailed description of all the methods. We believe this review will contribute to the development of this research area.

REFERENCES

[1] N. Roussopoulos, S. Kelley, and F. Vincent, "Nearest neighbor queries," *ACM SIGMOD Rec.*, vol. 24, no. 2, pp. 71–79, Jun. 1995.

[2] H. Xu, J. Wang, Z. Li, G. Zeng, S. Li, and N. Yu, "Complementary hashing for approximate nearest neighbor search," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 1631–1638.

[3] R. W. Hamming and W. L. Mammel, "A note on the location of the binary point in a computing machine," *IEEE Trans. Electron. Comput.*, vols. EC–14, no. 2, pp. 260–261, Apr. 1965.

[4] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.

[5] C. Roda and J. Thomas, "Attention aware systems: Theories, applications, and research agenda," *Comput. Hum. Behav.*, vol. 22, no. 4, pp. 557–587, Jul. 2006.

[6] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[7] J. L. Bentley and R. Sedgewick, "Fast algorithms for sorting and searching strings," in *Proc. 8th Annu. ACM-SIAM Symp. Discrete Algorithms*, 1997, pp. 360–369.

[8] J. S. Park, M.-S. Chen, and P. S. Yu, *An Effective Hash-Based Algorithm for Mining Association Rules*, vol. 24. New York, NY, USA: ACM Press, 1995.

[9] M. Ichino and H. Yaguchi, "Generalized minkowski metrics for mixed feature-type data analysis," *IEEE Trans. Syst., Man, Cybern.*, vol. 24, no. 4, pp. 698–708, Apr. 1994.

[10] J. Wang, W. Liu, S. Kumar, and S.-F. Chang, "Learning to hash for indexing big data—A survey," *Proc. IEEE*, vol. 104, no. 1, pp. 34–57, Dec. 2015.

[11] E. Lutwak, D. Yang, and G. Zhang, "On the lp-minkowski problem," *Trans. Amer. Math. Soc.*, vol. 356, no. 11, pp. 4359–4370, 2004.

[12] W. Kong, W.-J. Li, and M. Guo, "Manhattan hashing for large-scale image retrieval," in *Proc. 35th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, 2012, pp. 45–54.

[13] K. Weinberger, A. Dasgupta, J. Attenberg, J. Langford, and A. Smola, "Feature hashing for large scale multitask learning," Feb. 2009, *arXiv:0902.2206*. [Online]. Available: https://arxiv.org/abs/0902.2206

[14] M. Buckland and F. Gey, "The relationship between recall and precision," *J. Amer. Soc. Inf. Sci.*, vol. 45, no. 1, pp. 12–19, Jan. 1994.

[15] A. Gionis, P. Indyk, and R. Motwani, "Similarity search in high dimensions via hashing," in *Proc. VLDB*, 1999, vol. 99, no. 6, pp. 518–529.

[16] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, "Locality-sensitive hashing scheme based on *p*-stable distributions," in *Proc. 20th Annu. Symp. Comput. Geometry (SCG)*, 2004, pp. 253–262

[17] M. Raginsky and S. Lazebnik, "Locality-sensitive binary codes from shift-invariant kernels," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 1509–1517.

[18] M. Bawa, T. Condie, and P. Ganesan, "LSH forest: Self-tuning indexes for similarity search," in *Proc. 14th Int. Conf. World Wide Web (WWW)*, 2005, pp. 651–660.

[19] Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 1753–1760.

[20] J. Wang, S. Kumar, and S.-F. Chang, "Semi-supervised hashing for scalable image retrieval," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 3424–3431.

[21] J. Wang, S. Kumar, and S.-F. Chang, "Sequential projection learning for hashing with compact codes," in *Proc. 27th Int. Conf. Int. Conf. Mach. Learn.*, 2010, pp. 1127–1134.

[22] R.-S. Lin, D. A. Ross, and J. Yagnik, "SPEC hashing: Similarity preserving algorithm for entropy-based coding," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 848–854.

[23] S. Wang, Z. Huang, and X.-S. Xu, "A multi-label least-squares hashing for scalable image search," in *Proc. SIAM Int. Conf. Data Mining*, Jun. 2015, pp. 954–962

[24] Y. Gong, S. Lazebnik, A. Gordo, and F. Perronnin, "Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 12, pp. 2916–2929, Dec. 2013.

[25] W. Liu, J. Wang, S. Kumar, and S.-F. Chang, "Hashing with graphs," in *Proc. 28th Int. Conf. Int. Conf. Mach. Learn.*, 2011, pp. 1–8.

[26] W. Kong and W.-J. Li, "Isotropic hashing," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1646–1654.

[27] M. Norouzi and D. M. Blei, "Minimal loss hashing for compact binary codes," in *Proc. 28th Int. Conf. Mach. Learn. (ICML)*, 2011, pp. 353–360.

[28] C. Strecha, A. M. Bronstein, M. M. Bronstein, and P. Fua, "LDAHash: Improved matching with smaller descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 1, pp. 66–78, Jan. 2012.

[29] F. Shen, C. Shen, W. Liu, and H. T. Shen, "Supervised discrete hashing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 37–45.

[30] B. Kulis and T. Darrell, "Learning to hash with binary reconstructive embeddings," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 1042–1050.

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

[32] M. Kan, D. Xu, S. Shan, and X. Chen, "Semisupervised hashing via kernel hyperplane learning for scalable image search," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 24, no. 4, pp. 704–713, Apr. 2014.
[33] Y. Pan, T. Yao, H. Li, C.-W. Ngo, and T. Mei, "Semi-supervised Hashing with Semantic Confidence for Large Scale Visual Search," in *Proc. 38th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, 2015, pp. 53–62.
[34] S. Kumar and R. Udupa, "Learning hash functions for cross-view similarity search," in *Proc. 32nd Int. Joint Conf. Artif. Intell.*, Jun. 2011, pp. 1360–1365.
[35] J. Song, Y. Yang, Y. Yang, Z. Huang, and H. T. Shen, "Inter-media hashing for large-scale retrieval from heterogeneous data sources," in *Proc. Int. Conf. Manage. Data SIGMOD*, 2013, pp. 785–796.
[36] X. Zhu, Z. Huang, H. T. Shen, and X. Zhao, "Linear cross-modal hashing for efficient multimedia search," in *Proc. 21st ACM Int. Conf. Multimedia*, 2013, pp. 143–152.
[37] M. M. Bronstein, A. M. Bronstein, F. Michel, and N. Paragios, "Data fusion through cross-modality metric learning using similarity-sensitive hashing," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 3594–3601.
[38] Y. Zhen and D.-Y. Yeung, "Co-regularized hashing for multimodal data," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1376–1384.
[39] D. Zhang and W.-J. Li, "Large-scale supervised multimodal hashing with semantic correlation maximization," in *Proc. 28th AAAI Conf. Artif. Intell.*, Jun. 2014, pp. 2179–2183.
[40] D. Wang, X. Gao, X. Wang, and L. He, "Semantic topic multimodal hashing for cross-media retrieval," in *Proc. 24th Int. Joint Conf. Artif. Intell.*, Jun. 2015, pp. 3890–3895.
[41] Z. Lin, G. Ding, M. Hu, and J. Wang, "Semantics-preserving hashing for cross-view retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3864–3872.
[42] 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.
[43] J. Zhou, G. Ding, and Y. Guo, "Latent semantic sparse hashing for cross-modal similarity search," in *Proc. 37th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr. (SIGIR)*, 2014, pp. 415–424.
[44] J. Cheng, C. Leng, P. Li, M. Wang, and H. Lu, "Semi-supervised multi-graph hashing for scalable similarity search," *Comput. Vis. Image Understand.*, vol. 124, pp. 12–21, Jul. 2014.
[45] J. Wang, G. Li, P. Pan, and X. Zhao, "Semi-supervised semantic factorization hashing for fast cross-modal retrieval," *Multimedia Tools Appl.*, vol. 76, no. 19, pp. 20197–20215, Oct. 2017.
[46] Q.-Y. Jiang and W.-J. Li, "Deep cross-modal hashing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3232–3240.
[47] E. Yang, C. Deng, W. Liu, X. Liu, D. Tao, and X. Gao, "Pairwise relationship guided deep hashing for cross-modal retrieval," in *Proc. 31st AAAI Conf. Artif. Intell.*, Feb. 2017, pp. 1618–1625.
[48] 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.
[49] X. Zhang, H. Lai, and J. Feng, "Attention-aware deep adversarial hashing for cross-modal retrieval," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 591–606.
[50] 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.
[51] Z. Ji, W. Yao, W. Wei, H. Song, and H. Pi, "Deep multi-level semantic hashing for cross-modal retrieval," *IEEE Access*, vol. 7, pp. 23667–23674, 2019.
[52] T. Yao, Z. Zhang, L. Yan, J. Yue, and Q. Tian, "Discrete robust supervised hashing for cross-modal retrieval," *IEEE Access*, vol. 7, pp. 39806–39814, 2019.
[53] W. Gu, X. Gu, J. Gu, B. Li, Z. Xiong, and W. Wang, "Adversary guided asymmetric hashing for cross-modal retrieval," in *Proc. Int. Conf. Multimedia Retr. (ICMR)*, 2019, pp. 159–167.
[54] W. Cao, Q. Lin, Z. He, and Z. He, "Hybrid representation learning for cross-modal retrieval," *Neurocomputing*, vol. 345, pp. 45–57, Jun. 2019.
[55] K. Ding, B. Fan, C. Huo, S. Xiang, and C. Pan, "Cross-modal hashing via rank-order preserving," *IEEE Trans. Multimedia*, vol. 19, no. 3, pp. 571–585, Mar. 2017.
[56] D. Hu, F. Nie, and X. Li, "Deep binary reconstruction for cross-modal hashing," *IEEE Trans. Multimedia*, vol. 21, no. 4, pp. 973–985, Apr. 2019.
[57] S. Fu, R. Gutierrez-Osuna, A. Esposito, P. Kakumanu, and O. Garcia, "Audio/visual mapping with cross-modal hidden Markov models," *IEEE Trans. Multimedia*, vol. 7, no. 2, pp. 243–252, Apr. 2005.
[58] X. Chen, A. O. Hero, III, and S. Savarese, "Multimodal video indexing and retrieval using directed information," *IEEE Trans. Multimedia*, vol. 14, no. 1, pp. 3–16, Feb. 2012.
[59] A. Torralba, R. Fergus, and Y. Weiss, "Small codes and large image databases for recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.
[60] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics Intell. Lab. Syst.*, vol. 2, nos. 1–3, pp. 37–52, 1987.
[61] C. Ding and X. He, "K-means clustering via principal component analysis," in *Proc. 31st Int. Conf. Mach. Learn. (ICML)*, 2004, p. 29.
[62] R. Kothari and V. Jain, "Learning from labeled and unlabeled data," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, vol. 3, May 2002, pp. 2803–2808.
[63] Y. Grandvalet, J. Mariéthoz, and S. Bengio, "A probabilistic interpretation of svms with an application to unbalanced classification," in *Proc. Adv. Neural Inf. Process. Syst.*, 2006, pp. 467–474.
[64] C.-N. J. Yu and T. Joachims, "Learning structural SVMS with latent variables," in *Proc. ICML*, vol. 2, 2009, p. 5.
[65] A. P. Singh and G. J. Gordon, "Relational learning via collective matrix factorization," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2008, pp. 650–658.
[66] H. Lee, A. Battle, R. Raina, and A. Y. Ng, "Efficient sparse coding algorithms," in *Proc. Adv. Neural Inf. Process. Syst.*, 2007, pp. 801–808.
[67] W. Liu, J. He, and S.-F. Chang, "Large graph construction for scalable semi-supervised learning," in *Proc. 27th Int. Conf. Mach. Learn. (ICML)*, 2010, pp. 679–686.
[68] G. Shakhnarovich, "Learning task-specific similarity," Ph.D. dissertation, Dept. Elect. Eng. Comput. Sci., Massachusetts Inst. Technol., Cambridge, MA, USA, 2005.
[69] J. B. Copas and S. Haberman, "Non-parametric graduation using kernel methods," *J. Inst. Actuar.*, vol. 110, no. 01, pp. 135–156, Jun. 1983.
[70] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," May 2014, *arXiv:1405.3531*. [Online]. Available: https://arxiv.org/abs/1405.3531
[71] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
[72] J. Costa Pereira, E. Coviello, G. Doyle, N. Rasiwasia, G. R. G. Lanckriet, R. Levy, and N. Vasconcelos, "On the role of correlation and abstraction in cross-modal multimedia retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 3, pp. 521–535, Mar. 2014.
[73] M. J. Huiskes and M. S. Lew, "The MIR flickr retrieval evaluation," in *Proc. 1st ACM Int. Conf. Multimedia Inf. Retr. (MIR)*, 2008, pp. 39–43.
[74] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y. Zheng, "NUS-WIDE: A real-world web image database from National University of Singapore," in *Proc. ACM Int. Conf. Image Video Retr. (CIVR)*, 2009, p. 48.
[75] H. J. Escalante, C. A. Hernández, J. A. Gonzalez, A. López-López, M. Montes, E. F. Morales, L. Enrique Sucar, L. Villaseñor, and M. Grubinger, "The segmented and annotated IAPR TC-12 benchmark," *Comput. Vis. Image Understand.*, vol. 114, no. 4, pp. 419–428, Apr. 2010.
[76] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Zürich, Switzerland: Springer, 2014, pp. 740–755.
[77] A. Torralba, P. Murphy, W. T. Freeman, and M. A. Rubin, "Context-based vision system for place and object recognition," in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, vol. 2, 2003, p. 273.
[78] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.
[79] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," Sep. 2014, *arXiv:1409.1556*. [Online]. Available: https://arxiv.org/abs/1409.1556
[80] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," Jan. 2013, *arXiv:1301.3781*. [Online]. Available: https://arxiv.org/abs/1301.3781
[81] Q. Le and T. Mikolov, "Distributed representations of sentences and documents," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 1188–1196.

WENMING CAO (Member, IEEE) received the M.S. degree from the System Science Institute, Chinese Academy of Sciences, Beijing, China, in 1991, and the Ph.D. degree from the School of Automation, Southeast University, Nanjing, China, in 2003. From 2005 to 2007, he was a Postdoctoral Researcher with the Institute of Semiconductors, Chinese Academy of Sciences. He is currently a Professor with Shenzhen University, Shenzhen, China. His research interests include pattern recognition, image processing, and visual tracking.

WENSHUO FENG received the B.Eng. degree in electronic information engineering from Shenzhen University, Shenzhen, China, in 2018, where he is currently pursuing the master's degree in integrated circuit engineering. His current research interests include deep learning and multimodal retrieval.

QIUBIN LIN received the B.Eng. degree in electronic information engineering from Shenzhen University, Shenzhen, China, in 2017, where he is currently pursuing the master's degree in information and communication engineering. His current research interests include deep learning and multimodal retrieval.

GUITAO CAO received the M.S. degree from Shandong University, Jinan, China, in 2001, and the Ph.D. degree from Shanghai Jiao Tong University, Shanghai, China, in 2006. She is currently an Associate Professor with the School of Computer Science and Software Engineering, East China Normal University, Shanghai. Her research interests include image processing and pattern recognition, media analysis, and understanding.

ZHIHAI HE (Fellow, IEEE) was a Research Engineer with the David Sarnoff Research Center. He is currently a Professor with the Electrical Engineering and Computer Science Department, University of Missouri. He was named Fellow of the Institute of Electrical and Electronics Engineers (IEEE), in 2015, for his contributions to video communication and visual sensing technologies.