Fast Deep Asymmetric Hashing for Image Retrieval

Chuangquan Lin, Zhihui Lai, Jianglin Lu and Jie Zhou

December 3, 2021

Fast Deep Asymmetric Hashing for Image Retrieval

Chuangquan Lin[1], Zhihui Lai[1,2], Jianglin Lu[1], and Jie Zhou[1]

[1] Computer Vision Institute, College of Computer Science and Software Engineering, Shenzhen University, Shenzhen 518060, China
[2] Shenzhen Institute of Artificial Intelligence and Robotics for Society, Shenzhen

Abstract. Recently, by exploiting asymmetric learning mechanism, asymmetric hashing methods achieve superior performance in image retrieval. However, due to the discrete binary constraint, these methods typically rely on a special optimization strategy of discrete cyclic coordinate descent (DCC), which is time-consuming since it must learn the binary codes bit by bit. To address this problem, we propose a novel deep supervised hashing method called Fast Deep Asymmetric Hashing (FDAH), which learns the binary codes of training and query sets in an asymmetric way. FDAH designs a novel asymmetric hash learning framework using the inner product of the output of deep network and semantic label regression to approximate the similarity and minimize the discriminant reconstruction error between the deep representation and the binary codes. Instead of using the DCC optimization strategy, FDAH avoids using the quadratic term of binary variables and the binary code of all bits can be optimized simultaneously in one step. Moreover, by incorporating the semantic information in binary code learning and the quantization process, FDAH can obtain more discriminative and efficient binary codes. Extensive experiments on three well-known datasets show that the proposed FDAH can achieve state-of-the-art performance with less training time.

Keywords: Image Retrieval · Asymmetric Hashing · Deep Learning.

1 Introduction

As one of the most popular approximate nearest neighbor (ANN) [1] search techniques, hashing has attracted considerable attention in different scenarios, including sketch retrieval [21], large-scale clustering [26] and objective recognition [25]. By encoding data through a set of binary codes, hashing methods can reduce the memory storage and speed up retrieval with efficient pairwise comparison of Hamming distance.

With the rapid development of machine learning, learning-based hashing has become a hot topic, because it can greatly improve the retrieval performance by learning the hashing function from a large number of data. Generally, learning-based hashing methods can be categorized into unsupervised and supervised methods. Unsupervised hashing methods, including Spectral Hashing (SH) [23], Binary Reconstructive Embedding (BRE) [11], Iterative Quantization (ITQ) [6], Jointly Sparse Hashing (JSH) [13], aim at constructing hash functions by exploiting inherent structures of data. On the other hand, supervised hashing methods fully exploit labeled information to obtain more discriminative binary codes, such as Supervised Discrete Hashing (SDH) [20], Fast Supervised Discrete Hashing (FSDH) [7] and Column Sampling Based Discrete Supervised Hashing (COSDISH) [10]. However, the above-mentioned hashing methods learn hash functions based on hand-crafted features, which cannot perform feature learning to generate more effective binary codes. To address this problem, some hashing methods based on deep neural network have been proposed [24, 12, 3, 22, 4]. Some representative deep hashing methods including Deep Pairwise Supervised Hashing (DPSH) [15], Deep Supervised Discrete Hashing (DSDH) [14] and Deep Discrete Supervised Hashing (DDSH) [8] integrate deep feature learning and hash code learning into a end-to-end framework and then obtain a great retrieval performance.

Due to the high computation cost, most deep hashing methods will select a subset from the dataset for training, which cannot fully utilize the supervised information. Therefore, some deep asymmetric hashing methods have been proposed [19, 9, 27]. One of the representative methods is Asymmetric Deep Supervised Hashing (ADSH) [9]. By treating query set and training set in an asymmetric way, ADSH can fully exploit the supervised information during the iterative learning procedure. However, because of using discrete cyclic coordinate descent (DCC) algorithm [20], ADSH still needs high computation cost to solve discrete optimization with the increasing length of binary codes. Meanwhile, ADSH does not fully exploit the semantic information of data in binary codes learning and the quantization process, resulting in inevitable information loss. To address these problems, this paper proposes a novel deep hashing method called Fast Deep Asymmetric Hashing (FDAH) for image retrieval, which learns the binary codes of training and query sets in an asymmetric way. Specifically, we use the commonly-used objective function of asymmetric hashing and assume that the binary codes of training set can be obtained by regressing their semantic labels. As such, we can avoid using the quadratic term of binary variables and solve the discrete optimization with a closed-form solution instead of DCC algorithm. Moreover, we consider the accumulated quantization error and incorporate the semantic information in quantization process, which can reduce the inevitable information loss and obtain more discriminative and efficient binary codes. Extensive experiments on three well-known datasets show that the proposed FDAH can achieve state-of-the-art performance with less training time.

2 The Proposed Method

In this paper, boldface uppercase letters are used to denote matrices, e.g., $\mathbf{X}$, and boldface lowercase letters are used to denote vectors, e.g., $\mathbf{x}$. $\mathbf{X}_{ij}$ denote The i-th row and j-th column element of matrix X. The Frobenius norm and transpose

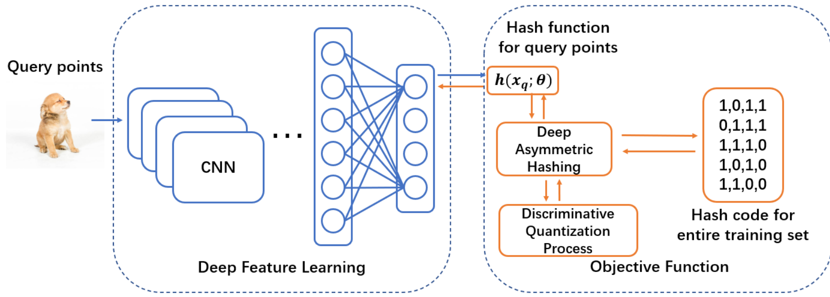


Fig. 1: The overview of FDAH

of matrix X are defined as $\|\mathbf{X}\|_F$ and $\mathbf{X}^T$, respectively. Furthermore, the binary function is presented by $sgn(\cdot)$, which outputs +1 for positive numbers and -1 for negative number. The Hadamard product is presented by $\odot$. $\mathbf{I}$ and $\mathbf{1}$ indicate an identity matrix and a matrix with all elements equaling to 1.

Suppose that the training set $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n$ includes n training samples, and the corresponding labels matrix are denoted by $\mathbf{Y} = \{\mathbf{y}_i\}_{i=1}^n \in \{0,1\}^{c\times n}$, where c is the number of classes and $\mathbf{Y}_{ji} = 1$ if $\mathbf{x}_i$ belongs to the j-th class. Meanwhile, the query set is denoted as $\mathbf{Q} = \{\mathbf{q}_j\}_{j=1}^m$ including m query samples, which are randomly sampled from the training set. The purpose of our designed model FDAH is to learn the binary codes $\mathbf{B} = \{\mathbf{b}_i\}_{i=1}^n \in \{-1,+1\}^{l\times n}$ for the whole training set and $\mathbf{U} = \{\mathbf{u}_j\}_{j=1}^m \in \{-1,+1\}^{l\times m}$ for the query set, respectively, where l is the length of binary codes.

2.1 The Idea and Model Formulation

Our method FDAH integrates deep feature learning and binary code learning into an end-to-end framework. The overview of FDAH framework is shown in Fig 1. The deep feature learning adopts a CNN model from [2], i.e., CNN-F model. Furthermore, the objective function of FDAH mainly contains two significant parts: deep asymmetric hashing part and discriminative quantization process part.

Deep Asymmetric Hashing. We attempt to learn the binary codes $\mathbf{B}$ of the whole training set $\mathbf{X}$ and the binary codes $\mathbf{U}$ of the query set $\mathbf{Q}$. Therefore, we consider the commonly-used objective function [16] of asymmetric hashing, that is:

$$\begin{aligned} &\min_{\mathbf{B},\mathbf{\Theta}} J_1 = \sum_{i=1}^{n}\sum_{j=1}^{m} ||\mathbf{b}_i^T\mathbf{u}_j - c\mathbf{S}_{ij}||^2 \\ &s.t. \quad \mathbf{B} \in \{-1,+1\}^{c\times n}, \mathbf{U} \in \{-1,+1\}^{c\times m} \end{aligned} \tag{1}$$

where S is the asymmetric semantic similarity matrix. Specifically, $\mathbf{S}_{ij} = +1$ if $\mathbf{x_i}$ and $\mathbf{q_j}$ belong to the same class, $\mathbf{S}_{ij} = -1$, otherwise.

To make full use of asymmetry, we attempt to directly learn the binary codes $\mathbf{B}$ of training set while the binary codes $\mathbf{U}$ of query set can be generated by training a deep hashing function. Thus, we set $\mathbf{U} = sgn(F(\mathbf{Q};\mathbf{\Theta}))$, where $F(\mathbf{Q};\mathbf{\Theta})$ is the output of the feature learning part, and Θ is the parameters of the neural network. Due to the non-differentiability of $sgn(\cdot)$ function, we decide to use $tanh(\cdot)$ function to replace the $sgn(\cdot)$ function for ease of optimization.

Moreover, because of the quadratic term of binary variables, the natural idea to solve the discrete optimization of the binary codes $\mathbf{B}$ is to use the discrete cyclic coordinate descent (DCC) algorithm, which is time-consuming. To tackle this problem, inspired by [7], we assume that the binary codes $\mathbf{B}$ of training set can be learned by regressing their semantic labels, i.e., $\mathbf{B} = sgn(\mathbf{W}^T\mathbf{Y})$, where $\mathbf{W} \in \mathbb{R}^{c\times l}$ is regression matrix. It is worth noting that integrating semantic information into the representation learning can generate more discriminative and efficient binary codes. Then we relax the $sgn(\cdot)$ function with its signed magnitude in our objective function. We can rewrite (1) as follows:

$$\min_{\mathbf{B},\mathbf{\Theta}} J_1 = \gamma_1 \sum_{i=1}^{n}\sum_{j=1}^{m}||(\mathbf{W}^T\mathbf{y}_i)^T tanh[F(\mathbf{q}_j;\Theta)] - c\mathbf{S}_{ij}||^2 \tag{2}$$

where γ_1 is a hyper-parameter.

Discriminative Quantization Process. Due to the relaxed strategy, we need to consider the accumulated quantization error in the binary code learning procedure. Thus, For the binary codes $\mathbf{B}$ of training set, we impose a discriminant term to keep $\mathbf{B}$ and $\mathbf{W}^T\mathbf{Y}$ as close as possible. Besides, we adopt an asymmetric graph regularization term [17] to minimize the distance between network outputs $tanh[F(\mathbf{q}_j;\Theta)]$ and the binary codes:

$$\begin{aligned}\min_{\mathbf{B},\mathbf{\Theta}} J_2 = &\gamma_2 \sum_{i=1}^{n}\sum_{j=1}^{m}\frac{1}{\tau_j}||\mathbf{b}_i - tanh(F(\mathbf{q}_j;\mathbf{\Theta}))||^2\mathbf{A}_{ij} \\ &+ \gamma_3 \sum_{i=1}^{n}||\mathbf{b}_i - \mathbf{W}^T\mathbf{y}_i||^2 \\ &s.t. \quad \mathbf{B} \in \{-1,+1\}^{c\times n}\end{aligned} \tag{3}$$

where γ_2, γ_3 are hyper-parameters, and τ_j is the total number of data points that have the same class with $\mathbf{q}_j$, which is designed to avoid class-imbalance effect. $\mathbf{A} \in \mathbb{R}^{n\times m}$ is an asymmetric affinity matrix, and if $\mathbf{x}_i$ and $\mathbf{q}_j$ belong to the same class, $\mathbf{A}_{ij} = 1$. Otherwise, $\mathbf{A}_{ij} = 0$. As can be seen from (3), we take full advantage of semantic information in the quantization process, which can improve the discriminative capabilities of the network model and reduce the inevitable information loss.

Overall Framework. Finally, by integrating J_1 and J_2 into a jointly framework, we obtain the final objective function of FDAH as follow:

$$\min_{\mathbf{B},\mathbf{\Theta}} J = J_1 + J_2 \quad s.t. \quad \mathbf{B} \in \{-1,+1\}^{c\times n} \tag{4}$$

2.2 Optimization Algorithm

In this part, we will solve the minimization problem (4) by using an iterative algorithm.

Given B and W, Update Θ. For simplicity, we define $\mathbf{z}_j = F(\mathbf{q}_j; \mathbf{\Theta})$ and $\tilde{\mathbf{u}}_i = tanh(F(\mathbf{q}_j; \mathbf{\Theta}))$. From (4), we rewrite the problem as:

$$\begin{aligned}\min_{\mathbf{\Theta}} J =& \gamma_1 tr(\tilde{\mathbf{U}}^T(\mathbf{W}^T\mathbf{Y})(\mathbf{W}^T\mathbf{Y})^T\tilde{\mathbf{U}} - 2\tilde{\mathbf{U}}^T\mathbf{W}^T\mathbf{Y}\tilde{\mathbf{S}}) \\ &+ \gamma_2 tr(\tilde{\mathbf{U}}\tilde{\mathbf{U}}^T - 2\mathbf{B}\tilde{\mathbf{A}}\tilde{\mathbf{U}}^T)\end{aligned} \tag{5}$$

where $\tilde{\mathbf{S}} = c\mathbf{S}$ and $\tilde{\mathbf{A}}_{ij} = \frac{1}{\tau_j}\mathbf{A}_{ij}$. We can update $\mathbf{\Theta}$ by using back-propagation (BP) algorithm [18]. Thus, we can compute the gradient of $\mathbf{Z}$:

$$\begin{aligned}\frac{\partial J}{\partial \mathbf{Z}} =& [2\gamma_1((\mathbf{W}^T\mathbf{Y})(\mathbf{W}^T\mathbf{Y})^T\tilde{\mathbf{U}} - 2\mathbf{W}^T\mathbf{Y}\tilde{\mathbf{S}}) \\ &+ 2\gamma_2(\tilde{\mathbf{U}} - \mathbf{B}\tilde{\mathbf{A}})] \odot (\mathbf{1} - \tilde{\mathbf{U}} \odot \tilde{\mathbf{U}})\end{aligned} \tag{6}$$

Given Θ and B, Update W. From (4), By taking the partial derivative with respect to $\mathbf{W}$ to be zero, we obtain:

$$\mathbf{W} = (\mathbf{Y}\mathbf{Y}^T)^{-1}(\gamma_1\mathbf{Y}\mathbf{S}\tilde{\mathbf{U}}^T + \gamma_3\mathbf{Y}\mathbf{B}^T)(\gamma_1\tilde{\mathbf{U}}\tilde{\mathbf{U}}^T + \gamma_3\mathbf{I})^{-1} \tag{7}$$

Given Θ and W, Update B. By expanding the objective function (4) and discarding the constant terms, we derive the following maximization problem:

$$\begin{aligned}&\max_{\mathbf{B}} tr(\gamma_2\mathbf{B}^T\tilde{\mathbf{U}}\tilde{\mathbf{A}}^T + \gamma_3\mathbf{B}^T\mathbf{W}^T\mathbf{Y}) \\ &\quad s.t. \quad \mathbf{B} \in \{-1, +1\}^{c\times n}\end{aligned} \tag{8}$$

Thus, $\mathbf{B}$ can be solved with a closed-form solution as follows:

$$\mathbf{B} = sgn(\gamma_2\tilde{\mathbf{U}}\tilde{\mathbf{A}}^T + \gamma_3\mathbf{W}^T\mathbf{Y}) \tag{9}$$

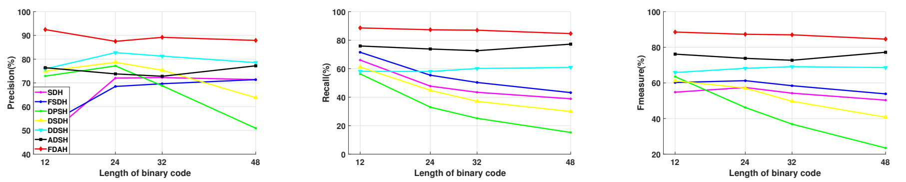
The same training strategy in ADSH [9] is adopted in our method. Specifically, we repeat the learning procedure for several times and each time we randomly sample a query set. After training, the learned neural network can be used to generate the binary codes of testing samples, i.e., $\mathbf{b}_{test} = sgn(F(\mathbf{x}_{test}; \mathbf{\Theta}))$, where $\mathbf{x}_{test}$ is a testing sample and $\mathbf{b}_{test}$ is its corresponding binary codes.

3 Experiments

In this part, we evaluate the proposed FDAH and baselines on three datasets: Fashion-MNIST, CIFAR-10 and NUS-WIDE.

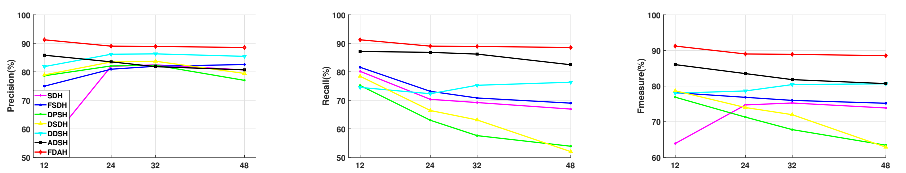
Table 1: The MAP (%) results with varying bits on three datasets. The best results are shown in bold face.

Method	Fashion-MNIST				CIFAR-10				NUS-WIDE			
	12	24	32	48	12	24	32	48	12	24	32	48
LSH	22.46	24.56	27.35	33.08	15.21	15.68	14.40	16.27	39.44	41.87	41.07	45.68
SH	35.56	31.57	31.84	29.46	20.27	18.27	17.94	17.53	41.91	41.00	40.63	43.21
BRE	33.85	42.90	42.41	44.33	18.26	21.26	23.13	23.80	46.03	47.49	46.68	51.72
ITQ	36.94	39.68	40.23	40.31	21.76	19.03	19.85	20.53	53.53	53.70	53.17	53.88
SDH	62.91	79.10	80.43	80.14	54.02	66.94	67.40	68.33	64.86	65.45	65.10	67.10
FSDH	77.70	79.86	80.69	81.10	60.95	65.73	66.38	68.41	57.64	58.21	67.19	58.33
DPSH	77.81	79.97	80.72	82.16	69.01	72.70	71.38	73.35	68.43	71.39	72.32	72.88
DSDH	79.67	81.58	82.40	82.59	72.02	77.41	79.86	80.72	67.18	69.34	70.13	70.12
DDSH	77.32	84.82	85.82	85.91	71.37	81.08	81.73	81.94	65.95	68.81	68.86	69.42
ADSH	91.36	93.35	93.93	94.22	87.06	91.20	92.93	93.46	76.70	80.28	**81.23**	**83.16**
FDAH	**94.18**	**94.19**	**94.39**	**94.48**	**93.66**	**93.32**	**93.59**	**94.31**	**78.77**	**80.34**	80.65	81.66



(a) (b) (c)

Fig. 2: Experimental results in (a) Precision, (b) Recall, and (c) F-measure of different methods on CIFAR-10 dataset.



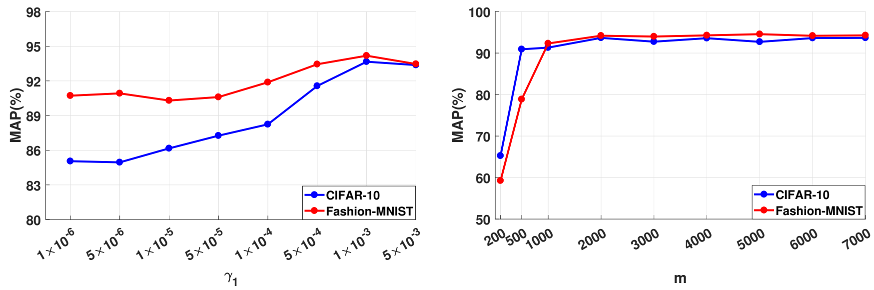
(a) (b) (c)

Fig. 3: Experimental results in (a) Precision, (b) Recall, and (c) F-measure of different methods on Fashion-MNIST dataset.

3.1 Datasets and Experimental Settings

The Fashion-MNIST includes 70,000 images which belong to 10 classes. From each class, we randomly select 6,000 images for training and the rest 1,000 images for testing. The CIFAR-10 contains 60,000 images from 10 classes. From each class, we randomly select 5,900 images for training and the rest 100 iamges for testing. The NUS-WIDE is a multi-labeled dataset which includes 21 classes, and we select more than 190,000 images for training and 2,100 for testing.

We compare our proposed method with some traditional hashing methods including LSH [5], SH [23], BRE [11], ITQ [6], SDH[20], FSDH [7] and some



(a) (b)

Fig. 4: The MAP results versus different hyper-parameters (a) γ_1 and (b) the size of query set m on CIFAR-10 and Fashion-MNIST datasets

representative deep methods including DPSH [15], DSDH [14], DDSH [8], ADSH [9]. For traditional hashing methods, we use the whole training set to learn the hashing function, and we first obtain deep features extracted by CNN-F model pre-trained on the ImageNet. For deep hashing methods, DPSH, DSDH and DDSH select 5,000 images from training set on Fashion-MNIST and CIFAR-10 datasets, and 10,500 images from training set on NUS-WIDE dataset for training. ADSH and our proposed FDAH select 2,000 imgaes on CIFAR-10 and Fashion-MNIST, and 5,000 images on NUS-WIDE as query set. For fair comparison, all the deep hashing methods iterate 150 times for convergence and apply the same network model [2], i.e., CNN-F model. The learning rate is tuned from $\{10^{-2}, ..., 10^{-6}\}$ and the batch size is 128. For FDAH, we set $\gamma_1 = 10^{-3}$, $\gamma_2 = 10$, and $\gamma_3 = 1$. Mean average precision (MAP), Precision rate, Recall rate and F-measure rate are adopted to evaluate the retrieval performance.

Table 2: The MAP (%) results and training time (in minute) of different methods with varying bits on CIFAR-10 dataset

Method	12bits	24bits	32bits	48bits
DPSH-A	92.01 334.2m	92.95 337.5m	93.16 348.1m	92.95 368.3m
DSDH-A	92.94 327.4m	93.83 350.6m	93.65 362.2m	94.25 391.6m
DDSH-A	75.89 276.2m	86.79 282.1m	90.85 292.7m	93.36 310.8m
ADSH	87.06 24.7m	91.20 30.9m	92.93 35.8m	93.46 47.6m
FDAH	93.66 10.5m	93.32 11.5m	93.59 11.7m	94.31 12.1m

3.2 Discussion

The MAP results of different methods are presented in Table 1. Obviously, by integrating feature learning and binary codes learning into a end-to-end framework, deep hashing methods can achieve better retrieval performance than traditional methods. We can find that the deep asymmetric hashing methods ADSH and our proposed FDAH can greatly outperform other deep symmetric hashing methods such as DPSH, DSDH and DDSH. The reason is that deep asymmetric methods can fully utilize the supervised information of the whole training set with the asymmetric learning mechanism. Compared with ADSH, FDAH can obtain a better performance in most cases since FDAH incorporates the semantic information in binary codes learning and the quantization process. ADSH can obtain higher accuracy with the increasing length of binary codes on NUS-WIDE dataset. The results on precision, recall and F-measure of different methods on CIFAR-10 and Fashion-MNIST datasets are shown in Fig. 2 and Fig. 3. As the figure shows, FDAH can always outperform the other methods on precision, recall and F-measure, which can be always around 90%.

Deep asymmetric hashing methods ADSH and FDAH adopt the whole training set for training to obtain high retrieval performance with the asymmetric mechanism. Therefore, we further test other deep hashing methods which utilize the whole training set. Table 2 shows the MAP results and training time of different methods on CIFAR-10. DPSH-A, DSDH-A and DDSH-A denote the corresponding deep hashing methods which utilize the whole training set. As Table 2 shows, DPSH-A, DSDH-A and DDSH-A obtain similarly high retrieval performance with much more training time. Because of using DCC algorithm, ADSH also need much training time as the length of binary codes increases. By using a closed-form solution instead of DCC algorithm, our proposed FDAH can achieve highest accuracy with less and steady training time.

Fig. 4 shows the sensitivity to hyper-parameters of the proposed FDAH on CIFAR-10 and Fashion-MNIST datasets. We shows the MAP results by tuning one of the parameters and fixing others. From Fig. 4 (a), we can see that FDAH obtains the best performance when $\gamma_1 = 10^{-3}$. Fig. 4 (b) presents the MAP results versus the size of query set m. FDAH can achieve stable performance when $m >= 2000$, because FDAH can utilize the whole training set when m is greater than 2000. Besides, FDAH is not sensitive to γ_2 and γ_3 in a range from 10^{-4} to 10^{2} in practice.

4 Conclusion

In this paper, we propose a novel deep hashing method called Fast Deep Asymmetric Hashing (FDAH). The proposed FDAH assumes that the binary codes of training set can be obtained by regressing their semantic labels and avoids using the quadratic term of binary variables in the final hashing loss. As a result, FDAH can learn the binary codes of all bits with a closed-form solution to speed up the training procedure. Moreover, FDAH can obtain more discriminative and efficient binary codes by incorporating the semantic information in binary codes learning and the quantization process. Extensive experiments on three well-known datasets show that the proposed FDAH can achieve state-of-the-art performance with less training time.

5 Acknowledgement

This work was supported in part by the Natural Science Foundation of China under Grant 61976145, Grant 62076164 and Grant 61802267, in part by the Guangdong Basic and Applied Basic Research Foundation (No.2021A1515011861), and in part by the Shenzhen Municipal Science and Technology Innovation Council under Grants JCYJ20180305124834854 and JCYJ20190813100801664.

References

1. Andoni, A., Razenshteyn, I.P.: Optimal data-dependent hashing for approximate near neighbors. In: STOC. pp. 793–801 (2015)
2. Chatfield, K., Simonyan, K., Vedaldi, A., Zisserman, A.: Return of the devil in the details: Delving deep into convolutional nets. In: BMVC (2014)
3. Chen, Y., Lai, Z., Ding, Y., Lin, K., Wong, W.K.: Deep supervised hashing with anchor graph. In: ICCV. pp. 9795–9803 (2019)
4. Cui, H., Zhu, L., Li, J., Yang, Y., Nie, L.: Scalable deep hashing for large-scale social image retrieval. vol. 29, pp. 1271–1284 (2020)
5. Datar, M., Indyk, P., Immorlica, N., Mirrokni, V.: Locality-sensitive hashing scheme based on p-stable distributions. In: ASCG. pp. 253–262 (2004)
6. Gong, Y., Lazebnik, S., Gordo, A., Perronnin, F.: Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval. IEEE Trans. Pattern Anal. Mach. Intell. **35**(12), 2916–2929 (2013)
7. Gui, J., Liu, T., Sun, Z., Tao, D., Tan, T.: Fast supervised discrete hashing. In: IEEE Trans. Pattern Anal. Mach. Intell. vol. 40, pp. 490–496 (2018)
8. Jiang, Q., Cui, X., Li, W.: Deep discrete supervised hashing. In: IEEE Trans. Image Process. vol. 27, pp. 5996–6009 (2018)
9. Jiang, Q., Li, W.: Asymmetric deep supervised hashing. In: AAAI. pp. 3342–3349 (2018)
10. Kang, W., Li, W., Zhou, Z.: Column sampling based discrete supervised hashing. In: AAAI. pp. 1230–1236 (2016)
11. Kulis, B., Darrell, T.: Learning to hash with binary reconstructive embeddings. In: NIPS. pp. 1042–1050 (2009)
12. Lai, H., Pan, Y., Liu, Y., Yan, S.: Simultaneous feature learning and hash coding with deep neural networks. In: CVPR. pp. 3270–3278 (2015)
13. Lai, Z., Chen, Y., Wu, J., Wong, W.K., Shen, F.: Jointly sparse hashing for image retrieval. IEEE Trans. Image Process. **27**(12), 6147–6158 (2018)
14. Li, Q., Sun, Z., He, R., Tan, T.: Deep supervised discrete hashing. In: NIPS. pp. 2482–2491 (2017)
15. Li, W., Wang, S., Kang, W.: Feature learning based deep supervised hashing with pairwise labels. In: IJCAI. pp. 1711–1717 (2016)
16. Liu, W., Wang, J., Ji, R., Jiang, Y., Chang, S.: Supervised hashing with kernels. In: CVPR. pp. 2074–2081 (2012)

17. Liu, W., Wang, J., Kumar, S., Chang, S.: Hashing with graphs. In: ICML. pp. 1–8 (2011)
18. Rumelhart, D.E., Hinton, G.E., Williams, R.J.: Learning representations by back propagating errors. In: Nature. vol. 323, pp. 533–536 (1986)
19. Shen, F., Gao, X., Liu, L., Yang, Y., Shen, H.T.: Deep asymmetric pairwise hashing. In: ACM MM. pp. 1522–1530 (2017)
20. Shen, F., Shen, C., Liu, W., Shen, H.T.: Supervised discrete hashing. In: CVPR. pp. 37–45 (2015)
21. Shen, Y., Liu, L., Shen, F., Shao, L.: Zero-shot sketch-image hashing. In: CVPR. pp. 3598–3607 (2018)
22. Wang, X., Shi, Y., Kitani, K.M.: Deep supervised hashing with triplet labels. In: ACCV. vol. 10111, pp. 70–84 (2016)
23. Weiss, Y., Torralba, A., Fergus, R.: Spectral hashing. In: NIPS. pp. 1753–1760 (2008)
24. Xia, R., Pan, Y., Lai, H., Liu, C., Yan, S.: Supervised hashing for image retrieval via image representation learning. In: AAAI. pp. 2156–2162 (2014)
25. Xie, G., Liu, L., Jin, X., Zhu, F., Zhang, Z., Qin, J., Yao, Y., Shao, L.: Attentive region embedding network for zero-shot learning. In: CVPR. pp. 9384–9393 (2019)
26. Zhang, Z., Liu, L., Shen, F., Shen, H.T., Shao, L.: Binary multi-view clustering. IEEE Trans. Pattern Anal. Mach. Intell. **41**, 1774–1782 (2019)
27. Zhao, S., Wu, D., Zhang, W., Zhou, Y., Li, B., Wang, W.: Asymmetric deep hashing for efficient hash code compression. In: ACM MM. pp. 763–771 (2020)