IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 21, 2020, accepted February 19, 2020, date of publication February 24, 2020, date of current version March 12, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2975883

Unsupervised Ensemble Hashing: Boosting Minimum Hamming Distance

YUFEI ZHA [1,2], ZHULING QIU [2], PENG ZHANG [1], (Member, IEEE), AND WEI HUANG [3]

[1] National Engineering Laboratory for Integrated Aero-Space-Ground-Ocean Big Data Application Technology, School of Computer Science, Northwestern Polytechnical University, Xi'an 710072, China
[2] Aeronautics Engineering College, Air Force Engineering University, Xi'an 710051, China
[3] School of Computer and Information Engineering, Jiangxi Normal University, Nanchang 330022, China

Corresponding author: Yufei Zha (zhayufei@126.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61773397, Grant 61971352, and Grant 61862043, in part by the Natural Science Basic Research Plan in Shaanxi Province of China under Grant 2018JM6015, and in part by the Fundamental Research Funds for the Central Universities under Grant 3102019ZY1003 and Grant 3102019ZY1004.

ABSTRACT Hashing aims at learning discriminative binary codes of high-dimensional data for the approximate nearest neighbor searching. However, the distance ranking obtained by traditional methods is not optimum in the Hamming space, and it degrades the performance for retrieval tasks. To tackle the above problem, an unsupervised ensemble hashing is proposed to improve the ranking accuracy by assembling the diverse hash tables independently in this paper. We observe that the higher the accuracy is the larger diversity the base learner has, and the more effective the ensemble method is. Based on this principle, two special ensembles hashing approaches are proposed to increase diversity by bootstrap sampling with data-dependent methods. Especially, the results are better when the minimum Hamming distance is large and the variance of the Hamming distance is small. This proposed method is conducted in the experiments and the results show that it can achieve about **10%-25%** performance compared with the baseline algorithm, which achieves competitive results with the state-of-the-art methods on the CIFAR-10 and LabelMe benchmarks.

INDEX TERMS Unsupervised hashing, ensemble method, accuracy and diversity, hamming distance, distance variance.

I. INTRODUCTION

In the machine learning, nearest neighbor search [1], [2] is one of the widely used technologies, and it has been applied in many areas, *e.g.* , data retrieval [3], [4], scene text recognition [5] and computer vision [6]. Practically, it's not necessary to output the exact nearest neighbors for each query on the large-scale datasets that are increasing from the Internet. Thus, hashing is an approximate nearest neighbor search approach which has attracted more and more attention for its higher speed and lower storage cost [7], [8]. The aim of hashing is to pursue a mapping which enables the high-dimensional data represented in a low-dimensional space with a short code consisting of a sequence of bits [9], [10], while the similarities are similar in both spaces.

Due to the low storage cost and fast query speed, hashing has been widely adopted for ANN search in large-scale datasets [11], [12]. In a word, hashing is an approach of transforming the data item to a low-dimensional representation, or equivalently a short code consisting of a sequence of bits [9]. The learning-based hashing methods can be divided into three main streams: semi-supervised [13]–[15] , supervised [16]–[19] and unsupervised methods [15], [20]–[23].

Most hashing methods are based on semi-supervised [13], [14], [24] and supervised methods [25], [26]. They are developed to improve the quality of hashing by incorporating supervisory information in form of class labels, including Semi-supervised hashing (SSH) [15], Binary Reconstructive Embedding (BRE) [27], minimal loss hashing (MLH) [28], Kernel-Based Supervised Hashing (KSH) [19], Two-Step Hashing (TSH) [18], Fast Supervised Hashing (FastHash) [17], and Supervised Discrete Hashing (SDH) [16]. Unfortunately, these methods cannot perform on the data that does not have any given label information in practice, and this limits the application scale.

To tackle the above problem, some unsupervised methods [20] have been developed to learn hash functions only using unlabeled data while preserving similarities of data in

The associate editor coordinating the review of this manuscript and approving it for publication was Joey Tianyi Zhou.

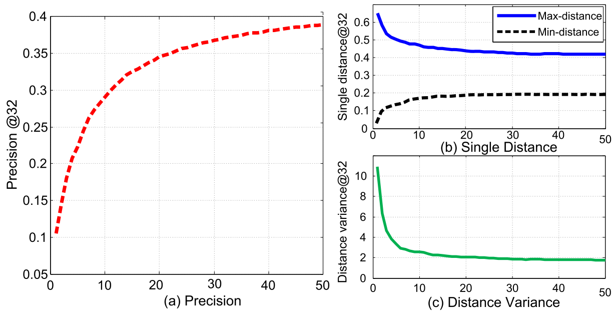


FIGURE 1. (a) Performance curve with 32 bits binary codes. The normalized minimum, maximum distance (b) and distance variance (c) curves for our method on the CIFAR-10 [36] dataset. The horizontal axis is the ensemble number.

the raw space. The representative algorithms in this category include Locality Sensitive Hashing (LSH) [29], Locality sensitive binary codes from shift-invariant kernels (SKLSH) [30], Spectral Hashing (SH) [22], PCA Hashing [15], Iterative quantization (ITQ) [31], Isotropic Hashing (IsoH) [22], Anchor graph hashing (AGH) [32], Harmonious Hashing [33], Learning Binary Codes with Bagging PCA (BPCAH) [21] and Sparse Projections for High-Dimensional Binary Codes (SP) [34]. Recently, they have achieved good improvement, but they cannot obtain a unique Hamming distance ranking since the diversity of the binary codes drawn from different learning procedures. This makes the retrieval performance unstable and degraded in large-scale data.

In this paper, we employ an ensemble method to integrate multiple hash tables independently generated by different base learners, so as to improve the ranking accuracy [35]. Our theoretical analysis reveals that the ensemble method is effective, especially when base learners simultaneously have high accuracy and large diversity. According to this, we employ bootstrap sampling to enhance the diversity of the data-dependent methods, such as ITQ [31]. It is interesting that the results are better when the minimum Hamming distance is large and the variance of the Hamming distances is small, and this can be seen in Fig. 1 on the CIFAR-10 [36] dataset with our method. Fig. 1 (a) shows that the performance continues to rise until convergence but it does not overfit. The ensemble performance will converge to bounding as we add more ensembles. The normalized minimum Hamming distance and distance variance respectively increase and decreases until convergence, and they reach a specific bound at last, and it is shown in Fig. 1 (b) and (c).

Our main contributions are as follows:

- An unsupervised ensemble hashing method is proposed to improve the Hamming distance ranking accuracy for unsupervised hashing;
- We analyze the proposed method theoretically, which indicates that it is critical that the base learner has high accuracy and large diversity simultaneously for ensembling;
- Two effective ensemble hashing approaches are proposed to increase the diversity by bootstrap sampling with data-dependent methods.

The rest of the paper is organized as follows. In Section II, we discuss the related work briefly. Section III describes the details of our unsupervised ensemble hashing method and two specific ensemble hashing methods designed. The experimental results are presented in Section IV. Finally, Section V concludes the paper.

II. RELATED WORK

Recently, many efforts have been done to solve the hashing problem, and we briefly review the most tightly related work in this section. Comprehensive reviews on hashing and benchmark evaluation can be found in [37].

A. UNSUPERVISED HASHING

The data-independent hashing methods projection functions are independent of training data. In locality-sensitive hashing (LSH) [29], after two adjacent data in the raw data space are transformed by the same map or projection, the probability that the two data points are still adjacent in the new space is large. There are many different variants of LSH [29]. It not only supports Euclidean space but also supports more dimensions, such as cosine similarity [38], Gaussian kernel [39] and shift-invariant kernels (SKLSH) [30]. Its advantage is that it is simple and easy to use, with a linear search time, but it requires a very long bit hash codes or a lot of hash tables to achieve the expected performance.

In data-dependent methods, principal component analysis (PCA) [40] as a linear orthogonal transform is often employed to learn the projection functions [15], [22], [31], [41], and it is derived from the data. However, the obtained binary hashing codes in terms of each projected data is not optimum for the different variances of each projected data. One simple way is to balance the variances with a random orthogonal transformation [41] and make them the same in the low-dimensional space. Gong *et al.* [31] proposes to preprocess the data by principal component analysis (PCA) [40] and pursue an orthogonal rotation matrix to minimize the quantization error when the data is mapped to the vertices of the binary hypercube. In order to enable the projected data to have equal variances, only the top eigenvectors are utilized to generate a piece of short but strong codes [21], while Isotropic Hashing (IsoH) [22] directly learns projection function to make the variances same. Additionally, locality preserving projection (LPP) [42] is used to replace the PCA [40] to obtain non-linear projection to preserve more information. Different from the above single-model based methods, the motivation of this study is how to ensemble these models to achieve better performance.

B. ENSEMBLE APPROACHES

Ensemble classification and regression approaches [43]–[47] have attracted a great deal of interest in recent years. The classification problem can be solved by combining the different ensembles non-linearly or majority voting, while linearly weighted ensembles can also optimize the regression problem. Bagging [43] and Boosting [48] are two common ensemble techniques. They resample data from all data to construct different training datasets, which are utilized to learn different classifiers. Different from bagging that is based on random sampling in all the data with replacement, the training set is chosen based on the performance of the earlier classifiers in the boosting method. More specially, examples that are incorrectly predicted by previous classifiers in a series of classifiers are chosen more often than examples that are correctly predicted.

Few hashing methods use ensemble techniques to improve retrieval performance. Different from complementary Hashing (CH) [49] that proposes to adopt the boosting-based approach to build complementary hash tables, BPCAH [21] exploits the bagging method to learn several pieces of diverse short codes and concatenate them into long codes.

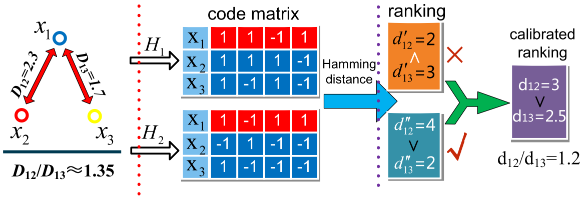
In recent works [24], [47], a bagging-boosting-based multi-hashing with query-adaptive re-ranking [24] is proposed for semi-supervised hashing. The variance on the projected dimensions is utilized for similarity-preserving mapping [26] and the samples are weighted in the training phase [50]. The ensemble hashing is proposed by training each hash function (or bit) independently from each other [47] and considering higher accuracy, larger diversity and the optimal weights for predictors simultaneously [51]. However, these methods pay less attention to both accuracy and diversity [51], which motivates us to extend the projection-based methods using an ensemble trick.

III. THE PROPOSED METHOD

The goal of hashing is to learn mapping functions to preserve similarities of samples between Hamming and Euclidean spaces. The following notations are used in the rest of this study for readers to understand. The samples are denoted as $\mathcal{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots \boldsymbol{x}_n\}$ with $\boldsymbol{x}_i \in \mathbb{R}^d$, and $\{H_l\}_{l=1}^{L}$ represents L hash tables, where each hash table is $H_l(\boldsymbol{x}) = \left\{h_i(\cdot)\right\}_{i=1}^{k}$ consisting of k hash functions. Here, $h_i(\boldsymbol{x}) = \text{sgn}\big(g(\boldsymbol{x})\big)$, where $g(\boldsymbol{x})$ is a prediction function and sgn() is a sign function. The output of the hash table is k-bit binary codes $\boldsymbol{B}_l \in \{1, -1\}^k$, so each sample is replaced with k binary codes. The Euclidean distance between samples $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ are denoted as D_{12}, while d_{12} is their distance in Hamming space.

A. OBSERVATION AND MOTIVATION

Our observation is that the Hamming distance rankings by the traditional methods are not unique, even conflict with each other, which is illustrated in Fig. 2. In the Euclidean space, $D(\boldsymbol{x}_1, \boldsymbol{x}_2) > D(\boldsymbol{x}_1, \boldsymbol{x}_3)$ shown in Fig. 2 (a), which indicates that $\boldsymbol{x}_1$ is more similar to $\boldsymbol{x}_3$ than to $\boldsymbol{x}_2$, which indicates that $\boldsymbol{x}_3$ is more similar to $\boldsymbol{x}_1$ than to $\boldsymbol{x}_2$, and the similarity proportion is 1.35. After they are mapped into the Hamming space with two different hash tables $H_1(\boldsymbol{x})$ and $H_2(\boldsymbol{x})$ shown in Fig. 2 (b), their distance rankings are distance rankings are conflicted: $d'_{12} < d'_{13}$ by $H_1(\boldsymbol{x})$ and $d''_{12} > d''_{13}$ by $H_2(\boldsymbol{x})$. Because the traditional methods focus on the similarities between samples and neglect the distance ranking.



(a) Euclidean space (b) Hamming space (c) Distance ranking

FIGURE 2. Our observations (see Section III-A for details). The Euclidean distances between samples x_1, x_2 and x_3 are shown in (a), where the similarity proportion of d_{12} and d_{13} is equal to 1.35. These samples are mapped into Hamming space by ITQ [31] method with two different hash tables $H_1(x)$ and $H_2(x)$, which is illustrated in (b). Their distance rankings are conflicted: $d'_{12} < d'_{13}$ by $H_1(x)$ and $d''_{12} > d''_{13}$ by $H_2(x)$. But the calibrated ranking ($d'_{12} > d'_{13}$) by our method is consistent ($d_{12}/d_{13} = 1.2$) with that in Euclidean space, which is shown in (c).

We define the hashing method that generates diverse hash tables as ambiguous methods in this study. Suppose two binary codes $\boldsymbol{B}_i = \text{sgn}[g_i(\mathcal{X})]$ and $\boldsymbol{B}_j = \text{sgn}[g_j(\mathcal{X})]$ are encoded by two hash tables H_i and H_j, the probability of the diversity between the binary codes $\boldsymbol{B}_i$ and $\boldsymbol{B}_j$ can be written as:

$$\begin{aligned}\mathbb{P}\{\boldsymbol{B}_i = \boldsymbol{B}_j\} &= \mathbb{P}\{\text{sgn}[g_i(\mathcal{X})] = \text{sgn}[g_j(\mathcal{X})]\}\\ &\propto exp(- \parallel \boldsymbol{W}_i\mathcal{X} - \boldsymbol{W}_j\mathcal{X} \parallel_F^2) \qquad (1)\end{aligned}$$

where $\| \cdot \|_F$ denotes the Frobenius norm and $\boldsymbol{W}$ is the projection function. Here, *exp* represents the exponent distribution and enable the probability values between 0 and 1. When $\boldsymbol{B}_i$ equals to $\boldsymbol{B}_j$, the $\boldsymbol{W}_i$ closes to the $\boldsymbol{W}_j$, so the probability approaches to 1. in contrast, the probability will be 0 when the $\boldsymbol{B}_i$ is different from $\boldsymbol{B}_j$. Fig. 3 illustrates the normalized average ambiguity of ambiguous hashing methods (ITQ [31], SKLSH [30] and LSH [29]), and they are calculated according to Eq. 1 on the CIFAR-10 [36] dataset.

Distance ranking benefits to achieve better performance for retrieval tasks through preserving the similarity from the Euclidean space into Hamming space. The distance ranking can be calibrated by combining the Hamming distances d' and d'', and similarity proportion (1.2) nears to the original value (1.35), which is shown in Fig. 2. This reveals that the calibrated ranking can preserve the neighborhood relation: not only achieve good results, but it is more accurate. This idea that ensembles results from multiple weak learners has been proved to be efficient in classification and regression problems [52]. Therefore, we are motivated to utilize an ensemble approach to improve retrieval performance and an ensemble hashing is proposed in this study.

B. OUR METHOD

As observed above, the Hamming distance measured by different hash tables may be not identical, while different hash tables may generate non-unique Hamming ranking.

Let $\{H_l\}_{l=1}^{L}$ denote L hash tables constructed by the base methods (such as ITQ [53], SKLSH [30] and so on), and k is the code length of one hash table $H_l = \left\{h_i(\cdot)\right\}_{i=1}^{k}$. For

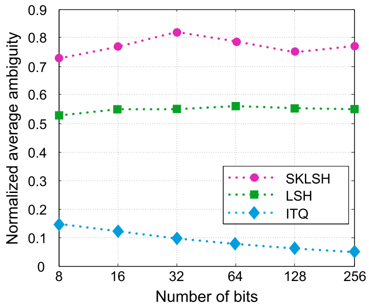


FIGURE 3. The normalized average ambiguity of different ambiguous hashing methods on the CIFAR-10 [36] dataset. The horizontal axis denotes the code length of each hash table. The normalized average ambiguity is shown in the vertical axis.

two samples $\boldsymbol{x}$ and $\boldsymbol{y}$, the Hamming distance evaluated by the l-th hash table can be denoted as $d_l(\boldsymbol{x}, \boldsymbol{y})$. Its corresponding similarity $f_l(\boldsymbol{x}, \boldsymbol{y})$ can be directly converted by the Hamming distance, and this is denoted as:

$$f_l(\boldsymbol{x}, \boldsymbol{y}) = \frac{k - d_l(\boldsymbol{x}, \boldsymbol{y})}{k}. \tag{2}$$

This reveals that the Hamming distance and the similarity are in one-to-one correspondence. Thus, we can acquire L diverse similarity $\{f_l\}_{l=1}^{L}$ corresponding to the Hamming distance. It is obvious that the individual similarity f_l is not accurate and robust, and it may be far away from the real similarity in the Euclidean space.

Let S denotes a real similarity between two samples. The similarity error of the individual can be expressed as $(f_l - S)^2$. Previous analysis has illustrated that the individual error $(f_l - S)^2$ is always large. Here, we propose to utilize the ensemble approach to reduce this error. For samples $\boldsymbol{x}$ and $\boldsymbol{y}$, define the ensemble Hamming distance and ensemble similarity between them as $d_L(\boldsymbol{x}, \boldsymbol{y})$ and $f_{ens}(\boldsymbol{x}, \boldsymbol{y})$, so $d_L(\boldsymbol{x}, \boldsymbol{y}) = \sum_{l=1}^{L} d_l(\boldsymbol{x}, \boldsymbol{y})$. Then we have

$$\begin{aligned} f_{ens}(\boldsymbol{x}, \boldsymbol{y}) &= \frac{L \times k - d_L(\boldsymbol{x}, \boldsymbol{y})}{L \times k} \\ &= \frac{1}{L}\left(\frac{L \times k - \sum_{l=1}^{L} d_l(\boldsymbol{x}, \boldsymbol{y})}{k}\right) \\ &= \frac{1}{L}\left(\sum_{l=1}^{L} \frac{k - d_l(\boldsymbol{x}, \boldsymbol{y})}{k}\right) \\ &= \frac{1}{L}\sum_{l=1}^{L} f_l(\boldsymbol{x}, \boldsymbol{y}), \end{aligned} \tag{3}$$

where k denotes the code length of individuals. This formula denotes that the ensemble similarity f_{ens} is a convex combination of the component individual similarity. The general form can be written as:

$$f_{ens} = \sum_{l} w_l f_l, \tag{4}$$

where w_l is the normalized weight coefficient, and $\sum_l w_l = 1$. When all individuals are treated equally, the weight can be denoted as $w_l = 1/L$.

Recall Fig. 2 (b), if the two ranking results are the same or both false, the individuals similarity $\{f_l\}_{l=1}^{L}$ is invariable or the similarity error $(f_l - S)^2$ of individuals is extremely large, and the ensemble method is not effective no longer.

Then, the similarity error can be decomposed by [45]:

$$\begin{aligned} &\sum_{l} w_l \big(f_l - S\big)^2 \\ &= \sum_{l} w_l \big(f_l - f_{ens} + f_{ens} - S\big)^2 \\ &= \sum_{l} w_l \big(f_l - f_{ens}\big)^2 + \big(f_{ens} - S\big)\Big[2\sum_{l} w_l f_l - \big(f_{ens} + S\big)\Big] \\ &= \sum_{l} w_l \big(f_l - f_{ens}\big)^2 + \big(f_{ens} - S\big)^2, \end{aligned} \tag{5}$$

where $(f_l - S)^2$ and $(f_{ens} - S)^2$ are the similarity error of individual and the convex-combined ensemble, respectively. The decomposition can be re-written as:

$$\big(f_{ens} - S\big)^2 = \sum_{l} w_l \big(f_l - S\big)^2 - \sum_{l} w_l \big(f_l - f_{ens}\big)^2. \tag{6}$$

The first term, $\sum_l w_l (f_l - S)^2$, is the weighted average error of the individuals. Its value depends on the accuracy of individuals. The second term, $\sum_l w_l (f_l - f_{ens})^2$, is the ambiguity term, which can be seen as the similarity variance of the individuals. Since this is always positive, it is subtracted from the first term, meaning the ensemble is guaranteed lower error than the average individual error. The larger the ambiguity term is, the lower the ensemble error is.

This is a very interesting result for ensemble research, providing a very simple expression for the effect due to error correlation in an ensemble. Eq. 6 shows that the error of the convex-combined ensemble will be less than or equal to the average error of the individuals, that is:

$$\big(f_{ens} - S\big)^2 \leq \sum_{i} w_i \big(f_i - S\big)^2. \tag{7}$$

In fact, one of the individuals may have lower error than the average, and even lower than the ensemble.

According to the decomposition, we can conclude two necessary principles for our ensemble method as follows:

- In terms of the first term of the decomposition, the lower the error of individuals is, the smaller the weighted average error of individuals is, and the better the convex-combined ensemble result should be. Therefore, it is necessary for individuals (weak learners) to have higher accuracy;
- The ambiguity term (the second term) denotes that larger diversity between individuals is necessary to reduce the ensemble error;
- The weight is used as the confidence of the individual. Optimal weights should be chosen carefully to enable each ensemble to decrease the ensemble error.

This denotes that if we increase the individual accuracy (the average error term) or enlarge the diversity (the ambiguity term), the ensemble results will be more effective.

Unfortunately, as the diversity of the individuals rises, so does the value of the average error term (the first term). This reveals that diversity itself is not enough, and we need to balance between diversity and individual accuracy, in order to achieve the low overall ensemble error.

C. WHAT MAKES THE SIMILARITY VARIANCE INCREASE?

According to the above analysis, increasing the similarity variance helps to reduce the ensemble error when the base learner is fixed. We note that similarity variance measures how far a set of individuals similarity (f_l) is spread out. A small variance indicates the individual's similarity tends to be close to the expected value (f_{ens}). In order to explore what makes the similarity variance increase, we have the following theorem.

Theorem 1: The similarity variance is proportional to the diversity of hash tables, i.e. $f_i - f_j \propto \| \boldsymbol{H}_i - \boldsymbol{H}_j \|_F^2$.

Proof: For samples $\boldsymbol{x}$ and $\boldsymbol{y}$, their binary codes generated by the i-th hash table can be written as $\boldsymbol{H}_i(\boldsymbol{x}) = sgn\big(\boldsymbol{x}\boldsymbol{W}^{(i)}\big)$ and $\boldsymbol{H}_i(\boldsymbol{y}) = sgn\big(\boldsymbol{y}\boldsymbol{W}^{(i)}\big)$, respectively. The Hamming distance between them can be computed as $d_i(\boldsymbol{x}, \boldsymbol{y}) = [k - \boldsymbol{H}_i(\boldsymbol{x}) \circ \boldsymbol{H}_i(\boldsymbol{y})]/2$, where " $\circ$ " represents inner product. Then we have the corresponding similarity as follows:

$$f_i(\boldsymbol{x}, \boldsymbol{y}) = \frac{k - d_i(\boldsymbol{x}, \boldsymbol{y})}{k} = \frac{k - [k - \boldsymbol{H}_i(\boldsymbol{x}) \circ \boldsymbol{H}_i(\boldsymbol{y})]/2}{k} = \frac{k + \boldsymbol{H}_i(\boldsymbol{x}) \circ \boldsymbol{H}_i(\boldsymbol{y})}{2k}. \tag{8}$$

Similarly, for the j-th hash table, we have $f_j(\boldsymbol{x}, \boldsymbol{y}) = [k + \boldsymbol{H}_j(\boldsymbol{x}) \circ \boldsymbol{H}_j(\boldsymbol{y})]/2k$. Therefore, the similarity diversity can be denoted as:

$$f_i(\boldsymbol{x}, \boldsymbol{y}) - f_j(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{2k}\Big[\boldsymbol{H}_i(\boldsymbol{x}) \circ \boldsymbol{H}_i(\boldsymbol{y}) - \boldsymbol{H}_j(\boldsymbol{x}) \circ \boldsymbol{H}_j(\boldsymbol{y})\Big]. \tag{9}$$

Like the same signed magnitude relaxation as in literature [16], the above function can be relaxed as:

$$\begin{aligned} &f_i(\boldsymbol{x}, \boldsymbol{y}) - f_j(\boldsymbol{x}, \boldsymbol{y}) \\ &= \frac{1}{2k}\Big[\big(\boldsymbol{x}\boldsymbol{W}^{(i)}\big)\big(\boldsymbol{y}\boldsymbol{W}^{(i)}\big)^{\mathrm{T}} - \big(\boldsymbol{x}\boldsymbol{W}^{(j)}\big)\big(\boldsymbol{y}\boldsymbol{W}^{(j)}\big)^{\mathrm{T}}\Big] \\ &= \frac{1}{2k}\Big[\boldsymbol{x}\big(\boldsymbol{W}^{(i)}\boldsymbol{W}^{(i)\mathrm{T}} - \boldsymbol{W}^{(j)}\boldsymbol{W}^{(j)\mathrm{T}}\big)\boldsymbol{y}^{\mathrm{T}}\Big]. \end{aligned} \tag{10}$$

According to the above equation, when the samples $\boldsymbol{x}$ and $\boldsymbol{y}$ are fixed, we find that the similarity diversity is determined by the hash tables. As an extreme example, if the hash tables are restricted to be orthogonal to each other $\boldsymbol{W}^{(i)^{\mathrm{T}}}\boldsymbol{W}^{(j)} = 0$, the diversity between two hash tables is the largest. It is obvious that the matrix $\boldsymbol{W}^{(i)}\boldsymbol{W}^{(i)^{\mathrm{T}}}$ is orthogonal to matrix $\boldsymbol{W}^{(j)}\boldsymbol{W}^{(j)^{\mathrm{T}}}$, yet the similarity diversity is also the largest. Certainly, if the generated hash tables are the same, namely, $\boldsymbol{W}^{(i)} = \boldsymbol{W}^{(j)}$, *i.e.* , the similarity f_i and f_j will be identical and the similarity variance is zero, so the ensemble method is useless. The larger the diversity of hash tables is, the bigger the similarity variance is, and the more effective the ensemble method is.

D. BOOTSTRAP

Theorem 1 reveals that the diversity of hash tables is important for the similarity variance. Motivated by the Learning Binary Codes with Bagging PCA (BPCAH) [21], we employ bootstrap sampling to increase the diversity of hash tables for data-dependent method. Especially, we randomly sample a small subset of p training data $\boldsymbol{X}^{(l)} = \{\boldsymbol{x}_1^{(l)}, \boldsymbol{x}_2^{(l)}, \ldots, \boldsymbol{x}_p^{(l)}\}$ to learn diverse hash tables each time. This process is repeated several times and the obtained short codes are concatenated into one piece of long codes. The objective function can be written as:

$$\begin{aligned} &\max_{\boldsymbol{W}^{(l)} \in \mathbb{R}^{d \times k}} \quad \frac{1}{p}\mathrm{tr}\Big(\boldsymbol{W}^{(l)^T}\boldsymbol{X}^{(l)}\boldsymbol{X}^{(l)^T}\boldsymbol{W}^{(l)}\Big) \\ &\text{s.t. } \boldsymbol{W}^{(l)^T}\boldsymbol{W}^{(l)} = \boldsymbol{I}_k, \end{aligned} \tag{11}$$

where $\boldsymbol{W}^{(l)} \in \mathbb{R}^{d \times k}$ denote the top PCA [40] vectors corresponding to the l-th individual, and k is the length of short codes, and tr() is the matrix trace, and $\boldsymbol{I}_k$ is a t-by-t identity matrix. The final projection matrix $\boldsymbol{W} = \big[\boldsymbol{W}^{(1)}, \boldsymbol{W}^{(2)}, \ldots, \boldsymbol{W}^{(L)}\big]$ can be obtained by repeating the above process with L times. If each block is regarded as a hash table, BPCAH [21] can be seen as a specific example of our method. However, our method is different from BPCAH [21] in following aspects:

- BPCAH [21] is just a special case of our method. We propose a flexible ensemble hashing framework and analyze the significance of accuracy and diversity in improving the ensemble performance. But the BPCAH is just a special case of our framework;
- Individuals based on different basic learners with higher accuracy can be selected to reduce the weighted average error in our method. Additionally, the individuals can be learned by the same base learner or not in our framework. But all the individuals of BPCAH [21] are learned by performing the same operation on a different subset of training data;
- In contrast to the deterministic method in BPCAH [21], ambiguous methods are utilized to generate diversity even without utilizing the bootstrap technique in this study.

E. TWO ENSEMBLE HASHING ALGORITHMS

The previous sections have revealed that the higher accuracy and the larger diversity the individuals have, the more effective the ensemble method is. According to these principles, the ensemble performance can be boosted by increasing the accuracy of individuals or enlarging the individual's diversity or meet both of them simultaneously. We now introduce two special designed ensemble cases in the study.

1) WEIGHTED BAGGING PCA HASHING (WBPCAH)

Different from the BPCAH [21], we utilize the bootstrap technique to generate diverse short codes by stronger individuals with higher accuracy, so BPCAH [21] can be seen as a specific case of our method. Then these pieces of short codes are concatenated into one piece of long codes to improve performance effectively.

Let $\{\lambda_t\}_{t=1}^k$ denote the top k eigenvalues after performing PCA [40] on training data and $\boldsymbol{a}$ is the transformation vector. $\{z_i = \boldsymbol{a}^T \boldsymbol{x}_i\}_{i=1}^n \in \mathbb{R}^k$ are the PCA-projected data, where $\bar{z} = \frac{1}{n}\sum z_i = 0$ and the corresponding variance is $\{\boldsymbol{v}_t\}_{t=1}^k$ with $\boldsymbol{v}_t = var(z^t)$. According to the definition of PCA, we have $\boldsymbol{v}_t / \sum_{t=1}^k \boldsymbol{v}_t = \lambda_t / \sum_{t=1}^k \lambda_t$.

In order to preserve the information of the samples as much as possible, we assign the weight of a hashing bit that has large variance. This can be denoted as:

$$g_i = \boldsymbol{v}_i / \sum\nolimits_{t=1}^{k} \boldsymbol{v}_t. \tag{12}$$

This weight is utilized to balance the variances of different dimensions, so as to increase the individual performance. Then, the final objective function can be re-written as:

$$\begin{aligned} &\max_{\boldsymbol{W}^{(l)} \in \mathbb{R}^{d\times k}} \quad \frac{1}{p}\mathrm{tr}\Big(\boldsymbol{G}^{(l)}\boldsymbol{W}^{(l)^T}\boldsymbol{X}^{(l)}\boldsymbol{X}^{(l)^T}\boldsymbol{W}^{(l)}\Big) \\ &\quad \text{s.t. } \boldsymbol{W}^{(l)^T}\boldsymbol{W}^{(l)} = \boldsymbol{I}_k, \quad \sum_i g_i = 1, \\ &\quad g_i = \boldsymbol{v}_i / \sum\nolimits_{t=1}^{k} \boldsymbol{v}_t. \end{aligned} \tag{13}$$

where $\boldsymbol{G}^{(l)}$ is a $k \times k$ weight matrix corresponding to the l-th individual, and its diagonal element is the weight g_i.

2) BAGGING PCA-ITQ (BPCA-ITQ)

ITQ [31] is a fast and competitive hashing method, and it tries to learn an orthogonal rotation matrix $\boldsymbol{R}$ to refine the initial projection matrix learned by PCA, so the quantization error of mapping the data to the vertices of binary hypercube is minimized. Its objective function can be written as:

$$\min \mathcal{Q}(\boldsymbol{B}, \boldsymbol{R}) = \| \boldsymbol{Y} - \boldsymbol{V}\boldsymbol{R} \|_F^2, \tag{14}$$

where $\boldsymbol{V} = \boldsymbol{X}^{(l)}\boldsymbol{W}^{(l)}$ denotes the PCA-projected data for the l-th individual, and $\boldsymbol{B}$ is binary codes matrix and $\| \cdot \|_F$ denotes the Frobenius norm. In practice, ITQ [31] outperforms both PCAH [15] and WPCAH with a large margin, so it is a good base learner as an individual for ensembling to achieve better accuracy than that of other learners. Additionally, ITQ [31] itself is an ambiguous method, and it can generate diversity without bootstrap technique. Next, we will evaluate the similarity variance between individuals.

We randomly select samples $\{\boldsymbol{x}_i \in \mathbb{R}^d\}_{i=1}^m$ and $\{\boldsymbol{y}_j \in \mathbb{R}^d\}_{j=1}^q$ from $\mathcal{X}$ as test sets, where $\boldsymbol{x}_i \neq \boldsymbol{y}_j$. Given samples $\boldsymbol{x}_i$ and $\boldsymbol{y}_j$, their similarity measured by the l-th hash table can be denoted as $f_l\big(\boldsymbol{x}_i, \boldsymbol{y}_j\big)$. The convex combination of the L hash tables is $f_{ens}\big(\boldsymbol{x}_i, \boldsymbol{y}_j\big) = \sum_{l=1}^L w_l f_l\big(\boldsymbol{x}_i, \boldsymbol{y}_j\big)$, where the weight w_l is set as $w_l = 1/L$ since all hash tables are learned by the same base learner. Then the average similarity variance can be written as:

$$\begin{aligned} \upsilon &= \frac{1}{mq}\sum_{i=1}^{m}\sum_{j=1}^{q}\sum_{l=1}^{L} w_l \Big(f_l\big(\boldsymbol{x}_i, \boldsymbol{y}_j\big) - f_{ens}\big(\boldsymbol{x}_i, \boldsymbol{y}_j\big)\Big)^2 \\ &= \frac{1}{Lmq}\sum_{i=1}^{m}\sum_{j=1}^{q}\sum_{l=1}^{L} \Big(f_l\big(\boldsymbol{x}_i, \boldsymbol{y}_j\big) - f_{ens}\big(\boldsymbol{x}_i, \boldsymbol{y}_j\big)\Big)^2. \end{aligned} \tag{15}$$

TABLE 1. Description of datasets.

	CIFAR-10 [36]	LabelMe [54]
Dimensionality	320	512
Size	60,000	22,019

Analysis and discuss. As stated above, it has demonstrated that the ensemble approach is effective to improve the retrieval performance, so does the ranking accuracy. In this section, we will discuss how does the Hamming distance ranking change as the ensembles increase in detail.

We randomly select M samples $\{\boldsymbol{x}_i \in \mathbb{R}^d\}_{i=1}^M$ from $\mathcal{X}$ as a test set. Given data $\boldsymbol{x}_i$, its Hamming distance to others can be denoted as $\gamma_{ij} = d_h(\boldsymbol{x}_i, \boldsymbol{x}_j), \forall j = 1, \ldots, M, j \neq i$, then the minimum Hamming distance can be written as:

$$\mu_i = \min\{\gamma_{ij}\}_{j=1}^M. \tag{16}$$

Then, this distance is normalized as $\hat{\mu}_i = \mu_i/(L \times k)$ and $\hat{\gamma}_{ij} = \gamma_{ij}/(L \times k)$, where k is the individual size, and L is the ensemble number. The average minimum Hamming distance of all M test data is

$$\mu = \frac{1}{M}\sum_{i=1}^{M} \hat{\mu}_i. \tag{17}$$

The average distance variance is

$$\gamma = \frac{1}{M}\sum_{i=1}^{M}\sum_{j=1}^{M}(\hat{\gamma}_{ij} - \bar{\gamma}_i)^2, \tag{18}$$

where $\bar{\gamma}_i = \frac{1}{M}\sum_{j=1}^M \hat{\gamma}_{ij}$ denotes the expected value.

IV. EXPERIMENTS

Datasets: To evaluate the effectiveness of our ensemble algorithms, we conduct extensive experiments on two real-world image datasets: CIFAR-10 [36] and LabelMe [54].

CIFAR-10 [36]: the dataset consists of 60K 32 × 32 color tiny images which are categorized into 10 classes, namely, airplane, automobile, bird, cat, deer, dog, frog, horse, ship and truck. Gray-scale GIST descriptors [55] are used to represent them and they are computed at 8 orientations and on 4 different scales, resulting in 320-dimensional feature vectors.

LabelMe [54]: this dataset consists of $22,019$ images, which are scaled to 32×32 pixels, and then represented by 512-dimensional GIST descriptors. We summarize the size and dimensionality of the datasets in Table 1. For each dataset, we randomly select $1,000$ data points as queries and use the rest as gallery database and training set.

Compared Methods: We compare the proposed BPCA-ITQ and WBPCAH with several state-of-the-art hashing algorithms, including locality sensitive hashing (LSH) [29], spectral hashing (SH) [22], principal component analysis based hashing (PCAH) [15], weighted PCA Hashing (WPCAH) [21], iterative quantization (ITQ) [31], learning binary codes with bagging PCA (BPCAH) [15] and Especially, LSH [29] is a data-independent method that does not use the training data. PCAH [15], WPCAH [21] and ITQ [31] can be regarded

TABLE 2. The similarity variance changing with the number of individuals for BPCAH [21] and WBPCAH on CIFAR-10 [36] and LabelMe [54] datasets.

# Ensemble size	Method	2	4	6	8	10	12	14	16
CIFAR-10 [36]	BPCAH	0.0034	0.0049	0.0056	0.0059	0.0061	0.0062	0.0063	**0.0063**
	WBPCAH	0.0022	0.0030	0.0034	0.0035	0.0036	0.0037	0.0037	**0.0037**
LabelMe [54]	BPCAH	0.0033	0.0052	0.0059	0.0062	0.0063	0.0065	0.0066	**0.0066**
	WBPCAH	0.0026	0.0038	0.0041	0.0042	0.0045	0.0046	0.0046	**0.0047**

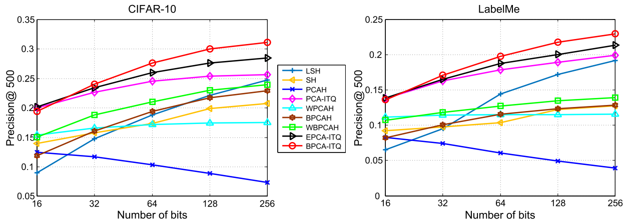


FIGURE 4. Precision of top 500 returned images with different numbers of bits on the two datasets.

as the base learners for our ensemble approach. BPCAH [21] can be seen as a specific example of our framework. Additionally, recent works such as URPH [35], DeepBit [56], HashGAN [57], are employed as the comparison methods. In this paper, there are two parameters to be set, the size of each bootstrap training set p and the size of each individual (ensemble) k. We set $p = 20\% \times n$ and $k = 16$ for all the comparisons, where n is the total number of data. The ensemble number is set as $L = 16$, and the ensemble code length can be denoted as $L \times k$.

Evaluation Protocols: To perform fair evaluation, we adopt the Hamming ranking search commonly used in the literature. All data in the database are ranked according to their Hamming distance to the query and the top K samples will be returned. The ground truth of each query instance is defined as its 50 nearest neighbors based on Euclidean neighbors [53]. The retrieval performance is measured with three widely used metrics: mean average precision (MAP) [3], precision of the top K returned examples and precision-recall curves. The precision, recall and MAP [3] are defined as follows:

$$\text{Precision} = \frac{\text{the number of retrieved relevant points}}{\text{the number of all retrieve points}}, \quad (19)$$

$$\text{Recall} = \frac{\text{the number of retrieved relevant points}}{\text{the number of all relevent points}}, \quad (20)$$

$$\text{MAP} = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{n_i}\sum_{j=1}^{n_i}\text{Precision}(R_{ij}). \quad (21)$$

where $q_i \in Q$ is a query, and n_i is the number of sample relevant to q_i in the dataset. The relevant samples are ordered as $\{x_1, x_2, \ldots, x_{n_i}\}$, and R_{ij} is the set of ranked retrieval results from the top result until getting to x_j. All the experiments are carried out on an Intel Core Duo 3.6GHz laptop of 6G memory with MATLAB2017a.

A. RESULTS

Fig. 4 evaluates the retrieval precision for top 500 returned images with the different numbers of bits on the two datasets. The proposed WBPCAH consistently performs better than the competitor BPCAH [21], although the advantage becomes smaller as the code size increases. The cause lies in the base learners, and WPCAH for our approach has higher accuracy than the PCAH [15], which is as a base learner of the BPCAH [21]. As the ensemble size increases, the performance of BPCAH [21] rises rapidly and almost reaches the performance of WBPCAH. This may be due to the fact that the similarity variance for BPCAH [21] is larger than our WBPCAH.

From Fig. 4, we can also find that the proposed BPCA-ITQ method already performs better than the compared methods. This may be because the bootstrap sampling is utilized to increase the diversity for BPCA-ITQ method. LSH [29], which is a data-independent method, also improves as the code size increases, and it almost reaches the performance of PCA-ITQ at 256 bits. Perhaps surprisingly, the performance of PCAH [15] consistently decreases as the code size increases. If each hash bit is regarded as an ensemble, our ensemble framework may be helpful in understanding this interesting phenomenon.

The experimental results that are shown in Table 2 reveals that accuracy itself is not enough, and we need to get the right balance between diversity and ensemble accuracy, in order to achieve the lowest overall ensemble error. The bias-variance decomposition for quadratic loss states that the generalization error of an estimator can be broken down into two components: bias and variance. These two usually work in opposition to each other: attempting to reduce the bias component will cause an increase in variance and vice versa. Summarily, the proposed methods exceed the baseline algorithms and achieve completive results with the recent works.

Fig. 5 illustrates the precision and precision-recall curves respectively using 64, 128 and 256 bits on the CIFAR-10 [36] dataset, which is also conformed from Fig. 4. Fig. 6 shows the precision and precision-recall curves on the LabelMe [54] dataset. The results illustrate that our proposed method BPCA-ITQ achieves superior performance to other methods.

MAP [3] is one of the most comprehensive criterions to evaluate the retrieval performance in the literature [31], [58], [59]. Table 3 shows the MAP [3] scores for all the algorithms on the two datasets. The BPCAH [21] and BWLH [51] are the unsupervised hashing, which are most related works to our method, and the results shows that our BPCA-ITQ has achieved the highest MAP [3] scores with different code lengths on all the datasets. At the same time, our method can get competitive performance to the recent state-of-the-art methods (such as, URPH [35], HashGAN [57]).

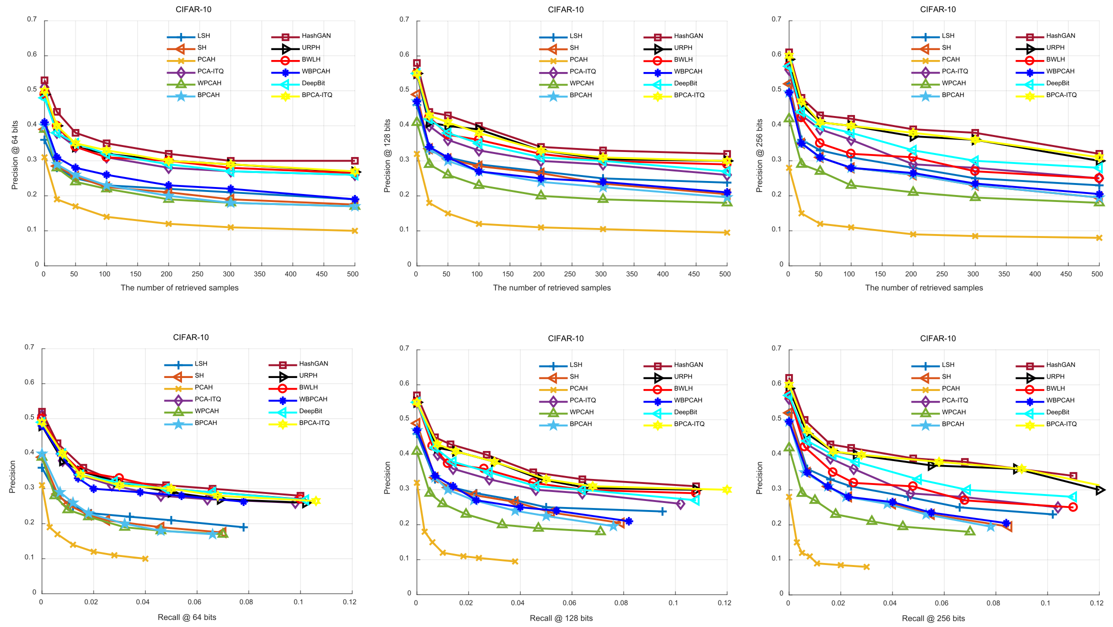


FIGURE 5. The precision and precision-recall curves using 64, 128 and 256 bits on the CIFAR-10 [36] dataset, respectively.

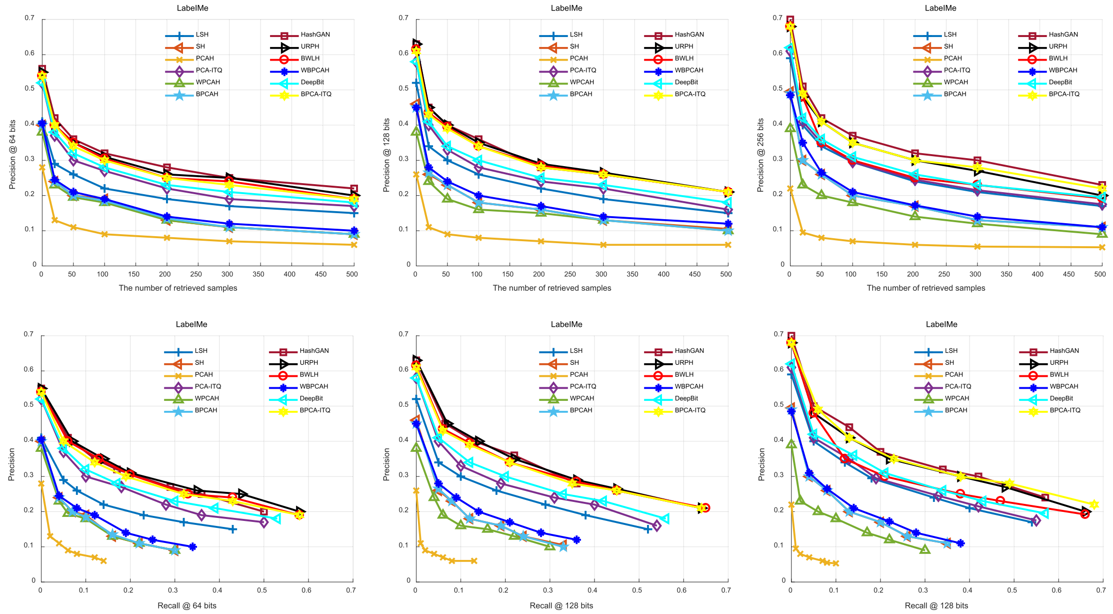


FIGURE 6. The precision and precision-recall curves respectively using 64, 128 and 256 bits on the LabelMe [54] dataset.

B. ABLATION STUDY

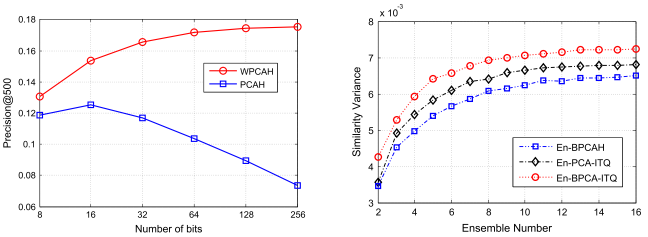
The results in Fig. 7 (a) illustrate that WPCAH has outperformed the traditional methods (*e.g.*, PCAH [15]) on all the bits. Therefore, if the WPCAH method is employed as a base learner, the accuracy will be higher than learning by the PCA and the lower error will boost the ensemble performance. The reason is that the bootstrap which is based on random sampling in the training data can increase the ambiguity of each individual.

To visualize the effects of the ensemble number, three ensemble hashing methods (Ensemble BPCAH, Ensemble PCA-ITQ, and Ensemble BPCA-ITQ) are employed to evaluate their similarity variances with respect to L on the CIFAR-10 [36] dataset. Here, we set bootstrap training set $p = 5000$ and code length $k = 16$ in the comparison.

The results in Fig. 7 (b) reveals that the methods that using PCA-ITQ as the base learner obtain larger similarity variance than that using BPCAH [21]. Especially,

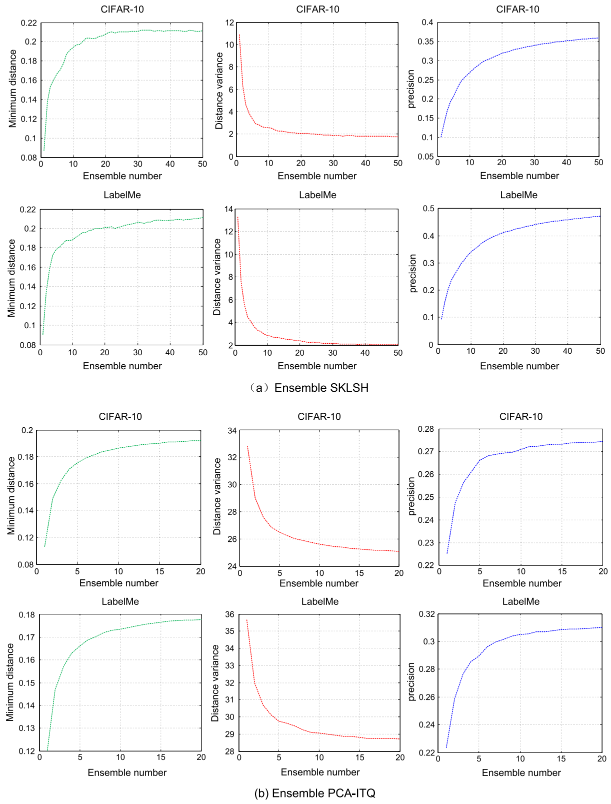
TABLE 3. Mean Average Precision (MAP) [3] on CIFAR-10 [36] and LabelMe [54] datasets which is shown in the first row. The code length is shown in the second row. The bold font presents the best result.

Method	CIFAR-10 [36]				LabelMe [54]			
# bits	64-bits	96-bits	128-bits	256-bits	64-bits	96-bits	128-bits	256-bits
SH [22]	0.1399	0.1647	0.1855	0.2277	0.1018	0.1435	0.1714	0.214
LSH [29]	0.2721	0.3045	0.3429	0.4264	0.2863	0.3462	0.3745	0.4109
PCAH [15]	0.0483	0.0030	0.0418	0.0357	0.0418	0.0345	0.0321	0.0244
PCA-ITQ [31]	0.1939	0.3996	0.4108	0.4217	0.3458	0.3615	0.3741	0.394
BPCAH [21]	0.1184	0.1415	0.1698	0.1837	0.1091	0.1376	0.1754	0.2043
WPCAH	0.1254	0.1325	0.1398	0.1425	0.1062	0.1198	0.1254	0.1295
WBPCAH	0.1521	0.1638	0.1812	0.1985	0.1452	0.1704	0.1967	0.2185
DeepBit [56]	0.2773	0.2978	0.3291	0.3476	0.2433	0.2645	0.3921	0.4123
HashGAN [57]	0.4812	0.5032	0.5348	0.5543	0.3542	0.3799	0.3958	0.4235
URPH [35]	0.4123	0.460	0.4876	0.5123	0.3953	0.3904	0.4721	0.4854
BWLH [51]	0.4089	0.4476	0.4342	0.4879	0.3812	0.3765	0.4587	0.4765
BPCA-ITQ	0.4156	0.448	0.4987	0.5264	0.3782	0.4067	0.4526	0.4978



(a) WPCAH (b) Ensemble BPCA-ITQ

FIGURE 7. Ablation study on the CIFAR-10 [36] dataset. Retrieval precision curves of PCAH [15] and the proposed WPCAH with various bits on the CIFAR-10 [36] dataset shown in (a). In (b), three methods: Ensemble BPCAH, Ensemble PCA-ITQ and Ensemble BPCA-ITQ, are employed to test the similarity variance, and the horizontal axis is the ensemble number.



(a) Ensemble SKLSH

(b) Ensemble PCA-ITQ

FIGURE 8. A series of normalized minimum distance (green curve), distance variance (red curve) and precision (blue curve) curves of two hashing methods (Ensemble SKLSH and Ensemble PCA-ITQ) on two datasets (CIFAR-10 [36] and LabelMe [54]).

the proposed Ensemble BPCA-ITQ consistently performs better than Ensemble PCA-ITQ since the bootstrap technique can increase the diversity between hash tables. Additionally, as the ensemble numbers increase, the similarity variance curves tend to be smooth and steady. Note that the special case BPCA-ITQ satisfies the higher accuracy and the larger diversity simultaneously, and this enables the method effective for ensembling to obtain better performance.

In order to explore the performance with respect to the ensemble number, we conduct experiments on two datasets (CIFAR-10 [36] and LabelMe [54]) with two hashing methods (Ensemble SKLSH and Ensemble PCA-ITQ), where the test set is $M = 2K$. The results are shown in Fig. 8, where the normalized minimum distance, distance variance, and precision curves are denoted as the green, red and blue curves respectively.

It can be seen that the normalized minimum Hamming and the precision curves are continually raising, while the distance variance is decreasing, as the ensembles increase. This shows that the minimum Hamming distance and distance variance are well relative to the ensemble results. In detail, the ensemble results are better when the minimum Hamming distance is large and the distance variance is small. The secret of ensemble hashing helps to understand why the ensemble method is effective to improve the ranking accuracy and to design effective hashing methods.

V. CONCLUSION

An ensemble method is used to calibrate the Hamming distance ranking on ambiguous methods in this paper. We conclude that high accuracy and large diversity are two necessary standards of our ensemble hashing, therefore we attempt to apply the bootstrap sampling to increase the diversity for data-dependent methods.

However, our method will suffer from severe redundancy of multiple hash tables and the multiple hash tables may also cause larger storage costs. We find that the Hamming distance closely related to the improvement of performance may be helpful in designing even more effective hashing methods in the future. Additionlly, neural network is powerful to represent the object with deep and hierarchical features, which benefits to generate hashing tables for accurate performance. It is complementary to this work and we will explore it in future works.

REFERENCES

[1] Z. Liu, F. Chen, and S. Duan, "Distributed fast supervised discrete hashing," *IEEE Access*, vol. 7, pp. 90003–90011, 2019.

[2] H. Xie, Z. Mao, Y. Zhang, H. Deng, C. Yan, and Z. Chen, "Double-bit quantization and index hashing for nearest neighbor search," *IEEE Trans. Multimedia*, vol. 21, no. 5, pp. 1248–1260, May 2019.

[3] 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.

[4] Z. Ye and Y. Peng, "Multi-scale correlation for sequential cross-modal hashing learning," in *Proc. ACM Multimedia Conf. Conf. (MM)*, 2018, pp. 852–860.

[5] H. Xie, S. Fang, Z.-J. Zha, Y. Yang, Y. Li, and Y. Zhang, "Convolutional attention networks for scene text recognition," *ACM Trans. Multimedia Comput., Commun., Appl.*, vol. 15, no. 1, pp. 1–17, Jan. 2019.

[6] Y.-Q. Li, Y.-F. Zha, B. Qin, J. Tian, and C. Liu, "Kernelised supervised context hashing," *IET Image Process.*, vol. 10, no. 12, pp. 986–995, Dec. 2016.

[7] J. Wang, T. Zhang, J. Song, N. Sebe, and H. T. Shen, "A survey on learning to hash," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 769–790, Apr. 2018.

[8] Y. Wu, Q. Sun, Y. Hou, J. Zhang, Q. Zhang, and X. Wei, "Deep covariance estimation hashing," *IEEE Access*, vol. 7, pp. 113223–113234, 2019.

[9] J. Wang, T. Zhang, J. Song, N. Sebe, and H. T. Shen, "A survey on learning to hash," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4 pp. 769–790, 2018.

[10] T. Yuan, W. Deng, and J. Hu, "Distortion minimization hashing," *IEEE Access*, vol. 5, pp. 23425–23435, 2017.

[11] A. Andoni, P. Indyk, H. L. Nguyễn, and I. Razenshteyn, "Beyond locality-sensitive hashing," in *Proc. 25th Annu. ACM-SIAM Symp. Discrete Algorithms*, Dec. 2013, pp. 1018–1028.

[12] Z. Zhou and W. Li, "Learning to hash for big data: Current status and future trends," *Chin. Sci. Bull.*, vol. 60, nos. 5–6, pp. 485–490, Feb. 2015.

[13] J. Zhang and Y. Peng, "SSDH: Semi-supervised deep hashing for large scale image retrieval," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 1, pp. 212–225, Jan. 2019.

[14] J. Zhang, Y. Peng, and M. Yuan, "SCH-GAN: Semi-supervised cross-modal hashing by generative adversarial network," *IEEE Trans. Cybern.*, vol. 50, no. 2, pp. 489–502, 2020.

[15] J. Wang, S. Kumar, and S.-F. Chang, "Semi-supervised hashing for large-scale search," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 12, pp. 2393–2406, Dec. 2012.

[16] 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.

[17] G. Lin, C. Shen, Q. Shi, A. van den Hengel, and D. Suter, "Fast supervised hashing with decision trees for high-dimensional data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1971–1978.

[18] G. Lin, C. Shen, D. Suter, and A. van den Hengel, "A general two-step approach to learning-based hashing," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 2552–2559.

[19] 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.

[20] J. Zhang, Y. Peng, and M. Yuan, "Unsupervised generative adversarial cross-modal hashing," in *Proc. 2nd AAAI Conf. Artif. Intell.*, 2018, pp. 539–546.

[21] C. Leng, J. Cheng, T. Yuan, X. Bai, and H. Lu, "Learning binary codes with bagging PCA," in *Proc. Mach. Learn. Knowl. Discovery Databases Eur. Conf. (ECML)*, 2014, pp. 177–192.

[22] Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2008, pp. 490–512.

[23] J. Zhang and Y. Peng, "Multi-pathway generative adversarial hashing for unsupervised cross-modal retrieval," *IEEE Trans. Multimedia*, vol. 22, no. 1, pp. 174–187, Jan. 2020.

[24] W. W. Y. Ng, X. Zhou, X. Tian, X. Wang, and D. S. Yeung, "Bagging–boosting-based semi-supervised multi-hashing with query-adaptive re-ranking," *Neurocomputing*, vol. 275, pp. 916–923, Jan. 2018.

[25] J. Zhang and Y. Peng, "Query-adaptive image retrieval by deep-weighted hashing," *IEEE Trans. Multimedia*, vol. 20, no. 9, pp. 2400–2414, Sep. 2018.

[26] S. Zhang, Y. Zha, Y. Li, H. Li, and B. Chen, "Image retrieval via balanced and maximum variance deep hashing," in *Proc. Comput. Vis. 2nd CCF Chin. Conf. (CCCV)*, 2017, pp. 622–637.

[27] B. Kulis and T. Darrell, "Learning to hash with binary reconstructive embeddings," in *Proc. Adv. Neural Inf. Process. Syst. 23rd Annu. Conf. Neural Inf. Process. Syst.*, 2009, pp. 1042–1050.

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

[29] A. Gionis, P. Indyk, and R. Motwani, "Similarity search in high dimensions via hashing," in *Proc. Int. Conf. Very Large Databases*, 2000, pp. 518–529.

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

[31] 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.

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

[33] B. Xu, J. Bu, Y. Lin, C. Chen, X. He, and D. Cai, "Harmonious hashing," in *Proc. Int. Joint Conf. Artif. Intell.*, 2013, pp. 1820–1826.

[34] Y. Xia, K. He, P. Kohli, and J. Sun, "Sparse projections for high-dimensional binary codes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3332–3339.

[35] S. Karaman, X. Lin, X. Hu, and S.-F. Chang, "Unsupervised rank-preserving hashing for large-scale image retrieval," in *Proc. Int. Conf. Multimedia Retr. (ICMR)*, Ottawa, ON, Canada, 2019, pp. 192–196.

[36] A. Krizhevsky, "Learning multiple layers of features from tiny images," M.S. thesis, Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, Apr. 2009.

[37] L. Chi and X. Zhu, "Hashing techniques: A survey and taxonomy," *ACM Comput. Surv.*, vol. 50, no. 1, pp. 11:1–11:36, 2017.

[38] Y. Mu and S. Yan, "Non-metric locality-sensitive hashing," in *Proc. 24th 745 AAAI Conf. Artif. Intell. (AAAI)*, 2010, pp. 539–544.

[39] B. Kulis and K. Grauman, "Kernelized locality-sensitive hashing," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 6, pp. 1092–1104, Jun. 2012.

[40] I. T. Jolliffe, *Principal Component Analysis*. New York, NY, USA: Springer-Verlag, 1989.

[41] H. Jegou, M. Douze, C. Schmid, and P. Perez, "Aggregating local descriptors into a compact image representation," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 3304–3311.

[42] X. He and P. Niyogi, "Locality preserving projections," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2003, pp. 153–160.

[43] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, Aug. 1996.

[44] T. Dietterich, "Ensemble methods in machine learning," in *Proc. 1st Int. Workshop Multiple Classifier Syst.*, 2000, pp. 1–15.

[45] A. Krogh and J. Vedelsby, "Neural network ensembles, cross validation, and active learning," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 1995, pp. 231–238.

[46] N. Ueda and R. Nakano, "Generalization error of ensemble estimators," in *Proc. Int. Conf. Neural Netw.*, 1996, pp. 90–95.

[47] M. A. Carreira-Perpiñán and R. Raziperchikolaei, "An ensemble diversity approach to supervised binary hashing," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2016, pp. 757–765.

[48] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," in *Proc. 13th Int. Conf. Mach. Learn.*, 1996, pp. 148–156.

[49] 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.

[50] Y. Li, Y. Zha, H. Li, S. Zhang, K. Tao, and Y. Yuan, "Learning weighted hashing on local structured data," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 211–215.

[51] H. Li and Y. Li, "An ensemble hashing framework for fast image retrieval," in *Proc. Adv. Internetwork., Data Web Technol.*, 2017, pp. 167–177.

[52] P. Viola and M. J. Jones, "Robust real-time face detection," *Int. J. Comput. Vis.*, vol. 57, no. 2, pp. 137–154, May 2004.

[53] F. X. Yu, S. Kumar, Y. Gong, and S. Chang, "Circulant binary embedding," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2014, pp. 946–954.

[54] 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.

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

[56] K. Lin, J. Lu, C.-S. Chen, and J. Zhou, "Learning compact binary descriptors with unsupervised deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1183–1192.

[57] K. G. Dizaji, F. Zheng, N. S. Nourabadi, Y. Yang, C. Deng, and H. Huang, "Unsupervised deep generative adversarial hashing network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 3664–3673.

[58] J.-P. Heo, Y. Lee, J. He, S.-F. Chang, and S.-E. Yoon, "Spherical hashing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2957–2964.

[59] W. Kong and W. Li, "Isotropic hashing," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2012, pp. 1655–1663.

YUFEI ZHA received the Ph.D. degree in information and communication engineering from Airforce Engineer University, Xian, China, in 2009. He is currently an Associate Professor with the School of Computer Science, Northwestern Polytechnical University, Xian. His current research interests include object detection, visual tracking, and machine learning.

ZHULING QIU received the B.E. degree in automation from Xidian University, in 2018. He is currently pursuing the master's degree in information and communication engineering with Airforce Engineer University, Xi'an, China. His current research interests include deep learning, computer vision, and object tracking.

PENG ZHANG (Member, IEEE) received the B.E. degree from Xian Jiaotong University, China, in 2001, and the Ph.D. degree from Nanyang Technological University, Singapore, in 2011. He is currently a Professor with the School of Computer Science, Northwestern Polytechnical University, China. He has published more than 80 high-ranked international conference and journal papers. His current research interests include object detection and tracking, computer vision, and pattern recognition. He is a member of ACM.

WEI HUANG received the B.Eng. and M.Eng. degrees from the Harbin Institute of Technology, China, in 2004 and 2006, respectively, and the Ph.D. degree from Nanyang Technological University, Singapore, in 2011. He worked as a Postdoctoral Research Fellow at the University of California San Diego, USA, and the Agency for Science Technology and Research, Singapore, till 2012. He has been acting as a Principal Investigator in 14 national/provincial grants, including three NSFC grants and two NSF key grants in Jiangxi Province. He has published more than 80 academic journal/conference papers. His research interests mainly include machine learning, pattern recognition, image processing, and computer vision. He received the Jiangxi Provincial Natural Science Award, in 2018, the Most Interesting Paper Award of ICME-ASMMC, in 2016, and the Best Paper Award of MICCAI-MLMI, in 2010. He was entitled the Provincial Young Scientist of the Jiangxi Province, in 2015.