Neurocomputing xxx (xxxx) xxx

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Robust supervised discrete hashing

Yao Xiao [a], Wei Zhang [a], Xiangguang Dai [a,*], Xiangqin Dai [a], Nian Zhang [b]

[a] Key Laboratory of Intelligent Information Processing and Control, Chongqing Three Gorges University, Wanzhou, Chongqing, 404100, China
[b] Department of Electrical and Computer Engineering, University of the District of Columbia Washington, D.C. 20008 USA

A R T I C L E I N F O

Article history:
Received 20 May 2021
Revised 3 August 2021
Accepted 7 September 2021
Available online xxxx

Keywords:
Robustness
Cauchy loss
Supervised discrete hashing
Image retrieval

A B S T R A C T

In this paper, we proposed a more robust supervised hashing framework based on the Cauchy loss function and Supervised Discrete Hashing (SDH) called Robust Supervised Discrete Hashing (RSDH), which can learn a robust subspace consisted of binary codes. The Cauchy loss is used to measure the error between the label matrix and the product of the decomposed matrices. RSDH can not only reduce the outliers and noise of the hashing codes, but also achieve the more satisfactory retrieval effect. Image retrieval experiments demonstrate that RSDH performs better than the other hashing methods.

1. Introduction

Recently, data processing techniques advance rapidly in some areas including fast internet, smart phone and multimedia data. For the high-dimensional data precise indexing and effective searching, it becomes an enormous challenge. Most previous data indexing techniques (*e.g.* B^{+}-tree [1] and B^{-}-tree [2]) were proposed to address this problem. However, these techniques may fail to deal with high-dimensional data due to disaster of dimensionality. To solve this problem, hashing techniques [3–7] were put forward and demonstrated effectiveness and efficiency in image retrieval [8–10], security protection [11], pattern recognition [12–15] and recommendation [16]. The purpose of hashing algorithms is to learn a Hamming space composed of binary codes (*i.e.* -1 and 1 or 0 and 1) from the original data space. The Hamming space has the following three properties: (1) remaining the similarity of data points. (2) reducing storage cost. (3) improving retrieval efficiency.

Typically, existing hashing methods can be approximately summarized into two categories: data-independent and data-dependent methods. Data-independent methods randomly produce a group of hash functions that do not need training data and project the original data space into a Hamming space by the hash functions. Some classical data-independent methods are Local-Sensitive Hashing (LSH) [17] and its variant [18]. To acquire a better retrieval performance, the LSH family requires a larger Hamming space than the original data space, which leads to high storage cost and low retrieval efficiency. In contrast, data-dependent hashing methods require training data to learn a Hamming space composed of fewer bits, which can achieve an excellent retrieval performance. Generally, data-dependent hashing methods can be approximately split into three groups, *i.e.* unsupervised hashing methods [19–24], semi-supervised methods [25,26] and supervised methods [27–31].

Unsupervised hashing methods [19–24] can learn hash codes without semantic labels. Spectral Hashing (SH) [19] generates hash codes by solving a continuously relaxed problem similar to Laplacian Eigenmap [32]. Anchor Graph Hashing (AGH) [20] utilizes the anchor graphs to construct a sparse adjacent graph. Semi-Supervised Hashing methods [25,26] employ the pairwise label information to regularize hashing functions. Binary Reconstructive Embedding (BRE) [26] learns hash functions by minimizing the reconstruction error between the Euclidean distances and the Hamming distances. Supervised hashing methods [27–31] exploit all labels of training data, which leads to a more excellent performance than other methods. Supervised Discrete Hashing (SDH) [27] learns directly the binary hash codes without relaxation. Kernel-Based Supervised Hashing (KSH) [28] relaxes the binary constraints to solve a successive optimization problem.

Most of hashing models [21,22,27,33,34] with the discrete constraints on the hash codes can be formulated as mixed-integer optimization problems (MIOP). To acquire a feasible solution from the MIOP, a general optimization structure can be summarized into three steps. Firstly, the MIOP is transformed into a relaxed problem

* Corresponding author.
E-mail addresses: xiaoyao227192@163.com (Y. Xiao), weizhang@sanxiau.edu.cn (W. Zhang), daixiangguang@163.com (X. Dai), 309376876@qq.com (X. Dai), nzhang@udc.edu (N. Zhang).

https://doi.org/10.1016/j.neucom.2021.09.077
0925-2312/

Table 1
Comparing the robustness of RSDH and other methods.

Methods	Objective function	Derivative
AGH	$\mathrm{tr}(Y^T LY)$	$YL + YL^T$
KSH	$\|\|S - \frac{1}{r}HH^T\|\|_F^2$	$\sum_{i,j} 2c(S_{i,j}, \frac{1}{r}HH^T)$
IMH	$\sum_{i=1}^{n} w(x_q, x_i)\|\|y_q - y_i\|\|_F^2$	$\sum_{i=1}^{n} 2w(x_q, x_i)c(y_i, y_q)$
SDH	$\|\|Y - BW\|\|_F^2$	$\sum_{i,j} 2c(Y_{ij}, BW)$
FSDH	$\|\|B - YW\|\|_F^2$	$\sum_{i,j} 2(B - YW)_{ij}(-B_{ij})$
RSDH	$\sum_{i,j} ln(1 + (\frac{Y-BW}{\gamma})_{ij}^2)$	$\sum_{i,j} \frac{2}{\gamma^2 + (Y-BW)_{ij}^2} c(Y_{ij}, BW)$

without any discrete constraints. Secondly, a solution of the continuous values is achieved by the relaxed problem. Thirdly, the solution of continuous values is rounded to the solution of binary values. Although, this strategy apparently exhibits its advantage to optimize MIOPs, the accumulating quantization error caused by the strategy decreases the effectiveness of the hash codes. To address this issue, Shen et al. [27] utilized the discrete cyclic coordinate descent algorithm (DCC) [35] to optimize directly hash codes without relaxation. However, this way the SDH becomes non-robust for unreliable and noisy environment. Recently, Luo et al. [33] proposed the $l_{2,p}$ loss function to control undependable binary codes and noise labeled samples. Actually, some loss functions [36,37] are more robust than the $l_{2,p}$ loss function in handling outliers and noise.

To further reinforce the robustness of hashing methods, we introduce the Cauchy loss to reduce the noise affection and the quantization error. Based on the framework of SDH, the Cauchy loss is utilized to measure the error between the label matrix and the corresponding decomposition matrix. In other words, the measuring function $||\cdot||_F^2$ is replaced by the Cauchy loss. In a word, the main contributions include:

- Based on the framework of SDH and the robustness of the Cauchy loss function, a robust supervised hashing method, called RSDH, is proposed to reduce the decomposed error and eliminate some outliers and noise of the hashing codes.
- Our problem is a mixed-integer optimization problem. The objective function is non-quadratic, thanks to the existence of the Cauchy loss function. We exploit the convex conjugation theory [38] and transform the non-quadratic objective function into an augmented loss function.
- Extensive experiments are carried out to evaluate the robustness and effectiveness of our method on four image datasets.

The remainder of this paper is written as follows. In Section 2, the SDH and the general optimization steps of the corresponding algorithm are reviewed. Then, the RSDH and its robustness analysis are introduced. Extensive experiments are gone in Section 4. Finally, the conclusion is summarized in Section 5.

Remarks: X represents data matrix, n represents training sample size, B represents hash codes, l represents the length of hash code, Y represents label matrix, W represents projection matrix for hash codes, $||\cdot||$ represents Frobenius norm, m represents the number of anchor points, $F(\cdot)$ represents embedding function, P represents projection matrix, γ, υ and λ represents penalty parameter, t represents iteration number, $\phi(x)$ represents an m-dimensional vector,

2. Supervised discrete hashing

Supposed that there are n example and each of those samples has d features. For each sample, we generate a short hashing codes (*e.g.* totally l bits) to remain the similarity. Therefore, any data matrix $X = \{x_1 \dots x_n\} \in R^{n\times d}$ can lead to hashing codes

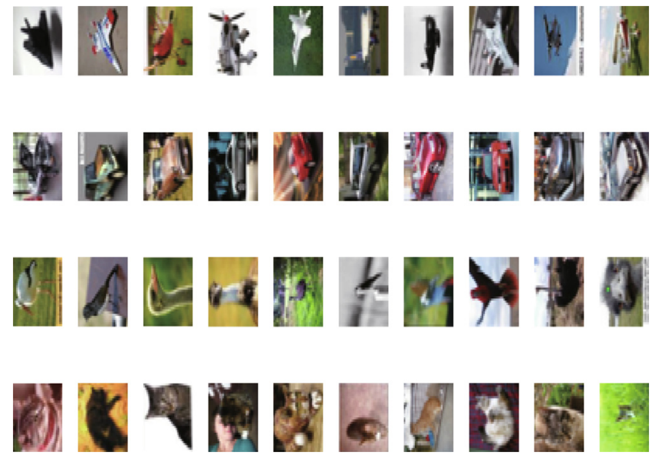


Fig. 1. A part of the sample images from CIFAR-10.

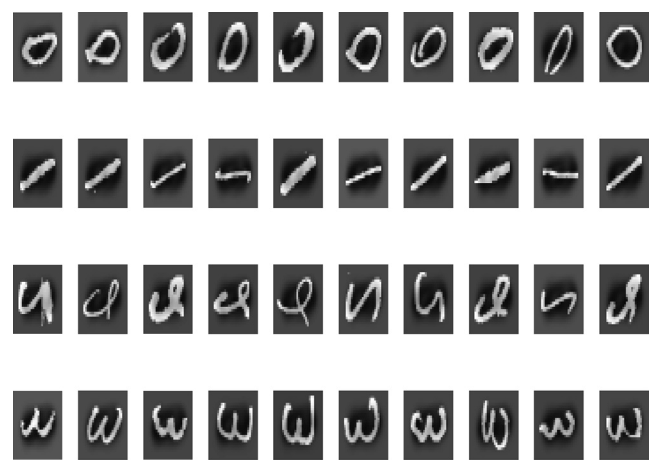


Fig. 2. A part of the sample images from MNIST.

$B = \{b_i\}_{i=1}^n \in \{-1, 1\}^{n\times l}$. The labels of all training instances are $Y = \{y_i\}_{i=1}^n \in R^{n\times c}$, where c represents the number of classes. Therefore, the problem of SDH can be summarized as follows:

Fig. 3. A part of the sample images from ORL.

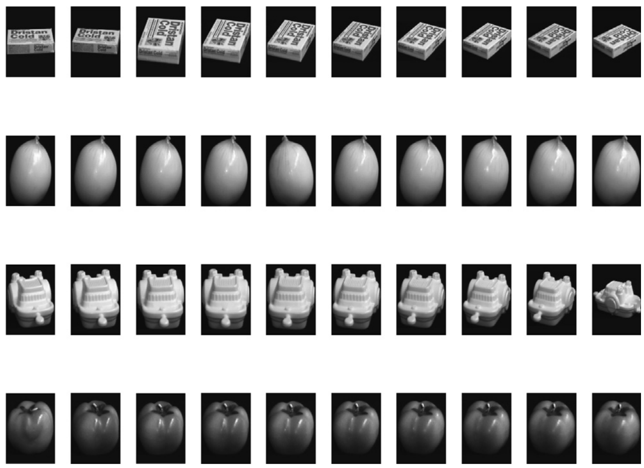


Fig. 4. A part of the sample images from COIL100.

$$\min_{B,F,W} ||Y-BW||_F^2 + \lambda||W||_F^2 + \nu||B-F(x)||_F^2 \quad s.t\ B \in \{-1,1\}^{n\times l}. \tag{1}$$

The nonlinear term $F(\cdot)$ is the approximation of hashing codes, which can be summarized as follows:

$$F(x) = P^T\phi(x), \tag{2}$$

where $\phi(x)$ represents an m-dimensional vector, which can be calculated by $\phi(x) = [exp(||x-a_1||^2)/\sigma, \ldots, exp(||x-a_m||^2)/\sigma]^T$, $\{a_j\}_{j=1}^m$ represents m random anchor points and σ is the kernel parameter.

The algorithm framework of problem (1) mainly follows the block coordinate descent algorithm [39]. Specifically, some block variables are updated while the remaining block variables are fixed. Repeatedly and alternatively updating P, W and B, the local solution of problem (1) can be achieved. The optimization procedure includes the following three parts: **Compute** P**:**

Fixing B and W, P can be calculated by

$$P = (\phi(x)^T\phi(x))^{-1}\phi(x)^T B. \tag{3}$$

Compute W**:**

Fixing B and F, W can be achieved by

$$W = (B^T B + \lambda I)^{-1} B^T Y. \tag{4}$$

Compute B**:**

Fixing P and W, B can be solved by the DCC algorithm [35]. To save space, the particulars of the optimization can be seen in [27].

3. Robust Supervised Discrete Hashing

To further reinforce the robustness and performance of SDH, we utilize the Cauchy loss to replace the Frobenius norm of SDH. Based on the Cauchy loss and SDH, the objective function can be described as follows:

$$\min_{B,F,W} \sum_{i,j} g\left(\left(\frac{Y-BW}{\gamma}\right)_{ij}^2\right) + ||W||_F^2 + \nu||B-F(x)||_F^2 \quad s.t\ B \in \{-1,1\}^{n\times l}, \tag{5}$$

where $g(u) = ln(1+u)$, $u = \left(\frac{Y-BW}{\gamma}\right)_{ij}^2$, $\nu = \frac{\nu}{\lambda}$and $\gamma = \sqrt{\lambda}\gamma$. ν and γ are penalty parameters.

Problem (5) denotes a mixed-integer optimization problem with three variables. Similar to SDH, the optimization problem solve iteratively by the block coordinate descent algorithm. Alternatively solving the following problems

$$\min_F ||B-F(x)||_F^2 \tag{6}$$

and

$$\min_B \sum_{i,j} g\left(\left(\frac{Y-BW}{\gamma}\right)_{ij}^2\right) + \nu||B-F(x)||_F^2 \tag{7}$$

and

$$\min_W \sum_{i,j} g\left(\left(\frac{Y-BW}{\gamma}\right)_{ij}^2\right) + ||W||_F^2 \tag{8}$$

until convergence. Problem (5) can achieve the local optimal solution. In the latter parts, we mainly discuss the optimization details for solving problems (6), (7) and (8).

Compute P**:**

The optimal solution of problem (6) is

$$P = (\phi(x)^T\phi(x))^{-1}\phi(x)^T B. \tag{9}$$

Compute B**:**

According to the conjugate function theory, problem (7) can be converted to a maximization form as follows:

$$\max_B \sum_{i,j} g\left(\left(\frac{Y-BW}{\gamma}\right)_{ij}^2\right) \tag{10}$$

where $f(u) = -\nu||B-F(x)||_F^2 - g(u)$. If the negative logarithmic function is convex, then $f(u)$ is convex. Therefore, we have

$$f^*(p) = \max_{u\in R_+}\{up - f(u)\}. \tag{11}$$

The optimal solution is $p^* = -1/(u+1)$. $f^*(p)$ can be written as:

$$f(u) = f^{**}(u) = \max_{p\in R_+}\{pu - f^*(p)\}. \tag{12}$$

Replacing u by $u = \left(\frac{Y-BW}{\gamma}\right)_{ij}^2$in formula (12), we have

$$f(u) = \max_{P_{ij}}\left\{\left(\frac{Y-BW}{\gamma}\right)_{ij}^2 - f^*(P_{ij})\right\}. \tag{13}$$

By combining (13) and (7), we have

$$\max_{B,P_{ij}} \sum_{i,j}\left\{\left(\frac{Y-BW}{\gamma}\right)_{ij}^2 - f^*(P_{ij})\right\} - \nu||B-F(x)||_F^2. \tag{14}$$

Problem (14) can be decomposed into the following two problems:

$$\max_{P_{ij}} \sum_{i,j}\left\{\left(\frac{Y-BW}{\gamma}\right)_{ij}^2 - f^*(P_{ij})\right\} \tag{15}$$

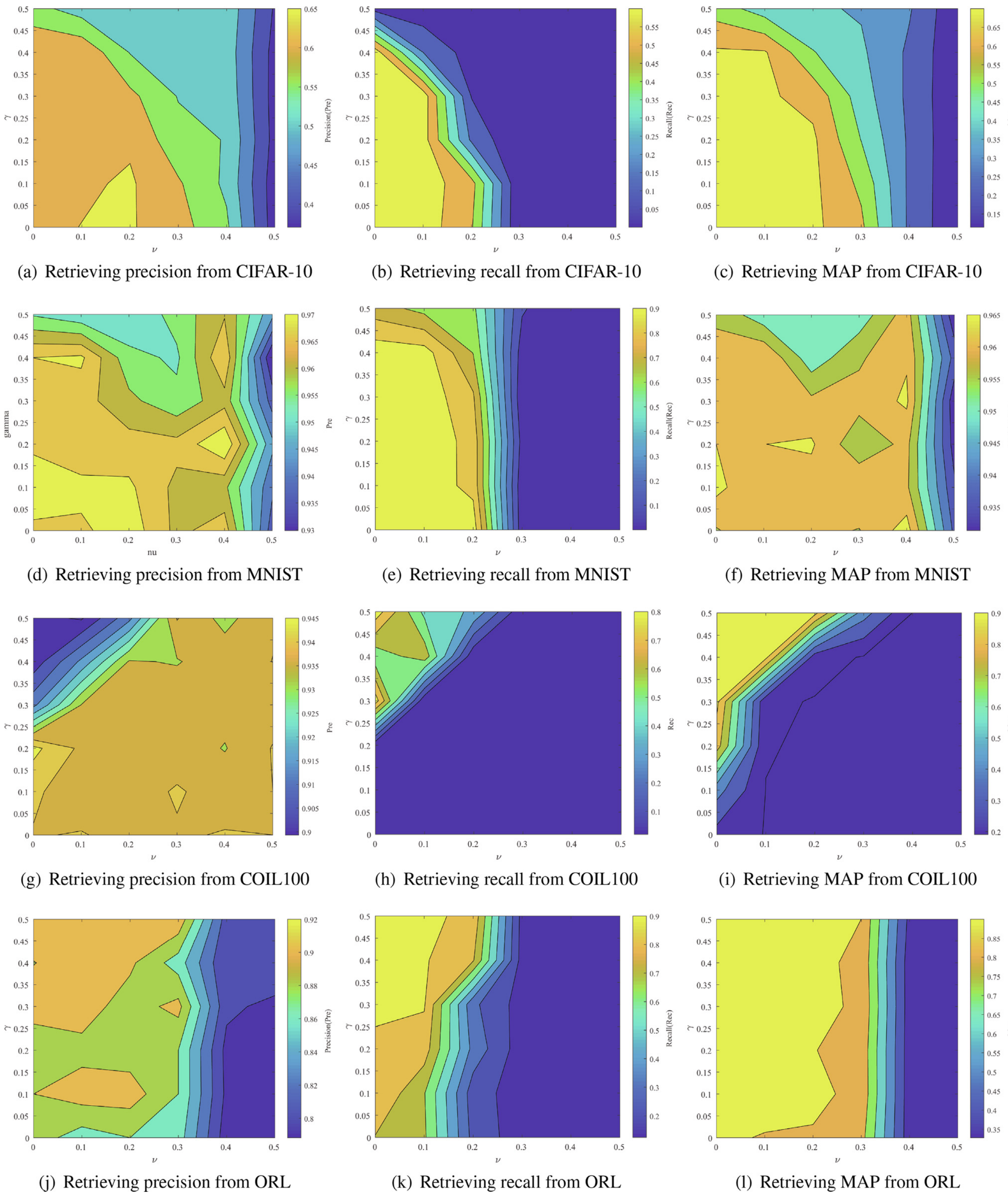
and

$$\max_B \sum_{i,j}\left\{\left(\frac{Y-BW}{\gamma}\right)_{ij}^2\right\} - \nu||B-F(x)||_F^2 \tag{16}$$

Problem (15) can be obtained by $p^* = -1/(u+1)$. Thus, P can be calculated as follows.

$$P_{ij} = -\frac{1}{\left(\frac{Y-BW}{\gamma}\right)_{ij}^2}. \tag{17}$$

For problem (16), we utilize DCC [35] to solve hash codes B. We suppose that



(a) Retrieving precision from CIFAR-10 (b) Retrieving recall from CIFAR-10 (c) Retrieving MAP from CIFAR-10

(d) Retrieving precision from MNIST (e) Retrieving recall from MNIST (f) Retrieving MAP from MNIST

(g) Retrieving precision from COIL100 (h) Retrieving recall from COIL100 (i) Retrieving MAP from COIL100

(j) Retrieving precision from ORL (k) Retrieving recall from ORL (l) Retrieving MAP from ORL

Fig. 5. Retrieving precision, recall and MAP from CIFAR-10, MNIST, COIL100 and ORL with different ν and γ. The third dimension represents precision, recall and MAP with different colors.

$$B = \begin{bmatrix} z^T \\ B\prime \end{bmatrix}, \; Y = \begin{bmatrix} y^T \\ Y\prime \end{bmatrix}, \; Q = \begin{bmatrix} q^T \\ Q\prime \end{bmatrix}, \; \frac{P}{\gamma^2} = \begin{bmatrix} d \\ D\prime \end{bmatrix} \tag{18}$$

where D represents the diagonal matrix of d, z^T represents the nth row of B, y^T represents the nth row of Y, q^T represents the nth row of Q. Thus, we rewrite problem (16) by

$$\begin{aligned} &\sum_{i,j}\left\{P_{ij}\left(\frac{Y-BW}{\gamma}\right)^2_{ij}\right\} \\ &= -2tr(YDB^TW^T) - tr(BWDW^TB^T) + constant \\ &= -2tr(y^TDzW) + constant \\ &= -2y^TDWz + constant. \end{aligned} \tag{19}$$

Table 2
Parameter settings of SDH, FSDH, RSDH, AGH, KSH and IMH.

parameters	datasets	
	CIFAR-10 and MNIST	ORL and COIL100
t	10	10
ν	10^{-5}	10^{-5}
γ	10^{-5}	1
anchor points	1000	200

Table 3
Retrieval results on CIFAR-10 when code length is 128. The best result are in bold.

Methods	Precision	Recall	Map
RSDH	**0.644**	**0.650**	**0.715**
SDH	0.310	0.064	0.458
FSDH	0.301	0.076	0.458
KSH	0.053	2.6e-4	0.383
AGH	0.243	3.8e-4	0.146
IMH	0.259	0.003	0.188

Table 4
Retrieval results on MNIST when code length is 128. The best result are in bold.

Methods	Precision	Recall	Map
RSDH	**0.966**	**0.966**	**0.973**
SDH	0.902	0.757	0.951
FSDH	0.904	0.762	0.949
KSH	0.669	0.273	0.907
AGH	0.813	0.003	0.574
IMH	0.859	0.083	0.760

Table 5
Retrieval results on ORL when code length is 64. The best result are in bold.

Methods	Precision	Recall	Map
RSDH	**0.875**	**0.880**	**0.904**
SDH	0.240	0.115	0.830
FSDH	0.300	0.140	0.840
KSH	0.030	0.009	0.604
AGH	0.332	0.102	0.373
IMH	0.460	0.202	0.468

Table 6
Retrieval results on COIL100 when code length is 64. The best result are in bold.

Methods	Precision	Recall	Map
RSDH	**0.912**	**0.917**	**0.916**
SDH	0.870	0.653	0.653
FSDH	0.892	0.682	0.682
KSH	0.537	0.080	0.080
AGH	0.693	0.265	0.265
IMH	0.843	0.225	0.225

Similarly, we have

$$\begin{aligned} & \nu\|B - F(X)\|_F^2 \\ = \; & -2tr(B^T Q) \\ = \; & -2q^T z, \end{aligned} \tag{20}$$

where $Q = \nu F(x)$. Combining (19) and (20), problem (16) can be written as:

$$\max_z(-y^T D W^T + q^T)z \tag{21}$$

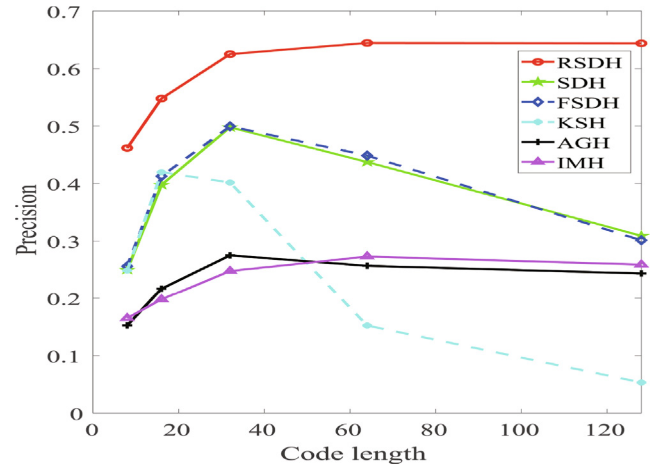


Fig. 6. The precision versus code length (8 to 128) on the CIFAR-10.

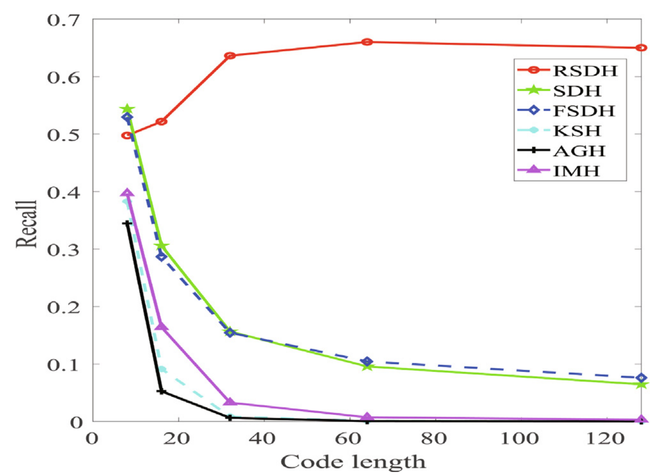


Fig. 7. The recall versus code length (8 to 128) on the CIFAR-10.

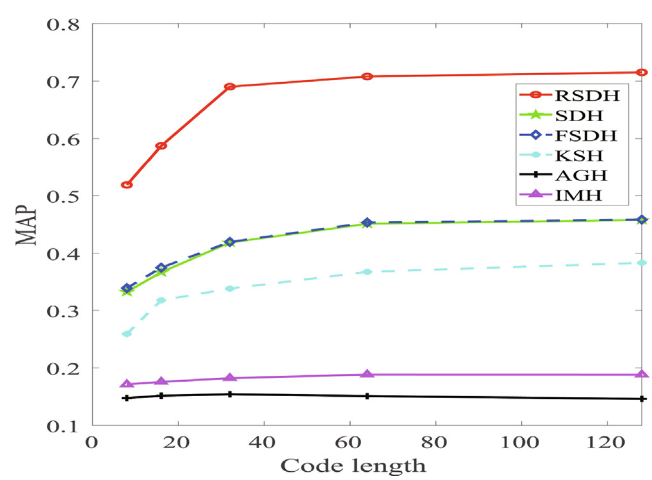


Fig. 8. The MAP versus code length (8 to 128) on the CIFAR-10.

Therefore, problem (7) can be solved by

$$z = sign(-WD^T y + q). \tag{22}$$

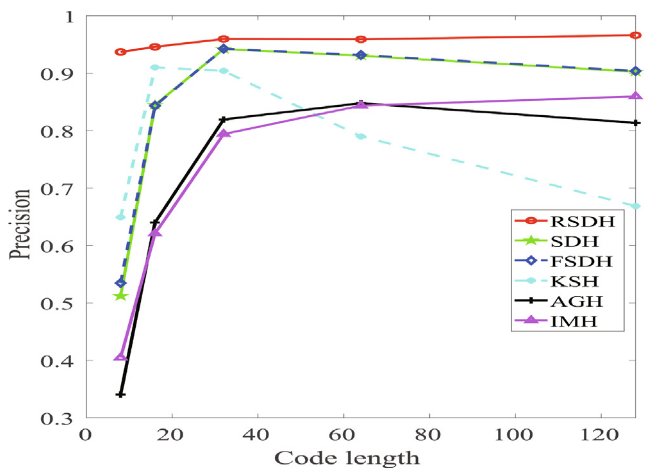


Fig. 9. The precision versus code length (8 to 128) on the MNIST.

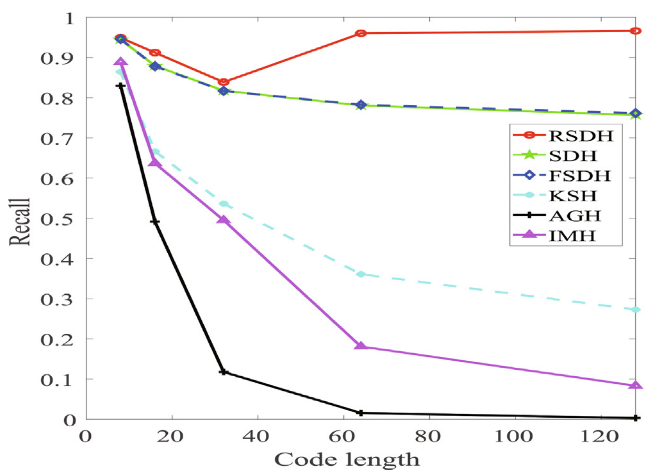


Fig. 10. The recall versus code length (8 to 128) on the MNIST.

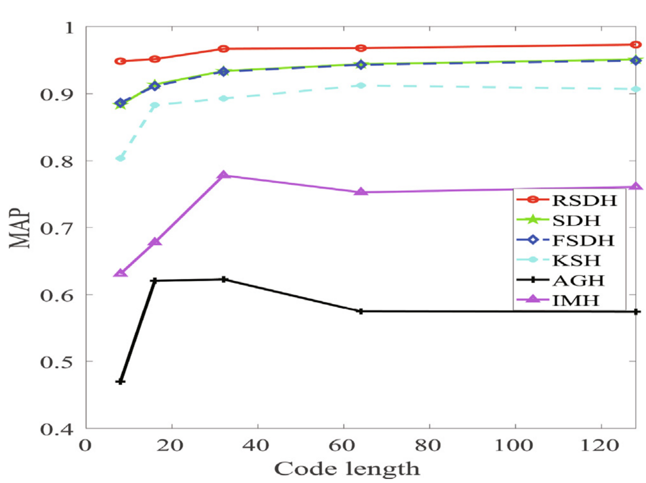


Fig. 11. The MAP versus code length (8 to 128) on the MNIST.

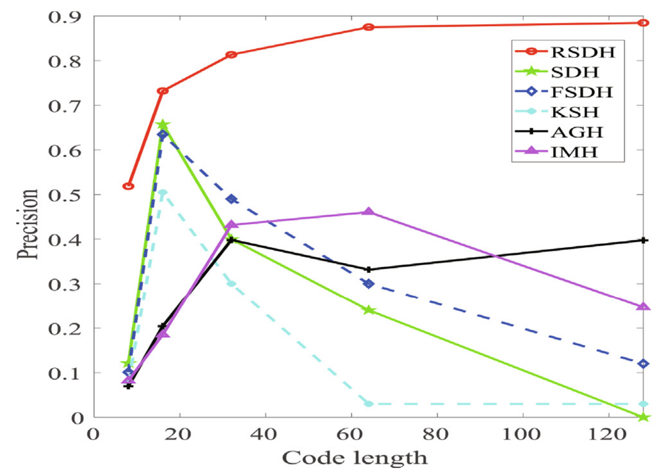


Fig. 12. The precision versus code length (8 to 128) on the ORL.

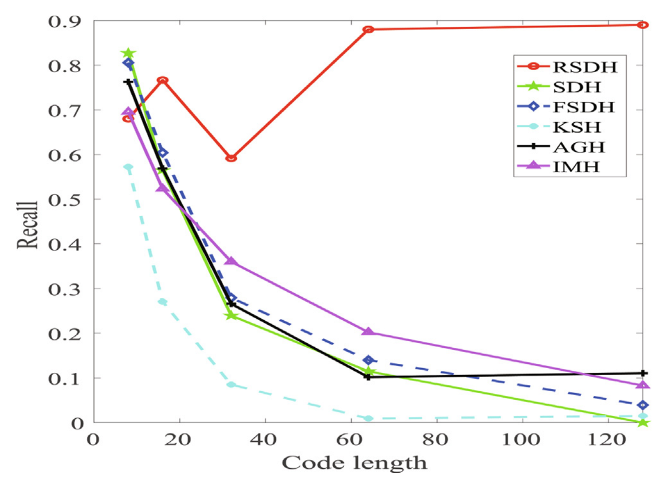


Fig. 13. The recall versus code length (8 to 128) on the ORL.

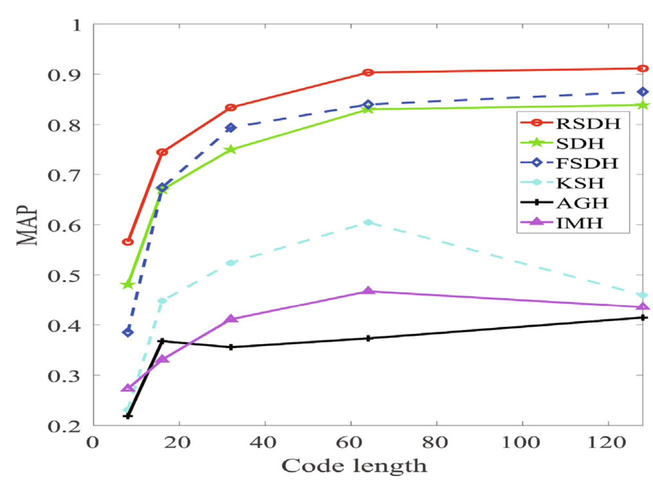


Fig. 14. The MAP versus code length (8 to 128) on the ORL.

The time complexity of B is $O(lc^2 + lc)$.

Compute W**:**

Similarly, problem (6) can also be transformed into the following two problems

$$\max_{P_{ij}} \sum_{i,j} \left\{ P_{ij} \left(\frac{Y - BW}{\gamma}\right)^2_{ij} - f^*(P_{ij}) \right\} \tag{23}$$

and

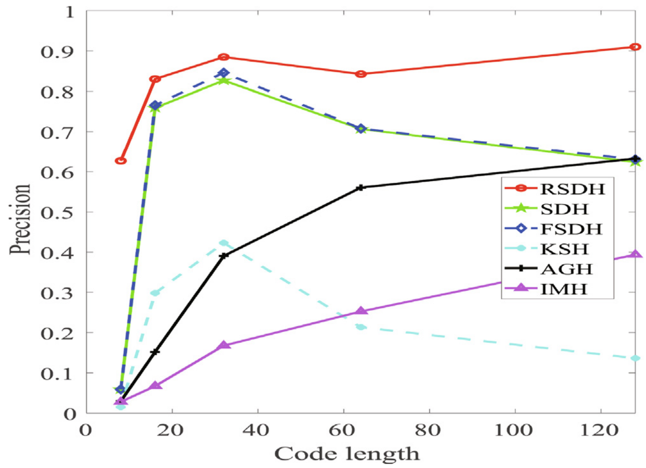


Fig. 15. The precision versus code length (8 to 128) on the COIL100.

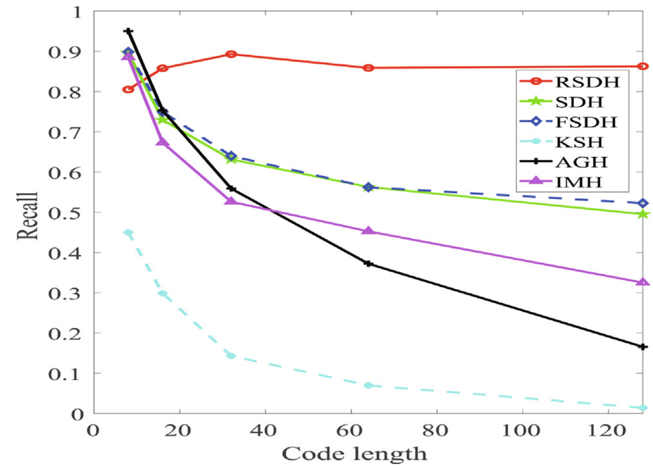


Fig. 16. The recall versus code length (8 to 128) on the COIL100.

$$\max_{W}\sum_{i,j}\left\{P_{ij}\left(\frac{Y-BW}{\gamma}\right)_{ij}^{2}\right\}-||W||_F^2 \tag{24}$$

For problem (23), we can also get the solution by $p^{*}=-1/(u+1)$. For problem (24), we have

$$\begin{aligned}\sum_{i,j}\left\{P_{ij}\left(\tfrac{Y-BW}{\gamma}\right)_{ij}^{2}\right\}-||W||_F^2 &= (Y-BW)K(Y-W^TB^T)-\gamma^2||W||_F^2\\ &= -2YKW^TB^T+BWKW^TB^T-\gamma^2||W||_F^2\\ &= -BKY+B^TKBW-\gamma^2W.\end{aligned} \tag{25}$$

where K represents the diagonal matrix of P. Thus, for problem (8), we have

$$W=(B^TKB-I\gamma^2)^{-1}B^TKY \tag{26}$$

The time complexity of W is $O(nl^2+ncl)$. Above all, we summarize the above-mentioned procedures in Algorithm 1.

Algorithm 1:RSDH

Input:X, l, t, λ,γ,ν
Output:hash code B.
Initialize B, Y and $\phi(x)$
Compute P in terms of (17).
Compute W in terms of (26).
Achieve maximum iterations (*i.e.* t):
  B: Compute B in terms of (22).
  F: Compute P in terms of (17).
  G: Compute W in terms of (26).

3.1. Robustness analysis

To better illustrate the robustness of RSDH, we compare several popular methods (*e.g.* SDH [27], FSDH [40], KSH [28], AGH [20] and IMH [23]) by using the sample-weighted procedure interpretation. [41].

Assumed that $F(X)$ represents the objection function of these problems and $f(t)=F(tX)$. Thus, the optimization of these problems is to find an X, bring $f\prime(1)=0$, in where $f\prime(t)$ represents the derivative of $f(t)$. Assume that $c(Y_{ij},X)=(Y-X)_{ij}(-X)_{ij}$ is the contribution of the jth entry of the ith training sample for the optimization procedure. For the sample-weighted procedure, we can explain it as the weighted contribution with about the noise.

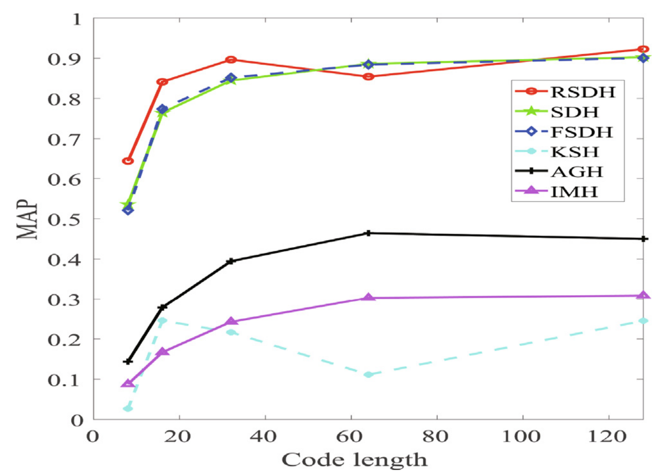


Fig. 17. The MAP versus code length (8 to 128) on the COIL100.

We compare the robustness of these problems by $f\prime(1)$ in Table 1. Note that robustness algorithms should allocate a small weight to the sample with large noise. Thus, according to Table 1, we can obtain the following conclusion: (1) KSH, SDH and FSDH are less robust than other methods for noise and outliers because they have constant weights. (2) RSDH is very robust to outliers and noise when the outliers and noise are too large, its weights will drop directly to zero.

4. Experiments

4.1. Experimental Setup

4.1.1. Compared Methods

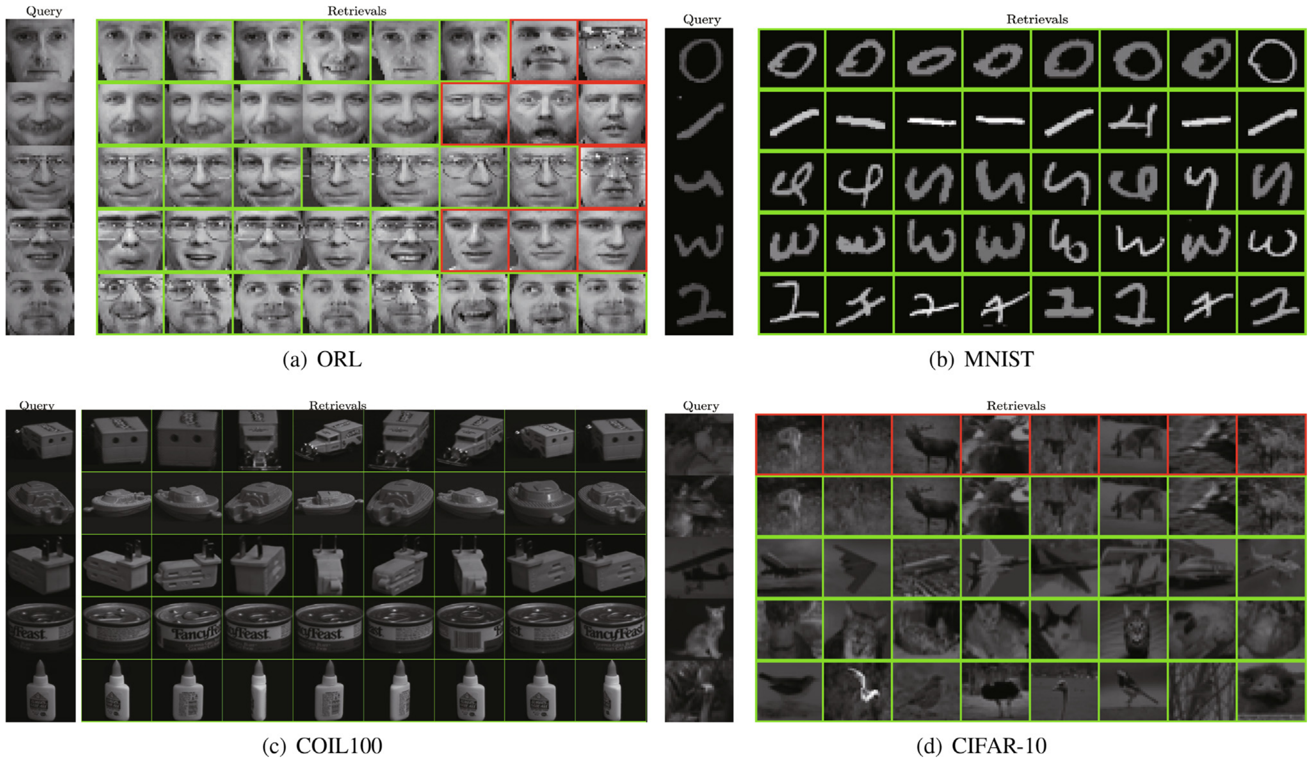
RSDH is compared with the other five methods (*i.e.* SDH [27], FSDH [40], KSH [28], AGH [20] and IMH [23]) in terms of image retrieval. All methods are summarized as follows:

RSDH is a robust discrete hashing method to reduce the noise affection and the quantization error.

SDH[1] learns directly the hash code without relaxations.

FSDH is based on SDH, which accelerates the convergence of the algorithm and improves performance.

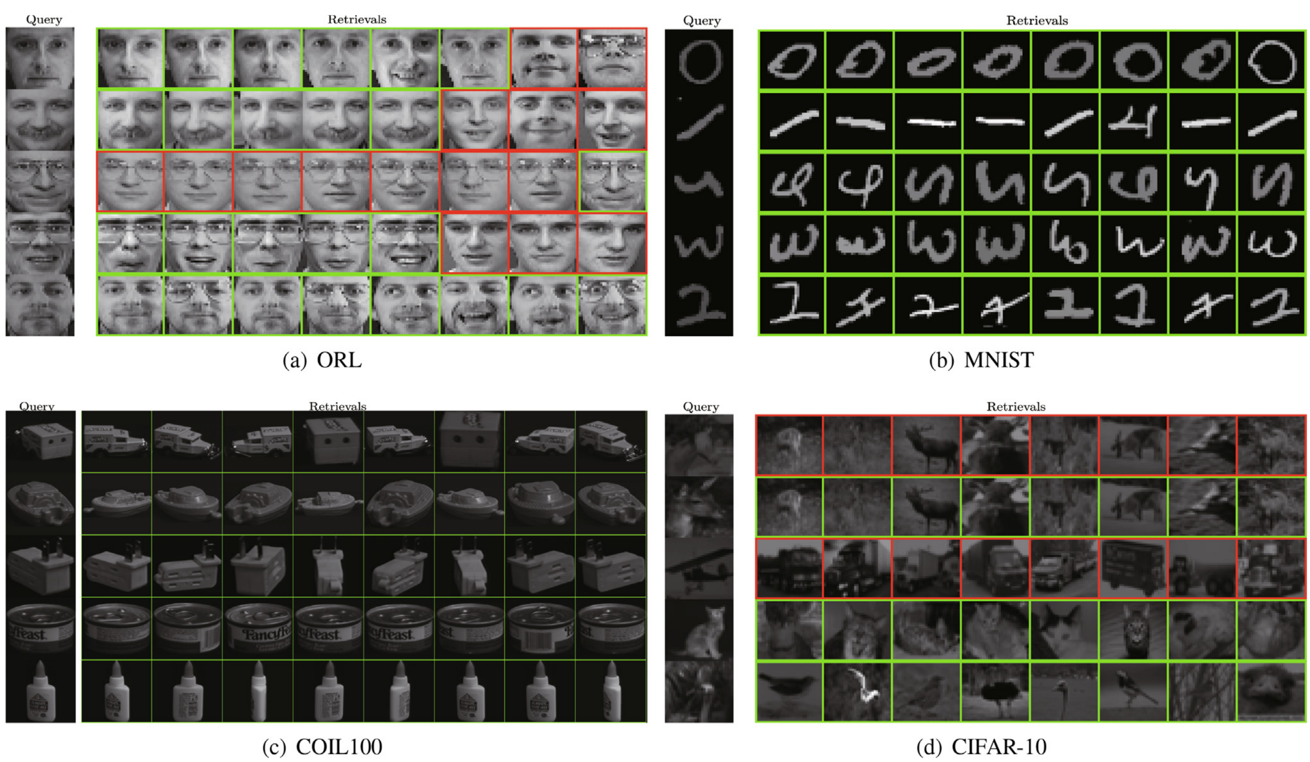
KSH relaxes the binary constraints to solve a successive optimization problem.



(a) ORL (b) MNIST

(c) COIL100 (d) CIFAR-10

Fig. 18. Retrievals achieved by RSDH.



(a) ORL (b) MNIST

(c) COIL100 (d) CIFAR-10

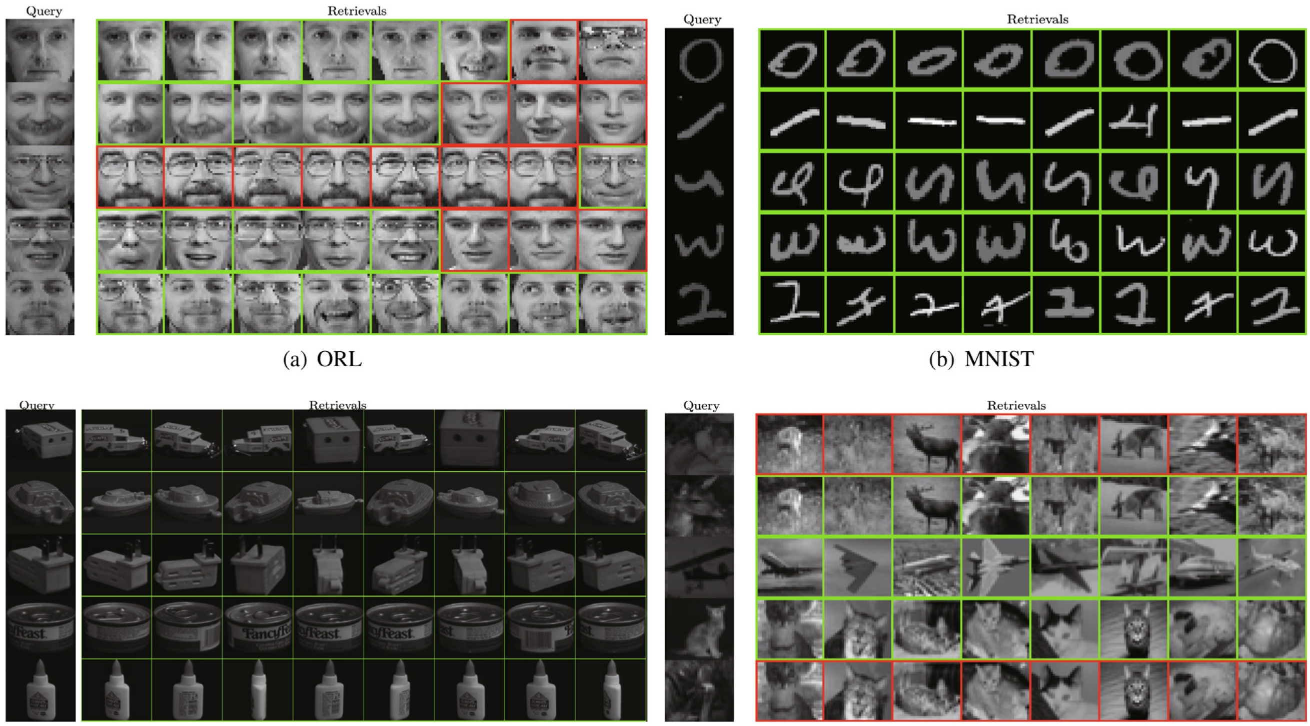
Fig. 19. Retrievals achieved by FSDH.

AGH utilizes the anchor graphs to construct a sparse adjacent graph.

IMH learns compact binary embeddings on the intrinsic manifolds.

4.1.2. Compared datasets

Four various datasets are used to assess the performance of the aforementioned methods. These datasets can be depicted as follows:



(a) ORL (b) MNIST

(c) COIL100 (d) CIFAR-10

Fig. 20. Retrievals achieved by SDH.

CIFAR-10[2] contains 60,000 manually mark images, in which each class contain 6000 instances. The GIST feature vector of dimension 512 represents each image. For this experiment, 59,000 data samples are utilized randomly to train and the rest is utilized to test.

MNIST[3] includes 70,000 images, and each instance consists of 784-dimensional. The handwritten digit consists of '0' to '9'. Each instance is a grayscale handwritten digital image that is cropped and normalized to 28×28.

ORL[4] contains sets of face images from 40 disparate persons, each person has 10 face images, totaling 400. These face images are taken with various times, various illumination, various face expression and various face detail.

COIL100[5] includes 20 different objects, where each object was photographed at different angles and has 72 postures.

To better illustrate the four datasets, we show some sample images in Figs. 1–4, respectively.

4.1.3. Parameters setup

To achieve a better retrieval performance, it is important to choose suitable parameters. These empirical parameters are shown in Table 1. Then, we will test suitable v and γ to achieve the ideal result. Fig. 5 means the retrieving recall, precision and MAP with different v and γ on four datasets. These figures denote three-dimensional images, in which the x-axis denotes v, y-axis denotes γ and the third dimension denotes retrieval results (*i.e.* precision, recall and MAP) showed by different colors. In terms of Fig. 5, we conclude that: 1) For ORL and COIL100, the larger γ and smaller v cause better retrieval performance. 2) For CIFAR-10 and MNIST, the smaller v and γ cause better retrieval effectiveness (see Table 2).

4.2. Image retrieval

The recall, precision Hamming ranking (MAP) are utilized as retrieval indices. The specific performance of our method is shown in Tables 3–6 and Figs. 6–17.

4.2.1. Cifar-10

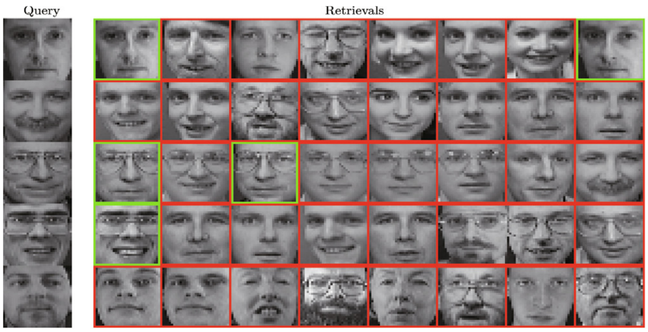
A part of the retrieval results are presented in Table 3. RSDH is superior to all other methods. SDH and FSDH perform similarly no matter what indices. To clearly show the retrieval effect, the precision, recall and MAP are shown in Figs. 6 to Fig. 8, respectively.

- As the code length increases, RSDH achieves better image retrieval performance. However, the other methods perform worse continuously. Apparently, RSDH leads to the relatively better image retrieval results than other methods.
- The performance of most methods on degraded as code length increases on the precision and recall, except RSDH.
- RSDH is always superior to all other methods on the recall when the code length exceeds 8.
- For SDH, FSDH, KSH, AGH and IMH, the retrieved result performs ideally on the recall aspect when the code length is equal to 8.
- Although the effectiveness of all methods tends to be stable on the MAP, RSDH always can be far ahead of other methods.

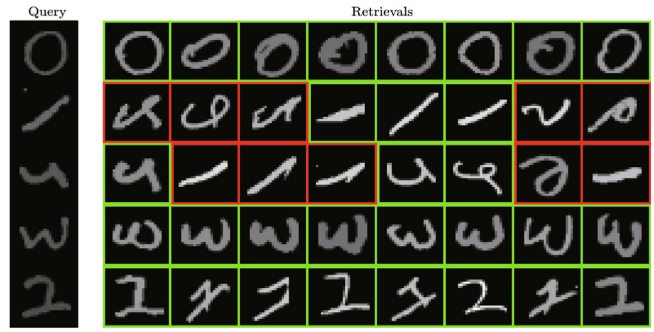
4.2.2. Mnist

Some experimental results are presented in Table 4. SDH and FSDH achieve the satisfactory precision and MAP, however, IMH, AGH and KSH perform poorly in recall. By comparison, RSDH outperforms all other methods in any indices. The precision, recall and MAP curves are shown in Figs. 9 to 11, respectively. Based on these Figures, we conclude that:

[2] http://www.cs.toronto.edu/ kriz/cifar.html
[3] http://yann.lecun.com/exdb/mnist/
[4] http://www.cad.zju.edu.cn/home/dengcai/Data/ORL/ORL.mat
[5] http://www.cad.zju.edu.cn/home/dengcai/Data/COIL100/COIL100.mat



(a) ORL

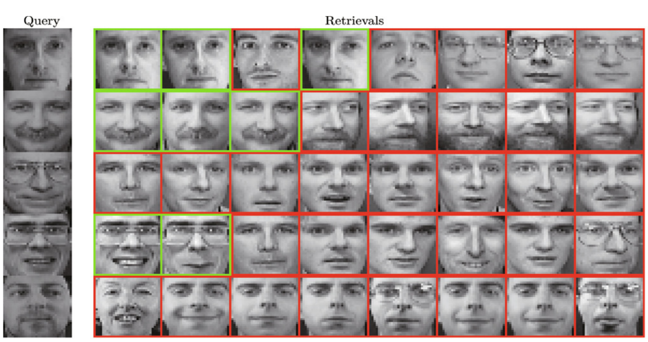


(b) MNIST

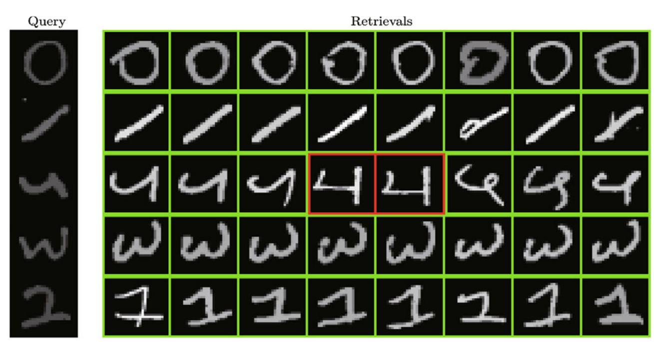
(c) COIL100

(d) CIFAR-10

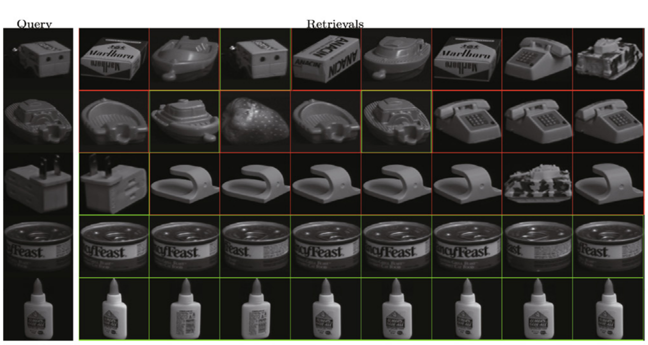
Fig. 21. Retrievals achieved by KSH.



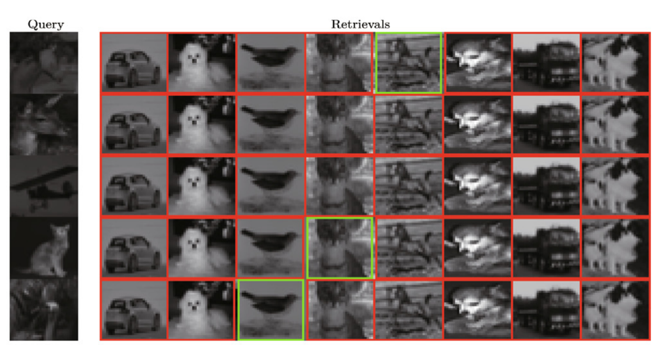
(a) ORL



(b) MNIST

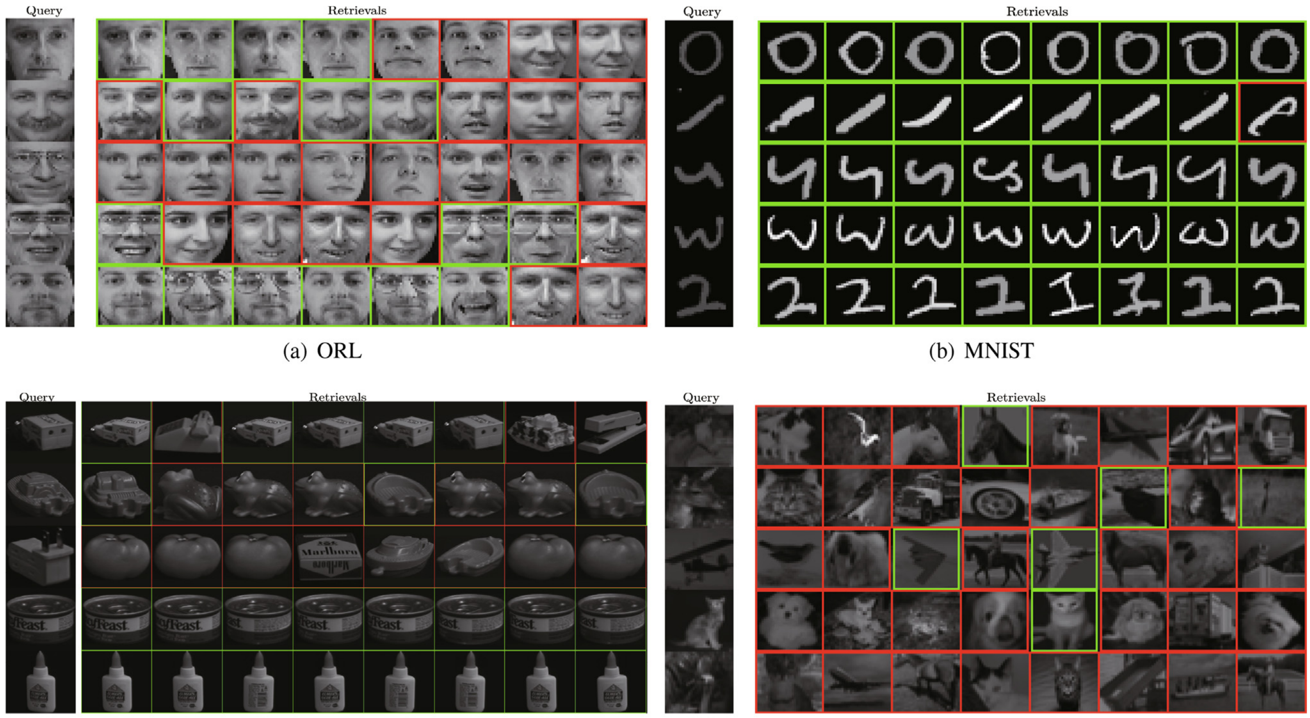


(c) COIL100



(d) CIFAR-10

Fig. 22. Retrievals achieved by IMH.



(a) ORL (b) MNIST

(c) COIL100 (d) CIFAR-10

Fig. 23. Retrievals achieved by AGH.

- In general, RSDH outperforms all other methods, no matter what the code length is.
- When the code length is 32, the effectiveness of FSDH and SDH are very close to RSDH.
- As code length increases, RSDH is generally stable in performance. However, some of the methods demonstrate performance degradation, such KSH.
- Although the performance of some methods tends to be stable on the MAP, RSDH always can be far ahead of other methods.
- When the code length is equal to 8, the performance of all methods was unsatisfactory on the precision, except RSDH.
- When the code length is equal to 8, all methods are very close on the recall. When the code length is 128, AGH is nearly 0.

4.2.3. Orl

A part of experimental results are presented in Table 5, RSDH performs best in image retrieval indices. To better explain the retrieval performance, Figs. 12 to 14 show the result of precision, recall and MAP. In terms of these Figures, we can obtain the following findings:

- When the code length is 8 bit, RSDH performs poorly than SDH, FSDH, AGH and IMH in the recall.
- RSDH outperforms other methods in two evaluating indicators (*i.e.* precision and MAP), no matter what the code length is.
- In general, the performance of RSDH performs the best in large code length.
- As code length increases, RSDH performs the best in recall.
- Although the performance of some methods tends to be stable on the MAP, RSDH always can be far ahead of other methods.

4.2.4. Coil100

Some experimental results are presented in Table 6. Obviously, when the code length is 64 bit, RSDH outperforms other methods in three evaluating indicators. The precision, recall and MAP versus the number of code length is shown in Figs. 15 to 17. According to these Figures, we summarized as below:

- When the code length is equal to 8, RSDH performs poorly than SDH, FSDH, AGH and IMH in recall.
- When the code length increases, the precision and recall of SDH, FSDH and KSH has been continuously declining.
- In general, when the code length is large, RSDH performs the best.
- Although the performance of some methods tends to be stable on the MAP, RSDH keeps ahead in general.
- When the code length is equal to 8, most methods are very close on the recall, except KSH.

4.2.5. Retrieval Examples

To illustrate intuitively the superiority of our method on image retrieval. Figs. 18–23 show some retrieval examples by RSDH, SDH, FSDH, KSH, AGH and IMH, respectively. For each algorithm, five queries and 8 top retrievals are utilized on CIFAR-10, MINIST, ORL and COIL100. If the query image is the same as the example image, the border presents green, otherwise, it presents red.

According to these retrieval results, RSDH performs the best than other methods, especially, this difference is particularly obvious on ORL and CIFAR-10.

Remarks: For this experiment, Hanming distance is utilized to measure similarity and hash bits are set to 64.

5. Conclusion

In this paper, Based on Cauchy loss and SDH, we present a more robust supervised hashing framework called Robust Supervised Discrete Hashing (RSDH) to reduce the decomposed error and eliminate some outliers and noise of the hashing codes. Due to the fact that the Cauchy loss is a non-quadratic function, the convex conjugation theory is utilized. The non-quadratic objective function is rewritten into an augmented loss function by adding an auxiliary variable. Then, the DCC algorithm is used to solve the augmented loss function. To illustrate the superiority of this method, we go on plentiful experiments on four various datasets and obtained considerable achievement.

Selecting an appropriate model is very important for large-scale data retrieval with noise or outliers in the data set of the real world. In the following work, we continue to enhance the robustness of this method framework.

CRediT authorship contribution statement

Yao Xiao: wrote the original draft. **Wei Zhang:** proposed the methodology of the paper. **Xiangguang Dai:** reviewed and modified the paper. **Xiangqin Dai:** implemented the algorithm of the paper and designed the software. **Nian Zhang:** proposed the methodology of the paper.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

This work is supported by Foundation of Chongqing Municipal Key Laboratory of Institutions of Higher Education ([2017]3), Foundation of Chongqing Development and Reform Commission (2017 [1007]), Scientific and Technological Research Program of Chongqing Municipal Education Commission (Grant Nos. KJQN201901218, KJQN201901203 and KJQN201801214), Natural Science Foundation of Chongqing (Grant Nos. cstc2019jcyj-bshX0101 and cstc2018jcyjA2453), Foundation of Chongqing Three Gorges University and National Science Foundation (NSF) (2011927) and DoD (W911NF1810475).

References

[1] H.V. Jagadish, B.C. Ooi, K.L. Tan, et al., idistance: An adaptive b+-tree based indexing method for nearest neighbor search, ACM Transactions on Database Systems (TODS) 30 (2) (2005) 364–397.
[2] C.S. Jensen, D. Lin, et al., Query and update efficient b+-tree based indexing of moving objects 30 (2004) 768–779..
[3] J. Wang, H.T. Shen, et al., Hashing for similarity search: A survey, arXiv preprint arXiv:1408.2927 (2014)..
[4] F. Shen, X. Zhou, et al., A fast optimization method for general binary code learning, IEEE Transactions on Image Processing 25 (12) (2016) 5610–5621.
[5] J. Masci, M.M. Bronstein, et al., Multimodal similarity-preserving hashing, IEEE Transactions on Pattern Analysis and Machine Intelligence 36 (4) (2013) 824–830.
[6] D. Song, W. Liu, et al., Top rank supervised binary coding for visual search (2015) 1922–1930..
[7] K. Li, G.-J. Qi, et al., Linear subspace ranking hashing for cross-modal retrieval, IEEE Transactions on Pattern Analysis and Machine Intelligence 39 (9) (2016) 1825–1838.
[8] J. Song, Y. Yang, et al., Multiple feature hashing for real-time large scale near-duplicate video retrieval (2011) 423–432..
[9] S. Zhang, M. Yang, et al., Query specific rank fusion for image retrieval, IEEE Transactions on Pattern Analysis and Machine Intelligence 37 (4) (2014) 803–815.
[10] S. Korman, S. Avidan, Coherency sensitive hashing, IEEE Transactions on Pattern Analysis and Machine Intelligence 38 (6) (2015) 1099–1112.
[11] P. Das, K. Karthik, et al., A robust alignment-free fingerprint hashing algorithm based on minimum distance graphs, Pattern Recognition 45 (9) (2012) 3373–3388.
[12] Y. Zhang, H. Lu, L. Zhang, et al., Combining motion and appearance cues for anomaly detection, Pattern Recognition 51 (2016) 443–452.
[13] J. Lu, V.E. Liong, et al., Learning compact binary face descriptor for face recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 37 (10) (2015) 2041–2056.
[14] J. Lu, V.E. Liong, et al., Cost-sensitive local binary feature learning for facial age estimation, IEEE Transactions on Image Processing 24 (12) (2015) 5356–5368.
[15] Z. Chen, J. Lu, et al., Nonlinear structural hashing for scalable video search, IEEE Transactions on Circuits and Systems for Video Technology 28 (6) (2017) 1421–1433.
[16] H. Zhang, F. Shen, et al., Discrete collaborative filtering (2016) 325–334..
[17] M. Datar, N. Immorlica, et al., Locality-sensitive hashing scheme based on p-stable distributions (2004) 253–262..
[18] S. Korman, S. Avidan, Coherency sensitive hashing, IEEE Transactions on Pattern Analysis and Machine Intelligence 38 (6) (2015) 1099–1112.
[19] Y. Weiss, A. Torralba, et al., Spectral hashing. 1 (2) (2008) 4..
[20] W. Liu, J. Wang, et al., Hashing with graphs (2011) 1–8..
[21] Y. Gong, S. Lazebnik, Iterative quantization: A procrustean approach to learning binary codes (2011) 817–824..
[22] Y. Gong, S. Lazebnik, et al., Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval, IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (12) (2013) 2916–2929.
[23] F. Shen, C. Shen, et al., Inductive hashing on manifolds (2013)..
[24] L. Van der Maaten, G. Hinton, Visualizing data using t-sne, Journal of Machine Learning Research 9 (11) (2008).
[25] J. Wang, S. Kumar, et al., Semi-supervised hashing for large-scale search, IEEE Transactions on Pattern Analysis and Machine Intelligence 34 (12) (2012) 2393–2406.
[26] B. Kulis, T. Darrell, Learning to hash with binary reconstructive embeddings 22 (2009) 1042–1050.
[27] F. Shen, C. Shen, et al., Supervised discrete hashing (June 2015)..
[28] W. Liu, J. Wang, et al., Supervised hashing with kernels (2012) 2074–2081..
[29] G. Lin, C. Shen, et al., Supervised hashing using graph cuts and boosted decision trees, IEEE Transactions on Pattern Analysis and Machine Intelligence 37 (11) (2015) 2317–2331.
[30] G. Lin, C. Shen, et al., Fast supervised hashing with decision trees for high-dimensional data (2014) 1963–1970..
[31] C. Strecha, A. Bronstein, M. Bronstein, P. Fua, Ldahash: Improved matching with smaller descriptors, IEEE Transactions on Pattern Analysis and Machine Intelligence 34 (1) (2012) 66–78.
[32] M. Belkin, P. Niyogi, Laplacian eigenmaps for dimensionality reduction and data representation, Neural computation 15 (6) (2003) 1373–1396.
[33] Y. Luo, Y. Yang, et al., Robust discrete code modeling for supervised hashing, Pattern Recognition 75 (2018) 128–135.
[34] X. Dai, X. Su, et al., Robust manhattan non-negative matrix factorization for image recovery and representation, Information Sciences 527 (2020) 70–87.
[35] T.T. Wu, K. Lange, et al., Coordinate descent algorithms for lasso penalized regression, Annals of Applied Statistics 2 (1) (2008) 224–244.
[36] A. Liutkus, D. Fitzgerald, et al., Cauchy nonnegative matrix factorization (2015) 1–5..
[37] N. Guan, T. Liu, et al., Truncated cauchy non-negative matrix factorization, IEEE Transactions on Pattern Analysis and Machine Intelligence 41 (1) (2017) 246–259.
[38] S. Boyd, S.P. Boyd, et al., Convex optimization (2004)..
[39] O.L. Mangasarian, Nonlinear programming (1994)..
[40] J. Gui, T. Liu, et al., Fast supervised discrete hashing, IEEE Transactions on Pattern Analysis and Machine Intelligence 40 (2) (2017) 490–496.
[41] T. Liu, D. Tao, On the robustness and generalization of cauchy regression (2014) 100–105..

Xiangguang Dai receieved the B.S. degree in computer science and technology and M.S. degree in computer Softerware and technology from College of Mobile Telecommunications, Chongqing University of Posts and Telecommunications in 2008 and Chongqing University in 2012, respectively, and the Ph.D degree in applied mathematics in Southwest University in 2018. His current search interests include neurodynamic optimization, nonnegative matrix factorization and hashing learning.

Xiangqin Dai graduated in social work and management from Shanxi University of Finance and Economics,Shan xi,China,in 2008,and received the M.Sc. degree in agric ultural engineering and information technology from t he Chongqing Three Gorges University,Chongqing,Chin a,in 2020. Her current research interests include net work security and artificial intelligence.

Dr. Nian Zhangis a Professor in the Department of Electrical and Computer Engineering at the University of the District of Columbia (UDC), Washington, D.C., USA. She received her Ph.D. degree in Computer Engineering from Missouri University of Science & Technology, USA. Her research interests include computational intelligence, machine learning, deep learning, supervised and unsupervised learning, classification, clustering, and optimization, neurodynamic optimization, and various application fields including big data science, time series prediction, biomedical engineering, and autonomous robot navigation. She was awarded numerous federal grants from the National Science Foundation, Department of Defense, and National Institutes of Health as the PI/Co-PI, accumulating over $4.5 million in research grant funds. Dr. Zhang serves as an *Associate Editor* for the IEEE Transactions on Cybernetics, IEEE Transactions on Neural Networks and Learning Systems, and Knowledge-based Systems. She also serves on the Editorial Board for the Complex & Intelligent Systems. In addition, Dr. Zhang served as a *Guest Editor* for the Knowledge-Based Systems, Control Engineering Practice, Complex & Intelligent Systems, International Journal of Computational Intelligence Systems, Computational Intelligence and Neuroscience, Cybernetics and Systems, International Journal on Artificial Intelligence Tools, Environmental Research Letters, Sensors, and International Journal of Systems, Control and Communications. Further, Dr. Zhang serves as the Vice Chair of the IEEE Computational Intelligence Society (CIS) Emergent Technologies Technical Committee and the Vice Chair of the IEEE CIS Adaptive Dynamic Programming and Reinforcement Learning Technical Committee. She regularly serves as the Program Chair, Publications Chair, and Special Sessions Chair of the IEEE technically co-sponsored conferences such as ICACI, ICICIP, ICIST, and ISNN. Dr. Zhang received the UDC's faculty recognition awards for Excellence in Research Award, Excellence in Teaching Award, and Outstanding Undergraduate Research Mentorship Award in three consecutive years.

Yao Xiao received the bachelor's degree in software engineering from the Yitong College, Chongqing University of Posts and Telecommunications, in 2020, and pursuing the master's degree in agricultural engineering and information technology from Chongqing Three Gorges University. His main research interests include neural networks, optimization algorithms, and nonnegative matrix factorization.

Wei Zhang receieved the B.S. degree in computer science and technology and M.S. degree in computer Softerware and technology from Chongqing University in 1992 and Southwest Normal University in 1999, respectively, and the Ph.D degree in Computer software and theory major in Chongqing University in 2003. His current search interests include neurodynamic optimization, nonnegative matrix factorization and hashing learning.