Learning Deep Binary Descriptor with Multi-Quantization

Yueqi Duan[1,2,3], Jiwen Lu[1,2,3,*], Ziwei Wang[1,4], Jianjiang Feng[1,2,3], Jie Zhou[1,2,3]
[1]Department of Automation, Tsinghua University, Beijing, China
[2]State Key Lab of Intelligent Technologies and Systems, Beijing, China
[3]Tsinghua National Laboratory for Information Science and Technology (TNList), Beijing, China
[4]Department of Physics, Tsinghua University, Beijing, China
duanyq14@mails.tsinghua.edu.cn; lujiwen@tsinghua.edu.cn; zw-wa14@mails.tsinghua.edu.cn;
jfeng@tsinghua.edu.cn; jzhou@tsinghua.edu.cn

Abstract

In this paper, we propose an unsupervised feature learning method called deep binary descriptor with multi-quantization (DBD-MQ) for visual matching. Existing learning-based binary descriptors such as compact binary face descriptor (CBFD) and DeepBit utilize the rigid sign function for binarization despite of data distributions, thereby suffering from severe quantization loss. In order to address the limitation, our DBD-MQ considers the binarization as a multi-quantization task. Specifically, we apply a K-AutoEncoders (KAEs) network to jointly learn the parameters and the binarization functions under a deep learning framework, so that discriminative binary descriptors can be obtained with a fine-grained multi-quantization. Extensive experimental results on different visual analysis including patch retrieval, image matching and image retrieval show that our DBD-MQ outperforms most existing binary feature descriptors.

1. Introduction

Feature description is a fundamental computer vision problem which is widely applicable in a number of applications, such as object recognition [11, 27], face recognition [29,32,43], image classification [15,26] and many others. There are two essential properties for an effective feature descriptor: strong discriminative power and low computational cost. On one hand, since real-world applications usually suffer from large intra-class variances, it is critical to extract desirable feature descriptors with high quality representation. On the other hand, mobile devices with limited computational capabilities and large amount of data require efficient feature descriptors with high computational speed and low memory cost.

*Corresponding author.

In recent years, deep convolutional neural network (CNN) has achieved state-of-the-art performance in various visual analysis tasks, and numerous discriminative CNN features have been proposed, such as AlexNet [23], VGG [32,41], GoogLeNet [44] and ResNet [17]. CNN features obtain high quality representation by training a feature learning model with large amount of labeled data to estimate extensive number of parameters. However, they suffer from heavy storage costs and low matching speed as they are high-dimensional real-valued descriptors.

Several local binary features have been proposed over the past decade. Representative binary features include local binary pattern (LBP) [1, 30] as well as its variants [36, 37], binary robust independent elementary feature (BRIEF) [6], binary robust invariant scalable keypoint (BRISK) [25], oriented FAST and rotated BRIEF (ORB) [38] and fast retina keypoint (FREAK) [2]. These methods reduce the computational cost by substituting the Euclidean distance with Hamming distance and compute the distances between binary codes using XOR operations.

Inspired by the fact that CNN features deliver strong discriminative power and binary features present low computational cost, DeepBit [26] learns deep compact binary descriptors in an unsupervised manner, which achieves the state-of-the-art in binary feature description. However, it simply utilizes the rigid sign function for binarization despite of data distributions. For many distributions, the handcrafted zero is not a reasonable threshold for binarization, which may lead to severe quantization loss. In order to address the limitation, we consider the binarization problem as a general multi-quantization task, where the sign function is a special case to cluster positives into one class and negatives into another. Specifically, we apply a K-AutoEncoders (KAEs) network and propose a deep binary descriptor with multi-quantization (DBD-MQ) learning method. Figure 1 illustrates the flowchart of the proposed approach. With the KAEs based multi-quantization, we jointly learn the param-

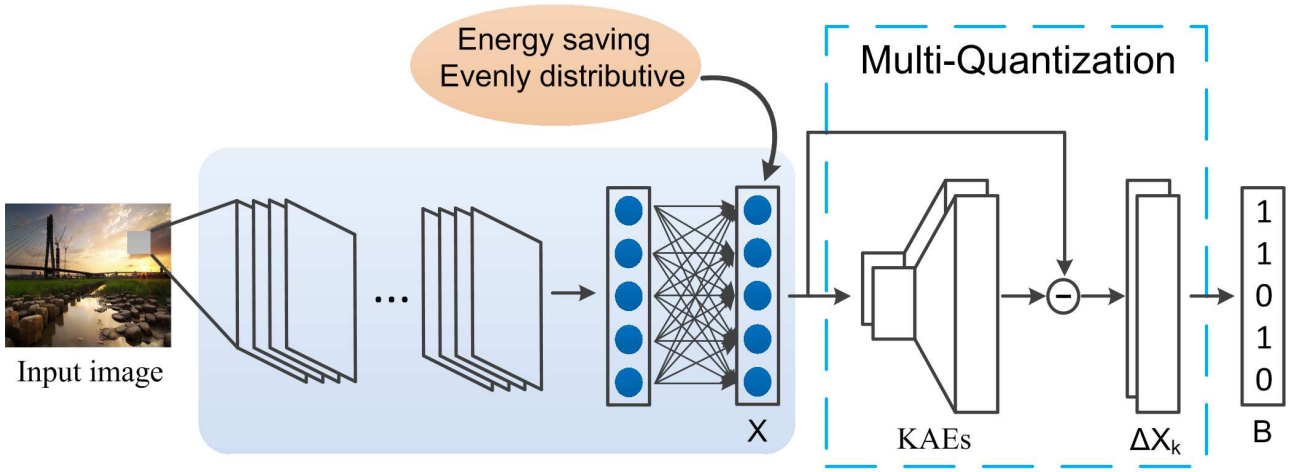


Figure 1. The flowchart of the proposed method. For each image patch from the training set, we first learn a real-valued feature vector with a pre-trained CNN by replacing the softmax layer with a fully connection layer. Then, we binarize the vectors with the K-Autoencoders (KAEs) based multi-quantization instead of the rigid sign function, which minimizes the reconstruction loss by controlling the residual features $\Delta\mathbf{X}_k$. K is equal to 2 in this figure for easy illustration. Lastly, we optimize the parameters iteratively with back-propagation in an unsupervised manner to obtain compact binary codes.

eters of the network and the binarization functions to obtain more discriminative binary codes. Extensive experimental results on three different visual analysis tasks including patch retrieval, image matching and image retrieval show the effectiveness of the proposed method.

2. Related Work

Binary Feature Descriptors: Binary feature descriptors have aroused extensive interest due to their efficiency of matching and storing in recent years. Earlier binary features include BRIEF [6], BRISK [25], ORB [38] and FREAK [2]. BRIEF directly utilized simple intensity difference tests to compute binary vectors in a smoothed image patch. BRISK leveraged a circular sampling pattern to obtain scale and rotation invariance. ORB shared the similar purpose by employing scale pyramids and orientation operators. FREAK referenced the human visual system by utilizing retinal sampling grid for fast computing. However, these methods have not shown remarkable performance because pairwise comparison of raw intensity is susceptible to scale and transformation. In order to address the limitation, several learning-based binary descriptors have been proposed [4, 45, 47, 51]. For example, Trzcinski *et al.* [47] proposed a D-BRIEF method by encoding similarity relationships to learn discriminative projections. Balntas *et al.* [4] presented a binary online learned descriptor (BOLD) by applying LDA criterion. However, these methods only employ pairwise learning, which are unfavorable to transfer the learned binary features into new applications.

In recent years, a number of unsupervised binary descriptor learning methods have been proposed, which project each local patch into a binary descriptor [26, 28, 29] For example, Lu *et al.* [29] proposed a compact binary face descriptor (CBFD) learning method to learn evenly-distributive and energy-saving local binary codes. They also presented a simultaneous local binary feature learning and encoding (SLBFLE) [28] method by jointly learning binary codes and the codebook in a one-stage procedure. Lin *et al.* [26] proposed a DeepBit by designing a CNN to learn compact binary codes in an unsupervised manner. However, these methods simply employ the rigid sign function for binarization, which is not optimal in many cases.

Deep Learning: There has been extensive work on deep learning in recent years [7, 17, 23, 31, 32, 41, 44], which achieves the state-of-the-art performance on many computer vision applications, such as object recognition [17, 41], object detection [13], face recognition [32, 43] and human action recognition [21]. With large amount of data, deep learning methods learn high-level hierarchical features by training powerful statistical models to obtain higher quality representation. In recent years, several deep binary codes learning methods have also been proposed [9, 24, 26, 50]. For example, Xia *et al.* [50] proposed a CNN hashing (CNNH) method by learning deep hashing codes and image representation in a supervised manner. Lai *et al.* [24] improved CNNH by presenting a one-stage deep binary codes learning procedure. Liong *et al.* [9] proposed a deep hashing (DH) method by learning multiple non-linear hierarchical transformations under three constraints. Lin *et al.* [26]

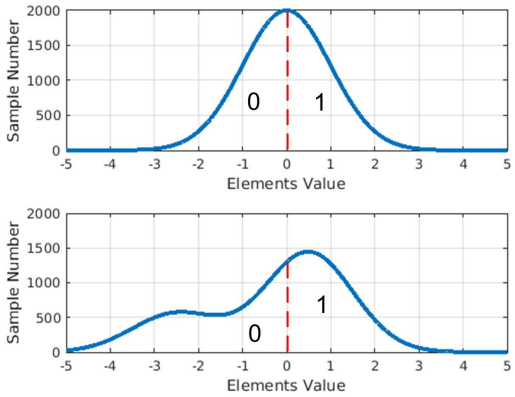


Figure 2. An illustration of binarizing with the sign function on two common distributions. Red dashes represent the threshold, and the coding results are shown in the figure. For both distributions, it is not very reasonable to set zero as the threshold.

presented a DeepBit method by learning compact deep binary codes in an unsupervised manner. However, most deep binary codes learning methods utilize the rigid sign function for binarization.

3. Proposed Approach

In this section, we first present the K-AutoEncoders based multi-quantization, and then propose the deep binary descriptor with multi-quantization (DBD-MQ) learning approach.

3.1. K-AutoEncoders Based Multi-Quantization

There have been a number of local binary code learning methods proposed in recent years [26,28,29], yet all of them utilize the rigid sign function to quantize each element of the real-valued vectors into binary codes. There are two key limitations of the sign function based binarization:

1) While existing local binary code learning methods attempt to learn evenly distributive elements, zero is still not the optimal threshold in many cases. We take the standard Gaussian distribution and the Gaussian mixture distribution as examples, which are shown in Figure 2. Both models contain the same number of positives and negatives. For the standard Gaussian distribution, as the threshold lies in the densest area, a large number of elements have to be separated into 0 and 1 even if their real-valued differences are small, which leads to large quantization loss. For the Gaussian mixture distribution, it is reasonable to separate different parts of the distribution with the threshold, yet zero may not be an ideal choice. Therefore, a fine-grained binarization strategy should be simultaneously learned with the local binary codes to obtain more optimal quantization.

2) Existing binarization approaches are applied on each bit separately, which ignore the holistic information from feature vectors, thereby are more susceptible to noise. The holistic feature vectors should provide prior knowledge for the binarization of each bit, so that elements from similar features have higher tendency to be quantized into the same binary codes, which deliver stronger robustness.

In order to address the above limitations, we propose a K-AutoEncoders (KAEs) based multi-quantization method. We formulate the binarization problem as a K-quantization task, where K is equal to 2^c in this work. Each element is clustered into one of K classes, which leads to a c-bit encoding. The conventional sign function is a special case which clusters negatives into one class and positives into another. As a 2-clustering approach, each element is quantized into a 1-bit binary code in this situation.

K-Means has been one of the most widely used clustering algorithms for over 50 years [19], which iteratively optimizes with a two-step procedure: 1) classifying each data point into a cluster, and 2) optimizing each cluster with corresponding data points. Inspired by the fact that K-means achieves outstanding performance in many quantization tasks, we train our KAEs with the similar iterative approach. In KAEs, we first associate each real-valued feature vector $\mathbf{x}_n$ with the AutoEncoder, which obtains the minimum reconstruction error:

$$k_n = \arg\min_k \varepsilon_{nk}, \tag{1}$$

where $\varepsilon_{nk} = ||\Delta \mathbf{x}_{nk}||_2$ is the reconstruction error of $\mathbf{x}_n$ with the kth AutoEncoder. Then, we utilize the corresponding $\mathbf{x}_n$ to update the parameters of the k_nth AutoEncoder. Figure 3 shows the detailed procedure of training the KAEs. The learned KAEs can be considered as K clustering centers, where each feature is clustered to the AutoEncoder with the minimum reconstruction error.

In order to quantize each element of the feature vectors into binary codes, we consider the element-wise quantization loss $\varepsilon_{nk}^{(i)} = |\Delta \mathbf{x}_{nk}^{(i)}|$, and the clustering approach of each element is formulated as follows:

$$k_n^{(i)} = \arg\min_k \varepsilon_{nk}^{(i)},\ k = 1, 2, \cdots, K \tag{2}$$

where the ith element of $\mathbf{x}_n$ is clustered into the $k_n^{(i)}$th AutoEncoder. Each element is clustered to the AutoEncoder with the minimum element-wise reconstruction error, so that the total quantization loss is minimized.

As one of the main purposes of binary code learning is to reduce the storage costs, we simply encode K clusters into c-bit binary codes to balance the accuracy and the binary length without special encoding strategies. Having clustered real-valued elements into K classes, we obtain

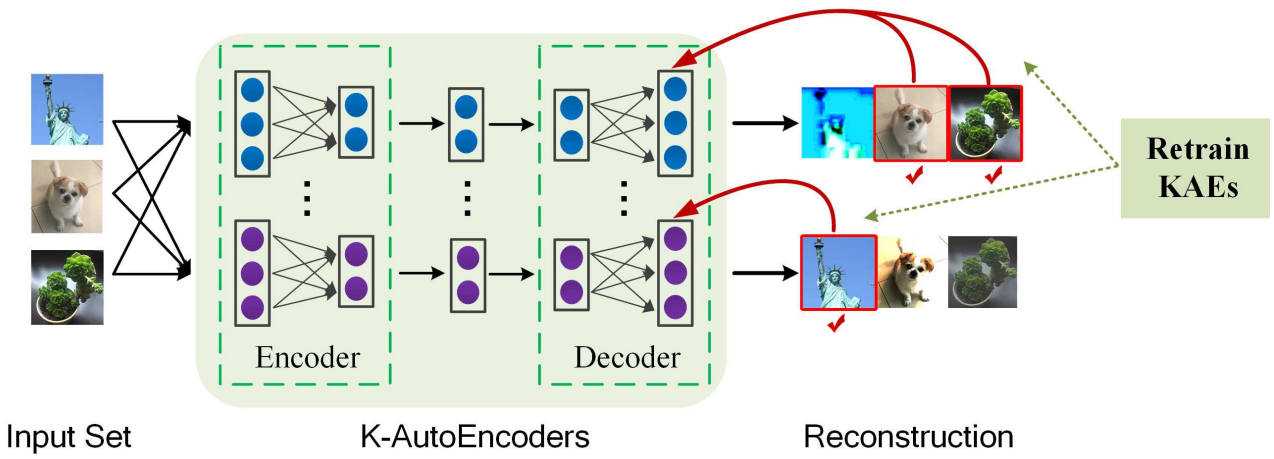


Figure 3. A detailed explanation of training KAEs. For each image from the input set, we first encode and decode its CNN feature with all KAEs. Then, we associate each feature with the AutoEncoder obtaining the minimum reconstruction loss, which is highlighted with a red box. Lastly, we utilize the corresponding features to train the associate AutoEncoder. These steps are executed iteratively until convergence.

the corresponding binary codes for each element, which are concatenated into the binary descriptor.

3.2. Learning Deep Binary Descriptor with Multi-Quantization

We initialize the CNN with the pre-trained 16 layers VGGNet [41] trained on the ImageNet dataset, which replaces the softmax layer with a fully connection layer. Let $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N]$ be the CNN features of N images, where $\mathbf{x}_n \in \mathbb{R}^d$ $(1 \leq n \leq N)$ is the nth feature of the input images. The objective function of our approach to learn the parameters of the holistic deep neural network with KAEs is shown as follows:

$$\begin{aligned} \min_{\mathbf{X},\mathbf{W}_k} J &= J_1 + \lambda_1 J_2 + \lambda_2 J_3 \\ &= \sum_{n=1}^{N} \varepsilon_{nk_n}^2 + \lambda_1 \sum_{k=1}^{K} \sum_{l} ||\mathbf{W}_k^{(l)}||_F^2 \\ &- \lambda_2 \mathrm{tr}((\mathbf{X}-\mathbf{U})^T(\mathbf{X}-\mathbf{U})), \end{aligned} \tag{3}$$

where $\mathbf{W}_k^{(l)}$ represents the parameters of the lth layer of the kth AutoEncoder, and $\mathbf{U} \in \mathbb{R}^{d \times N}$ is the mean feature of $\mathbf{X}$ repeating N times.

J_1 aims to minimize the reconstruction error of the features. This term not only directs the projection parameters of KAEs, but also leads to better real-valued features with the minimum quantization loss. J_2 is the regularization term for KAEs to prevent from overfitting. The physical meaning of J_3 is to enlarge the variance of the learned features. The first term J_1 may lead to similar features for all input patches, which harms the discriminativeness of the learned feature, while the third term J_3 maximizes the variance of each element of the features, so that each element of descriptors contains more information from the training patches.

As it is not convex to simultaneously optimize CNN and KAEs, we use an iterative approach to update one by fixing the others.

Learning $\mathbf{W}_k$ with a fixed $\mathbf{X}$: when $\mathbf{X}$ is fixed, the objective function (3) can be rewritten as follows:

$$\min_{\mathbf{W}_k} J = \sum_{n=1}^{N} \varepsilon_{nk_n}^2 + \lambda_1 \sum_{k=1}^{K} \sum_{l} ||\mathbf{W}_k^{(l)}||_F^2, \tag{4}$$

and we apply stochastic gradient descent approach to update $\mathbf{W}_k$.

Learning $\mathbf{X}$ with fixed $\mathbf{W}_k$: when the parameters of the KAEs are fixed, the objective function (3) can be rewritten as follows:

$$\min_{\mathbf{X}} J = \sum_{n=1}^{N} \varepsilon_{nk_n}^2 - \lambda_2 \mathrm{tr}((\mathbf{X}-\mathbf{U})^T(\mathbf{X}-\mathbf{U})). \tag{5}$$

Similarly, the stochastic gradient descent approach with back-propagation is applied to train the network iteratively, and we learn effective and discriminative local binary codes in an unsupervised manner. Algorithm 1 details the approach of the proposed DBD-MQ.

In the training procedure, we simultaneously learn the parameters of CNN and the KAEs to obtain energy-saving and evenly-distributive binary descriptors. In the test procedure, for each local patch, we first learn its real-valued feature representation using the learned CNN, and then quantize each element into binary codes with the learned KAEs using (2), which are concatenated into a longer binary descriptor as the final representation. As the dimension of

Algorithm 1: DBD-MQ

Input: Training image set, parameters λ_1 and λ_2, and iteration number T.

Output: Projection parameters of CNN $\mathbf{W}$, and parameters of KAEs $\mathbf{W}_k$.

```
Initialize pre-trained CNN features X and KAEs W_k.
for iter = 1, 2, ..., T do
    loop
        Cluster each x_n into an AutoEncoder using (1).
        Update W_k with corresponding x_n using (4).
    end loop until convergence
    Update CNN with W_k fixed using (5).
end for
return W and W_k.
```

features is relatively small, we utilized the term of J_2 to prevent from overfitting instead of dropout, by fixing λ_1 as 0.001 and λ_2 as 1.0, respectively. Moreover, we rotate each image by -10, -5, 0, 5, 10 degrees for data augmentation. For each image, we first reshape its size into 256×256 by following [26], and then crop it into 224×224 to remove the background information.

3.3. Discussion

The proposed DBD-MQ improves the conventional sign function based binary codes learning methods in the following two aspects:

1) Instead of employing a hand-crafted threshold, the proposed DBD-MQ simultaneous learns the parameters of CNN and KAEs to minimize the quantization loss. With the fine-grained multi-quantization, we cluster similar elements of real-valued descriptors into the same class and obtain more energy-saving binary descriptors.

2) KAEs are learned from holistic feature vectors, minimizing the reconstruction error of similar real-valued descriptors in the corresponding AutoEncoder. Therefore, elements from similar features vectors belonging to the same AutoEncoder have higher tendency to be quantized into the same class, as the total reconstruction error is small in this AutoEncoder. Unlike existing binarization approaches [26, 29] which quantize each bit separately, the holistic real-valued descriptors provide strong prior knowledge for the binarization of each element, which enhances the robustness and stableness of the learned binary descriptors.

4. Experiments

We evaluated the proposed DBD-MQ method on three challenging datasets including CIFAR-10 [22], Brown [5] and Oxford [34] datasets. We conducted experiments on three different visual analysis tasks, including patch retrieval on CIFAR-10, image patch matching on Brown and image retrieval on Oxford. We compared the proposed method with several state-of-the-art local descriptors to evaluate the effectiveness of DBD-MQ.

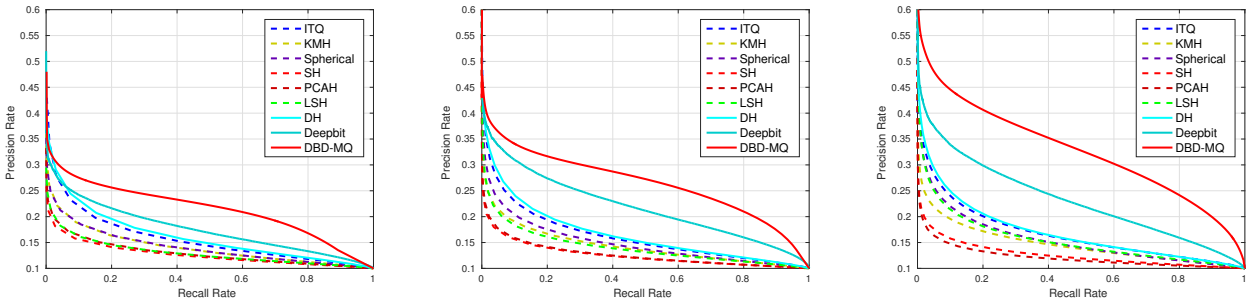
4.1. Results on Patch Retrieval

The CIFAR-10 dataset [22] contains 10 subjects with 6000 images for each class. The image size is 32×32, with 50,000 training images and the other 10,000 test images. In the experiments, we followed the standard evaluation protocol [22], and tested the proposed DBD-MQ under different binary length: 16 bits, 32 bits and 64 bits.

Parameter Analysis: We first tested the dimensions of layers of each AutoEncoder by using cross validation under different binary length. For 16-bit DBD-MQ, the dimensions for each AutoEncoder were empirically set as [16 $\rightarrow$ 12 $\rightarrow$ 8 $\rightarrow$ 12 $\rightarrow$ 16] with cross validation. For 32-bit, the dimensions were set as [32 $\rightarrow$ 24 $\rightarrow$ 16 $\rightarrow$ 24 $\rightarrow$ 32]. For 64-bit, the dimensions were set as [64 $\rightarrow$ 50 $\rightarrow$ 32 $\rightarrow$ 50 $\rightarrow$ 64]. Moreover, we utilized the ReLU function as the nonlinear units.

We tested the mean average precision (mAP) under different number of AutoEncoders K, with the structure of AutoEncoders fixed as [16 $\rightarrow$ 12 $\rightarrow$ 8 $\rightarrow$ 12 $\rightarrow$ 16]. Figure 5 shows that the best result was obtained when K is equal to 4. Although the binary lengths are 16, 32, 64 and 128 respectively when K is set as 2, 4, 8 and 16, they share the same original real-valued feature vectors. In other words, they share the same original information and use different lengths of binary codes to represent each element, which differ from the sign function based methods under different binary lengths. The learned binary codes preserve more information when K is increasing. However, the mean average precision will decrease if the searching space is too large. Therefore, the mean average precision increases at first, and then decreases when K is too large. Most binary codes utilize a one-bit code to present each real-valued element. Hence, we apply $K = 2$ to all the following experiments for a fair comparison.

Comparison with the State-of-the-Art Unsupervised Hashing Methods: We compared the proposed DBD-MQ method with several state-of-the-art unsupervised hashing approaches on this image retrieval task, where deep hashing (DH) and DeepBit are two latest deep binary codes learning methods. Table 1 illustrates the mean average precision (mAP) of the proposed method compared with several state-of-the-art unsupervised hashing methods. Among previous unsupervised hashing methods, DeepBit delivers outstanding mAP, yet our DBD-MQ improves the performance by 2.10%(= 21.53% - 19.43%), 1.64%(= 26.50% - 24.86%) and 4.12%(= 31.85% - 27.73%) with 16 bits, 32 bits and



(a) 16 bits (b) 32 bits (c) 64 bits

Figure 4. Precision/Recall curves of the Cifar-10 dataset compared with the state-of-the-art unsupervised hashing methods under varying binary lengths (a) 16 bits, (b) 32 bits and (c) 64 bits.

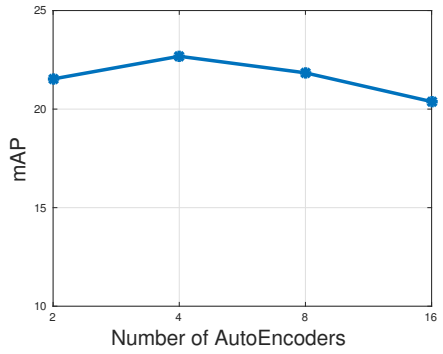


Figure 5. The mean average precision (mAP) performance (%) under varying number of AutoEncoders.

64 bits respectively. The main reason is that DeepBit simply applies rigid sign function for binarization thereby suffering from severe quantization loss. Our DBD-MQ simultaneously learns the features and the fine-grained quantization function in an end-to-end network, so that the learned binary codes are more compact and deliver stronger discriminative power for each bit. Figure 4 illustrates the Precision/Recall curves of the proposed DBD-MQ and the state-of-the-art unsupervised hashing methods. We observe that the proposed DBD-MQ consistently outperforms other approaches.

Evaluation of Different Binarization Strategies: One of the most significant contributions of the proposed DBD-MQ is the application of KAEs for fine-grained binarization. In the previous experiments, we obtained state-of-the-art performance compared with the state-of-the-art unsupervised hashing approaches, yet it could not directly show the effectiveness of our multi-quantization. In order to better evaluate our KAEs, we conducted an experiment to compare different binarization strategies. We fixed all other parameters and simply changed our KAEs with sign functions for binarization to test the mean average precision performance on CIFAR-10. Table 2 shows the experimental results. As the only difference between these two methods is the binarization strategy, this experiment shows that the fine-grained multi-quantization approach outperforms the rigid sign function under all three binary lengths. Moreover, we observe that with the increase of binary length, the improvement of KAEs becomes more significant. On one hand, KAEs minimize the quantization loss for each bit, so that the learned binary codes are more compact and longer descriptors benefit more from the fine-grained multi-quantization. On the other hand, longer descriptors are able to train better KAEs, so that the holistic descriptors provide more precise prior knowledge for the binarization of each element.

Table 1. The mean average precision (mAP) performance (%) of top 1,000 returned images compared with different state-of-the-art unsupervised hashing methods under different binary code length.

Method	16 bits	32 bits	64 bits
KMH [16]	13.59	13.93	14.46
SphH [18]	13.98	14.58	15.38
SpeH [49]	12.55	12.42	12.56
SH [39]	12.95	14.09	13.89
PCAH [48]	12.91	12.60	12.10
LSH [3]	12.55	13.76	15.07
PCA-ITQ [14]	15.67	16.20	16.64
DH [9]	16.17	16.62	16.96
DeepBit [26]	19.43	24.86	27.73
DBD-MQ	**21.53**	**26.50**	**31.85**

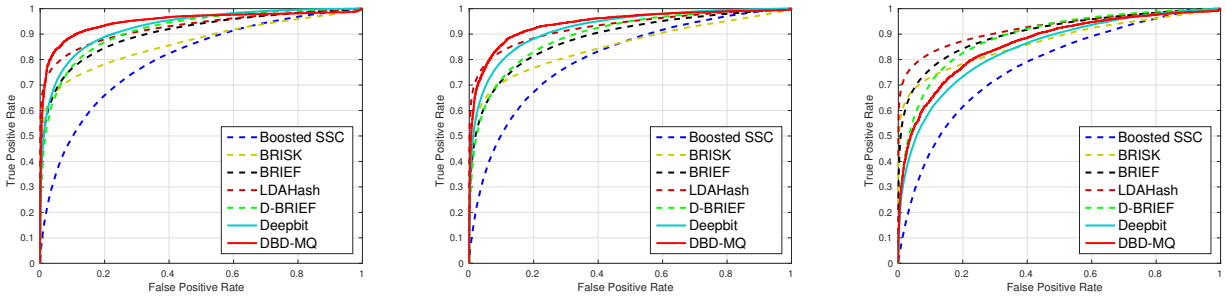
Table 2. The mean average precision (mAP) performance (%) of different binarization strategies on the Cifar-10 dataset under different binary code length.

Binarization	16 bits	32 bits	64 bits
KAEs	**21.53**	**26.50**	**31.85**
Sign	19.16	23.89	26.90
ΔmAP	2.37	2.61	4.95

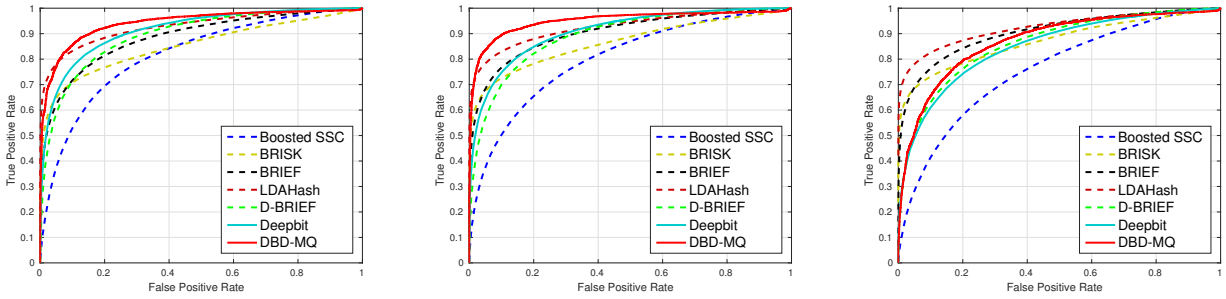
Computational Time: Our hardware configuration comprises of a 2.8-GHz CPU and a 32G RAM. As we applied a very deep VGG convolutional network to initialize our CNN, we utilized a Tesla K80 GPU for acceleration.

Table 3. 95% error rates (ERR) compared with the state-of-the-art binary descriptors on Brown dataset (%), where Boosted SSC, Brisk, BRIEF and DeepBit are unsupervised binary feature and LDAHash, D-BRIEF, BinBoost and RFD are supervised. The real-valued feature SIFT is provided for reference.

Train Test	Yosemite Noter Dame	Yosemite Liberty	Notre Dame Yosemite	Notre Dame Liberty	Liberty Notre Dame	Liberty Yosemite	Average ERR
SIFT [27] (128 bytes)	28.09	36.27	29.15	36.27	28.09	29.15	31.17
Boosted SSC [40] (16 bytes)	72.20	71.59	76.00	70.35	72.95	77.99	73.51
BRISK [25] (64 bytes)	74.88	79.36	73.21	79.36	74.88	73.21	75.81
BRIEF [6] (32 bytes)	54.57	59.15	**54.96**	59.15	54.57	**54.96**	56.23
DeepBit [26] (32 bytes)	29.60	34.41	63.68	32.06	26.66	57.61	40.67
LDAHash [42] (16 bytes)	51.58	49.66	52.95	49.66	51.58	52.95	51.40
D-BRIEF [47] (4 bytes)	43.96	53.39	46.22	51.30	43.10	47.29	47.54
BinBoost [45] (8 bytes)	14.54	21.67	18.96	20.49	16.90	22.88	19.24
RFD [10] (50-70 bytes)	11.68	19.40	14.50	19.35	13.23	16.99	15.86
DBD-MQ (32 bytes)	**27.20**	**33.11**	57.24	**31.10**	**25.78**	57.15	**38.59**



(a) Yosemite-Notre Dame (b) Yosemite-Liberty (c) Notre Dame-Yosemite



(d) Notre Dame-Liberty (e) Liberty-Notre Dame (f) Liberty-Yosemite

Figure 6. ROC curves of the proposed method compared with several methods on the Brown dataset, under all the combinations of training and test of Liberty, Notre Dame and Yosemite.

We tested the total computational time of extracting features from a pair of images in CIFAR-10 and computing their similarities. It took 0.043s for the proposed DBD-MQ, while HOG [8] and SIFT [27] took 0.028s and 0.051s, respectively. For the storage cost, a 32-bit DBD-MQ descriptor requires 4 bytes memory for each image patch, while 9 bytes are required for HOG and 128 bytes for SIFT. This shows that our DBD-MQ is more suitable for scalable visual matching and search in practical applications.

4.2. Results on Image Matching

We evaluated the proposed DBD-MQ on the Brown dataset [5], including Liberty, Notre Dame and Yosemite where each of them contains more than 400,000 image patches. For each dataset, there are 200,000 to 400,000 training images and 100,000 test pairs with half of them matched (positive) and the others mismatched (negative). In the experiments, we followed the settings in [46] where all six training and test combinations were used: Yosemite-Notre Dame, Yosemite-Liberty, Notre Dame-Yosemite, Notre Dame-Liberty, Liberty-Notre Dame and Liberty-Yosemite. We fixed the binary length as 256, applying the KAEs with the structure of [256 $\rightarrow$ 160 $\rightarrow$ 100 $\rightarrow$ 60 $\rightarrow$ 100 $\rightarrow$ 160 $\rightarrow$ 256].

Comparison with the State-of-the-Arts: Table 3 shows the 95% error rates (ERR) of the proposed DBD-MQ com-

Table 4. 95% error rates (ERR) of different binarization strategies on the Brown dataset (%).

Train	Yosemite	Yosemite	Notre Dame	Notre Dame	Liberty	Liberty	Average
Test	Noter Dame	Liberty	Yosemite	Liberty	Notre Dame	Yosemite	ERR
KAEs	**27.20**	**33.11**	**57.24**	**31.10**	**25.78**	**57.15**	**38.59**
Sign	29.84	36.13	60.42	32.97	28.52	59.04	41.15
ΔERR	2.64	3.02	3.18	1.87	2.74	1.89	2.56

pared with several state-of-the-art descriptors, and Figure 6 shows the ROC curves on all six training and test combinations. Among these compared approaches, Boosted SSC [40], BRISK [25], BRIEF [6] and DeepBit [26] are unsupervised binary descriptors while LDAHash [42], D-BRIEF [47], BinBoost [45] and RFD [10] are supervised. The real-valued SIFT [27] is provided for reference. Among the existing unsupervised binary descriptors, DeepBit obtains outstanding results due to its strong discriminative power. However, DeepBit employs the rigid sign function for binarization, while the proposed DBD-MQ learns fine-grained KAEs to minimize the quantization loss, leading to better performances on all six experiments. Our DBD-MQ also achieves better average 95% error rate than supervised approaches. As an unsupervised manner, DBD-MQ fits for the applications where it is difficult to collect label information, while supervised approaches fail to work in such scenarios.

Evaluation of Different Binarization Strategies: Similar to the experiment designed on the CIFAR-10 dataset, we conducted an additional experiment to evaluate the effectiveness of the proposed multi-quantization based binarization. Table 4 shows the experimental results of different binarization strategies on the brown dataset. We find that the proposed KAEs based method outperforms the conventional sign function on all the experiments of the Brown dataset, which shows the effectiveness of binarization with multi-quantization.

4.3. Results on Image Retrieval

The Oxford dataset [34] contains 5,062 images of Oxford landmarks collected from Flickr, where 11 locations are manually generated comprehensive ground truth, represented by 5 bounding boxes for each as queries. We need to retrieve all the image of the same place with the 55 queries. We followed the experimental settings in [33] by training on the Paris dataset [35] and learning a 256-centroid vocabulary. We set the length of the binary codes as 128, applying the KAEs of $[128 \rightarrow 90 \rightarrow 60 \rightarrow 90 \rightarrow 128]$.

Table 5 shows the image retrieval results on the Oxford dataset. The SIFT descriptor [27] is listed as a baseline method. As our DBD-MQ only exploits raw RGB patches as the input without any pre-processing, the result of CKN [33] is reported with the raw input for a fair comparison. AlexNet [23] is one of the most popular convolutional neural networks, which consists of 7 layers. We evaluate the mean average precision of the output after ReLU of all 5 convolutional layers. Our DBD-MQ obtains encouraging result on the Oxford dataset. CKN extracts patch-level descriptors using an unsupervised CNN, while the proposed DBD-MQ learns energy-saving and evenly-distributive binary descriptors, which deliver stronger discriminative power. Moreover, as a binary descriptor learning method, the proposed DBD-MQ has higher efficiency to store and compute on image retrieval tasks compared with real-valued descriptors.

Table 5. The mean average precision (mAP) performance (%) of different approaches on the Oxford dataset.

Method	mAP
SIFT [27]	43.7
BoW 200k-D [20]	36.4
AlexNet-conv1 [23]	18.8
AlexNet-conv2 [23]	12.5
AlexNet-conv3 [23]	33.3
AlexNet-conv4 [23]	34.3
AlexNet-conv5 [23]	33.4
PhilippNet [12]	38.3
CKN-raw [33]	23.0
DBD-MQ	**38.9**

5. Conclusion

In this paper, we have proposed a deep binary descriptor learning with multi-quantization (DBD-MQ) method. Unlike most existing binary codes learning methods which utilize the rigid sign function for binarization, our DBD-MQ simultaneously learns the parameters of CNN and KAEs, replacing the sign function with the fine-grained multi-quantization to minimize the quantization loss. The proposed DBD-MQ outperforms most existing unsupervised binary descriptors on three widely used datasets.

Acknowledgements

This work was supported in part by the National Key Research and Development Program of China under Grant 2016YFB1001001, the National Natural Science Foundation of China under Grants 61672306, 61572271, 61527808, 61373074 and 61373090, the National 1000 Young Talents Plan Program, the National Basic Research Program of China under Grant 2014CB349304, the Ministry of Education of China under Grant 20120002110033, and the Tsinghua University Initiative Scientific Research Program.

References

[1] T. Ahonen, A. Hadid, and M. Pietikainen. Face description with local binary patterns: Application to face recognition. *TPAMI*, 28(12):2037–2041, 2006.

[2] A. Alahi, R. Ortiz, and P. Vandergheynst. FREAK: Fast retina keypoint. In *CVPR*, pages 510–517, 2012.

[3] A. Andoni and P. Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In *FOCS*, pages 459–468, 2006.

[4] V. Balntas, L. Tang, and K. Mikolajczyk. BOLD-binary online learned descriptor for efficient image matching. In *CVPR*, pages 2367–2375, 2015.

[5] M. Brown, G. Hua, and S. Winder. Discriminative learning of local image descriptors. *TPAMI*, 33(1):43–57, 2011.

[6] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. BRIEF: Binary robust independent elementary features. In *ECCV*, pages 778–792, 2010.

[7] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *BMVC*, pages 1–12, 2015.

[8] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, volume 1, pages 886–893, 2005.

[9] V. Erin Liong, J. Lu, G. Wang, P. Moulin, and J. Zhou. Deep hashing for compact binary codes learning. In *CVPR*, pages 2475–2483, 2015.

[10] B. Fan, Q. Kong, T. Trzcinski, Z. Wang, C. Pan, and P. Fua. Receptive fields selection for binary feature description. *TIP*, 23(6):2583–2595, 2014.

[11] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *CVPR*, volume 2, pages 264–271, 2003.

[12] P. Fischer, A. Dosovitskiy, and T. Brox. Descriptor matching with convolutional neural networks: a comparison to sift. *arXiv preprint arXiv:1405.5769*, 2014.

[13] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, pages 580–587, 2014.

[14] Y. Gong, S. Lazebnik, A. Gordo, and F. Perronnin. Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval. *TPAMI*, 35(12):2916–2929, 2013.

[15] K. Grauman and T. Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *ICCV*, volume 2, pages 1458–1465, 2005.

[16] K. He, F. Wen, and J. Sun. K-means hashing: An affinity-preserving quantization method for learning binary compact codes. In *CVPR*, pages 2938–2945, 2013.

[17] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[18] J.-P. Heo, Y. Lee, J. He, S.-F. Chang, and S.-E. Yoon. Spherical hashing. In *CVPR*, pages 2957–2964, 2012.

[19] A. K. Jain. Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31(8):651–666, 2010.

[20] H. Jegou, F. Perronnin, M. Douze, J. Sánchez, P. Perez, and C. Schmid. Aggregating local image descriptors into compact codes. *TPAMI*, 34(9):1704–1716, 2012.

[21] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *TPAMI*, 35(1):221–231, 2013.

[22] A. Krizhevsky and G. Hinton. Learning multiple layers of features from tiny images. *Master Thesis*, 2009.

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012.

[24] H. Lai, Y. Pan, Y. Liu, and S. Yan. Simultaneous feature learning and hash coding with deep neural networks. In *CVPR*, pages 3270–3278, 2015.

[25] S. Leutenegger, M. Chli, and R. Siegwart. BRISK: Binary robust invariant scalable keypoints. In *ICCV*, pages 2548–2555, 2011.

[26] K. Lin, J. Lu, C.-S. Chen, and J. Zhou. Learning compact binary descriptors with unsupervised deep neural networks. In *CVPR*, pages 1183–1192, 2016.

[27] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[28] J. Lu, V. Erin Liong, and J. Zhou. Simultaneous local binary feature learning and encoding for face recognition. In *ICCV*, pages 3721–3729, 2015.

[29] J. Lu, V. E. Liong, X. Zhou, and J. Zhou. Learning compact binary face descriptor for face recognition. *TPAMI*, 37(10):2041–2056, 2015.

[30] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *TPAMI*, 24(7):971–987, 2002.

[31] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *CVPR*, pages 1717–1724, 2014.

[32] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *BMVC*, volume 1, pages 1–12, 2015.

[33] M. Paulin, M. Douze, Z. Harchaoui, J. Mairal, F. Perronin, and C. Schmid. Local convolutional features with unsupervised training for image retrieval. In *ICCV*, pages 91–99, 2015.

[34] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*, pages 1–8, 2007.

[35] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *CVPR*, pages 1–8, 2008.

[36] X. Qi, R. Xiao, C. Li, Y. Qiao, J. Guo, and X. Tang. Pairwise rotation invariant co-occurrence local binary pattern. *TPAMI*, 36(11):2199–2213, 2014.

[37] X. Qian, X. Hua, P. Chen, and L. Ke. PLBP: An effective local binary patterns texture descriptor with pyramid representation. *PR*, 44(10):2502–2515, 2011.

[38] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. ORB: An efficient alternative to sift or surf. In *ICCV*, pages 2564–2571, 2011.

[39] R. Salakhutdinov and G. Hinton. Semantic hashing. *International Journal of Approximate Reasoning*, 50(7):969–978, 2009.

[40] G. Shakhnarovich. *Learning task-specific similarity*. PhD thesis, Massachusetts Institute of Technology, 2005.

[41] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, pages 1–14.

[42] C. Strecha, A. Bronstein, M. Bronstein, and P. Fua. LDA-Hash: Improved matching with smaller descriptors. *TPAMI*, 34(1):66–78, 2012.

[43] Y. Sun, X. Wang, and X. Tang. Deep learning face representation from predicting 10,000 classes. In *CVPR*, pages 1891–1898, 2014.

[44] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, pages 1–9, 2015.

[45] T. Trzcinski, M. Christoudias, P. Fua, and V. Lepetit. Boosting binary keypoint descriptors. In *CVPR*, pages 2874–2881, 2013.

[46] T. Trzcinski, M. Christoudias, and V. Lepetit. Learning image descriptors with boosting. *TPAMI*, 37(3):597–610, 2015.

[47] T. Trzcinski and V. Lepetit. Efficient discriminative projections for compact binary descriptors. In *ECCV*, pages 228–242, 2012.

[48] J. Wang, S. Kumar, and S.-F. Chang. Semi-supervised hashing for scalable image retrieval. In *CVPR*, pages 3424–3431, 2010.

[49] Y. Weiss, A. Torralba, and R. Fergus. Spectral hashing. In *NIPS*, pages 1753–1760, 2009.

[50] R. Xia, Y. Pan, H. Lai, C. Liu, and S. Yan. Supervised hashing for image retrieval via image representation learning. In *AAAI*, pages 2156–2162, 2014.

[51] S. Zhang, Q. Tian, Q. Huang, W. Gao, and Y. Rui. USB: ultrashort binary descriptor for fast visual matching and retrieval. *TIP*, 23(8):3671–3683, 2014.