IPDH: An Improved Pairwise-Based Deep Hashing Method for Large-Scale Image Retrieval

WEI YAO, FEIFEI LEE, (Member, IEEE), LU CHEN, CHAOWEI LIN, SHUAI YANG, HANQING CHEN, AND QIU CHEN, (Member, IEEE)

1School of Optical-Electrical and Computer Engineering, University of Shanghai for Science and Technology, Shanghai 200093, China
2Graduate School of Engineering, Kogakuin University, Tokyo 163-8677, Japan

Corresponding authors: Feifei Lee (feifeilee@ieee.org) and Qiu Chen (q.chen@ieee.org)

This work was supported in part by the Programme for Professor of Special Appointment (Eastern Scholar) at Shanghai Institutions of Higher Learning, and in part by the JSPS KAKENHI under Grant 15K00159.

ABSTRACT Hashing technique has been extensively utilized in approximate nearest neighbor (ANN) search for large-scale image retrieval by virtue of its storage simplicity and computational efficiency. Recently, many researches show that hashing methods based on deep neural networks (DNNs) can improve retrieval accuracy by simultaneously learning both deep feature representation and hashing functions in an end-to-end framework. Most deep supervised hashing methods aim to preserve the distance or similarity between data points using the similarity relationships constructed based on semantic labels of images, while ignoring the classification ability of the generated hash codes. However, the semantic labels themselves carry more information than the corresponding similarity labels. We propose an Improved Pairwise-based Deep Hashing (IPDH) method to generate hash codes with powerful classification ability by exploring the global distribution of semantic labels. Specifically, the proposed IPDH method aims to minimize the information loss generated during the process of classification prediction to ensure that the output predicted labels of the network model has a similar distribution with those from the original semantic labels. Comprehensive experiments show that the proposed IPDH method can obtain better improvement than other state-of-the-art algorithms.

INDEX TERMS Hashing method, large-scale image retrieval, deep neural network, classification ability, information loss.

I. INTRODUCTION

With the development of information technology, the massive and high-dimensional multimedia information resources on the web have greatly promoted the development of large-scale visual search [1]–[4]. Traditional Text-Based Image Retrieval (TBIR) [5] generally queries images in the form of keywords, and its development has been relatively mature. Due to the limitation of controlled vocabulary, the TBIR system cannot efficiently deal with ever-changing images. Therefore, Content-Based Image Retrieval (CBIR) [6], [7] has received extensive attention in people’s lives, which can further explore the semantic content of multimedia data resources. Compared with the traditional linear scanning method, Approximate Nearest Neighbor (ANN) search is usually used to ensure the real-time response speed. In other words, it is to find data points most similar to the target data by calculating the similarity between data points from the database.

It can be considered that the closer the original data are, the higher the calculated similarity between the data is. However, these similarity-based retrieval methods cannot deal with high-dimensional image data. Hashing technique [8] has more advantages by virtue of its computational efficiency and retrieval quality. It can transform high-dimensional media data into binary hash codes while preserving the original similarity (metric similarity in the original feature space or semantic similarity based on labels).

Existing hashing methods consist of two categories: data-independent and data-dependent methods. Data-independent hashing aims to learn the compact representation of high-dimensional image features by using a set of random projections, and the process of generating hash functions is independent of the distribution characteristics of the data.
Locality Sensitive Hashing [9] is a representative data-independent method, which constructs hash functions by a random hyperplane instead of deeply exploring the data structure. Researches show that the LSH method achieves a lower recall accuracy when the hash bits are shorter. Data-dependent hashing exploits the distribution of data points to learn hash functions, thereby obtaining hash codes with strong discriminating ability and high distance keeping property. Hence, data-dependent hashing methods can obtain great improvement and are widely used. They can be further subdivided into unsupervised and supervised methods according to whether to use image labels or not. Unsupervised hashing methods aim to learn hash functions from unlabeled data, and strive to maintain the original similarity between data points. However, supervised hashing methods integrate the supervision information into the hash learning process, which can improve hash learning ability better than unsupervised methods.

Whether for unsupervised or supervised hashing methods, choosing an appropriate method to learn image representation is a key issue. Traditional hand-crafted feature extraction methods such as SIFT-based [10], HOG-based [11] and GIST-based [12] methods usually lose the key semantic information and cannot express image characteristics well. Recently, hashing methods based on deep learning [13], [14] have proved that Deep Convolutional Neural Networks (DCNNs) [15] has advantages in learning both feature representations and hash coding effectively. These methods integrating DCNNs into the learning of hash functions are called deep hashing methods, which greatly promote the development of image retrieval.

Recent researches have proved that deep supervised hashing methods can learn feature representation and hash functions simultaneously by integrating feature extraction and hash coding into an end-to-end learning framework. Therefore, the end-to-end learning hashing algorithms based on DCNNs show the state-of-the-art results [16]–[19].

Most supervised deep hashing algorithms use appropriate metrics to model the similarity between the original samples. In pairwise hashing methods, absolute distance is often used, such as DHN [20], DSH [21], DPSH [22] and DSDH [23]. These methods can preserve the original distance similarity in Hamming space. In other words, pairwise hashing methods strive to reduce the distance between similar images while increasing the distance between dissimilar images in Hamming space. In the triplet hashing methods, the relative distance between dissimilar image pairs and similar image pairs is considered, such as NINH [13], DRSC [24], DTS [25]. These methods try to keep the Hamming distance between similar images shorter than those from dissimilar images and vice versa. Others try to learn discriminative binary hash codes to improve classification capability based on the semantic labels, such as DLBC [26] and DHCQ [27].

In pairwise hashing methods, pairs of images and corresponding similarity labels are utilized as the training inputs of the Deep Convolutional Neural Networks (DCNNs). The key issue is how to construct and optimize the relationship between input images and the generated hash codes to ensure that the generated hash codes can preserve the original similarity of images as well as minimize the quantization error in the optimization process. Some recent pairwise-based deep hashing methods, such as DHN [20], HashNet [28], DCH [29], focus on optimizing the relationship between the similarity labels of images and generated hash codes constructed in a Bayesian framework. The discrete optimization problem also can be optimized in the Bayesian framework. However, while considering to maintain the similarity between original image pairs, these hashing methods do not attach much importance to the classification capability of generated binary hash codes. Hence, we propose an Improved Pairwise-based Deep Hashing (IPDH) method based on DCNNs to generate discriminated binary codes for efficient Hamming space retrieval in the pairwise manner. Inspired by [30], we propose a novel classification loss using Jensen-Shannon (JS) divergence to constrain the predicted labels to have a similar distribution with the original image semantic labels. Due to its symmetry and finite value, the JS divergence can be used to optimize the distance similarity between the distributions of the original semantic labels and the predicted labels.

In general, the contributions of this article are as follows:

1. We propose a novel robust classification metric based on JS divergence to obtain optimal hash codes with high classification capability by optimizing the relationship between the semantic labels of images and the predicted labels learned by DCNNs.
2. We propose an Improved Pairwise-based Deep Hashing (IPDH) method, which can simultaneously learn and optimize both the classification quality and the distance-based similarity in the learning process.
3. Comprehensive experiments on three extensively used datasets indicate that the proposed IPDH method can obtain better improvement than the state-of-the-art image retrieval methods.

The rest of this article is organized as follows. In Section II, we review some related works about traditional and deep supervised hashing algorithms for image retrieval. In Section III, we describe the details of our proposed novel deep supervised hashing algorithm. In Section IV, we detail the experimental results and analysis. Finally, conclusions are given in Section V.

II. RELATED WORKS

Recently, hashing methods have been extensively utilized in image retrieval by virtue of its small storage space and fast computational speed. Here we will review some traditional and deep supervised hashing algorithms.

A. TRADITIONAL HASHING METHODS

Wang et al. [31] has given a comprehensive survey on hash learning, in which hashing algorithms can be subdivided into unsupervised methods and supervised methods.
Unsupervised hashing methods aim to learn the hash functions by exploring the distribution characteristics of unlabeled data. Kernelized LSH (KLSH) [32] is an extension of LSH, which can learn hash functions by using arbitrary kernel functions while maintaining the similarity of original samples. Spectral Hashing (SH) [33] can learn binary compact codes by transforming the hash learning process into a graph partition problem. By calculating the eigenvectors of the graph Laplacian, a relaxed solution can be found to solve the graph partition problem. Anchor Graph Hashing (AGH) [34] is capable to capture the intrinsic neighborhood structure automatically based on the massively large dataset. Then Discrete Graph Hashing (DGH) [35] extends AGH to learn similarity-preserving hash codes in the discrete Hamming space by introducing a tractable alternating optimization method. Iterative Quantization (ITQ) [36] first utilizes principal component analysis (PCA) [37] projections to reduce the dimensionality of the original dataset. Then a rotation matrix of zero-centered data is randomly initialized to minimize the quantization loss.

Supervised hashing methods aim to generate optimal compact binary codes by simultaneously using semantic information (e.g., semantic labels) and feature representation of images. Representative supervised methods contain Minimal Loss Hashing (MLH) [38], Supervised Hashing with Kernels (KSH) [39], and Binary Reconstruction Embedding (BRE) [40]. Minimal Loss Hashing (MLH) can preserve the original similarity in the binary space by optimizing pairwise hinge-like loss functions. Supervised Hashing with Kernels (KSH) introduces a kernel function to solve the inseparable problem of the original data. Meanwhile, KSH can also learn hash functions by using the inner product to calculate the Hamming distance between data pairs. Binary Reconstructive Embedding (BRE) aims to generate hash codes by introducing a coordinate-descent strategy to minimize the reconstruction error directly.

B. DEEP SUPERVISED HASHING METHODS

In recent years, hashing algorithms based on deep neural networks have played an increasingly significant role in image retrieval tasks. Deep hashing methods is capable to learn various feature representations of images. Compared with the traditional hand-crafted methods, higher retrieval performance can be achieved in deep neural networks architecture. According to the usage of image semantic labels, deep hashing methods can be divided into three categories: point-wise, triplet-wise and pairwise.

Lin et al. [26] first introduce a latent layer into the deep neural networks architecture to learn deep image representation and the hash functions in a point-wise manner, and then utilize the class labels to train the network model. Yang et al. [41] propose a SSDH method to learn hash functions by simultaneously optimizing the classification error as well as other constraints on the hash codes. Wang et al. [42] extends SSDH to learn efficient image representation in a point-wise manner and add the top layer information to calculate the distance for the similarity retrieval process. Lu et al. [43] propose an IDSH method which can quickly learn hash functions by introducing a Divide-and-Encode Module and a Batch Normalization (BN) layer. A center loss is also applied to improve the retrieval effectiveness. Jin et al. [44] propose a DOH method to learn ranking-based compact hash codes by simultaneously learning the global semantic and local spatial information. A spatial attention module is introduced to obtain the local spatial information as well as the local discriminability.

Network in Network Hashing (NINH) [13], a typical triplet-wise hashing algorithm, introduces a joint optimizing process to learn image feature representation and hash functions simultaneously. Then Zhao et al. describe DRSH [45] to utilize triplet ranking loss as multi-level similarity information to learn hashing functions. Lai et al. [14] propose a novel hashing method to learn instance-aware representations for multi-label images, in which the learned image representations can be organized in multiple groups of features, and each group corresponds to a category. The instance-aware representations can also guide the learning process for both category-aware hashing and semantic hashing. A representative cross-modal method is Triplet-based Deep Hashing (TDH) [46], which can preserve the original semantic similarity by combining graph regularization into the hash learning process.

In pairwise supervised hashing algorithms, pairwise semantic information is used to learn the image representation. Shen et al. introduce Supervised Discrete Hashing (SDH) [47], which directly generates nonlinear hash codes under the discrete constraint by minimizing (maximizing) the Hamming distance across similar (dissimilar) image pairs. Xie et al. introduce a two-step method called Convolutional Neural Network Hashing (CNNH) [48] to learn the feature representation and hashing functions. However, the learned image representation cannot guide the updating process of generated binary hash codes. Deep Pairwise-supervised Hashing (DPSH) [22] first constructs the relationship between pairwise labels and hash codes using a Bayesian framework, and then learn hash functions by optimizing the relationship. DHN [20] introduces a novel bimodal Laplacian prior (unnormalized) for the continuous representations during the discrete optimization process, which can limit the generated hashing codes to a fixed range with great probability. Thus the quantization error can be minimized to a large extent in the Bayesian framework. HashNet [28] introduces a weighting parameter to solve the data imbalance problem by weighting the training pairs according to the importance of misclassifying that pair. A non-smooth sigh function is also introduced to address the non-convex optimization problem. As the training proceeds, the initial smoothed objective function will become more and more non-smooth. Different probability distribution functions are also used to measure the posterior estimation between the semantic similarity and hash codes. Deep Cauchy Hashing (DCH) [29] learns hash functions by designing a novel pairwise cross-entropy loss.
based on Cauchy distribution in an end-to-end architecture. The pairwise cross-entropy can ensure the high probability of similar image pairs with small Hamming distance.

Compared with the pairwise based hashing methods, the point-wise based hashing learns hash codes by transforming the retrieval into a classification problem without considering the similarity order among neighbors. And the triplet-wise based hashing costs high computational complexity for large-scale datasets. Hence, we propose an end-to-end pairwise based hashing method, which attaches much importance to the classification ability of generated compact hash codes.

III. THE PROPOSED METHOD
We will further introduce our IPDH method in detail. We propose an end-to-end deep supervised architecture to simultaneously learn hash functions and feature representation of images. In view of many hashing methods, AlexNet [15] is used as the basic architecture to learn the deep image representation in this article. We first add a hashing layer to generate similarity-preserving binary codes, and then introduce a classification layer to obtain optimal prediction vectors for classification. The generated compact hash codes should preserve the original similarity in the Hamming Space and guarantee a high probability that similar images will be classified into the same class after being mapped by hash functions simultaneously.

A. THE PROPOSED OVERALL FRAMEWORK
Figure 1 shows the framework of the proposed IPDH method. Most generalized deep supervised hashing algorithms mainly use AlexNet [15] as the basic framework. For comparison, AlexNet is also utilized to learn deep feature representation in the proposed method. Table 1 shows the configurations of the CNN. The original eighth classification layer in AlexNet is replaced with a hashing layer to generate compact binary codes as image representation. Follow the [45], feature fusion is carried out for the first two fully-connected layers (FC6, FC7). Then the obtained fused features are sent to the hashing layer to obtain diverse feature representation. A classification layer (FC9) is introduced to the network to get optimal hash codes with classification ability. We use the original image pixels and similarity labels in the form of \((x_i, x_j, s_{ij})\) as the inputs of the deep architecture.

The network parameters are initialized based on the parameters of pre-training model on the ImageNet [49] dataset. The network parameters can be fine-tuned by the stochastic gradient descent (SGD) method. The hash learning process can be described as such a pipeline: (1) a feature extraction subnetwork to extract image feature representation; (2) a hash layer to project high-dimensional image features into compact hash codes; (3) a pairwise similarity loss to preserve the original similarity in the Hamming space; (4) a pairwise quantization loss to control the quality of discrete hashing; (5) and a classification loss to learn the hash codes with optimal classification ability. Our algorithm can also be applied to other deep network architectures including GoogleNet [50] and ResNet [51].

B. DEEP HASHING FUNCTIONS
Suppose the training set is defined as \(X = \{x_i\}_{i=1}^N \in \mathbb{R}^{d \times N}\), where \(x_i\) denotes the d-dimensional feature vector for each

---

TABLE 1. Configurations of the CNN.

| Type          | Filter size/stride |
|---------------|-------------------|
| Convolution 1 | 11*11/4           |
| Max Pool 1    | 3*3/2             |
| Convolution 2 | 5*5/2             |
| Max Pool 2    | 3*3/2             |
| Convolution 3 | 3*3/1             |
| Convolution 4 | 3*3/1             |
| Convolution 5 | 3*3/1             |
| Max Pool 3    | 3*3/2             |
data point. In addition, the similarity matrix \( s_{ij} \) is defined to indicate the pairwise similarity between image pairs of points \( x_i \) and \( x_j \), where \( s_{ij} = 1 \) if \( x_i \) and \( x_j \) are similar, otherwise \( s_{ij} = 0 \).

Hash coding aims to generate a set of binary codes \( B \in \{-1, 1\}^{K \times N} \), where the \( i \)th column \( b_i \in \{-1, 1\}^K \) corresponds to the hash codes projected by the \( i \)th data point \( x_i \). The hash function is defined in the form of \([h_1(\cdot), \ldots, h_K(\cdot)]\) to generate hash codes. In general, the goal of hashing method is to learn hash functions to map original image samples to a set of hash codes.

### C. PAIRWISE SIMILARITY LOSS

We try to perform deep hashing in a Bayesian learning framework by preserving similarity of pairwise images. The generated hash codes must satisfy the constraints limited by the similarity labels in Hamming space.

The inner product is utilized to calculate the pairwise similarity instead of linear Hamming distance. The inner product of the similarity labels in Hamming space.

\[
\langle b_i, b_j \rangle = b_i^T b_j
\]

is designed to maximize the alignment between each pair of examples. The pairwise similarity loss is defined as follows

\[
J_1 = -\log p(S|B) = -\sum_{s_{ij} \in S} (s_{ij} \omega_{ij} - \log (1 + e^{\omega_{ij}}))
\]

The above formula guarantees the Hamming distance of two dissimilar (similar) points to be as large (small) as possible, which is exactly the goal of hashing methods based on pairwise similarity.

The pairwise similarity loss term has been widely utilized in pairwise-based image retrieval tasks. In addition, the idea of optimizing the pairwise similarity loss during the training process has also been applied to other tasks. Shu et al. [52] propose to hierarchically learn to transfer the semantic knowledge from web texts to images. An empirical loss on co-occurrence pairs is designed to maximize the alignment between each pair of examples. Tang et al. propose a generalized deep transfer networks (DTNs) [52] for knowledge propagation in heterogeneous domains. The empirical loss on co-occurrence pairs is also introduced to maximize the alignment between each pair of examples.

### D. PAIRWISE QUANTIZATION LOSS

In practical application, discrete hash codes are used for similarity calculation. However, it is difficult to optimize discrete hash coding in the CNNs. So the continuous form of hash coding is utilized to avoid gradient disappearance in the back-propagation process. We define the output of hash layer as \( u_i \) and make \( b_i = \text{sgn}(u_i) \), which can be represented as:

\[
u_i = V^T \theta(x_i, \theta) + F
\]

where \( \theta(x_i, \theta) \) represents the output of the layer, \( \theta \) denotes all the parameters of the FC7 layer, \( V \) and \( F \) denote the weight matrix and bias term respectively. Hence, the pairwise quantization loss is introduced to narrow the gap between discrete and continuous hash codes. The optimization problem is defined as follows:

\[
J_2 = \sum_{i=1}^{n} ||b_i - u_i||^2 + \lambda \|V\|^2
\]

where \( \|\cdot\| \) is the L2-norm of vectors as well as the Frobenius norm of matrices, \( \lambda \) is the regularization parameter.

### E. CLASSIFICATION LOSS

Most supervised hashing methods utilize the pairwise similarity to keep the original distance similarity between images instead of fully exploiting the available label information. As mentioned in [26], it is a reasonable assumption that the generated compact hash codes should also be optimal for classification due to the semantic labels conveying more information than similarity alone.

In the linear classification task, we can define the multiclass classification problem as follows:

\[
y = W^T b = \left[w_1^T b, \ldots, w_c^T b\right]^T
\]

where \( y \in \mathbb{R}^c \) is the label vector and \( w_k \in \mathbb{R}^K \), \( k = 1, \ldots, c \) is the classification vector for class \( k \). Most multi-class classification problems utilize Mean Squared Error [54] to measure classification error.

Zhang et al. propose a novel hashing method [30] to maintain the global distribution of data during hashing embedding by using the JS divergence to measure the difference between the distributions of the original samples and the generated hash codes. Inspired by [30], we propose an assumption that the learning of hash codes with optimal classification ability needs fully exploration of the semantic labels. Different from the Mean Squared Error used in the traditional classification problem, we try to constrain the distribution variation between the image semantic labels and the output predicted classification vectors.

Information divergence plays an important role in measuring difference between probability distributions, of which KL divergence is commonly used. The definition of the KL divergence between two distributions P and Q is:

\[
\text{KL}(P||Q) = \sum P(x) \log \frac{P(x)}{Q(x)}
\]
The above formula is non-negative. If \( P = Q \), the formula is equal to zero. It can be considered that KL divergence can quantify the information loss between two distributions.

The JS divergence would be a better measure for the difference between distributions due to its characteristics. It can be denoted as:

\[
\text{JS}(P||Q) = \frac{1}{2} \text{KL}(P(x)||\frac{P(x)+Q(x)}{2}) + \frac{1}{2} \text{KL}(Q(x)||\frac{P(x)+Q(x)}{2})
\]

(7)

Obviously, the JS divergence calculates the average distribution and average distance between probability distributions. In other words, the JS divergence measures the variation of two distributions under the same prior condition.

The JS divergence has two advantages compared with the KL divergence. The first one is symmetry, which means that the JS divergence can quantify the information loss from Q to P as well as from P to Q. The KL divergence is non-negative and asymmetric. Compared with the KL divergence, the JS divergence has a definite upper bound and its value range is \([0, 1]\). The definite upper bound makes the measure of similarity between distributions more accurate and helps the optimization process to converge. We also compare the training loss based on the KL divergence and JS divergence, the comparison curves on CIFAR-10 are shown as in Figure 2 where the hash code length is 64-bit.

**FIGURE 2.** The comparison curves of training loss on CIFAR-10 dataset.

We can learn from Figure 2 that the training loss based on the JS divergence decreases faster than that of the KL divergence. In other words, the JS divergence can help the optimization process to converge.

Hence, we use the JS divergence to measure the distribution variation between image semantic labels and output predictive classification vectors in this article. The symmetry not only guarantees that images with similar semantic labels should obtain similar predicted labels, but also that similar predicted labels should correspond to images with similar semantic labels. The upper bound of the JS divergence can help the optimization process to converge.

For the original image semantic labels with M classes \( L = [l_i]_{i=1}^N \) where the \( i \)th column \( l_i \) corresponds to the semantic label of the \( i \)th sample \( x_i \), the similarity of \( l_i \) to \( l_j \) can be defined as the conditional probability \( p_{ij} \). We use the Euclidean distance to calculate similarities between labels and the Gaussian distribution is utilized for fitting the distribution. In detail, if the nearest neighbors are selected according to the proportion of the probability density function, \( l_i \) would select \( l_j \) as its neighbor under the Gaussian distribution, centered at \( l_i \). The conditional probability \( p_{ij} \) is relatively high for the similar labels, on the other hand, the \( p_{ij} \) will be very close to zero for the dissimilar labels. Hence, we define the conditional probability as:

\[
p_{ij} = \frac{\exp \left( -\|l_i - l_j\|^2/2\sigma_i^2 \right)}{\sum_{k\neq i} \exp \left( -\|l_i - l_k\|^2/2\sigma_i^2 \right)}
\]

(8)

Similarly, the conditional probability \( p_{ij} \) is represented as:

\[
p_{ij} = \frac{\exp \left( -\|l_j - l_i\|^2/2\sigma_j^2 \right)}{\sum_{k\neq j} \exp \left( -\|l_j - l_k\|^2/2\sigma_j^2 \right)}
\]

(9)

It can be concluded that the joint probability distribution of the above two conditional probabilities is defined as \( P_{ij} = \frac{p_{ij} + p_{ji}}{2n} \).

The set of predicted classification labels with M classes in this article is defined as: \( L' = \{l'_i\}_{i=1}^N \), where the \( i \)th column \( l'_i \) corresponds to the predicted label of the \( i \)th sample \( x_i \). We choose the Cauchy distribution to fit the probability distribution of the predicted classification labels. The Cauchy distribution has more advantages in converting distance to probability, which can alleviate the crowding problem. Therefore, the probability distribution \( Q_{ij} \) is defined as follows:

\[
Q_{ij} = \left(\frac{1 + \|l'_i - l'_j\|^2}{\sum_{k\neq m} \left(1 + \|l'_k - l'_m\|^2\right)^{-1}}\right)^{-1}
\]

(10)

Since the difference between \( P_{ij} \) and \( Q_{ij} \) is measured by the JS divergence, the optimization problem can be defined as follows:

\[
J_3 = \sum JS(P_{ij}|Q_{ij}) = \sum_i \sum_j \left(\frac{1}{2}P_{ij}\log\frac{P_{ij}}{v_{ij}} + \frac{1}{2}Q_{ij}\log\frac{Q_{ij}}{v_{ij}}\right)
\]

(11)

\( v_{ij} = \frac{P_{ij} + Q_{ij}}{2} \).

In conclusion, the overall objective function is represented as:

\[
J = J_1 + \beta J_2 + \gamma J_3
\]

(12)

where \( \beta, \gamma \) are hyper parameters that control importance of the \( J_2 \) and \( J_3 \) terms.
F. OVERALL OPTIMIZATION PROBLEM

In this section, we will summarize the overall optimization problem of our proposed algorithm. The overall optimization problem consists of three key parts: pairwise similarity loss, pairwise quantization loss and classification loss. The overall optimization problem is redefined as follows:

\[
\min J' = - \sum_{sij \in S} (s_{ij} \omega_{ij} - \log (1 + e^{\omega_{ij}})) + \beta \sum_{i=1}^{n} \|b_{i} - u_{i}\|_{2}^{2} + \lambda \|V\|^{2} + \gamma \sum_{i} \sum_{j} \left( \frac{1}{2} P_{ij} \log \frac{P_{ij}}{V_{ij}} + \frac{1}{2} Q_{ij} \log \frac{Q_{ij}}{V_{ij}} \right)
\]

where \(\omega_{ij} = \frac{1}{2} u_{i}^{T} u_{j}, b_{i} = \text{sgn}(u_{i}) = \text{sgn}(V^{T} \theta(x_{i}, \theta) + F)\).

The first part is pairwise similarity loss. Given the training set with pairwise similarity labels as \(\{(x_{i}, x_{j}, s_{ij}) : s_{ij} \in S\}\), pairwise similarity loss aims to ensure that the generated hash codes can preserve the similarity between original images in the Hamming space by optimizing the relationship constructed between similarity labels and generated hash codes in a Bayesian framework.

The second part is pairwise quantization loss. The hyper parameters \(\beta\) controls the importance of this term, which is set to 0.01. As mentioned in section (III) (D), it is difficult to directly optimize discrete hash coding in the learning process. We first introduce an auxiliary variable \(u_{i}\) to avoid gradient disappearance in the back-propagation process, where \(u_{i}\) denotes the continuous form of output of the hash layer. Then we make \(b_{i} = \text{sgn}(u_{i})\) to obtain the discrete hash codes, where \(\text{sgn}(\cdot)\) denotes a sign function. Pairwise quantization loss aims to optimize the hash learning process by reducing the quantization error between continuous hash codes and discrete hash codes.

The third part is classification loss, which can ensure that the generated compact hash codes have strong classification ability by fully exploiting the semantic labels information. The hyper parameters \(\gamma\) controls the importance of this term, which is set to 1. Different from the Mean Squared Error (MSE) used in traditional classification problems, we try to constrain the distribution variation between the image semantic labels and the output predicted classification vectors based on the JS divergence.

These parameters can be learned by an alternating strategy in this article, which means that when optimizing one parameter, the other parameters are fixed. We can first optimize \(b_{i}\) by reducing the difference between continuous and discrete hash codes and optimize the rest parameters using the SGD as well as back-propagation algorithm.

IV. EXPERIMENTS

We verify our method on three massive public datasets: CIFAR-10, NUS-WIDE and MS-COCO. We first briefly introduce these datasets, and then present the experimental settings. Experimental results containing the evaluations and comparisons with other state-of-the-art hashing algorithms are given in the section (C). In the last section, we show the discussions.

A. DATASETS

CIFAR-10 dataset [55] contains 60,000 color images corresponding to 10 classes, each class consists of 6000 images. Each image corresponds to only one class. According to [22], [29], we randomly sample 500 images per class as the training set, 100 images per class as the query dataset, and the remaining images as the database.

NUS-WIDE [56] is a multi-label dataset containing nearly 270,000 color images. Each image is associated with one or more of the 81 semantic labels. We randomly select 5,000 images as query dataset, while the remaining images are utilized as the database, and randomly sample 10,000 images from the database as the training set.

MS-COCO is a popular dataset [57] for image recognition, captioning and segmentation. MS-COCO consists of 40504 validation images and 82783 training images, each of which is associated with one or more of 80 semantic labels. We randomly select 5,000 images as query points, while the remaining images are utilized as the database, and randomly select 10,000 images from the database for training.

According to [22], [29], [58], [59], the similar pairs are constructed using image semantic concepts: when the similarity \(s_{ij} = 1\), the two images share at least one label and are considered to be similar. Otherwise the similarity \(s_{ij} = 0\), the two images do not share any labels and are considered to be dissimilar.

B. EXPERIMENTAL SETTINGS

We perform experiments on the proposed hashing algorithm using the open source Pytorch [60] framework by using a single NVIDIA Tesla K40c GPU. The weights of convolutional layers and the first two fully-connected layers (FC6, FC7) are initialized by the model parameters pre-trained on the ImageNet [49] dataset, and the remaining layers are initialized randomly. Then the SGD and back-propagation can be utilized to update the network parameters.

As the inputs of the network framework, the images from different datasets are resized to 256 × 256 first and then cropped into the size of 224 × 224. The initial learning rate is set to 0.05. A “step” strategy is adopted for the learning rate, which is reduced to one-tenth of the original learning rate every 75 epochs, for a total of 300 epochs.

In the testing process, we follow the same standard evaluation approach as previous work [29] for Hamming space retrieval. The test process can be divided into two steps: 1) calculate the similarities of data points and return data points in the retrieval space within Hamming radius 2; 2) re-rank the returned data points in an ascending order according to the similarities calculated between the query image and data points. Similar to most previous work [20], [28], [29], we use mean Average Precision (mAP) to evaluate
our proposed IPDH method together with several benchmarks. In addition, the precision curves and mAP curves under different lengths of hash bits are also utilized to evaluate the performance of our proposed IPDH method as well as state-of-the-art hashing methods.

C. EXPERIMENTAL RESULTS

1) EXPERIMENTAL RESULTS ON CIFAR-10 DATASET

We compare our method with several hashing algorithms, which can be categorized into two parts: traditional and deep supervised hashing methods. Traditional supervised hashing algorithms include SDH [47], KSH [40], [39] and ITQ-CCA [61], while deep hashing methods contain CNNH [48], DNNH [14], DHN [20], HashNet [28] and DCH [29].

**TABLE 2.** mAP on CIFAR-10 dataset for different conditions (%).

| Hash codes | DCH | IPDH |
|------------|-----|------|
| 16-bit     | 0.7901 | 0.8085 |
| 32-bit     | 0.7979 | 0.8122 |
| 48-bit     | 0.8071 | 0.8694 |
| 64-bit     | 0.7936 | 0.8776 |

**TABLE 3.** mAP on NUS-WIDE dataset for different conditions (%).

| Hash codes | KL divergence | JS divergence |
|------------|---------------|---------------|
| 48-bit     | 0.7913        | 0.8694        |
| 64-bit     | 0.7990        | 0.8776        |

Table 2(a) shows the mean Average Precision (mAP) on CIFAR-10 dataset of proposed IPDH method and the primary compared method DCH [29], which introduces the Cauchy distribution to measure the posterior estimation between the semantic similarity and hash codes without further exploring the classification ability of the generated hash codes. Figure 3 describes the precision and mAP curves of our IPDH method and other algorithms on CIFAR-10 dataset in case of different length hash codes.

We can see from the table and figure that although other deep hashing methods have obtained great performance than these traditional hashing methods, our method obtains much better improvement. Table 2(a) shows that the mAP of our IPDH method is increased by 1.43 – 8.4% compared with the related deep hashing methods.

In addition, the retrieval performance under different constraints is also compared here. As mentioned in section (III) (E), both the KL divergence and JS divergence can be utilized to measure information loss. However, compared with the KL divergence, the JS divergence has symmetry and a certain upper bound, which can make the similarity measure between distributions more accurate. We also perform experiments to compare the results using these two divergences based on our method. As shown in Table 2(b), the JS divergence is better than the KL divergence in measuring the distribution variation between the image semantic labels and the output predictive classification vectors for our method.

2) EXPERIMENTAL RESULTS ON NUS-WIDE DATASET

NUS-WIDE is a large-scale dataset with complicated images compared with the CIFAR-10 dataset. We perform experiments on NUS-WIDE dataset to evaluate our method and several hashing algorithms.

Table 3(a) indicates the variation between proposed IPDH method and DCH [29] method, which introduces the Cauchy
distribution to measure the posterior estimation between the semantic similarity and hash codes. Figure 4 describes the precision and mAP curves of the IPDH method and other algorithms on NUS-WIDE dataset in case of different length hash codes.

Both the table and the figure can indicate the advantage of our method. In addition to the case where the hash code length is 32-bit, Table 3(a) shows that the mAP based on our method increases by 0.4 - 7.72% compared with the related deep hashing methods. The retrieval performance under different constraints is also compared here. As mentioned in section (III) (E), both the KL divergence and the JS divergence can be used to measure information loss. Table 3(b) shows that the JS divergence is better than the KL divergence in measure the distribution variation between the image semantic labels and the output predictive classification vectors for our method due to its symmetry and a definite upper bound.

3) EXPERIMENTAL RESULTS ON MS-COCO DATASET

MS-COCO is also a massive multi-label dataset, which is suitable for image classification, image segmentation, objective detection and other tasks. We also compare the performance of our method with other hashing algorithms.

Table 4(a) presents the mean Average Precision (mAP) on MS-COCO dataset of proposed IPDH method and the primary compared method DCH [29], which introduces the Cauchy distribution to measure the posterior estimation between the semantic similarity and hash codes without further exploring the classification ability of generated hash codes. From the table, we can learn that the mAP of our method increases by 4.45 - 8.36% compared with other deep hashing algorithms.

Figure 5 indicates the precision and mAP curves of the IPDH method and other hashing algorithms on MS-COCO dataset based on different length hash codes. We can draw a conclusion that our method can achieve great improvement compared with other hashing algorithms. Table 4(b) indicates the difference between the KL divergence and the JS divergence in measuring the distribution variation between the image semantic labels and the output predictive classification vectors for our method. Compared with the method by using the KL divergence, the mAP by using the JS divergence is increased by 0.01% in the case of 48-bit hash codes, while the mAP by using the KL divergence is increased by 0.32% compared with the method using the JS divergence based on the 64-bit hash codes.

D. DISCUSSIONS

Compared with the traditional hashing methods, the end-to-end supervised hashing methods based on Deep Convolutional Neural Networks (DCNNs) have achieved great improvement in image retrieval tasks. Recently, some pairwise-based deep supervised hashing methods try to preserve the original similarity and optimize the quantization loss simultaneously in a Bayesian framework during the hashing embedding process, such as DHN [20], HashNet [28], DCH [29], etc. On the one hand, these methods focus on optimizing the relationship between the similarity labels of images and generated hash codes constructed in a Bayesian framework. On the other hand, these methods also try to reduce the quantization error between continuous hash codes and discrete hash codes during the discrete optimization process.

| Hash codes | mAP based on different length hash codes (%) |
|------------|---------------------------------------------|
| 16-bit     | 0.7010                                      | 0.7455                      |
| 32-bit     | 0.7576                                      | 0.7625                      |
| 48-bit     | 0.7251                                      | 0.7768                      |
| 64-bit     | 0.7013                                      | 0.7849                      |

| Hash codes | mAP based on different divergences (%) |
|------------|----------------------------------------|
| 48-bit     | 0.7800                                 | 0.7768                      |
| 64-bit     | 0.7848                                 | 0.7849                      |
Although these methods can achieve higher performance than previous hashing methods, they ignore the classification ability of generated hash codes during the learning process. In order to improve the classification ability of the generated hash codes, we propose a novel robust classification metric based on the JS divergence, which can obtain optimal hash codes with high classification capability by optimizing the relationship between the semantic labels of images and the predicted labels learned by DCNNs. The proposed IPDH method can not only learn the similarity-preserving hash codes but also improve the classification ability of the generated binary codes in a pairwise manner.

In order to verify the performance of our propose network model, we conduct experiments on three widely used datasets: CIFAR-10, NUS-WIDE and MS-COCO. We compare the performance of our method with several hashing algorithms including traditional hashing methods and deep supervised hashing methods. The experimental results are given in the form of tables and figures. We also compare the difference between the KL divergence and the JS divergence in measuring the distribution variation between the image semantic labels and the output predictive classification vectors for our method. In general, we perform several experiments mentioned above to evaluate the performance of our proposed method for large-scale image retrieval.

**V. CONCLUSION**

In this article, we have proposed an Improved Pairwise-based Deep Hashing (IPDH) method for large-scale image retrieval. The deep framework can learn the similarity-preserving hash codes with optimal classification ability in a pairwise manner. This means that both the pairwise similarity supervision and semantic label information can be learned based on the framework. In order to improve the classification ability of the learned binary codes, we try to limit the distribution variation between the image semantic labels and the output predicted classification vectors by using the JS divergence. Comprehensive experiments indicate that proposed IPDH method can obtain better improvement compared with the state-of-the-art methods.

**ACKNOWLEDGMENT**

(Wei Yao and Feifei Lee contributed equally to this work.)

**REFERENCES**

[1] A. Khodaskar and S. Ladhake, “Promising large scale image retrieval by using intelligent semantic binary code generation technique,” *Procedia Comput. Sci.*, vol. 48, pp. 282–287, Jan. 2015.

[2] X. Zhang, Z. Li, L. Zhang, W.-Y. Ma, and H.-Y. Shum, “Efficient indexing for large scale visual search,” in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Kyoto, Japan, Sep. 2009, pp. 1103–1110.

[3] X. Liu, L. Huang, C. Deng, B. Lang, and D. Tao, “Query-adaptive hash code ranking for large-scale multi-view visual search,” *IEEE Trans. Image Process.*, vol. 25, no. 10, pp. 4514–4524, Oct. 2016.

[4] X. Liu, Z. Li, C. Deng, and D. Tao, “Distributed adaptive binary quantization for fast nearest neighbor search,” *IEEE Trans. Image Process.*, vol. 26, no. 11, pp. 5324–5336, Nov. 2017.

[5] D. Cai, X. He, Z. Li, W.-Y. Ma, and J.-R. Wen, “Hierarchical clustering of WWW image search results using visual, textual and link information,” in *Proc. 12th Ann. ACM Int. Conf. Multimedia*, New York, NY, USA, 2004, pp. 952–959.

[6] M. Danish, R. Rawat, and R. Sharma, “A survey: Content based image retrieval based on color, texture, shape & neuro fuzzy,” *Int. J. Eng. Res. Appl.*, vol. 3, no. 5, pp. 839–844, Sep. 2013.

[7] C. Patil and V. Dalal, “Content based image retrieval using combined features,” in *Proc. Int. Conf. Workshop Emerg. Trends Technol. (ICWET)*, Mumbai, India, 2011, pp. 102–105.

[8] 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, Jan. 2016.

[9] A. Gionis, P. Indyk, and R. Motwani, “Similarity search in high dimensions via hashing,” in *Proc. 25th Int. Conf. Very Large Data Bases*, Edinburgh, U.K., 1999, pp. 518–529.

[10] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[11] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, San Diego, CA, USA, Jun. 2005, pp. 886–893.

[12] 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.

[13] H. Lai, Y. Pan, Y. Liu, and S. Yan, “Simultaneous feature learning and hash coding with deep neural networks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 3270–3278.

[14] 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.
WEI YAO received the B.S. degree in automation from the Qilu Institute of Technology, Jinan, China, in 2018. She is currently pursuing the M.S. degree in control science and engineering with the University of Shanghai for Science and Technology, Shanghai, China. Her research interests include computer vision and deep learning.

FEIFEI LEE (Member, IEEE) received the Ph.D. degree in electronic engineering from Tohoku University, Japan, in 2007. She is currently a Professor with the University of Shanghai for Science and Technology. Her research interests include pattern recognition, video indexing, and image processing.

LU CHEN received the B.S. degree from the Industrial and Commercial College, Anhui University of Technology, Ma’anshan, China, in 2016. He is currently pursuing the M.S. degree in control science and engineering with the University of Shanghai for Science and Technology, Shanghai, China. His research interests include computer vision and deep learning.

CHAOWEI LIN received the B.S. degree in mechanical design, manufacturing, and automation from the Guangdong University of Technology, Guangzhou, Guangdong, China, in 2018. He is currently pursuing the M.S. degree in control engineering with the University of Shanghai for Science and Technology, Shanghai, China. His research interests include computer vision and deep learning.

SHUAI YANG received the B.S. degree in automation from the Changzhou Institute of Technology, Changzhou, China, in 2018. He is currently pursuing the M.S. degree in control theory and control engineering with the University of Shanghai for Science and Technology, Shanghai, China. His research interests include computer vision and deep learning.

HANQING CHEN received the B.S. degree in opto-electronics information science and engineering from the Changzhou Institute of Technology, Changzhou, China, in 2018. He is currently pursuing the M.S. degree in control engineering with the University of Shanghai for Science and Technology, Shanghai, China. His research interests include computer vision and deep learning.

QIU CHEN (Member, IEEE) received the Ph.D. degree in electronic engineering from Tohoku University, Japan, in 2004. Since then, he has been an Assistant Professor and an Associate Professor with Tohoku University. He is currently a Professor with Kogakuin University. His research interests include pattern recognition, computer vision, information retrieval, and their applications. He serves on the editorial boards for several journals, as well as committees for a number of international conferences.

***