Hierarchical Hashing Learning for Image Set Classification

Yuan Sun, Xu Wang, Dezhong Peng, Zhenwen Ren, Xiaobo Shen

***Abstract*—With the development of video network, image set classification (ISC) has received a lot of attention and can be used for various practical applications, such as video based recognition, action recognition, and so on. Although the existing ISC methods have obtained promising performance, they often have extreme high complexity. Due to the superiority in storage space and complexity cost, learning to hash becomes a powerful solution scheme. However, existing hashing methods often ignore complex structural information and hierarchical semantics of the original features. They usually adopt a single-layer hashing strategy to transform high-dimensional data into short-length binary codes in one step. This sudden drop of dimension could result in the loss of advantageous discriminative information. In addition, they do not take full advantage of intrinsic semantic knowledge from whole gallery sets. To tackle these problems, in this paper, we propose a novel Hierarchical Hashing Learning (HHL) for ISC. Specifically, a coarse-to-fine hierarchical hashing scheme is proposed that utilizes a two-layer hash function to gradually refine the beneficial discriminative information in a layer-wise fashion. Besides, to alleviate the effects of redundant and corrupted features, we impose the $\ell_{2,1}$ norm on the layer-wise hash function. Moreover, we adopt a bidirectional semantic representation with the orthogonal constraint to keep intrinsic semantic information of all samples in whole image sets adequately. Comprehensive experiments demonstrate HHL acquires significant improvements in accuracy and running time. We will release the demo code on https://github.com/sunyuan-cs.**

***Index Terms*—image set classification, hierarchical hashing, bidirectional semantic representation.**

I. Introduction

WITH the rapid development of video equipment, a great deal of video sequences are viewed as image set. Thereinto, each sequence can be treated as a set of images. Image set classification (ISC) [1–4] has received widespread research due to its various fields applications in video-based surveillance and action recognition. Different from one-shot classification task, the main goal of ISC is to recognize each probe set from the known classes in gallery sets. Since multiple images from the same image set enjoy comprehensive complementary information and more intra-class variations of object, ISC has more promising performance in terms of overcoming image appearance variations. The key steps of the ISC are: (1) to model image set efficiently and take advantage of the comprehensive information of image set; (2) to formulate a distance criterion and accurately calculate the similarity. However, these large appearance variations also raise some great challenges, for example, lighting conditions, arbitrary poses, occlusion, and viewpoints).

For ISC tasks, theses existing methods obtain gratifying classification performance. With the increase of data volume, these ISC methods based on real-value representation can easily lead to extremely high running time and space complexity. Consequently, it is urgent to design an effective ISC method to achieve fast and accurate classification. Some studies [5–7] indicate that the hashing technique is a powerful scheme to support efficient storage and fast recognition for high-dimensional image data. Recently, there has been growing interest in learning to hash community [3]. The existing hashing methods usually are designed for the image retrieval and cross-modal retrieval tasks. Without loss of generality, hashing methods often project the feature data into low-dimensional binary hash codes, meanwhile preserving the feature similarity and structural information in Hamming space. Since hashing methods adopt binary codes to store high-dimensional data, space cost can be greatly reduced. For out-of-samples, it can generate new hash codes by the learned hash function, and then compute Hamming distance to achieve similarity search. Due to the XOR bit-wise operations, the retrieval process can be further accelerated. Some promising representative hashing methods include [8–10].

However, learning to hash usually is used for image retrieval or multimodal retrieval. In the field of ISC, it is less touched. Zhang et al [11] proposed to learn binary codes for each image in image set by maximizing inter-class and minimizing intra-class Hamming distances. Sun et al [12] proposed to learn binary codes of image set by feature and semantic views consensus. It is still an open question that how to reduce the information loss of hashing projection and preserve the similarity in whole image sets. Existing hashing methods often ignore complex structural information and hierarchical semantics of the original features, and adopt a single-layer hashing strategy to transform high-dimensional images into low-dimensional Hamming space in one step. To see the insight deeply, we provide more descriptions at subsection 'Motivation'. Based on our analysis, the sudden one-step dimension reduction could result in the important discriminative information loss. Moreover, they do not take full advantage of intrinsic semantic knowledge from whole gallery sets.

To balance the discriminative information extracting and preserving, we propose a **Hierarchical Hashing Learning**

Y. Sun, X. Wang and D. Peng are with College of Computer Science, Sichuan University, Chengdu, China, 610044. (e-mail: pengdz@scu.edu.cn)

Z. Ren is with Guangdong Laboratory of Artificial Intelligence and Digital Economy, Shenzhen, China, 518060, and the Department of National Defence Science and Technology, Southwest University of Science and Technology. (e-mail: rzw@njust.edu.cn)

X. Shen is with the College of Computer Science, Nanjing University of Science and Technology, Nanjing, China, 210094.

Corresponding authors: Dezhong Peng; Zhenwen Ren.

Manuscript received Nov. 30, 2022.

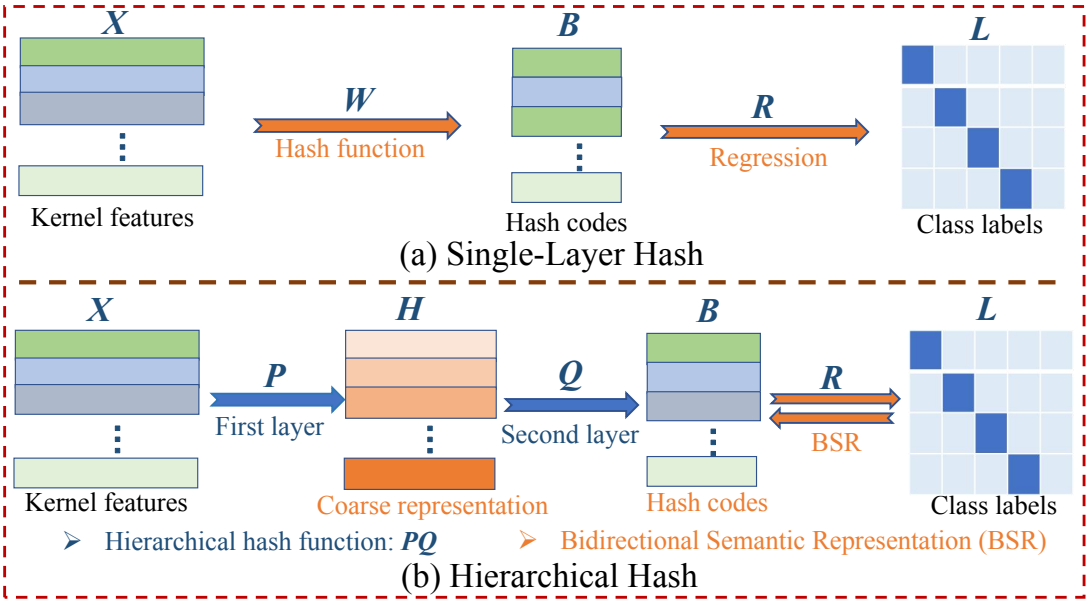


Fig. 1: (a) and (b) are the frameworks of single-layer hashing and hierarchical hashing, respectively. The former adopts a single-layer hash function to directly learn low-dimensional binary codes from high-dimensional kernel features, and then utilizes unidirection regression to preserve label information. The latter adopts a coarse-to-fine hierarchical hashing scheme that uses a two-layer hash function to refine beneficial discriminative information step by step. To be specific, the first layer hash function is used to learn coarse representation to gather beneficial information, and the second layer hash function is used to learn fine hash codes to enhance discrimination. Besides, bidirectional semantic representation is used to preserve similarity, thereby reducing the distribution difference between the feature and Hamming spaces.

(HHL) method for ISC tasks. Specifically, as shown in Fig. 1, we propose a coarse-to-fine hierarchical hashing scheme that uses two smaller hash mapping matrices with the $\ell_{2,1}$ norm constraint instead of a single-layer hashing to learn discriminative hash codes gradually. Then we propose a bidirectional semantic representation with the orthogonal constraint to express semantic similarity, thereby reducing the distribution difference between the feature and Hamming spaces. To summarize, this paper has the following main contributions:

- We propose an elegant HHL framework for fast ISC tasks. To the best of our knowledge, this work is the first to develop hierarchical hashing to boost the discrimination power of the learned hash codes
- To mitigate the loss of important information caused by a considerable drop in dimension, we propose a coarse-to-fine hierarchical hashing scheme that uses the two-layer hash function to gradually refine the relative important discriminative information.
- To preserve intrinsic similarity of all images in whole gallery sets as much as possible, we propose a bidirectional semantic representation strategy, thereby reducing the distribution difference between the feature and Hamming spaces.
- To efficiently solve the binary hierarchical hashing problem, we develop an iterative optimization algorithm. A plentiful experiments on three benchmark datasets demonstrate that HHL outperforms some state-of-the-art comparison methods.

II. Related Work

In this section, we review some ISC methods and image hashing methods.

A. Image set classification

Over the past few decades, the researchers proposes a mass of real-valued representation methods to handle ISC. The related traditional ISC methods mainly fall into the following categories. (1) Manifold: discriminant analysis on riemannian manifold of gaussian Distributions (DARG) [13], and multiple riemannian manifold-valued descriptors (MRMD) [14]; (2) Point-to-point: sparse approximated nearest points (SANP) [15], regularized nearest points (RNP) [16], and sparse projection learning (SPL) [17]; (3) Regression: dual linear regression classification (DLRC) [18], pairwise linear regression classification (PLRC) [19], and prototype discriminative learning (PDL) [20]; (4) Multiple kernel: multiple kernel dimensionality reduction (MKDR) [21]; (5) Binary representation: simultaneous feature and sample reduction (SFSR) [11]. Although most researchers use mathematical models to model each image set, deep learning also gradually apply to the ISC task due to the strong non-linear representation ability. For example, Huang et al proposed a deep network method on grassmann manifolds [22]. Wang et al [1] proposed a deep learning network on symmetric positive definite manifold.

B. Image hashing

Generally speaking, learning to hash is used for image retrieval task [23] or multi-modal retrieval [24, 25]. As shown in

Fig. 1, the existing hashing methods usually use a single-layer hash function to transform original high-dimensional data into a set of compact binary hash codes, meanwhile preserving the similarity of the original data in the Hamming space. Representative discrete hashing methods include SDH [26], COSDSH [27], FSDH [28], FSSH [29], SSLH [30], RSLH [31], and SCDH [32]. SDH learns the hash codes directly from the original image data via a hash function, and then reconstructs the label matrix from the hash codes. COSDSH tries to directly learn the hash codes under the supervision of semantic information by the sample strategy. FSDH learns hash codes via regressing the class label to preserve semantic similarities. FSSH uses a pre-computed intermediate term to avoid using the large similarity matrix in order to better learn hash codes. SSLH utilizes mutual regression to reduce quantitative loss, and then proposes a robust estimator to improve the robustness. RSLH learns short-length hash codes and preserves semantic information by mutual regression and asymmetric pairwise similarity. SCDH directly learns hash codes from the original samples via imposing strong constraints (*i.e.*, the bit balance and decorrelation). Recently, many advanced deep image hashing methods are proposed, for example, weighted generative adversarial networks (WeGAN) [33] utilizes the uncertain relationships between images and labels to improve the quality of binary codes.

More specifically, to achieve fast image retrieval, image hashing encodes all samples into more compact binary codes $\boldsymbol{B}$ in Hamming space. Without loss of generality, the hashing prototype can be defined as follows:

$$\min_{\boldsymbol{B},\boldsymbol{W}} \|\boldsymbol{B}-\boldsymbol{X}\boldsymbol{W}\|_F^2+\alpha\|\boldsymbol{W}\|_F^2 \quad s.t.\ \boldsymbol{B}\in\{-1,1\}^{n\times l} \tag{1}$$

where α is the regularization parameter for preventing trivial solution, $\boldsymbol{X}\in\mathbb{R}^{n\times d}$ is the original image feature, and $\boldsymbol{W}\in\mathbb{R}^{d\times l}$ is a single-layer hash function. According to relevant literatures [34], the quality of hash code can be improved by integrating bit balance and decorrelation constraints. Bit decorrelation $\boldsymbol{B}\boldsymbol{B}^T=n\boldsymbol{I}$ requires the hash bits to be uncorrelated, which encourages the generation of compact hash codes. Bit balance $\boldsymbol{B}\boldsymbol{1}=\boldsymbol{0}$ represents each bit has a half chance of $+1$ or -1. On one hand, it can avoid trivial solutions. On the other hand, according to hash theory [35], when hash bit is balanced, the entropy and the amount of information will reach a maximum, indicating the hash code is high quality. Thus, image hashing usually uses these two constraints to obtain more efficient hash codes.

$$\begin{aligned}&\min_{\boldsymbol{B},\boldsymbol{W}}\|\boldsymbol{B}-\boldsymbol{X}\boldsymbol{W}\|_F^2+\alpha\|\boldsymbol{W}\|_F^2\\&s.t.\ \boldsymbol{B}\in\{-1,1\}^{n\times l},\boldsymbol{B}\boldsymbol{1}=\boldsymbol{0},\boldsymbol{B}\boldsymbol{B}^T=n\boldsymbol{I}\end{aligned} \tag{2}$$

Since the hashing methods can be executed offline, and the kernel trick usually is used to capture non-linear structure among image samples in advance. Concretely, a commonly used RBF kernel strategy can be adopted to map original features into kernel features. Mathematically, each image can be denoted as follows

$$\phi(\boldsymbol{X}_{ij})=[exp(\frac{\|\boldsymbol{X}_{ij}-a_1\|_2^2}{-2\sigma^2}),...,exp(\frac{\|\boldsymbol{X}_{ij}-a_h\|_2^2}{-2\sigma^2})]^\top \tag{3}$$

where a is the anchor point, h is the number of anchors in total, and $\sigma=\frac{1}{nh}\sum_{i=1}^{n}\sum_{j=1}^{h}\|x_i-a_j\|_2$ is the kernel width. The d-dimension features can be mapped into h-dimension kernel features.

III. Proposed Method

We describe the proposed HHL the proposed HHL. First, we introduce the problem definition, notations and the motivation in Sections III-A and III-C. Then, Sections III-D to III-I give the framework, optimization, classification criterion, and some other analysis.

A. Problem definition

For ISC task, we give some important terminologies. Image set is collected from video frames or multiple unordered images with a certain degree of correlation. Each image usually appears with large intra-class variations. Gallery sets are used for training, which contain one or more image sets from each class. Probe set is used for testing, which contains multiple images from the same class. ISC aims to compute the similarities between probe set and gallery sets, thereby predicting the class of the probe set.

B. Notations

Throughout this paper, we denote the uppercase bold font characters as matrices and the lowercase bold font characters as vectors. For image set setting, we denote $\boldsymbol{X}=[\boldsymbol{X}_1,\boldsymbol{X}_2,\cdots,\boldsymbol{X}_k]^\top\in\mathbb{R}^{n\times d}$ as whole gallery sets consisted of k gallery sets from c classes, where $\boldsymbol{X}_i=[\boldsymbol{x}_{i1},\boldsymbol{x}_{i2},\cdots,\boldsymbol{x}_{iq}]^\top\in\mathbb{R}^{q\times d}$ represents the i-th gallery set and the dimension of each image sample is d. $\boldsymbol{X}_{ij}$ represents the j-th image from the i-th gallery set. The number of all image samples in whole gallery sets is $n=kq$. In addition, suppose probe set be $\boldsymbol{Y}=[\boldsymbol{y}_1,\boldsymbol{y}_2,\cdots,\boldsymbol{y}_m]^\top\in\mathbb{R}^{m\times d}$, where the number of all samples is m. Let $\boldsymbol{L}\in\{0,1\}^{n\times c}$ is the ground-truth label of all image samples in whole gallery sets. Similarly, we also use the RBF strategy to extract nonlinear kernel features [36–38] in advance. For the sake of convenience in writing, we use $\boldsymbol{V}$ to represent $\phi(\boldsymbol{X})$.

C. Motivation

As previously mentioned, the existing hashing methods often learn a single-layer hash function to transform high-dimensional data with size $\mathbb{R}^{n\times h}$ into binary codes with size $\mathbb{R}^{n\times l}$ in one step. We notice that the dimension suddenly drops from h to l, which could result in a large volume of the kernel discriminative information loss and make the error of binary representation magnified. Besides, the original features usually include complex structural information and hierarchical semantics. They are difficult to extract by only adopting the single-layer hashing scheme.

We give a simple introduction on the discriminative information loss of existing hashing methods in the progress of learning to binary codes. Since the essence of learning to hash is subspace learning and binary codes are orthogonal,

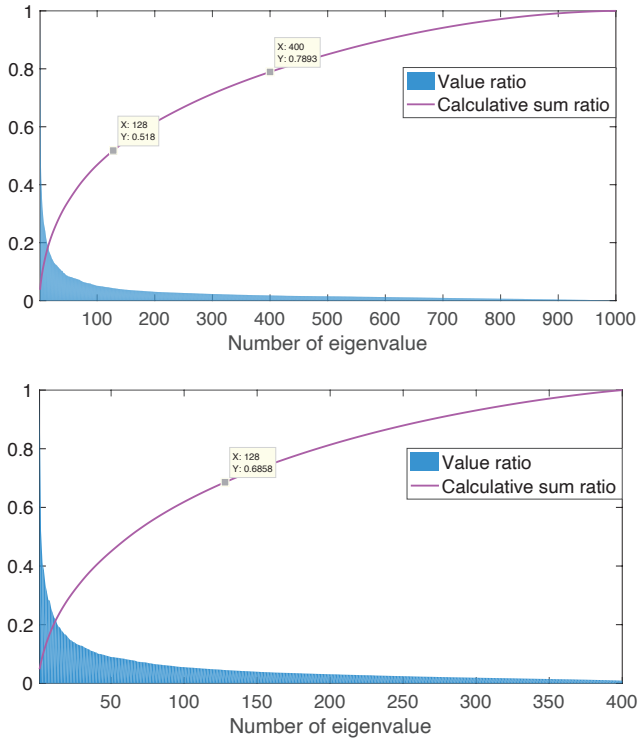


Fig. 2: Eigenvalue distribution of the kernel matrix on YTC. First, we sort the eigenvalues from large to small. Then, we denote the blue curve as the ratio of each eigenvalue to the first eigenvalue, and the purple curve as the ratio of the sum of the ordered topN eigenvalues to the total eigenvalues. For the first-layer, the hash matrix only keeps 51.8% kernel information when setting dimension as 128 (*i.e.*, $l = 128$). For the second-layer, we first map kernel features into 400-dimension representation and then map ones into 128-dimension binary codes. The hash matrix can keep $78.93\% \times 68.58\% = 54.13\%$ kernel information. Compared with single-layer hash mapping, two-layer hash mapping preserves more kernel information (*i.e.*, $54.13\% > 51.8\%$).

we use singular value decomposition (SVD) to show eigenvalue distribution. According to the literature [39], the larger the eigenvalue, the more discriminative information carried by the eigenvector. We hope that l bits binary codes can preserve most of the discriminative information by choosing the eigenvectors corresponding to top-l eigenvalues. Ideally, eigenvalues except the top-l ones are zeros. We thus roughly estimate the total amount of discriminative information in the corresponding eigenvector by the eigenvalue. Motivated by the deep architecture [40, 41], to effectively avoid the abrupt drop of dimension and alleviate the discriminative information loss, we adopt a coarse-to-fine hierarchical hashing scheme that utilizes a progressive way to learn a multi-level hash function. That is, we employ multiple intermediary matrices to extract the discriminative information from kernel features. In this paper, we only focus on the two-layer hash function. We draw a brief illustration on YTC dataset in Fig. 2, which shows the eigenvalue distributions of the first-layer representation and the second-layer ones. From the results, the sudden one-step dimension reduction could result in the loss of important discriminative information. Furthermore, such hierarchical hashing scheme can preserve the main information of kernel features and reduce the information loss.

D. Model framework

We propose a novel hierarchical hashing learning method, including two components: the hierarchical hashing strategy and the bidirectional semantic representation.

(1) Hierarchical hashing strategy

We propose a coarse-to-fine hierarchical hashing scheme to learn discriminative hash codes. More precisely, we use two smaller hash mapping matrix $\mathbf{W} = \mathbf{PQ}$, where $\mathbf{P} \in \mathbb{R}^{h\times r}$, $\mathbf{Q} \in \mathbb{R}^{r\times l}$, $l < r < h$, and r is the dimension of discriminative details. $\boldsymbol{P}$ is first-layer hash mapping matrix used to gather important discriminant information. $\boldsymbol{Q}$ is viewed as second-layer hash mapping matrix used to learn discrimination hash codes. In the first-layer hash mapping stage, the redundant features can be refined and squeezed out as much as possible, thereby improving the quality of binary codes in the second-layer hash mapping stage. Hence, $\mathbf{PQ}$ acts as a hierarchical hash mapping matrix, which can promote the corresponding hash representation error to be minimized. Note here, although a new parameter r is introduced, r is easy to tune since r is a positive integer greater than l, and has explicit meaning, *i.e.*, r is the number of selected important features for extracting recognition information. r can be varied from the range of (l, h). Due to $\mathrm{rank}(\mathbf{P}) \leq r < h$ and $\mathrm{rank}(\mathbf{Q}) \leq l < r$, $\mathbf{PQ}$ also has latent low-rank property. Overall, the proposed strategy iteratively extracts the beneficial feature information into smaller matrices via two steps, and learns the final hash codes $\boldsymbol{B}$. The optimization problem can be written as follows:

$$\begin{aligned} &\min_{\boldsymbol{B},\boldsymbol{P},\boldsymbol{Q}} \|\boldsymbol{B} - \boldsymbol{VPQ}\|_F^2 + \alpha\|\boldsymbol{PQ}\|_F^2 \\ &s.t.\ \boldsymbol{B} \in \{-1, 1\}^{n\times l} \end{aligned} \tag{4}$$

Moreover, in a real world application, due to redundant noisy and outliers, these troublesome features easily affect binary codes in one-step process. Inspired by [42], to eliminate the redundant and corrupted features, we consider that the two-layer hash function $\boldsymbol{PQ}$ should be row-sparsity. Because $\ell_{2,1}$ norm [42] has the excellent row-sparsity property, we impose the regularization term for $\boldsymbol{PQ}$, that is, $\|\boldsymbol{PQ}\|_{2,1}$. Compared with the $\mathcal{F}$-norm, the negative effects of noisy or corrupted data can be alleviated. Besides, we impose bit balance and bit decorrelatation constraints on hash codes. The problem can be transformed as

$$\begin{aligned} &\min_{\boldsymbol{B},\boldsymbol{P},\boldsymbol{Q}} \|\boldsymbol{B} - \boldsymbol{VPQ}\|_F^2 + \alpha\|\boldsymbol{PQ}\|_{2,1}^2 \\ &s.t.\ \boldsymbol{B} \in \{-1, 1\}^{n\times l}, \boldsymbol{B1} = \boldsymbol{0}, \boldsymbol{BB}^T = n\boldsymbol{I} \end{aligned} \tag{5}$$

Because of the $\ell_{2,1}$ norm constraint, problem (5) has the ability to adaptively assign large weights, such that the extractive important information can be mapped into hash codes. Moreover, the $\ell_{2,1}$ norm constraint can avoid the trivial solution, that is, preventing $\boldsymbol{PQ} = \boldsymbol{I}$. The hierarchical hash mapping function involved $\ell_{2,1}$ norm assures the spareness.

(2) Bidirectional semantic representation

Our goal is to keep the underlying geometrical structure information from all images in Hamming space. Since both hash codes and label are binary, we adopt the linear auto-encoder scheme that converts the two to each other to preserve the similarity information. Hence, the problem can be written as follows:

$$\begin{aligned}\min_{\boldsymbol{B},\boldsymbol{R},\boldsymbol{H}} \ & \|\boldsymbol{L}-\boldsymbol{B}\boldsymbol{R}\|_F^2+\alpha\|\boldsymbol{B}-\boldsymbol{L}\boldsymbol{H}\|_F^2 \\ & +\lambda(\|\boldsymbol{R}\|_F^2+\|\boldsymbol{H}\|_F^2) \quad s.t.\ \boldsymbol{B}\in\{-1,1\}^{n\times l}\end{aligned} \tag{6}$$

where $\boldsymbol{R}$ and $\boldsymbol{H}$ are the regression matrices, and λ is the regularization parameter. To preserve the semantic similarity information well (we conduct some analysis in subsection 'Similarity preserving analysis'), we use the same regression matrix (*i.e.*, $\boldsymbol{R}=\boldsymbol{H}^\top$) between hash codes and semantic label. Hence, the semantic representation error produced in the bidirectional regression can be minimized. This bidirectional regression strategy could make the model more stable and precise, which can express as:

$$\begin{aligned}\min_{\boldsymbol{B},\boldsymbol{R}} & \|\boldsymbol{L}-\boldsymbol{B}\boldsymbol{R}\|_F^2+\lambda\|\boldsymbol{B}-\boldsymbol{L}\boldsymbol{R}^\top\|_F^2 \\ & s.t.\ \boldsymbol{B}\in\{-1,1\}^{n\times l}\end{aligned} \tag{7}$$

The two processes are mutually promoted, which can generate hash codes from class labels, as well as reconstruct the label matrix. Therefore, the intrinsic semantic structure from original data and discrimination property from the label information can be better inherited. In order to simplify problem (7) and avoid the introduction of a new balance parameter λ, we further impose the orthogonal constraint to ensure bidirectional transformation and reconstruction mechanism (satisfy both $\|\boldsymbol{L}-\boldsymbol{B}\boldsymbol{R}\|_F^2$ and $\|\boldsymbol{B}-\boldsymbol{L}\boldsymbol{R}^\top\|_F^2$). The problem can further simplify as:

$$\min_{\boldsymbol{B},\boldsymbol{R}} \|\boldsymbol{L}-\boldsymbol{B}\boldsymbol{R}\|_F^2 \quad s.t.\ \boldsymbol{R}\boldsymbol{R}^\top=\boldsymbol{I},\boldsymbol{B}\in\{-1,1\}^{n\times l} \tag{8}$$

(3) Overall HHL model

Utilizing the aforementioned insights into a unified objective function, the proposed HHL can be written as:

$$\begin{aligned}\min_{\boldsymbol{B},\boldsymbol{P},\boldsymbol{Q},\boldsymbol{R}} & \|\boldsymbol{B}-\boldsymbol{V}\boldsymbol{P}\boldsymbol{Q}\|_F^2+\alpha\|\boldsymbol{P}\boldsymbol{Q}\|_{2,1}^2+\beta\|\boldsymbol{L}-\boldsymbol{B}\boldsymbol{R}\|_F^2 \\ & s.t.\ \boldsymbol{R}\boldsymbol{R}^\top=\boldsymbol{I},\boldsymbol{B}\in\{-1,1\}^{n\times l} \\ & \boldsymbol{B}\mathbf{1}=\mathbf{0},\boldsymbol{B}\boldsymbol{B}^T=n\boldsymbol{I}\end{aligned} \tag{9}$$

To summarize, problem (9) has the following properties:

- The first term iteratively extracts the beneficial feature information via a two-layer mapping function to refine the important discriminative information. More importantly, the hash mapping matrix $\boldsymbol{W}=\boldsymbol{P}\boldsymbol{Q}$ has the hierarchy and latent low-rank property, hence the beneficial feature information can be extracted well.
- The second term encourages the hash mapping matrix $\boldsymbol{P}\boldsymbol{Q}$ to be row-sparsity, thereby filtering out or weakening the redundant and corrupted features.
- The third term utilizes orthogonal constraint to build linear bidirectional regression, which can promote hash codes inheriting intrinsic semantic structure and discrimination property from the label information.

E. Optimization

In this subsection, we efficiently use the iterative optimization strategy to solve problem (9). To be specific, we fix other variables, and then solve the variable to be solved and ensure that it is globally optimal in each step.

▶ **Update $\boldsymbol{P}\boldsymbol{Q}$-subproblem:** Fixing the variables $\boldsymbol{W}$ and $\boldsymbol{B}$, $\boldsymbol{P}\boldsymbol{Q}$-subproblem of (9) can be depicted as

$$\min_{\boldsymbol{P},\boldsymbol{Q}} \|\boldsymbol{B}-\boldsymbol{V}\boldsymbol{P}\boldsymbol{Q}\|_F^2+\alpha\|\boldsymbol{P}\boldsymbol{Q}\|_{2,1}^2 \tag{10}$$

With simple algebraic manipulations, problem (9) can be transformed as

$$\min_{\boldsymbol{P},\boldsymbol{Q}} \|\boldsymbol{B}-\boldsymbol{V}\boldsymbol{P}\boldsymbol{Q}\|_F^2+\alpha\,\mathrm{Tr}(\boldsymbol{Q}^\top\boldsymbol{P}^\top\boldsymbol{M}\boldsymbol{P}\boldsymbol{Q}) \tag{11}$$

where $\boldsymbol{M}$ is an auxiliary diagonal matrix, and m_{ii} is represented as

$$m_{ii}=\frac{1}{2\|(\boldsymbol{P}\boldsymbol{Q})_i\|_2},i=1,\cdots,h \tag{12}$$

Then, we solve the $\boldsymbol{P}$-subproblem and $\boldsymbol{Q}$-subproblem from problem (11).

(1) Update $\boldsymbol{Q}$-subproblem: By fixing $\boldsymbol{P}$ in problem (11), we take its derivative with respect to $\boldsymbol{Q}$ to zero. The $\boldsymbol{Q}$ can be obtained via

$$\boldsymbol{Q}=(\boldsymbol{P}^\top\boldsymbol{S}_t\boldsymbol{P})^{-1}\boldsymbol{P}^\top\boldsymbol{V}^\top\boldsymbol{B} \tag{13}$$

where $\boldsymbol{S}_t=\boldsymbol{V}^\top\boldsymbol{V}+\alpha\boldsymbol{M}$.

(2) Update $\boldsymbol{P}$-subproblem: By fixing $\boldsymbol{Q}$, problem (11) can be transformed as

$$\begin{aligned}\min_{\boldsymbol{P}} \mathrm{Tr}(&\boldsymbol{B}^\top\boldsymbol{B}-\boldsymbol{B}^\top\boldsymbol{V}\boldsymbol{P}\boldsymbol{Q}-\boldsymbol{Q}\boldsymbol{P}\boldsymbol{V}\boldsymbol{B} \\ &+\boldsymbol{Q}^\top\boldsymbol{P}^\top\boldsymbol{V}^\top\boldsymbol{V}\boldsymbol{P}\boldsymbol{Q})+\alpha\,\mathrm{Tr}(\boldsymbol{Q}^\top\boldsymbol{P}^\top\boldsymbol{M}\boldsymbol{P}\boldsymbol{Q})\end{aligned} \tag{14}$$

Mathematically, problem (14) is equivalent to

$$\begin{aligned}\min_{\boldsymbol{P}} &\mathrm{Tr}(\boldsymbol{B}^\top\boldsymbol{B}-\boldsymbol{B}^\top\boldsymbol{V}\boldsymbol{P}\boldsymbol{Q}-\boldsymbol{Q}\boldsymbol{P}\boldsymbol{V}\boldsymbol{B}) \\ &+\mathrm{Tr}(\boldsymbol{Q}^\top\boldsymbol{P}^\top(\boldsymbol{V}^\top\boldsymbol{V}+\alpha\boldsymbol{M})\boldsymbol{P}\boldsymbol{Q})\end{aligned} \tag{15}$$

By introducing $\boldsymbol{S}_t$ into problem (15), then it becomes

$$\min_{\boldsymbol{P}} \mathrm{Tr}(-2\boldsymbol{B}^\top\boldsymbol{V}\boldsymbol{P}\boldsymbol{Q})+\mathrm{Tr}(\boldsymbol{Q}^\top\boldsymbol{P}^\top\boldsymbol{S}_t\boldsymbol{P}\boldsymbol{Q}) \tag{16}$$

By substituting $\boldsymbol{Q}^*$ into problem (16), we have

$$\begin{aligned}&\min_{\boldsymbol{P}} \mathrm{Tr}(-2\boldsymbol{B}^\top\boldsymbol{V}\boldsymbol{P}\boldsymbol{Q})+\mathrm{Tr}(\boldsymbol{P}^\top\boldsymbol{S}_t\boldsymbol{P}\boldsymbol{Q}\boldsymbol{Q}^\top) \\ &=\max_{\boldsymbol{P}} \mathrm{Tr}(\boldsymbol{P}^\top\boldsymbol{S}_t\boldsymbol{P})^{-1}\boldsymbol{P}^\top\boldsymbol{S}_b\boldsymbol{P}\end{aligned} \tag{17}$$

where $\boldsymbol{S}_b=\boldsymbol{V}^\top\boldsymbol{B}\boldsymbol{B}^\top\boldsymbol{V}$.

The two auxiliary variables ($\boldsymbol{S}_b$ and $\boldsymbol{S}_t$) in problem (17) can be equivalent to the between-class and within-class scatter matrices, respectively, used in LDA method [7]. Therefore, the optimal solution $\boldsymbol{P}$ contains the eigenvectors corresponding to the top r eigenvalues of $\boldsymbol{S}_t^{-1}\boldsymbol{S}_b$. Note here, we compute $\boldsymbol{V}^\top\boldsymbol{V}$ in advance for $\boldsymbol{S}_t$.

▶ **Update $\boldsymbol{R}$-subproblem:** Fixing the variables $\boldsymbol{P}$, $\boldsymbol{Q}$ and $\boldsymbol{B}$, $\boldsymbol{R}$-subproblem of (9) can be reduced as

$$\min_{\boldsymbol{R}} \beta\|\boldsymbol{L}-\boldsymbol{B}\boldsymbol{R}\|_F^2 \quad s.t.\ \boldsymbol{R}\boldsymbol{R}^\top=\boldsymbol{I} \tag{18}$$

We can compute the SVD of $\boldsymbol{B}^\top\boldsymbol{L}$ (*i.e.*, $\boldsymbol{B}^\top\boldsymbol{L}=\boldsymbol{U}\boldsymbol{S}\boldsymbol{D}^\top$), and we can obtain $\boldsymbol{R}=\boldsymbol{U}\boldsymbol{D}^\top$.

► **Update B-subproblem:** Fixing the variables $\boldsymbol{P}$, $\boldsymbol{Q}$ and $\boldsymbol{W}$, we can obtain

$$\begin{aligned} &\min_{\boldsymbol{B}} \|\boldsymbol{B}-\boldsymbol{V}\boldsymbol{P}\boldsymbol{Q}\|_F^2+\beta\|\boldsymbol{L}-\boldsymbol{B}\boldsymbol{R}\|_F^2 \\ &s.t.\ \boldsymbol{B}\in\{-1,1\}^{n\times l}, \boldsymbol{B}\mathbf{1}=\mathbf{0}, \boldsymbol{B}\boldsymbol{B}^T=n\boldsymbol{I} \end{aligned} \tag{19}$$

Adopting matrix manipulation, problem (19) can further transformed as

$$\begin{aligned} &\max_{\boldsymbol{B}} \boldsymbol{B}^\top(\boldsymbol{V}\boldsymbol{P}\boldsymbol{Q}+\beta\boldsymbol{L}\boldsymbol{R}^\top) \\ &s.t.\ \boldsymbol{B}\in\{-1,1\}^{n\times l}, \boldsymbol{B}\mathbf{1}=\mathbf{0}, \boldsymbol{B}\boldsymbol{B}^T=n\boldsymbol{I} \end{aligned} \tag{20}$$

Let $\boldsymbol{E}=\boldsymbol{V}\boldsymbol{P}\boldsymbol{Q}+\beta\boldsymbol{L}\boldsymbol{R}^\top$, and we can rewritten into

$$\begin{aligned} &\max_{\boldsymbol{B}} \boldsymbol{B}^\top\boldsymbol{E} \quad s.t.\ \boldsymbol{B}\in\{-1,1\}^{n\times l}, \\ &\boldsymbol{B}\mathbf{1}=\mathbf{0}, \boldsymbol{B}\boldsymbol{B}^T=n\boldsymbol{I} \end{aligned} \tag{21}$$

Afterwards, we can update $\boldsymbol{B}$ through Theorem 1.

Theorem 1. [32] *Given the optimization problem:*

$$\max_{\boldsymbol{B}} \boldsymbol{B}^\top\boldsymbol{E} \quad s.t. \boldsymbol{B}\mathbf{1}_n=\mathbf{0}_l, \boldsymbol{B}\boldsymbol{B}^\top=n\boldsymbol{I}_l \tag{22}$$

$\boldsymbol{B}$ can be solved by

$$\boldsymbol{B}=\sqrt{n}[\boldsymbol{U},\widetilde{\boldsymbol{U}}][\boldsymbol{D},\widetilde{\boldsymbol{D}}]^T \tag{23}$$

Here, $\boldsymbol{U}=[\boldsymbol{U}_1,\boldsymbol{U}_2,...,\boldsymbol{U}_{\hat{l}}]$ and $\boldsymbol{D}=[\boldsymbol{D}_1,\boldsymbol{D}_2,...,\boldsymbol{D}_{\hat{l}}]$ are calculated by the SVD of $\boldsymbol{J}\boldsymbol{E}$, where $\boldsymbol{J}=\boldsymbol{I}_n-\frac{1}{n}\mathbf{1}_n\mathbf{1}_n^\top$, i.e.,

$$\boldsymbol{J}\boldsymbol{E}=\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{U}^\top=\sum\nolimits_{i=1}^{\hat{l}}\sigma_i u_l v_l^T \tag{24}$$

where eigenvalues is ordered by $0\geq\sigma_{\hat{l}}\geq\cdots\geq\sigma_1$. $\widetilde{\boldsymbol{U}}\in R^{n\times(l-\hat{l})}$ and $\widetilde{\boldsymbol{V}}\in R^{l\times(l-\hat{l})}$ can be obtained by the Gram-Schmidt process. Note that $\widetilde{\boldsymbol{U}}$ and $\widetilde{\boldsymbol{D}}$ will be empty if $\hat{l}=l$. The proof process is shown in .

Algorithm 1 Hierarchical Hashing Learning (HHL)

Input: Gallery sets $\boldsymbol{X}$ with label matrix $\boldsymbol{L}$, probe set $\boldsymbol{Y}$, and the number of anchors h.

Parameter: α, β, and l;
Initialize $\boldsymbol{W}$, $\boldsymbol{P}$, $\boldsymbol{Q}$, and $\boldsymbol{B}$;
Obtain kernel features $\boldsymbol{V}$ by kernel trick;
repeat
  Calculate $\boldsymbol{Q}$ via performing Eq.(13);
  Calculate $\boldsymbol{P}$ via performing Eq.(17);
  Calculate $\boldsymbol{R}$ via performing Eq.(18);
  Calculate $\boldsymbol{B}$ via performing Eq.(21);
until Satisfy the stop criteria $\|\boldsymbol{B}^{t+1}-\boldsymbol{B}^t\|_F^2/\|\boldsymbol{B}^t\|_F^2\leq\epsilon$ or reach the the number of maximum iteration;
Obtain hash codes $\boldsymbol{B}^Y$ of probe set $\boldsymbol{Y}^k$ via Eq.(25);

Output: Perform image set classification.

F. Classification criterion

In classification phase, when the new probe set $\boldsymbol{Y}$ arrives, we learn binary codes $\boldsymbol{B}^Y$ via the two-layer hash function $\boldsymbol{P}\boldsymbol{Q}$ as follows

$$\boldsymbol{B}^Y=\boldsymbol{Y}^k\boldsymbol{P}\boldsymbol{Q} \tag{25}$$

where $\boldsymbol{Y}^k$ is kernel features. After the hash codes of gallery sets and probe set (*i.e.*, $\boldsymbol{B}$ and $\boldsymbol{B}^Y$) are learned, we compute the Hamming distance between them, and assign label of probe set as the class corresponding to the minimum Hamming distance. Afterwards, each image for probe set casts one only vote and the class corresponding to the gallery set with the most votes is the label of the probe set. The progress of our proposed HHL is summarized in Algorithm 1.

G. Similarity preserving analysis

To show the similarity preserving of our HHL, we conduct conduct the theoretical analysis. The semantic similarity preserving is usually written as follows

$$\min_{\boldsymbol{B}} \|\boldsymbol{B}\boldsymbol{B}^\top-\boldsymbol{S}\|^2 \quad s.t. \boldsymbol{B}\in\{-1,+1\}^{n\times l} \tag{26}$$

Although the similarity preserving has achieved the promising performance, the paired similarity matrix $\boldsymbol{S}$ with size $n\times n$ may lead to expensive complexity cost in the training process, and the asymmetric optimization scheme will lead to higher quantitative loss. We thus use inner product of class labels to replace $\boldsymbol{S}$, and the problem (26) can be further transformed as

$$\min_{\boldsymbol{B}} \|\boldsymbol{B}\boldsymbol{B}^\top-\boldsymbol{S}\|^2=\|\boldsymbol{B}\boldsymbol{B}^\top-\boldsymbol{L}\boldsymbol{L}^\top\|^2 \tag{27}$$

By simple matrix operations, we can obtain

$$\boldsymbol{L}^\top\boldsymbol{B}\boldsymbol{B}^\top\boldsymbol{B}=\boldsymbol{L}^\top\boldsymbol{L}\boldsymbol{L}^\top\boldsymbol{B} \tag{28}$$

Whereupon, we can derive that

$$\boldsymbol{L}^\top\boldsymbol{B}(\boldsymbol{B}^\top\boldsymbol{B}+\lambda\boldsymbol{I})=(\boldsymbol{L}^\top\boldsymbol{L}+\lambda\boldsymbol{I})\boldsymbol{L}^\top\boldsymbol{B} \tag{29}$$

Then, we can further obtain

$$\boldsymbol{L}^\top\boldsymbol{B}(\boldsymbol{B}^\top\boldsymbol{B}+\lambda\boldsymbol{I})^{-1}=(\boldsymbol{L}^\top\boldsymbol{L}+\lambda\boldsymbol{I})^{-1}\boldsymbol{L}^\top\boldsymbol{B} \tag{30}$$

Review the problem (6), we observe that the left and right sides of the problem (30) are exactly the solutions of their hash functions. We can obtain

$$\begin{cases} \boldsymbol{R}=\boldsymbol{L}^\top\boldsymbol{B}(\boldsymbol{B}^\top\boldsymbol{B}+\lambda\boldsymbol{I})^{-1} \\ \boldsymbol{H}^\top=(\boldsymbol{L}^\top\boldsymbol{L}+\lambda\boldsymbol{I})^{-1}\boldsymbol{L}^\top\boldsymbol{B} \\ \boldsymbol{R}=\boldsymbol{H}^\top \end{cases} \tag{31}$$

Hence, the similarity preserving implies that both $\|\boldsymbol{L}-\boldsymbol{B}\boldsymbol{R}\|^2$ and $\|\boldsymbol{B}-\boldsymbol{L}\boldsymbol{R}\|^2$ are satisfied. Obviously, if problem (8) can be satisfied, we can easily get problem (26). In other words, the proposed bidirectional semantic representation strategy can preserve semantic similarity well, while both high complexity and the asymmetric computation process are avoided.

H. Computational complexity analysis

The four constants of our HHL, n, h, r, and l, are the number of samples in whole gallery sets, the total of anchors, the dimensionality of intermediate state, and the hash length. The computational complexity in Algorithm 1 depends on four subproblems alternating minimization, including $\boldsymbol{P}$-subproblem, $\boldsymbol{Q}$-subproblem, $\boldsymbol{W}$-subproblem, and $\boldsymbol{B}$-subproblem, which are repeated until satisfying the convergence stop criterion. Specifically, the updating of $\boldsymbol{P}$, $\boldsymbol{Q}$, $\boldsymbol{W}$, and $\boldsymbol{B}$ have the computational complexity of $\mathcal{O}(h^3)$, $\mathcal{O}(r^3)$, $\mathcal{O}(l^2n)$ and $\mathcal{O}(l^2n)$, respectively. Since $h<n$ and $l<r<<n$, the final computational complexity of Algorithm 1 approximates to $\mathcal{O}(n)$. It indicates that HHL is scalable to larger datasets.

I. Convergence analysis

The convergence analysis of the proposed HHL is given as follows. For ease of expression, we formulate $\mathcal{L}(\boldsymbol{P}, \boldsymbol{Q}, \boldsymbol{R}, \boldsymbol{B})$ as problem (9). According to the previous solution analysis, it is obvious that

$$\begin{aligned} &\mathcal{L}(\boldsymbol{Q}^{t+1}, \boldsymbol{P}^{t+1}, \boldsymbol{R}^{t+1}, \boldsymbol{B}^{t+1}) \\ &\leq \mathcal{L}(\boldsymbol{Q}^{t+1}, \boldsymbol{P}^{t+1}, \boldsymbol{R}^{t+1}, \boldsymbol{B}^{t}) \\ &\cdots \\ &\leq \mathcal{L}(\boldsymbol{Q}^{t}, \boldsymbol{P}^{t}, \boldsymbol{R}^{t}, \boldsymbol{B}^{t}) \end{aligned} \tag{32}$$

where superscript t represents the t-th iteration optimization. The objective value monotonically decreases progressively at each iteration. Each subproblem is all convex and has the closed-form solution. Based on the bounded monotone convergence theorem, our HHL is convergent. In Section IV-F, we experimentally demonstrate the convergence.

IV. Experiment

To evaluate the recognition performance of our proposed HHL, we conduct enormous experiments for ISC tasks on Mac with i7 and 16GB RAM. We record the best average classification accuracies and standard deviation to evaluate performance. The bold indicates the best result.

A. Image set datasets

For ISC tasks, we perform some experiments on three benchmark image set datasets, including Honda, YTC, ETH-80. These image sets contain high intra-class variations with expression deformations, illumination variations, pose, etc. To be fair, we take five-fold cross validation. The three datasets are introduced in detail as follows:

Honda dataset is collected in a controlled indoor environment, which has 59 video sequences from 20 subjects. There are about 12-645 images in each sequence, which have various variations contained sport speeds, expressions and rotations. Following [43], we resize each image as 20×20 and adopt to histogram equalization. For experiment setting, we randomly choose an image from each subject as a gallery set and the remainder as probe sets. Since the size limitation of image sets from each class, we do not perform muiti-fold cross validation, but repeat the experiment five times.

YTC dataset is collected under unconstrained reality environments, which has 81973 images from 1910 sequences of 47 subjects. Since each set enjoys large expression and pose variations, it is very challenging to recognize the identity of probe set. Following [2], we resize all images into 30×30 and extract LBP features. We randomly select 3 sets from 9 sets of each subject as gallery sets and the others as probe sets.

ETH-80 dataset has 3280 images consisted of 80 objects from 8 classes and each set has 41 images from different views. Each image is resized into 10×10, and is converted to grayscale. Following [44], we resieze all images into 10×10 and extract grayscale. We choose 5 classes of image sets as gallery sets randomly, and the rest sets as probe sets.

B. Comparison methods

We compare our HHL with a large number of state-of-the-art methods: **some shallow ISC methods** (SANP [15], RNP [16], DLRC [18], PLRC [19], DARG [13], PDL [20], MKDR [21], SPL [17], and MRMD [21]); and **some hashing methods** (SDH [26], COSDSH [27], FSDH [28], FSSH [29], SSLH [30], RSLH [31], SCDH [32], and POPSH [45]); and **some deep learning ISC methods** (SPDNet [22], SPDNet [46], GEMKML [47], and SymNet [1]). For a fair comparison, the parameters involved in these comparison methods have been carefully adjusted according to the recommendations of the respective authors.

C. Comparison with shallow ISC methods

For fairness, we follow the shallow ISC common setup. That is, we set 50, 100, and 200 frames on the Honda and YTC datasets, respectively. And we set 15, 25, and 41 frames on the ETH-80 dataset. To obtain the best result, we choose 128 bits codes. The average classification accuracies and the standard deviations are given in Table I. Through analyzing these tables, we can achieve some important conclusions:

- On the three image set datasets, we find that our proposed HHL has no short board. Compared with traditional shallow ISC methods, the performance improvements of HHL are significant in all of the cases (three different frames), which demonstrates its performance advantages. As the number of frames increasing, the classification performance of all methods improves greatly, which shows large size image sets can provide complementary discriminant information.
- On the Honda dataset collected in a controlled environment, the difficulty of ISC is relatively low. According to the experiment results, all of the comparison methods achieved satisfactory results, especially when the number of frames is 200, several methods achieved 100% recognition rate. It is worth mentioning that HHL achieves the best performance (up to 100% recognition rate) in all of the cases. Our HHL achieves such exciting performance up to 100% recognition rate. This may be because that Honda is the low difficulty dataset and the proposed can improve the quality of hash codes.
- On the more challenging YTC dataset, collected in an unconstrained real-world environment, HHL still outperforms these competitors. Experiment results indicate that HHL can filter out or weaken redundant and corrupted features. It is gratifying that HHL obtains the best performance in different size. This shows that the hierarchical hashing strategy can enhance discrimination of hash codes, and effectively resist noise and even outliers
- Since the ETH-80 dataset has much few images in each set, large appearance and view angle variations, the recognition performance of all methods is not very satisfactory. We observe that HHL gains best classification accuracies, compared with all competitors in small size image sets (frames $[15, 25, 41]$).

TABLE I: Classification performance (%) compared with traditional ISC methods on the first three datasets.

Method	Honda			YTC			ETH80		
	Frames 50	Frames 100	Frames 200	Frames 50	Frames 100	Frames 200	Frames 15	Frames 25	Frames 41
SANP [15]	84.6 ± 2.6	92.3 ± 1.2	94.9 ± 0.6	56.7 ± 5.5	61.9 ± 8.1	65.4 ± 6.8	62.5 ± 2.5	63.0 ± 4.1	65.0 ± 3.9
RNP [16]	87.2 ± 3.1	94.9 ± 1.1	97.4 ± 1.2	58.4 ± 6.9	63.2 ± 8.4	65.4 ± 7.2	64.5 ± 5.7	65.0 ± 8.2	69.5 ± 5.7
DLRC [18]	88.2 ± 2.7	90.5 ± 2.1	84.1 ± 1.5	58.9 ± 8.6	61.4 ± 7.1	67.3 ± 7.6	64.0 ± 6.1	66.5 ± 5.8	75.5 ± 6.8
PLRC [19]	87.2 ± 2.5	97.4 ± 1.7	100.0 ± 0.0	61.7 ± 8.2	65.6 ± 7.9	66.8 ± 7.5	73.5 ± 5.1	76.7 ± 4.6	79.7 ± 6.5
DARG [13]	94.9 ± 1.5	97.4 ± 1.6	100.0 ± 0.0	66.4 ± 6.4	66.6 ± 7.6	67.0 ± 7.1	70.0 ± 7.1	76.7 ± 6.8	79.4 ± 7.3
PDL [20]	87.2 ± 2.8	94.9 ± 1.1	97.4 ± 0.7	63.9 ± 6.8	65.7 ± 7.7	67.1 ± 7.6	65.0 ± 5.0	70.5 ± 4.8	74.5 ± 4.1
MKDR [21]	92.3 ± 1.2	94.9 ± 1.0	97.4 ± 0.7	68.7 ± 8.6	72.6 ± 7.8	78.2 ± 7.4	74.5 ± 5.1	81.2 ± 4.5	88.3 ± 6.2
SPL [17]	92.3 ± 1.3	94.9 ± 0.9	97.4 ± 1.0	67.8 ± 3.9	68.7 ± 3.4	69.5 ± 3.6	72.2 ± 3.2	78.4 ± 5.1	83.6 ± 2.6
MRMD [14]	94.9 ± 2.1	97.4 ± 1.2	100.0 ± 0.0	68.2 ± 4.1	**74.8 ± 4.5**	76.3 ± 5.2	75.3 ± 4.2	81.4 ± 3.7	**88.4 ± 5.1**
Our HHL	**100.0 ± 0.0**	**100.0 ± 0.0**	**100.0 ± 0.0**	**68.9 ± 9.0**	74.7 ± 10.1	**79.7 ± 10.2**	**80.9 ± 3.1**	**82.2 ± 2.8**	87.5 ± 3.8

TABLE II: Classification performance (%) compared with hashing methods on Honda.

Method	Frames 50			Frames 100			Frames 200		
	32 bits	64 bits	128 bits	32 bits	64 bits	128 bits	32 bits	64 bits	128 bits
SDH [26]	90.7 ± 2.8	91.3 ± 2.1	92.1 ± 1.8	93.4 ± 2.1	94.3 ± 1.2	96.7 ± 2.4	96.2 ± 1.8	97.6 ± 1.4	99.1 ± 0.4
COSDSH [27]	83.2 ± 4.8	84.4 ± 4.2	87.5 ± 4.8	85.2 ± 4.3	86.3 ± 3.5	88.9 ± 3.6	86.9 ± 4.7	88.7 ± 3.8	90.3 ± 3.2
FSDH [28]	95.2 ± 1.3	97.4 ± 1.7	100.0 ± 0.0	96.3 ± 1.6	98.2 ± 0.7	100.0 ± 0.0	97.3 ± 1.6	99.1 ± 0.3	100.0 ± 0.0
FSSH [29]	95.2 ± 1.7	97.8 ± 1.1	99.3 ± 0.4	96.8 ± 0.9	98.7 ± 0.6	100.0 ± 0.0	95.8 ± 1.2	97.7 ± 1.2	100.0 ± 0.0
SSLH [30]	89.6 ± 3.1	92.3 ± 2.5	92.3 ± 0.9	93.2 ± 1.1	94.5 ± 2.1	97.4 ± 1.3	95.3 ± 2.1	97.2 ± 1.7	100.0 ± 0.0
RSLH [31]	90.3 ± 2.2	93.1 ± 1.4	94.1 ± 0.6	94.2 ± 2.1	95.7 ± 1.6	98.2 ± 0.8	95.8 ± 1.3	97.8 ± 0.7	100.0 ± 0.0
SCDH [32]	90.2 ± 2.4	91.2 ± 2.1	94.9 ± 1.3	94.2 ± 2.2	95.7 ± 1.3	97.4 ± 0.9	96.4 ± 1.4	98.5 ± 1.2	100.0 ± 0.0
POPSH [45]	88.2 ± 1.3	90.5 ± 1.8	92.3 ± 1.0	93.4 ± 1.7	94.2 ± 2.1	97.4 ± 1.1	94.2 ± 1.7	98.5 ± 0.8	100.0 ± 0.0
Our HHL	**100.0 ± 0.0**	**100.0 ± 0.0**	**100.0 ± 0.0**	**100.0 ± 0.0**	**100.0 ± 0.0**	**100.0 ± 0.0**	**100.0 ± 0.0**	**100.0 ± 0.0**	**100.0 ± 0.0**

TABLE III: Classification performance (%) compared with hashing methods on YTC.

Method	Frames 50			Frames 100			Frames 200		
	32 bits	64 bits	128 bits	32 bits	64 bits	128 bits	32 bits	64 bits	128 bits
SDH [26]	58.5 ± 8.3	64.3 ± 8.9	67.2 ± 9.1	66.6 ± 9.3	72.0 ± 8.9	74.0 ± 9.3	72.4 ± 9.6	76.2 ± 9.1	78.0 ± 8.9
COSDSH [27]	20.6 ± 1.2	29.7 ± 6.7	30.7 ± 3.9	24.0 ± 2.1	32.2 ± 8.6	34.0 ± 4.5	25.8 ± 2.7	34.7 ± 8.0	57.2 ± 7.1
FSDH [28]	58.0 ± 8.6	64.6 ± 8.8	67.8 ± 10.5	66.2 ± 10.5	71.8 ± 9.6	74.2 ± 9.7	72.8 ± 8.3	76.7 ± 9.4	78.1 ± 10.1
FSSH [29]	57.9 ± 8.5	59.7 ± 8.6	58.4 ± 9.2	60.1 ± 10.5	64.0 ± 8.9	58.7 ± 8.0	62.5 ± 9.8	64.3 ± 9.1	67.1 ± 9.7
SSLH [30]	56.8 ± 7.4	59.4 ± 8.9	57.7 ± 7.2	61.4 ± 9.7	63.1 ± 10.4	60.4 ± 10.9	63.3 ± 10.2	66.4 ± 9.2	69.5 ± 4.6
RSLH [31]	57.5 ± 9.3	62.1 ± 9.3	62.9 ± 9.9	58.9 ± 9.9	63.6 ± 9.5	67.4 ± 9.3	63.1 ± 8.2	65.8 ± 9.4	69.3 ± 9.5
SCDH [32]	60.8 ± 9.1	65.6 ± 9.3	65.6 ± 8.8	66.8 ± 9.5	69.3 ± 9.8	69.4 ± 9.8	70.1 ± 9.9	72.3 ± 9.3	73.6 ± 9.3
POPSH [45]	61.2 ± 8.6	61.3 ± 9.4	61.8 ± 9.2	69.2 ± 8.9	69.4 ± 9.7	69.5 ± 9.0	74.7 ± 9.1	74.9 ± 9.3	74.1 ± 9.8
Our HHL	**67.3 ± 8.6**	**68.3 ± 9.1**	**68.9 ± 9.0**	**73.3 ± 10.0**	**74.0 ± 10.9**	**74.7 ± 10.1**	**78.9 ± 10.2**	**79.3 ± 10.6**	**79.7 ± 10.2**

D. Comparison with hashing methods

We compare our HHL with some hashing methods for the sake of fairness. To gain insight into the effect of hash length on classification performance, we set the hash length to 32, 64, and 128 bits, respectively. The experiments results on three datasets are given in Table II, Table III, and Table IV. Besides, we use some addition metrics (*e.g.*, precision, recall, and F1score) to show the effectiveness of our HHL in Table V. According to all experiments results, we have the following observations:

- Compared with all hashing methods, experiments results show HHL consistently outperforms these competitors. With the hash length increases, the classification accuracy is significantly improved. Moreover, the larger the number of frames in image set, the better the classification performance. For the metrics of precision, recall, and F1score, our HHL still obtains the best results. Results demonstrate that hierarchical hashing strategy and bidirectional regression semantic representation can improve the quality of hash codes and the discrimination ability.
- On Honda, competitors can achieve better classification scores only with long length hash codes and high frames in image set. Surprisingly, our HHL method achieves 100% classification accuracy in all cases. This shows our methods has a strong discriminatory power.
- On YTC, HHL is still superior than state-of-the-art methods, which shows that HHL can resist noise and enhance discriminant ability under complex conditions. We also find that classification accuracy of image set with large frames and short hash length is higher than

TABLE IV: Classification performance (%) compared with hashing methods on ETH-80.

Method	Frames 15			Frames 25			Frames 41		
	32 bits	64 bits	128 bits	32 bits	64 bits	128 bits	32 bits	64 bits	128 bits
SDH [26]	61.5 ± 7.6	67.0 ± 5.7	69.5 ± 2.1	62.5 ± 6.4	66.0 ± 9.6	68.0 ± 3.7	66.5 ± 4.9	69.5 ± 6.2	72.5 ± 6.1
COSDSH [27]	53.0 ± 5.7	52.5 ± 6.4	59.5 ± 5.1	52.3 ± 4.2	59.5 ± 2.1	62.0 ± 10.5	62.3 ± 2.6	67.5 ± 3.6	69.7 ± 4.2
FSDH [28]	64.5 ± 6.9	67.5 ± 4.3	67.5 ± 9.2	66.0 ± 1.4	67.5 ± 4.0	69.0 ± 7.4	68.0 ± 6.9	69.5 ± 8.6	72.0 ± 8.7
FSSH [29]	65.5 ± 2.9	66.7 ± 2.9	69.2 ± 6.4	67.7 ± 1.8	67.5 ± 6.3	70.4 ± 5.5	71.6 ± 5.6	72.5 ± 5.2	75.4 ± 3.5
SSLH [30]	64.8 ± 1.9	70.4 ± 3.5	74.9 ± 3.1	65.0 ± 4.4	71.9 ± 4.1	75.5 ± 2.6	65.0 ± 4.9	74.0 ± 5.7	75.8 ± 4.7
RSLH [31]	64.1 ± 1.6	62.8 ± 2.2	62.5 ± 1.9	65.9 ± 4.9	68.1 ± 4.4	66.5 ± 5.4	67.4 ± 3.5	69.2 ± 5.3	72.3 ± 3.5
SCDH [32]	66.0 ± 3.8	67.5 ± 4.7	66.5 ± 5.2	68.0 ± 2.7	69.5 ± 4.8	68.5 ± 5.5	70.5 ± 8.9	71.5 ± 4.5	76.0 ± 6.8
POPSH [45]	65.6 ± 8.6	65.9 ± 5.4	66.0 ± 5.5	64.5 ± 5.3	67.7 ± 8.1	68.7 ± 5.9	69.1 ± 5.6	69.8 ± 5.0	70.1 ± 2.8
Our HHL	**72.6 ± 2.8**	**77.1 ± 4.0**	**80.9 ± 3.1**	**76.3 ± 4.2**	**79.8 ± 4.4**	**82.2 ± 2.8**	**77.7 ± 6.0**	**82.6 ± 5.4**	**87.5 ± 3.8**

TABLE V: The performance (%) of precision, recall, and F1-score compared with the frames 50 on the YTC dataset.

Method	32 bits			64 bits			128 bits		
	precision	recall	F1-socre	precision	recall	F1-socre	precision	recall	F1-socre
SDH [26]	59.3 ± 8.4	57.0 ± 8.5	58.1 ± 9.2	65.2 ± 7.8	63.4 ± 8.5	64.3 ± 7.9	69.7 ± 9.1	67.3 ± 6.5	68.5 ± 8.3
COSDSH [27]	22.9 ± 4.5	26.0 ± 5.2	24.3 ± 4.3	33.4 ± 5.4	33.3 ± 4.7	33.3 ± 4.5	36.7 ± 3.4	34.6 ± 4.2	35.5 ± 5.4
FSDH [28]	58.5 ± 7.5	57.5 ± 8.4	58.0 ± 7.1	66.4 ± 6.5	64.3 ± 7.1	65.3 ± 6.3	70.4 ± 9.4	67.2 ± 6.7	68.7 ± 8.4
FSSH [29]	59.2 ± 9.2	57.9 ± 9.6	58.3 ± 8.4	60.4 ± 7.5	59.7 ± 8.4	60.1 ± 7.8	60.0 ± 7.4	58.4 ± 9.3	59.4 ± 5.8
SSLH [30]	59.2 ± 9.2	57.2 ± 7.3	58.1 ± 8.7	62.3 ± 10.4	58.6 ± 7.8	60.4 ± 8.8	63.4 ± 9.4	60.0 ± 9.1	61.7 ± 7.7
RSLH [31]	59.5 ± 7.8	58.3 ± 9.4	58.9 ± 8.3	61.2 ± 7.2	59.9 ± 6.9	60.5 ± 9.5	63.2 ± 8.8	61.8 ± 8.1	62.5 ± 7.4
SCDH [32]	62.0 ± 9.4	60.8 ± 8.7	61.3 ± 9.7	66.2 ± 10.2	65.6 ± 9.5	65.9 ± 8.4	67.2 ± 7.8	65.6 ± 8.4	66.4 ± 9.6
POPSH [45]	64.6 ± 7.9	61.4 ± 8.3	63.0 ± 8.7	65.7 ± 8.5	61.6 ± 9.2	63.5 ± 9.7	66.4 ± 8.1	61.8 ± 9.9	64.0 ± 10.1
Our HHL	**69.0 ± 9.8**	**66.3 ± 9.0**	**67.6 ± 9.4**	**69.8 ± 9.6**	**67.6 ± 8.8**	**68.7 ± 9.2**	**71.4 ± 9.3**	**68.9 ± 8.8**	**70.1 ± 9.0**

TABLE VI: Classification performance (%) compared with deep ISC methods.

Method	Honda			YTC			ETH80		
	Frames 50	Frames 100	Frames 200	Frames 50	Frames 100	Frames 200	Frames 15	Frames 25	Frames 41
SPDNet [22]	93.1 ± 2.1	97.4 ± 1.6	98.7 ± 1.1	65.3 ± 1.6	67.4 ± 2.0	70.4 ± 3.5	80.2 ± 1.6	81.2 ± 3.1	86.4 ± 3.7
DISH [46]	94.7 ± 0.9	97.8 ± 1.4	100.0 ± 0.0	68.4 ± 1.5	**75.0 ± 2.1**	76.2 ± 1.7	80.7 ± 1.7	81.2 ± 2.3	86.2 ± 2.7
GEMKML [47]	96.1 ± 1.1	98.5 ± 1.3	100.0 ± 0.0	68.1 ± 2.5	74.8 ± 3.2	76.1 ± 2.6	79.8 ± 3.2	80.4 ± 1.9	85.4 ± 2.1
SymNet [1]	96.9 ± 1.7	100.0 ± 0.0	100.0 ± 0.0	67.9 ± 1.2	74.5 ± 1.4	75.5 ± 8.9	80.7 ± 3.5	82.1 ± 2.1	**90.2 ± 1.6**
Our HHL	**100.0 ± 0.0**	**100.0 ± 0.0**	**100.0 ± 0.0**	**68.9 ± 9.0**	74.7 ± 10.1	**79.7 ± 10.2**	**80.9 ± 3.1**	**82.2 ± 2.8**	87.5 ± 3.8

that of the image set with small frames and long hash length. It shows image set with large frames contains rich complementary information, which can effectively improve the discrimination.

- On ETH-80, HHL again outperformed all hashing learning methods. Since ETH-80 is also more challenging, classification scores of all comparative methods are less than 80%, and that of HHL is less than 90%.

E. Comparison with deep ISC methods

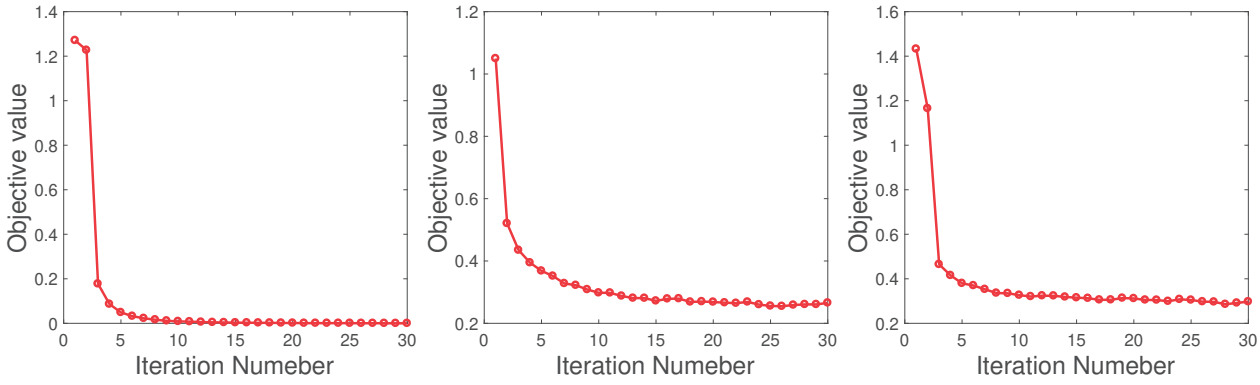
To further evaluate the effectiveness of our HHL, we perform some experiments compared with deep ISC methods on three datasets. The average classification performance is shown in Table VI. We can see that our HHL can be comparable with or superior to these deep methods. It indicates that the hierarchical hashing scheme and bidirectional semantic representation scheme of our HHL can efficiently improve the recognition performance.

F. Convergence analysis

Our HHL proposes an alternating iterative optimization scheme to iteratively update all variables. In order to empirically demonstrate the convergence, we record the relative error of two successive computed hash codes $\boldsymbol{B}$ by using the stop criterion $\|\boldsymbol{B}^{t+1} - \boldsymbol{B}^{t}\|_F^2 / \|\boldsymbol{B}^{t}\|_F^2 \leq \epsilon$ in each iterator. The experimental results are plotted in Fig. 3. We can see that the relative error is fast decreasing close to a stable point on all evaluated datasets.

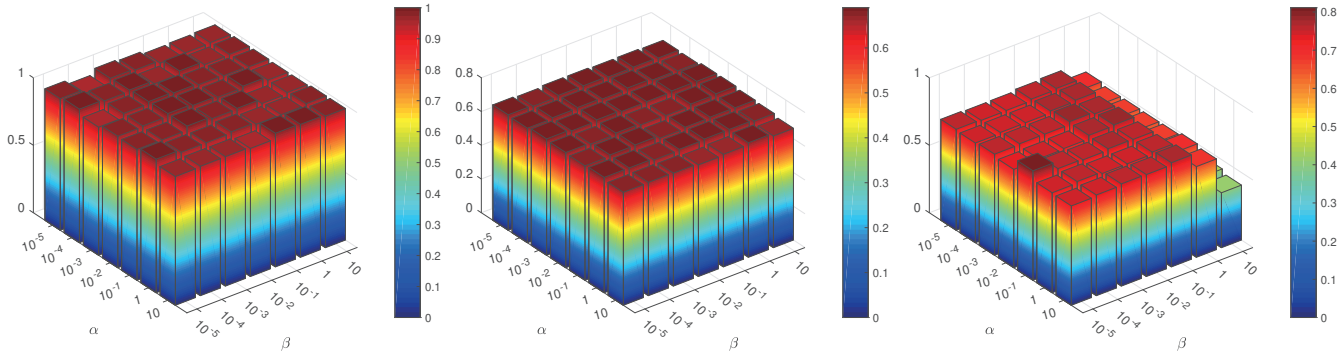
G. Parameter sensitivity analysis

To verify the stability of the proposed HHL, we perform experiments with two essential parameters (α and β) involved in HHL to be tuned. To obtain the best classification performance, both α and β are searched from 10^{-5} to 10^{2} with a step of 10 by leveraging a grid search strategy. As shown in Fig. 5 (zoom in for best view), it is easy to see HHL works well for a wide range of values α and β on the three datasets, which shows that HHL exhibits distinguished stability and sensitivity. Furthermore, to explore the impact of dimension values



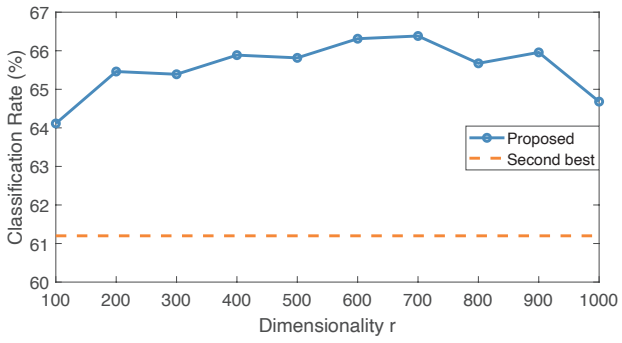
(a) Honda (b) YTC (c) ETH80

Fig. 3: Convergence properties on three datasets.



(a) Honda (b) YTC (c) ETH-80

Fig. 4: Parameter sensitivity analysis of our HHL *w.r.t.* α, β, and r on the three datasets.



(a) Performance *w.r.t.* dimension

Fig. 5: Parameter sensitivity analysis of our HHL *w.r.t.* α, β, and r on the three datasets.

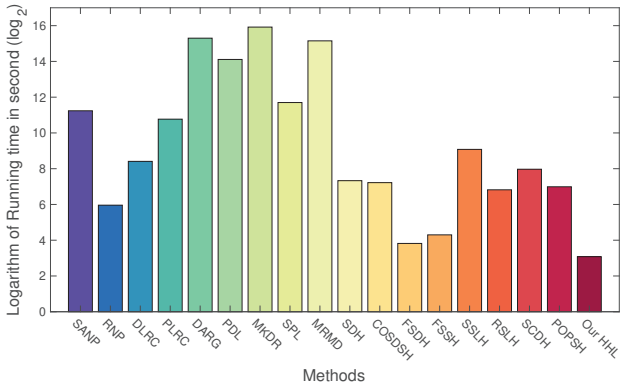
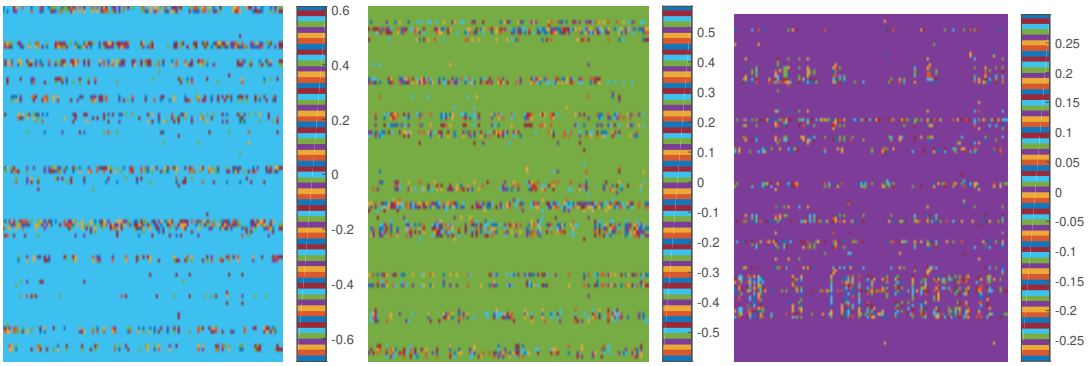


Fig. 6: Running time comparison of different methods on the YTC dataset (in logarithm of second).

on performance, we draw the performance *w.r.t.* dimension curve in Fig. 5 with 32 bits on YTC. From the results, our proposed HHL consistently outperforms competitors. We can see that the appropriate dimensions are beneficial to improve classification performance, and are generally set between the bit length and the feature dimension, *i.e.*, $l < r < h$. This could be because setting a larger value will destroy the original structure information, and setting a smaller value is difficult to alleviate the information loss.

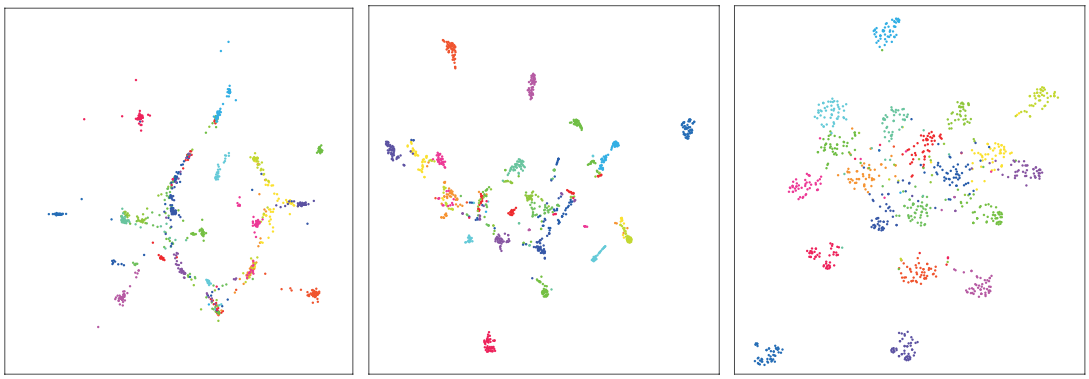
H. Time cost comparison

In Section III-G, we give the complexity analysis of our HHL, which show HHL is linear to the size of whole gallery sets. To further evaluate the computational efficiency of HHL, we take the challenging YTC data set with frames 50 as an example, and Fig. 6 reports the running time of all compared methods with recognizing 282 probe sets. We take the logarithm of the running time of all compared methods



(a) Honda (b) YTC (c) ETH-80

Fig. 7: The visualization of the hash function (*i.e.*, $\boldsymbol{PQ}$). For vividly comparison, we choose colormap of 'Lines'.



(a) Original kernel (b) Coarse representation (c) Hash codes

Fig. 8: The t-SNE visualization of each layer data on Honda.

TABLE VII: Ablation study on the three datasets.

Method	Honda: Frames 50			YTC: Frames 50			ETH-80: Frames 15		
	32 bits	64 bits	128 bits	32 bits	64 bits	128 bits	32 bits	64 bits	128 bits
HHL_LH	99.0 ± 0.8	99.8 ± 0.1	100.0 ± 0.0	66.6 ± 8.7	67.5 ± 8.0	68.2 ± 9.8	71.0 ± 3.8	73.4 ± 3.1	74.9 ± 1.2
HHL_HL	98.5 ± 1.4	99.0 ± 0.4	100.0 ± 0.0	66.1 ± 9.0	67.0 ± 8.6	67.7 ± 10.1	72.4 ± 1.7	74.4 ± 1.7	80.0 ± 3.7
NHHL	99.5 ± 0.2	99.5 ± 0.3	99.5 ± 0.4	59.3 ± 7.2	65.6 ± 8.9	66.2 ± 9.4	71.1 ± 1.4	76.4 ± 2.8	77.6 ± 3.5
HHL	**100.0 ± 0.0**	**100.0 ± 0.0**	**100.0 ± 0.0**	**67.3 ± 8.6**	**68.3 ± 9.1**	**68.9 ± 9.0**	**72.6 ± 2.8**	**77.1 ± 4.0**	**80.9 ± 3.1**

for better illustration. As can be seen, shallow ISC methods often have a great high running time due to modeling an image set based on real-value representation. And hashing methods have the low running time cost since it is a powerful scheme to reduce computational complexity by binary representation. Since some hashing methods (such as SSLH, RSLH, and SCDH) adopt the discrete cyclic coordinate descent (DCC) strategy to optimize hash codes bit by bit, they have a higher running time. Overall, the time advantage of our HHL is very evident, which verifies its computational efficiency. Consequently, it is well suited to handle large-scale ISC task that exists extensively within real-life applications.

I. Visualization analysis

In this section, we visualize the learned hash function $\boldsymbol{PQ}$ for the three datasets in Fig. 7. It is obvious that the two-layer hash function has the row-sparsity property, since the $\ell_{2,1}$ norm imposed on the hash mapping matrix (*i.e.*, $\|\boldsymbol{PQ}\|_{2,1}$) has the row-sparsity property. The $\ell_{2,1}$ norm term has the potential to adaptively select the important features and remove the redundant features during learning. Hence, the $\ell_{2,1}$ norm term makes the learned hash function have better interpretability for features. To evaluate the strength of each layer representation, we visualize the original kernel data (*i.e.*, $\boldsymbol{V}$), the first-layer representation (*i.e.*, $\boldsymbol{VP}$), and hash codes (*i.e.*, $\boldsymbol{B}$) with 32 bits using t-SNE tool on Honda as shown in Fig. 8. We can see that the first-layer representation is more discriminative than the kernel features, which indicates that the hierarchical representation can well reveal the underlying structure information. Moreover, hash codes are more compact and discrimination than others and have high the between-class and high within-class scatters. Accordingly, the hierarchical

hashing can help to distinguish different classes so as to greatly improve the classification performance.

J. Ablation study

To investigate the contributions of the proposed components, we further perform ablation study for the proposed HHL. Specifically, we design three addition variants of our HHL: 1) Removing orthogonal constraint and regressing labels to hash codes (*i.e.*, HHL_LH); 2) Removing orthogonal constraint and regressing hash codes to labels (*i.e.*, HHL_HL); and 3) Only using the single-layer hash function, that is, non-hierarchical hashing learning (*i.e.*, NHHL). The experiment results with short frames (50 or 15) are given in Table VII. Both the hierarchical hashing scheme and the bidirectional semantic representation can greatly improve classification performance and validate the effectiveness of the proposed method.

V. Conclusion

In this paper, we propose a novel hierarchical hashing learning (HHL) method. To be specific, we propose a layer-by-layer scheme to learn discriminative hash codes, thereby gradually gathering and refining the relative important discriminative information. To alleviate the effects of redundant and corrupted features, we impose the $\ell_{2,1}$ norm on the two-layer hash function. Moreover, we propose the bidirectional semantic representation to fully extract the supervised knowledge and preserve semantic similarity. Extensive experiments are conducted to adequately prove the effectiveness of the proposed HHL.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (Grant nos. 62106209) and the Sichuan Science and Technology Program (Grant no. 2021YJ0083).

References

[1] R. Wang, X.-J. Wu, and J. Kittler, "Symnet: A simple symmetric positive definite manifold deep learning method for image set classification," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 5, pp. 2208–2222, 2022.

[2] D. Wei, X. Shen, Q. Sun, and X. Gao, "Discrete metric learning for fast image set classification," *IEEE Transactions on Image Processing*, vol. 31, pp. 6471–6486, 2022.

[3] B. Uzun, H. Cevikalp, and H. Saribas, "Deep discriminative feature models (ddfms) for set based face recognition and distance metric learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–14, 2022.

[4] R. Wang, X.-J. Wu, T. Xu, C. Hu, and J. Kittler, "Deep metric learning on the spd manifold for image set classification," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–1, 2022.

[5] M. Lin, R. Ji, S. Chen, X. Sun, and C.-W. Lin, "Similarity-preserving linkage hashing for online image retrieval," *IEEE Transactions on Image Processing*, vol. 29, pp. 5289–5300, 2020.

[6] S. He, B. Wang, Z. Wang, Y. Yang, F. Shen, Z. Huang, and H. T. Shen, "Bidirectional discrete matrix factorization hashing for image search," *IEEE transactions on cybernetics*, vol. 50, no. 9, pp. 4157–4168, 2019.

[7] Y. Wang, Z.-D. Chen, X. Luo, and X.-S. Xu, "High-dimensional sparse cross-modal hashing with fine-grained similarity embedding," in *Proceedings of the Web Conference 2021*, 2021, pp. 2900–2909.

[8] M. Meng, H. Wang, J. Yu, H. Chen, and J. Wu, "Asymmetric supervised consistent and specific hashing for cross-modal retrieval," *IEEE Transactions on Image Processing*, vol. 30, pp. 986–1000, 2020.

[9] X. Liu, X. Wang, and Y.-m. Cheung, "Fddh: Fast discriminative discrete hashing for large-scale cross-modal retrieval," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.

[10] X. Liu, Z. Hu, H. Ling, and Y.-m. Cheung, "Mtfh: A matrix tri-factorization hashing framework for efficient cross-modal retrieval," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 3, pp. 964–981, 2019.

[11] M. Zhang, R. He, D. Cao, Z. Sun, and T. Tan, "Simultaneous feature and sample reduction for image-set classification," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, 2016.

[12] Y. Sun, D. Peng, H. Huang, and Z. Ren, "Feature and semantic views consensus hashing for image set classification," in *Proceedings of the 30th ACM International Conference on Multimedia*, 2022, pp. 2097–2105.

[13] W. Wang, R. Wang, Z. Huang, S. Shan, and X. Chen, "Discriminant analysis on riemannian manifold of gaussian distributions for face recognition with image sets," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 2048–2057.

[14] R. Wang, X.-J. Wu, K.-X. Chen, and J. Kittler, "Multiple riemannian manifold-valued descriptors based image set classification with multi-kernel metric learning," *IEEE Transactions on Big Data*, 2020.

[15] Y. Hu, A. S. Mian, and R. Owens, "Face recognition using sparse approximated nearest points between image sets," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 10, pp. 1992–2004, 2012.

[16] M. Yang, P. Zhu, L. Van Gool, and L. Zhang, "Face recognition based on regularized nearest points between image sets," in *2013 10th IEEE international conference and workshops on automatic face and gesture recognition (FG)*. IEEE, 2013, pp. 1–7.

[17] W. Zhu, B. Peng, H. Wu, and B. Wang, "Query set centered sparse projection learning for set based image classification," *Applied Intelligence*, vol. 50, no. 10, pp. 3400–3411, 2020.

[18] L. Chen, "Dual linear regression based classification for face cluster recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*,

2014, pp. 2673–2680.

[19] Q. Feng, Y. Zhou, and R. Lan, "Pairwise linear regression classification for image set retrieval," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4865–4872.

[20] W. Wang, R. Wang, S. Shan, and X. Chen, "Prototype discriminative learning for face image set classification," in *Asian Conference on Computer Vision*. Springer, 2016, pp. 344–360.

[21] W. Yan, Q. Sun, H. Sun, Y. Li, and Z. Ren, "Multiple kernel dimensionality reduction based on linear regression virtual reconstruction for image set classification," *Neurocomputing*, vol. 361, pp. 256–269, 2019.

[22] Z. Huang and L. Van Gool, "A riemannian network for spd matrix learning," in *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[23] Y. Cao, S. Chen, J. Gui, H. Qi, Z. Li, and C. Liu, "Hash learning with variable quantization for large-scale retrieval," *IEEE Transactions on Circuits and Systems for Video Technology*, 2021.

[24] X. Nie, W. Jing, C. Cui, C. J. Zhang, L. Zhu, and Y. Yin, "Joint multi-view hashing for large-scale near-duplicate video retrieval," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 10, pp. 1951–1965, 2019.

[25] C. Zheng, L. Zhu, X. Lu, J. Li, Z. Cheng, and H. Zhang, "Fast discrete collaborative multi-modal hashing for large-scale multimedia retrieval," *IEEE Transactions on Knowledge and Data Engineering*, vol. 32, no. 11, pp. 2171–2184, 2019.

[26] F. Shen, C. Shen, W. Liu, and H. Tao Shen, "Supervised discrete hashing," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 37–45.

[27] W.-C. Kang, W.-J. Li, and Z.-H. Zhou, "Column sampling based discrete supervised hashing," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 30, no. 1, 2016.

[28] J. Gui, T. Liu, Z. Sun, D. Tao, and T. Tan, "Fast supervised discrete hashing," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 2, pp. 490–496, 2017.

[29] X. Luo, L. Nie, X. He, Y. Wu, Z.-D. Chen, and X.-S. Xu, "Fast scalable supervised hashing," in *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, 2018, pp. 735–744.

[30] X. Liu, X. Nie, Q. Zhou, X. Xi, L. Zhu, and Y. Yin, "Supervised short-length hashing." in *IJCAI*, 2019, pp. 3031–3037.

[31] X. Liu, X. Nie, Q. Dai, Y. Huang, L. Lian, and Y. Yin, "Reinforced short-length hashing," *IEEE Transactions on Circuits and Systems for Video Technology*, 2020.

[32] Y. Chen, Z. Tian, H. Zhang, J. Wang, and D. Zhang, "Strongly constrained discrete hashing," *IEEE Transactions on Image Processing*, vol. 29, pp. 3596–3611, 2020.

[33] Y. Wang, L. Zhang, F. Nie, X. Li, Z. Chen, and F. Wang, "Wegan: Deep image hashing with weighted generative adversarial networks," *IEEE Transactions on Multimedia*, vol. 22, no. 6, pp. 1458–1469, 2019.

[34] Y. Chen, H. Zhang, Z. Tian, J. Wang, D. Zhang, and X. Li, "Enhanced discrete multi-modal hashing: more constraints yet less time to learn," *IEEE Transactions on Knowledge and Data Engineering*, 2020.

[35] J. Wang, T. Zhang, N. Sebe, H. T. Shen *et al.*, "A survey on learning to hash," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, no. 4, pp. 769–790, 2017.

[36] Z. Ren, M. Mukherjee, J. Lloret, and P. Venu, "Multiple kernel driven clustering with locally consistent and selfish graph in industrial iot," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 4, pp. 2956–2963, 2020.

[37] Z. Ren, S. X. Yang, Q. Sun, and T. Wang, "Consensus affinity graph learning for multiple kernel clustering," *IEEE Transactions on Cybernetics*, vol. 51, no. 6, pp. 3273–3284, 2020.

[38] Z. Ren and Q. Sun, "Simultaneous global and local graph structure preserving for multiple kernel clustering," *IEEE transactions on neural networks and learning systems*, vol. 32, no. 5, pp. 1839–1851, 2020.

[39] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem," *Neural computation*, vol. 10, no. 5, pp. 1299–1319, 1998.

[40] J. Liu, X. Liu, S. Wang, S. Zhou, and Y. Yang, "Hierarchical multiple kernel clustering," in *Proceedings of the aaai conference on artificial intelligence*, vol. 35, no. 10, 2021, pp. 8671–8679.

[41] L. Zhu, X. Lu, Z. Cheng, J. Li, and H. Zhang, "Deep collaborative multi-view hashing for large-scale image search," *IEEE Transactions on Image Processing*, vol. 29, pp. 4643–4655, 2020.

[42] Z. Ren, Q. Sun, B. Wu, X. Zhang, and W. Yan, "Learning latent low-rank and sparse embedding for robust image feature extraction," *IEEE Transactions on Image Processing*, vol. 29, pp. 2094–2107, 2019.

[43] X. Gao, S. Niu, D. Wei, X. Liu, T. Wang, F. Zhu, J. Dong, and Q. Sun, "Joint metric learning-based class-specific representation for image set classification," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–15, 2022.

[44] G. Zhang, J. Yang, Y. Zheng, Z. Luo, and J. Zhang, "Optimal discriminative feature and dictionary learning for image set classification," *Information Sciences*, vol. 547, pp. 498–513, 2021.

[45] Z. Zhang, X. Zhu, G. Lu, and Y. Zhang, "Probability ordinal-preserving semantic hashing for large-scale image retrieval," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 15, no. 3, pp. 1–22, 2021.

[46] J. Feng, S. Karaman, and S.-F. Chang, "Deep image set hashing," in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2017, pp. 1241–1250.

[47] R. Wang, X.-J. Wu, and J. Kittler, "Graph embedding multi-kernel metric learning for image set classification with grassmannian manifold-valued features," *IEEE Transactions on Multimedia*, vol. 23, pp. 228–242, 2020.