Greedy Asymmetric Deep Supervised Hashing for Image Retrieval: Postprint

Authors: Zhao Xinxin, Li Yang, Robust seedlings, Wang Jiabao, Zhang Rui

Date: 2022-05-10T00:00:00+00:00

Abstract

In recent years, deep supervised hashing retrieval methods have been successfully applied in numerous image retrieval systems. However, existing methods still suffer from several limitations: First, most deep hashing learning methods adopt a symmetric strategy to train the network, but this training strategy is usually time-consuming and difficult to apply in large-scale hashing learning processes; Second, there exists a discrete optimization problem in the hashing learning process, and existing methods relax this problem but cannot guarantee obtaining the optimal solution. To address the aforementioned issues, we propose a greedy asymmetric deep supervised hashing image retrieval method, which fully combines the advantages of greedy algorithms and asymmetric strategies to further improve hashing retrieval performance. This paper compares with 17 state-of-the-art methods on two commonly used datasets. On the CIFAR-10 dataset under 48-bit conditions, the mAP improves by 1.3% compared to the best-performing method; on the NUS-WIDE dataset under all bits, the mAP improves by an average of 2.3%. Experimental results on both datasets demonstrate that the proposed method can further improve hashing retrieval performance.

Full Text

Preamble

Greedy-Asymmetric Deep Supervised Hashing for Image Retrieval

Zhao Xinxin, Li Yang, Miao Zhuang†, Wang Jiabao, Zhang Rui
(Command & Control Engineering College, Army Engineering University of PLA, Nanjing 210007, China)

Abstract: In recent years, deep supervised hashing retrieval methods have been successfully applied to numerous image retrieval systems. However, existing approaches still suffer from several limitations. First, most deep hashing learning methods employ symmetric strategies for network training, which is typically time-consuming and difficult to scale to large-scale hashing learning processes. Second, the hashing learning process involves discrete optimization problems that existing methods address through relaxation, yet this makes it difficult to guarantee optimal solutions. To overcome these issues, this paper proposes a greedy-asymmetric deep supervised hashing method for image retrieval that fully leverages the advantages of both greedy algorithms and asymmetric strategies to further improve hashing retrieval performance. We compare our method with 17 state-of-the-art approaches on two commonly used datasets. On CIFAR-10 at 48 bits, our method achieves a 1.3% mAP improvement over the best-performing existing method. On NUS-WIDE across all bit lengths, it achieves an average mAP improvement of 2.3%. Experimental results on both datasets demonstrate that the proposed method can further enhance hashing retrieval performance.

Keywords: asymmetric strategy; greedy algorithm; supervised hashing; image retrieval

0 Introduction

With the explosive growth of image data, finding required information from massive datasets has become a critical challenge. In large-scale image retrieval, hashing methods [1] have gained widespread attention for approximate nearest neighbor search based on hash features due to their computational and storage efficiency [2, 3].

Numerous hashing retrieval methods have been proposed in recent years, which can be categorized into data-independent [4] and data-dependent approaches [5]. Data-independent methods primarily rely on random projections to construct hash functions without depending on training data, resulting in relatively low retrieval accuracy [6]. Data-dependent methods leverage various machine learning techniques to learn hash functions. Compared to data-independent methods, data-dependent approaches can achieve higher accuracy with shorter hash codes [7], and thus have been more widely adopted.

Motivated by the success of deep neural networks in various tasks such as image classification [8], object detection [9], and face recognition [10, 11], researchers have attempted to address hashing problems through deep learning, leading to the development of deep supervised hashing methods. These methods extract image features through deep neural networks while simultaneously performing hash learning [12, 13], integrating feature extraction and hash learning into a unified end-to-end framework [14] that significantly improves retrieval performance compared to traditional methods.

Despite substantial progress, deep supervised hashing methods still face several limitations [15, 16]. Most existing deep supervised hashing methods adopt symmetric strategies during training, such as CNNH [17] (Convolutional Neural Networks Hashing), DPSH [18] (Deep Pairwise Supervised Hashing), and DHN

[19] (Deep Hashing Network). Symmetric strategy refers to using the same deep neural network to extract deep features for both query samples and database samples, followed by the same deep hash function to generate hash codes for both. However, training with symmetric strategies is typically time-consuming, making it difficult to efficiently leverage supervision information for large-scale datasets. For instance, the storage and computational cost of DPSH [18] is $O(n^2)$, where n is the number of database samples, while DTSH [20] (Deep Supervised Hashing with Triplet) incurs even higher training costs. Due to the time-consuming nature of symmetric strategy training, most existing methods only sample a subset from the entire database to construct the training set for hash function learning, discarding the remaining database samples. Although this accelerates network training, it leads to insufficient utilization of supervision information [21].

Early research on asymmetric hashing [22, 23] employed asymmetric distance metrics to preserve similarity between images, where binary hash codes for both training and query sets were generated by the same hash function that needed to be learned. Inspired by this asymmetric concept, subsequent works proposed several deep asymmetric hashing methods, including Deep Asymmetric Pairwise Hashing [24], Nonlinear Asymmetric Multi-valued Hashing [25], Collaborative learning for extremely low bit Asymmetric Hashing [26], Deep Asymmetric Hashing with Dual Semantic Regression and Class Structure Quantization [27], and Asymmetric Supervised Deep Discrete Hashing for Image Retrieval [28]. Among them, the ADSH [21] method employs an asymmetric approach for deep hash learning. Asymmetric strategy means that feature extraction is performed only for query samples, not for database samples. During deep hash function learning, only query sample hash codes are learned, while database sample hash codes are learned directly. This asymmetric approach effectively addresses the problem of insufficient supervision information utilization.

Although ADSH can fully leverage supervision information and enable efficient network training through its asymmetric approach, it faces a challenge: during hash code learning, the sign function (which is non-differentiable) is required to map deep features to binary hash codes, making the optimization problem NP-hard. ADSH addresses this by using the tanh function as a relaxation during training [29] and adding a penalty term to the loss function to generate features that are as discrete as possible [18, 30], then applying the sign function in the testing phase to obtain true binary hash codes. While this enables network training, it introduces quantization errors.

To better solve discrete optimization problems, the Greedy Hash [31] method utilizes greedy principles by strictly applying the sign function during forward propagation to maintain discrete constraints on network outputs, while during backpropagation, gradients from the hash layer are fully transmitted to previous layers, thereby avoiding gradient vanishing.

Inspired by both ADSH and Greedy Hash, this paper proposes a greedy-asymmetric deep supervised hashing method for image retrieval. By simultaneously applying greedy algorithms and asymmetric strategies to the hash function learning process, our approach can both fully utilize supervision information and better solve discrete optimization problems. Specifically, our main contributions include three aspects:

a) We propose a greedy-asymmetric deep supervised hashing method for image retrieval that fully combines the advantages of greedy algorithms and asymmetric strategies to further improve hash retrieval performance.

b) We design a greedy-asymmetric pairwise loss function consisting of two components: greedy loss and asymmetric pairwise loss. The greedy loss addresses discrete optimization by maintaining discrete constraints on network outputs during forward propagation while fully transmitting gradients from the hash layer to the network output layer during backpropagation. The asymmetric pairwise loss improves hash code learning efficiency by employing asymmetric strategies to learn query samples and database samples differently.

c) Experimental results on two datasets demonstrate that the greedy-asymmetric deep supervised hashing method can further improve hash retrieval performance.

1 Feature Extraction

[Figure 1: see original paper] illustrates the model architecture of our greedy-asymmetric deep supervised hashing method. The model comprises two main components: a feature extraction module and a loss function module. The feature extraction module extracts deep image features through a backbone network, while the loss function module maps these deep features to hash codes through hash learning. As shown in Figure 1, training images first obtain deep features during the feature extraction stage, then acquire binary hash codes through the hash layer, and finally are guided by our designed loss function to generate hash codes that preserve image similarity. By integrating both components into a unified end-to-end structure, our model enables mutual feedback between modules during training, resulting in more robust hash encoding.

Notably, our designed loss function includes two components: greedy loss and asymmetric pairwise loss. As indicated by the greedy loss in Figure 1, after obtaining image hash features, the sign function directly maps them to binary hash codes, and the greedy principle solves the discrete optimization problem (detailed implementation in Section 2.1). As shown by the asymmetric pairwise loss in Figure 1, this component simultaneously utilizes supervision information from both query images and database images to train the network, enabling the learning of binary hash codes for database images during training (detailed implementation in Section 2.2).

For feature extraction, we employ AlexNet [8] as the backbone network, which consists of 5 convolutional layers and 3 fully connected layers (structure detailed in Table 1). To obtain image hash features, we add a hash coding layer after the FC3 layer of AlexNet, which maps deep features to $\mathbb{R}^c$ space, where c is the hash code length.

Let $X = \{x_i\}_{i=1}^m$ denote the query image set, and $Z = \{z_i\}_{i=1}^n$ denote the database image set containing n samples. After processing through the backbone network, we obtain hash features $h_i = f(x_i; \theta)$ for query images and $v_i = f(z_i; \theta)$ for database images, where $f(\cdot)$ represents the backbone network function and θ denotes its parameters.

2 Loss Function

As shown in the loss function stage of Figure 1, we design a loss function to learn optimal hash codes, expressed as: $L = L_1 + \lambda L_2$, where L_1 is the greedy loss, L_2 is the asymmetric pairwise loss, and λ is a hyperparameter. The greedy loss L_1 addresses gradient vanishing issues during optimization [30], while the asymmetric pairwise loss L_2 fully utilizes database label information and enables efficient network training [21].

2.1 Greedy Loss

To obtain image hash codes, the sign function is typically applied after the hash coding layer to map deep features to binary hash codes. However, since the sign function is non-differentiable, the optimization becomes an NP-hard problem. Traditional methods use tanh or sigmoid functions for relaxation [29], which enables network training but yields suboptimal solutions. To better address this issue, we propose utilizing greedy algorithms that strictly apply the sign function during forward propagation to maintain discrete constraints on network outputs, while during backpropagation, gradients from the hash layer are fully transmitted to previous layers, avoiding gradient vanishing and effectively solving the discrete optimization problem.

The core challenge of greedy loss is how to leverage greedy algorithms to solve discrete optimization problems. Let $b_i \in \{-1, 1\}^{c \times 1}$ represent the hash code of h_i. Specifically, our greedy loss adopts the cross-entropy loss form.

During discrete optimization, if we completely ignore discrete constraints and use gradient descent algorithms, we can obtain the solution for the $(t+1)$-th iteration as $b_i^{t+1} = b_i^t - \eta \frac{\partial L}{\partial b_i^t}$, where η represents the learning rate. However, the solution obtained using this equation does not satisfy $b_i^{t+1} \in \{-1, 1\}^{c \times 1}$. Without considering discrete constraints, the solution from this equation represents a continuous optimal solution. The greedy principle posits that the discrete point closest to the continuous optimal solution, as shown in equation (3), is the desired discrete optimal solution.

Equation (3) can be implemented through two steps: forward propagation and backpropagation. The forward propagation process is shown in equation (4), which implements a new hash layer using the sign function during forward propagation:

$$b_i^{t+1} = \mathrm{sign}(h_i^{t+1})$$

During backpropagation, we add a penalty term p to the greedy loss and make this penalty term approach zero as closely as possible. From $\frac{\partial p}{\partial h_i^{t+1}} = 0$, we can derive:

$$\frac{\partial L}{\partial h_i^{t+1}} = \frac{\partial L}{\partial b_i^{t+1}} \frac{\partial b_i^{t+1}}{\partial h_i^{t+1}} = \frac{\partial L}{\partial b_i^{t+1}}$$

This shows that during backpropagation, the greedy principle can fully transmit gradients from the hash layer to the network output layer. By separately implementing forward and backpropagation processes, our greedy loss effectively solves the discrete optimization problem and obtains precise hash codes.

2.2 Asymmetric Pairwise Loss

The asymmetric pairwise loss serves to train the network using asymmetric strategies during training. This not only enables full utilization of database supervision information but also allows efficient network training. Asymmetric strategy refers to processing query images and database images differently. For query images, deep features are extracted through the backbone network, and query image hash codes are generated using deep hash functions. For database images, their hash codes are learned directly.

To obtain hash codes that preserve similarity between query images and database images, a common approach [29] minimizes the difference between supervised similarity information of training images and database images and the similarity of their hash code inner products:

$$L_2 = \min_{\theta, V} \sum_{i=1}^{m} \sum_{j=1}^{n} [S_{ij} c - b_i^T v_j]^2$$

where $B = \{b_i\}_{i=1}^m \in \{-1, 1\}^{m \times c}$ represents the hash code set learned by the hash function for the training image set, and $V = \{v_j\}_{j=1}^n \in \{-1, 1\}^{n \times c}$ represents the directly learned hash code set for database images containing n samples. $S \in \mathbb{R}^{m \times n}$ is the similarity matrix where $S_{ij} = 1$ if two images are similar and $S_{ij} = -1$ otherwise.

Since equation (9) contains the sign function (which is non-differentiable), gradients for parameter θ cannot be directly backpropagated. We use the tanh function to replace the sign function for relaxation:

$$L_2 = \min_{\theta,V} \sum_{i=1}^{m} \sum_{j=1}^{n} [S_{ij}c - \tanh(f(x_i;\theta))^T v_j]^2$$

In practical applications, if only a database Z is given without a specified query set X, we can randomly sample m data points from Z as the query set. Let $\Omega = \{1, 2, \dots, n\}$ denote all indices of the database, and $\Gamma \subseteq \Omega$ denote the index set of sampled data. We also obtain the corresponding similarity matrix S_Ω for the sampled dataset. Equation (10) can then be rewritten as:

$$L_2 = \min_{\theta,V} \sum_{i\in\Gamma} \sum_{j\in\Omega} [S_{ij}c - \tanh(f(z_i;\theta))^T v_j]^2$$

To keep $\tanh(f(z_i;\theta))$ and v_i as close as possible, we add an additional constraint to equation (11), which is reasonable since v_i approximates the hash code of $\tanh(f(z_i;\theta))$. The constraint can be rewritten as:

$$L_2 = \min_{\theta,V} \sum_{i\in\Gamma} \sum_{j\in\Omega} [S_{ij}c - \tanh(f(z_i;\theta))^T v_j]^2 + \gamma \sum_{i\in\Omega} || \tanh(f(z_i;\theta)) - v_i||^2$$

where γ is a hyperparameter.

We employ an alternating optimization strategy to learn parameters θ and V in equation (12), which is also applicable to equation (10). Specifically, in each iteration, only one parameter is learned while other parameters remain fixed, and this process repeats for multiple iterations. The detailed parameter update process using alternating optimization strategy can be found in the learning algorithm section of ADSH [21].

3 Experiments

To validate the effectiveness of our proposed method, we conduct experiments on two commonly used public datasets and compare with 17 state-of-the-art methods.

3.1 Datasets

CIFAR-10 Dataset. The CIFAR-10 dataset [32] contains 10 classes, with 6,000 samples per class, totaling 60,000 color images of size 32$×$32. For CIFAR-10, two images are considered similar if they share the same label.

NUS-WIDE Dataset. The NUS-WIDE dataset [33] consists of 269,648 web images with tags. It is a multi-label dataset where each image may have multiple tags. We select images from the 10 most common classes. For NUS-WIDE, two images are defined as similar if they share at least one common tag.

3.2 Experimental Settings

Our model is implemented in PyTorch. The mini-batch size is set to 128, and Adam [34] is used as the optimizer with weight decay of 0.0005 and maximum iterations of 50. For data splitting:

- **CIFAR-10:** We randomly sample 200 images per class (2,000 total) as the training set, and 100 images per class (1,000 total) as the query set. The remaining 59,000 images (excluding the 1,000 query images) serve as the database.
- **NUS-WIDE:** We randomly sample 2,000 images from the database as the training set and 1,000 images as the query set, with all images excluding the query set serving as the database.

For evaluation metrics, we use the most common metrics in image retrieval: mAP (mean Average Precision) and PR (Precision-Recall).

Since different deep neural networks have varying feature extraction capabilities that affect retrieval performance, we use the AlexNet model pre-trained on ImageNet [35] as the backbone network for fair comparison.

3.3.1 Retrieval Performance Comparison

We evaluate our greedy-asymmetric deep supervised hashing method on CIFAR-10 and NUS-WIDE datasets, comparing with 17 methods including 7 traditional hashing methods and 10 deep hashing methods. Traditional methods include unsupervised ITQ [2] and supervised methods: Lin:Lin [36], LFH [37], FastH [38], SDH [39], COSDISH [40], KADGH [41]. Deep hashing methods include: DSH [1], DSDH [15], DHN [19], DPSH [18], DTSH [20], Greedy Hash [31], ADSH [21], SDNMSH [42], ASDDH [27], and TransHash [43].

Table 2 presents the image retrieval mAP accuracy on CIFAR-10 and NUS-WIDE, where bold indicates the best performance, underline indicates the second-best, and * indicates our reproduced results. The results show that supervised methods generally outperform unsupervised methods, and deep hashing methods outperform traditional methods. Our method significantly outperforms all other methods across all hash code lengths. This is because the proposed greedy-asymmetric loss better preserves image feature information, thereby improving hash retrieval performance. On CIFAR-10, our method achieves a 1.4% improvement over TransHash at 48 bits. On NUS-WIDE, it achieves an average 2.3% improvement across different bit settings. The results demonstrate strong performance on both datasets, with particularly notable improvements on the single-label CIFAR-10 dataset.

To further illustrate the superiority of our method, we plot PR curves at 12 bits on both datasets in Figure 2 [Figure 2: see original paper] and Figure 3 [Figure 3: see original paper]. The larger the area under the PR curve, the better the performance. The results clearly show that our greedy-asymmetric deep supervised hashing method significantly outperforms all other methods.

3.3.2 Hyperparameter Analysis

The greedy-asymmetric loss is defined as $L = L_1 + \lambda L_2$, where L_1 is greedy loss, L_2 is asymmetric pairwise loss, and λ is a hyperparameter. To analyze λ' s impact on retrieval performance, we conduct parameter analysis on CIFAR-10 and NUS-WIDE. Figures 4 [Figure 4: see original paper] and 5 [Figure 5: see original paper] show retrieval accuracy at different λ values (0.1, 1, 10, 100, 1000) for 12-bit and 48-bit settings. The results indicate that λ has minimal impact on retrieval performance for both datasets. Figure 4 shows that CIFAR-10 achieves good performance when $0.1 < \lambda < 1$, with the best results at $\lambda = 1$ across different bit settings. Figure 5 shows that NUS-WIDE achieves good performance when $0.1 < \lambda < 10$, also with optimal results at $\lambda = 1$. Based on these findings, we set $\lambda = 1$ for all experiments.

3.3.3 Retrieval Result Visualization

To further demonstrate our method' s effectiveness, Figure 6 [Figure 6: see original paper] visualizes the top-10 retrieval results for each class on CIFAR-10. The results show that our method achieves an average top-10 retrieval accuracy of 98% on CIFAR-10, further validating its excellent performance on single-label datasets.

4 Conclusion

To address the discrete optimization problem and insufficient utilization of supervision information in hash function learning, we propose a greedy-asymmetric deep supervised hashing method for image retrieval. By simultaneously applying greedy algorithms and asymmetric strategies to the hashing learning process, our approach can fully transmit hash layer gradients to the network output layer to solve discrete optimization problems while efficiently training the network with full supervision information. Comparative experiments with 17 methods on two public datasets validate the effectiveness of our approach. Although the proposed method achieves significant improvements in retrieval performance, there remains room for enhancement in multi-label retrieval tasks. Future work will focus on further improving the method' s adaptability to multi-label data.

References

[1] Liu H M, Wang R P, Shan S G, et al. Deep supervised hashing for fast image retrieval [C]// Proc of the 2016 IEEE Conference on Computer Vision and Pattern Recognition. Washington: IEEE Computer Society, 2016: 2064-2072.
[2] Gong Y C and Lazebnik S. Iterative quantization: a procrustean approach to learning binary codes [C]// Proc of the 2011 IEEE Conference on Computer

Vision and Pattern Recognition. Washington: IEEE Computer Society, 2011: 817-824.
[3] Andoni A and Razenshteyn I. Optimal data dependent hashing for approximate near neighbors [C]// Proc of the 47th Annual ACM Symposium on the Theory of Computing. New York: ACM Press, 2015.
[4] Andoni A and Indyk P. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions [C]// Proc of the 2006 Annual IEEE Symposium on Foundations of Computer Science. Piscataway, NJ: IEEE Press, 2006: 459-468.
[5] Kong W H and Li W J. Isotropic hashing [C]// Proc of the 2012 Neural Information Processing Systems. Cambridge, MA: MIT Press, 2012.
[6] Atkinson M P, Orlowska M E, Valduriez P, et al. Similarity search in high dimensions via hashing [C]// Proc of 25th International Conference on Very Large Data Bases. San Francisco: Morgan Kaufmann, 1999: 518-529.
[7] Liu W, Wang J, Ji R R, et al. Supervised hashing with kernels [C]// Proc of the 2012 IEEE Conference on Computer Vision and Pattern Recognition. Washington: IEEE Computer Society, 2012: 2074-2081.
[8] Krizhevsky A, Sutskever I and Hinton G. ImageNet classification with deep convolutional neural networks [C]// Proc of the 2012 Neural Information Processing Systems. Cambridge, MA: MIT Press, 2012.
[9] Szegedy C, Liu W, Jia Y Q, et al. Going deeper with convolutions [C]// Proc of the 2015 IEEE Conference on Computer Vision and Pattern Recognition. Washington: IEEE Computer Society, 2015: 7-12.
[10] Sun Y, Chen Y H, Wang X G, et al. Deep learning face representation by joint identification-verification [C]// Proc of the 2014 Neural Information Processing Systems. Cambridge, MA: MIT Press, 2014.
[11] Taigman Y, Yang M, Ranzato M, et al. DeepFace: closing the gap to human-level performance in face verification [C]// Proc of the 2014 IEEE Conference on Computer Vision and Pattern Recognition. Washington: IEEE Computer Society, 2014: 1701-1708.
[12] Zhao F, Huang Y Z, Wang L, et al. Deep semantic ranking based hashing for multi-label image retrieval [C]// Proc of the 2015 IEEE Conference on Computer Vision and Pattern Recognition. Washington: IEEE Computer Society, 2015: 1556-1664.
[13] Zhang R M, Lin L, Zhang R, et al. Bit-Scalable deep hashing with regularized similarity learning for image retrieval and person re-identification [J]. IEEE Trans on Image Processing, 2015, 24 (12): 4766-4779.
[14] Lai H J, Pan Y, Liu Y, et al. Simultaneous feature learning and hash coding with deep neural networks [C]// Proc of the 2015 IEEE Conference on Computer Vision and Pattern Recognition. Washington: IEEE Computer Society, 2015: 3270-3278.
[15] Li Q, Sun Z N, He R, et al. Deep supervised discrete hashing [J]. IEEE Trans on Image Processing, 2018: 27 (12): 5996-6009.
[16] Luo X, Chen C, Zhang H S, et al. A Survey on Deep Hashing Methods [J]. Arxiv, 2020, abs/2003.03369.
[17] Xia H K, Pan Y, Lai H J, et al. Supervised hashing for image retrieval

via image representation learning [C]// Proc of the 28th AAAI Conference on Artificial Intelligence. New York: AAAI Press, 2014: 2156-2162.
[18] Li W J, Wang S and Kang W C. Feature learning based deep supervised hashing with pairwise Labels [C]// Proc of the 25th International Joint Conference on Artificial Intelligence. New York: AAAI Press, 2016.
[19] Zhu H, Long M S, Wang J M, et al. Deep hashing network for efficient similarity retrieval [C]// Proc of the 30th AAAI Conference on Artificial Intelligence. New York: AAAI Press, 2016: 2415-2421.
[20] Wang X F, Shi Y and Kitani K M. Deep supervised hashing with triplet labels [C]// Proceedings of the 13th Asian Conference on Computer Vision. Taipei, Taiwan: Springer, 2016: 70-84.
[21] Jiang Q Y and Li W J. Asymmetric deep supervised hashing [C]// Proc of the 32th AAAI Conference on Artificial Intelligence. New York: AAAI Press, 2018: 3342-3349.
[22] Dong W, Charikar M, Li K, Asymmetric distance estimation with sketches for similarity search in high-dimensional spaces [C]// Proc of the Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. New York: ACM Press, 2008.
[23] Gordo A, Perronnin F, Gong Y C, et al. Asymmetric distances for binary embeddings [J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 2014, 36 (1): 33-47.
[24] Shen F M, Gao X, Liu L, et al. Deep asymmetric pairwise hashing [C]// Proc of the 2017 ACM on Multimedia Conference. New York: ACM Press, 2017: 1522-1530.
[25] Da C, Meng G F, Xiang S M, et al. Nonlinear asymmetric multi-valued hashing [J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 2019, 44 (11): 2660-2676.
[26] Luo Y D, Huang Z, Li Y, et al. Collaborative learning for extremely low bit asymmetric hashing [J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 2021, 33 (12): 3675-3685.
[27] Lu J L, Wang H L, Zhou J, et al. Deep Asymmetric Hashing with Dual Semantic Regression and Class Structure Quantization [J]. Information Sciences, 2022, 589: 235-249.
[28] Gu Guanghua, Huo Wenhua, Su Mingyue, et al. Asymmetric Supervised Deep Discrete Hashing Based Image Retrieval [J]. Journal of Electronics & Information Technology, 2021, 43 (12): 3530-3537.
[29] Zhu H and Gao S H. Locality constrained deep supervised hashing for image retrieval [C]// Proc of the 26th International Joint Conference on Artificial Intelligence. San Francisco: Morgan Kaufmann, 2017: 3567-3573.
[30] Yang H F, Lin K and Chen C S. Supervised learning of semantics-preserving hashing via deep convolutional neural networks for large-scale image search [J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 2017, 40 (2): 437-451.
[31] Su S P, Zhang C, Han K, et al. Greedy Hash: towards fast optimization for accurate hash coding in CNN [C]// Proc of the 2018 Neural Information Processing Systems. Cambridge, MA: MIT Press, 2018.

[32] Krizhevsky A and Hinton G. Learning multiple layers of features from tiny images [D]. Technical Report TR-2009. University of Toronto, 2009.
[33] Chua T S, Tang J H, Hong R C, et al. NUS-WIDE: a real-world web image database from national university of singapore [C]// Proc of the 8th ACM International Conference on Image and Video Retrieval. New York: ACM Press, 2009: 1-9.
[34] Kingma D P and Ba J. Adam: a method for stochastic optimization [EB/OL]. (2020-02-25). https://arxiv.org/pdf/1412.6980.pdf.
[35] Russakovsky O, Deng J, Su H, et al. ImageNet large scale visual recognition challenge [J]. International Journal of Computer Vision, 2015, 115 (3): 211-252.
[36] Neyshabur B, Srebro N, Salakhutdinov R, et al. The power of asymmetry in binary hashing [C]// Proc of the 2013 Neural Information Processing Systems. Cambridge, MA: MIT Press, 2013: 2823-2831.
[37] Zhang P C, Zhang W, Li W J, et al. Supervised hashing with latent factor models [C]// Proc of the 37th International ACM SIGIR Conference on Research and Development in Information Retrieval. New York: ACM Press, 2014: 173-182.
[38] Lin G S, Shen C H, Shi Q F, et al. Fast supervised hashing with decision trees for high-dimensional data [C]// Proc of the 2014 IEEE Conference on Computer Vision and Pattern Recognition. Washington: IEEE Computer Society, 2014: 1971-1978.
[39] Shen F M, Shen C H, Liu W, et al. Supervised discrete hashing [C]// Proc of the 2015 IEEE Conference on Computer Vision and Pattern Recognition. Piscataway, NJ: IEEE, 2015: 37-45.
[40] Kang W C, Li W J and Zhou Z H. Column sampling based discrete supervised hashing [C]// Proc of the 30th AAAI Conference on Artificial Intelligence. New York: AAAI Press, 2016: 1230-1236.
[41] Shi X S, Xing F Y, Xu K D, et al. Asymmetric discrete graph hashing [C]// Proceedings of the 31th AAAI Conference on Artificial Intelligence. New York: AAAI Press, 2017: 2541-2547.
[42] Zhang Zhisheng, Qu Huaijing, Xu Jia, et al. Sparse differential network and multi-supervised hashing for efficient image retrieval [J]. Application Research of Computers, 2021, 39 (6): 1-8.
[43] Chen Y B, Zhang S, Liu F X, et al. TransHash: Transformer-based Hamming Hashing for Efficient Image Retrieval [J]. Arxiv, 2021, abs/2105.01823.

Note: Figure translations are in progress. See original paper for figures.