Learning Deep Binary Descriptor with Multi-Quantization

Yueqi Duan, Jiwen Lu, *Senior Member, IEEE*, Ziwei Wang, Jianjiang Feng, *Member, IEEE*, and Jie Zhou, *Senior Member, IEEE*

Abstract—In this paper, we propose an unsupervised feature learning method called deep binary descriptor with multi-quantization (DBD-MQ) for visual analysis. 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, which usually suffer from severe quantization loss. In order to address the limitation, we propose a deep multi-quantization network to learn a data-dependent binarization in an unsupervised manner. More specifically, we design a K-Autoencoders (KAEs) network to jointly learn the parameters of feature extractor and the binarization functions under a deep learning framework, so that discriminative binary descriptors can be obtained with a fine-grained multi-quantization. As DBD-MQ simply allocates the same number of quantizers to each real-valued feature dimension ignoring the elementwise diversity of informativeness, we further propose a deep competitive binary descriptor with multi-quantization (DCBD-MQ) method to learn optimal allocation of bits with the fixed binary length in a competitive manner, where informative dimensions gain more bits for complete representation. Moreover, we present a similarity-aware binary encoding strategy based on the earth mover's distance of Autoencoders, so that elements that are quantized into similar Autoencoders will have smaller Hamming distances. Extensive experimental results on six widely-used datasets show that our DBD-MQ and DCBD-MQ outperform most state-of-the-art unsupervised binary descriptors.

Index Terms—Binary descriptor, unsupervised learning, deep learning, competitive learning, multi-quantization, K-Autoencoders

1 Introduction

Feature description is a fundamental computer vision problem which is widely applicable in a number of applications, such as object recognition [18], [43], face recognition [45], [49], [65], image classification [23], [40] 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.

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 [37], VGG [49], [62], GoogLeNet [66], ResNet [26] and DenseNet [29]. 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. Meanwhile, several binary features have been proposed over the past decade due to their efficiency. Representative binary features include local binary pattern (LBP) [1], [47] as well as its variants [53], [54], binary robust independent elementary feature (BRIEF) [9], binary robust invariant scalable keypoint (BRISK) [39], oriented FAST and rotated BRIEF (ORB) [56] and fast retina keypoint (FREAK) [2]. These methods reduce the computational cost by substituting the Euclidean distance with Hamming distance and computing the distances between binary codes using XOR operations.

Inspired by the fact that CNN features present strong discriminative power and binary representations benefit from low computational cost, a number of deep binary descriptor learning methods have been proposed, which achieve the state-of-the-art results in binary representation [32], [40], [42], [61], such as DeepBit [40], textual-visual deep binaries (TVDB) [61] and supervised structured binary code (SUBIC) [32]. For binary representation, binarization is an essential step to enhance the efficiency of the descriptors at the cost of quantization loss. However, most existing deep binary descriptors simply utilize the rigid sign function for binarization despite of data distributions. For many distributions, the hand-crafted zero is not a reasonable threshold for binarization, which may lead to severe quantization loss. Fig. 1 shows that the sign function is not proper for all the three distributions of the real-valued feature dimensions.

• *The authors are with the Department of Automation, Tsinghua University, State Key Lab of Intelligent Technologies and Systems, and Beijing National Research Center for Information Science and Technology (BNRist), Beijing, 100084, China. E-mail: {duanyq14, zw-wa14}@mails.tsinghua.edu.cn, {lujiwen, jfeng, jzhou}@tsinghua.edu.cn.*

Manuscript received 24 Feb. 2018; revised 26 June 2018; accepted 16 July 2018. Date of publication 22 July 2018; date of current version 11 July 2019.
(Corresponding author: Jiwen Lu.)
Recommended for acceptance by R. Manmatha.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TPAMI.2018.2858760

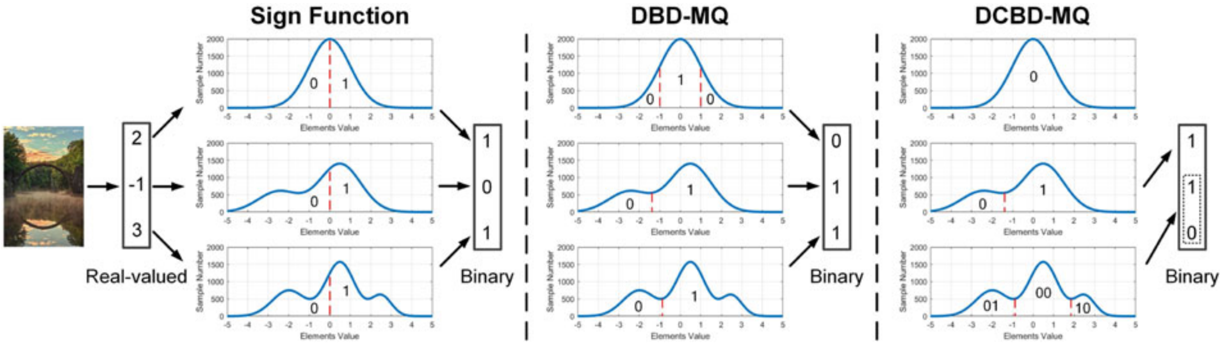


Fig. 1. An illustration of binarizing three real-valued feature dimensions under varying distributions with the sign function, DBD-MQ, and DCBD-MQ, fixing the total number of bits as three. In the figure, red dashes represent the threshold and we show the binarization results with binary codes. For all the three distributions, it is not very reasonable to simply employ the sign function by setting the threshold as zero. Compared with the sign function, DBD-MQ learns a data-dependent binarization to reduce the quantization loss. While the sign function and DBD-MQ evenly allocate bits to the feature dimensions (1 bit per dimension), DCBD-MQ exploits the elementwise diversity of informativeness by adaptively learning the allocation of bits with the fixed total binary length.

In order to address these limitations, we propose a deep multi-quantization network to learn data-dependent binarization functions in an unsupervised manner. For each real-valued element, we determine its binary code based on the quantization result, where the sign function is a special case to quantize positives into one class and negatives into another. Fig. 1 shows the data-dependent binarization results of varying distributions. Compared with the hand-crafted threshold, multi-quantization exploits the distributions of each feature dimension and obtains fine-grained binarization results. More specifically, we propose a K-Autoencoders (KAEs) network for data-dependent binarization and present a deep binary descriptor with multi-quantization (DBD-MQ) learning method. Fig. 2 illustrates the flowchart of the proposed approach. With the KAEs based multi-quantization, we jointly learn the parameters of the network and the binarization functions to obtain more discriminative binary codes.

While DBD-MQ learns data-dependent binarization functions, it allocates the same number of bits to each real-valued feature dimension despite of elementwise diversity of informativeness. Inspired by the fact that the discriminative dimensions deserve more bits for complete description, we further propose a deep competitive binary descriptor with multi-quantization (DCBD-MQ) learning method by encouraging elementwise contest for quantizers with the fixed total binary length. Through the competition, discriminative dimensions gain more bits for representation while some uninformative dimensions are eliminated. Fig. 1 shows that the third dimension grabs one more bit from the first dimension due to its discriminativeness. Once a real-valued feature dimension is quantized into multiple bits as shown in the third distribution of Fig. 1, the binary encoding for quantizers would be uncertain where different pairs of quantizers may have varying Hamming distances. In order to obtain a similarity-aware binary encoding strategy, we present an earth mover's distance (EMD) [57] based similarity measurement for Autoencoders, so that similar quantizers would be encoded into binary codes with smaller Hamming distances. Extensive experimental results on the CIFAR-10 [36], Brown [8], HPatches [6], Paris [52], Oxford [51] and INRIA Holidays [33] datasets show the effectiveness of the proposed methods.

This paper is an extended version of our conference paper [15], where we make the following new contributions:

(1) We further propose a new DCBD-MQ method based on DBD-MQ in the conference version by adaptively

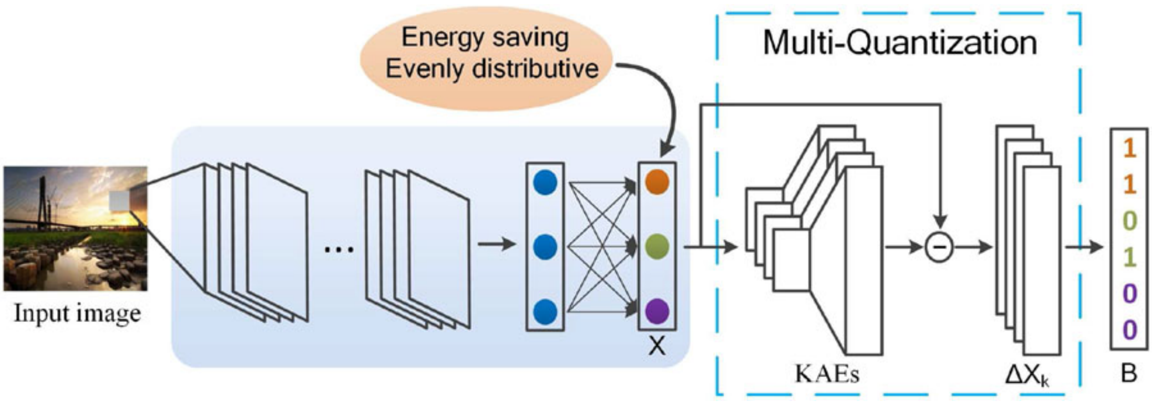


Fig. 2. The flowchart of the proposed DBD-MQ. For each image patch from the training set, we first learn a real-valued feature vector with a pre-trained CNN by replacing the softmax layer with a fully connection layer. Then, we binarize the vectors with the K-Autoencoders (KAEs) based multi-quantization instead of the rigid sign function, which minimizes the reconstruction loss by controlling the residual features $\Delta \mathbf{X}_k$. K is equal to 4 in this figure for 2-bit binary encoding, where each feature dimension is quantized to two bits with the same color. Lastly, we optimize the parameters iteratively with back-propagation in an unsupervised manner to obtain compact binary codes. (Best viewed in color.)

TABLE 1
Comparison of the Proposed Approaches to the Widely-Used Binary Representations

Method	Type	Supervision	Compactness	Binarization	Allocation of bits
LBP [47]	Hand-crafted	No	No	Threshold	Even
BRIEF [9]	Hand-crafted	No	No	Threshold	Even
BRISK [39]	Hand-crafted	No	No	Threshold	Even
ORB [56]	Hand-crafted	No	No	Threshold	Even
FREAK [2]	Hand-crafted	No	Yes	Threshold	Even
CBFD [45]	Shallow	No	Yes	Threshold	Even
CA-LBFL [14]	Shallow	No	Yes	Threshold	Even
D-BRIEF [70]	Shallow	Yes	Yes	Threshold	Even
BOLD [7]	Shallow	Yes	No	Threshold	Even
BinBoost [68]	Shallow	Yes	Yes	Threshold	Even
DeepBit [40]	Deep	No	Yes	Threshold	Even
TVDB [61]	Deep	Yes	No	Threshold	Even
SUBIC [32]	Deep	Yes	No	Data-dependent	Even
BDNN [12]	Deep	Both	Yes	Data-dependent	Even
DBD-MQ [15]	Deep	No	Yes	Data-dependent	Even
DCBD-MQ	Deep	No	Yes	Data-dependent	Competitive

learning the allocation of bits for the real-valued feature dimensions with the fixed total binary length, so that discriminative dimensions grab more bits from the uninformative ones for complete description.

(2) We present a similarity-aware binary encoding strategy for multiple bits by designing an EMD based similarity measurement of Autoencoders, so that similar quantizers have smaller Hamming distances.

(3) We conduct extensive experiments on more public benchmark datasets to demonstrate the effectiveness of the proposed methods, which include the latest image patch dataset with three baseline visual analysis tasks.

2 Background

In this section, we briefly review two related topics: binary representation and deep learning.

2.1 Binary Representation

Binary representations have aroused extensive interest due to their efficiency of matching and storing in recent years. Earlier binary features include BRIEF [9], BRISK [39], ORB [56] 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 [7], [68], [70], [75]. For example, Trzcinski et al. [70] proposed a D-BRIEF method by encoding similarity relationships to learn discriminative projections. Balntas et al. [7] 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 [13], [14], [22], [40], [44], [45], [58], [73]. For example, Salakhutdinov and Hinton [58] proposed a semantic hashing (SH) approach by learning binary codes with Restricted Boltzmann Machines (RBM). Weiss et al. [73] presented a Spectral hashing (SpeH) method through spectral graph partitioning. Lu et al. [45] proposed a compact binary face descriptor (CBFD) to learn evenly-distributive and energy-saving local binary codes. They also presented a simultaneous local binary feature learning and encoding (SLBFLE) [44] method by jointly learning binary codes and the codebook in a one-stage procedure. Lin et al. [40] proposed a DeepBit by designing a CNN to learn compact binary codes in an unsupervised manner. Duan et al. [14] presented a context-aware local binary feature learning (CA-LBFL) approach to exploit contextual bitwise interaction. Table 1 shows an overview of the widely-used binary representations, where *compactness* represents whether the redundancy is removed in the binary representation. We observe that most of these methods utilize a hand-crafted threshold for binarization, which ignore the distributions of the real-valued feature dimensions and the allocation of bits.

2.2 Deep Learning

There has been extensive work on deep learning in recent years [10], [26], [29], [37], [48], [49], [62], [66], which achieves the state-of-the-art performance in many computer vision applications, such as object recognition [26], [29], [62], object detection [20], [21], [55], face recognition [49], [65] and human action recognition [35], [63]. With large amount of data, deep learning methods learn high-level hierarchical features by training powerful statistical models to obtain higher quality representation. However, most deep features are high-dimensional and real-valued, which require strong computational capabilities.

In recent years, several deep binary representation learning methods have also been proposed [12], [16], [32], [38],

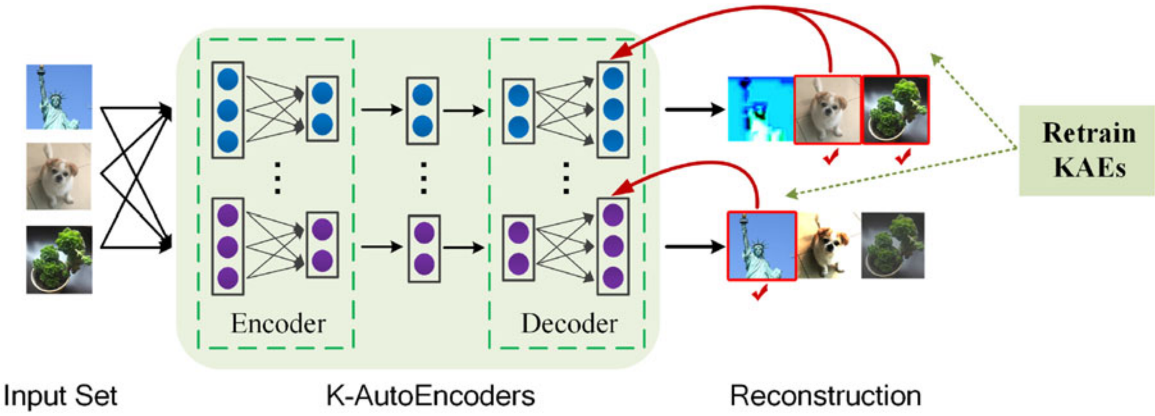


Fig. 3. A detailed explanation of training KAEs. For each image from the input set, we first encode and decode its CNN feature with all KAEs. Then, we associate each feature with the Autoencoder obtaining the minimum reconstruction loss, which is highlighted with a red box. Lastly, we utilize the corresponding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

[40], [41], [42], [61], [74], [76], where the recent survey paper [72] presented an exhaustive review. For example, Xia et al. [74] proposed a CNN hashing (CNNH) method by learning deep hashing codes and image representation in a supervised manner. Lai et al. [38] improved CNNH by presenting a one-stage deep binary representation learning procedure. Liong et al. [16] proposed a deep hashing (DH) method by learning multiple non-linear hierarchical transformations under three constraints. Shen et al. [61] presented textual-visual deep binaries (TVDB) to exploit the detailed semantics with an integrated deep architecture. Jain et al. [32] proposed a supervised structured binary code (SUBIC) with a one-hot block structure. Compared with existing deep binary representation learning methods, the proposed DCBD-MQ learns data-dependent binarization with competitive allocation of bits, which fully exploits the varying distributions of the real-valued feature dimensions to minimize the quantization loss.

3 Deep Binary Descriptor with Multi-Quantization

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 [40], [44], [45], yet all of them utilize the rigid sign function to quantize each dimension 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 distributions as examples, which are shown in Fig. 1. All the 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 distributions, 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 the elements in each dimension from different features are more possible to be quantized into the same binary codes if their holistic feature vectors are similar.

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 DBD-MQ. In the training procedure of KAEs, we quantize the holistic feature vectors to K Autoencoders for parameter optimization. In the test procedure for binarization, each feature dimension is clustered into one of K classes, which leads to c-bit encoding per dimension. The conventional sign function is a special case which clusters negatives into one class and positives into another. As a 2-clustering approach, each feature dimension 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 [31], 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:

$$k_n = \arg\min_k \varepsilon_{nk}, \qquad (1)$$

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. Fig. 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 dimension of the feature vectors into binary codes, we consider the elementwise quantization loss $\varepsilon_{nk}^{(i)} = |\Delta \mathbf{x}_{nk}^{(i)}|$, and the clustering approach of each dimension is formulated as follows:

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

where the ith dimension of $\mathbf{x}_n$ is clustered into the $k_n^{(i)}$th Autoencoder. Each feature dimension is clustered to the Autoencoder with the minimum elementwise 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 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 the corresponding binary codes for each feature dimension, which are concatenated into the binary descriptor.

3.2 DBD-MQ

We initialize the CNN with the pre-trained 16 layers VGGNet [62] trained on the ImageNet dataset, which replaces the softmax layer with a fully connection layer. Fig. 2 shows the flowchart of the proposed DBD-MQ. Let $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N]$ 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:

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

where $\mathbf{W}_k$ represents the parameters 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 dimension of the features, so that each dimension 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 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:

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

and we apply stochastic gradient descent (SGD) 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 SGD 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.

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 parameters of
  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.
```

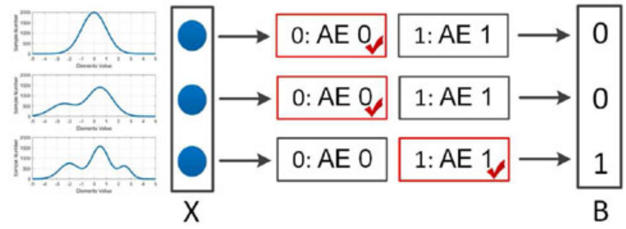
In the training procedure, we simultaneously learn the parameters of CNN and the KAEs to obtain energy-saving and evenly-distributive binary descriptors. In the test procedure, for each local patch, we first learn its real-valued feature representation using the learned CNN, and then quantize each feature dimension into binary codes with the learned KAEs using (2), which are concatenated into a longer binary descriptor as the final representation. Fig. 4a shows an example of binary encoding with the learned KAEs. As the dimension of 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 [40], and then crop it into 224×224 to remove the background information.

3.3 Discussion

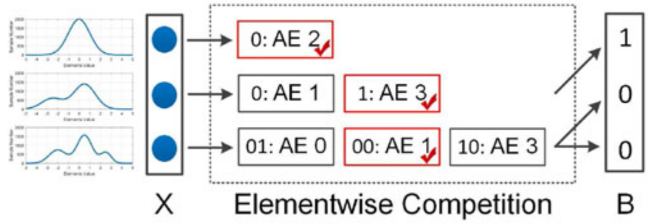
Our DBD-MQ improves the conventional sign function based binary representation learning methods in the following two aspects:

(1) Instead of employing a hand-crafted threshold, the proposed DBD-MQ simultaneously 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) The parameters of KAEs are learned from holistic feature vectors, minimizing the reconstruction error of similar real-valued descriptors in the corresponding Autoencoder. Therefore, elements from similar



(a) DBD-MQ



(b) DCBD-MQ

Fig. 4. Examples of data-dependent binarization in (a) DBD-MQ, and (b) DCBD-MQ. For DBD-MQ, we set $K = 2$ for easy illustration, where each dimension is binarized into one bit. We quantize each real-valued element to the Autoencoder with the minimum reconstruction loss according to (2), and obtain the binary code through the quantization result. For DCBD-MQ, there are four KAEs in the original set, where the real-valued dimensions compete for more Autoencoders from the set with the fixed total binary length. Based on the informativeness of each feature dimension, the first dimension only obtains *AE 2* with 0 bit for representation, the second dimension receives *AE 1* and *AE 3* with 1 bit, and the third dimension gains *AE 0*, *AE 1* and *AE 3* with 2 bits.

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 [40], [45] which quantize each bit separately, the holistic real-valued descriptors provide strong prior knowledge for the binarization of each feature dimension, which enhances the robustness and stability of the learned binary descriptors.

4 Deep Competitive Binary Descriptor with Multi-Quantization

In this section, we first propose the deep competitive binary descriptor with multi-quantization (DCBD-MQ) learning method, and then present the earth mover's distance (EMD) based similarity-aware binary encoding for KAEs.

4.1 DCBD-MQ

While DBD-MQ learns data-dependent binarization for real-valued features, it allocates the same number of bits for each feature dimension, which ignores the elementwise diversity of informativeness. With the fixed total binary length, discriminative dimensions deserve more bits for fully representation as shown in Fig. 1. In order to address the limitation, we further propose a DCBD-MQ learning approach by encouraging elementwise competition for Autoencoders. Different from DBD-MQ which uses all the KAEs to quantize each real-valued feature dimensions, elements in DCBD-MQ fight for more Autoencoders from the original KAEs set, so that more informative dimensions gain more Autoencoders and result in more bits for representation. Fig. 4b shows an example of elementwise competition in DCBD-MQ.

Let $\mathbb{K} = \{1, 2, \ldots, K\}$ be the original set of KAEs, where $\mathbb{K}_i \subset \mathbb{K}$ represents the K_i Autoencoders picked by the ith dimension. We define a binary matrix $\mathbf{C} \in \{0,1\}^{d\times K}$ to register the allocation of Autoencoders, where $C_{ik} = 1$ only if $k \in \mathbb{K}_i$ and $K_i = \sum_{k=1}^{K} C_{ik}$. DBD-MQ can be seen as a special case of DCBD-MQ when all the elements in $\mathbf{C}$ are ones. Note that K and K_i can be any positive integers in DCBD-MQ rather than 2^c, and the number of bits for the ith dimension is determined by the shortest binary encoding of K_i Autoencoders:

$$c_i = \lceil \log{}_2 K_i \rceil, \tag{6}$$

where $\lceil x \rceil$ is the minimum integer greater than or equal to x.

We define the objective function for DCBD-MQ as follows:

$$\begin{aligned}
\min_{\mathbf{X},\mathbf{W}_k,\mathbf{C}} J &= J_1 + \lambda_1 J_2 + \lambda_2 J_3 + \lambda_3 J_4 \\
&= \sum_{n=1}^{N}\left(\varepsilon^2_{nk_n} + \alpha\left(\sum_{i=1}^{d}\varepsilon^{(i)}_{nk_n^{(i)}}\right)^2\right) \\
&\quad + \lambda_1\sum_{k=1}^{K}||\mathbf{W}_k||_F^2 - \lambda_2 \mathrm{tr}((\mathbf{X}-\mathbf{U})^T(\mathbf{X}-\mathbf{U})) \\
&\quad + \lambda_3\left(\sum_{k=1}^{K}\left(\frac{||\mathbf{C}||_F}{K} - \sum_{i=1}^{d} C_{ik}\right)^2 - \beta\sum_{i=1}^{d} r_i^2\right), \\
\text{subject to} &\quad \sum_{i=1}^{d} c_i = d,
\end{aligned} \tag{7}$$

where

$$\varepsilon^{(i)}_{nk_n^{(i)}} = \min_{k\in\mathbb{K}_i}\varepsilon^{(i)}_{nk}, \tag{8}$$

represents the minimum reconstruction loss of the ith dimension $x_n^{(i)}$ in $\mathbf{x}_n$ among the Autoencoders in $\mathbb{K}_i$, and $r_i = 2^{c_i} - K_i$ is the remaining number of Autoencoders that can be used with c_i bits. For example, the third dimension in Fig. 4b gains three Autoencoders ($K_i = 3$), which requires two bits for representation ($c_i = 2$), and one more Autoencoder can further be used without increasing the binary length ($r_i = 1$).

Compared with (3), the objective function of DCBD-MQ modifies J_1 and add J_4 for competitive binarization. In J_1, we simultaneously minimize the reconstruction losses of the real-valued features and elements for elementwise selection of Autoencoders. In J_4, the first term encourages each Autoencoder to be selected by the same number of elements, and the second term prevents from redundant Autoencoders which make little contribution under the same binary length. We set λ_1, λ_2, λ_3, α and β as 0.004, 0.4, 10, 0.1 and 0.1, respectively. Similarly, we employ an iterative training strategy to update one with the others fixed.

Learning $\mathbf{W}_k$ *fixing* $\mathbf{X}$ *and* $\mathbf{C}$: when $\mathbf{X}$ and $\mathbf{C}$ are fixed, we can rewrite the objective function (7) as follows:

$$\min_{\mathbf{W}_k} J = \sum_{n=1}^{N}\left(\varepsilon^2_{nk_n} + \alpha\left(\sum_{i=1}^{d}\varepsilon^{(i)}_{nk_n^{(i)}}\right)^2\right) + \lambda_1\sum_{k=1}^{K}||\mathbf{W}_k||_F^2, \tag{9}$$

and we also employ SGD to update $\mathbf{W}_k$.

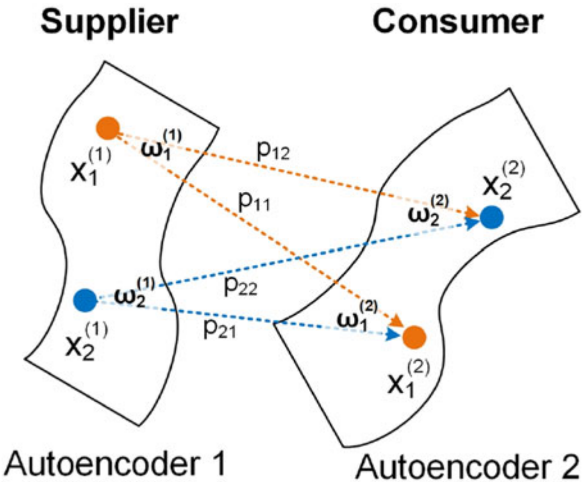


Fig. 5. An example of pointwise distances between Autoencoders. In the figure, the points in the same color are reconstructed from the same original samples, and p_{n_1,n_2} and $\omega_n^{(k)}$ show the weights and the total supply/capacity, respectively. We exploit all the pointwise distances through the reconstruction results to completely describe the distance between Autoencoders. (Best viewed in color.)

Learning $\mathbf{C}$ *fixing* $\mathbf{W}_k$ *and* $\mathbf{X}$: when the parameters of KAEs and CNN are fixed, we can rewrite the objective function (7) to learn the elementwise allocation of Autoencoders as follows:

$$\begin{aligned}\min_{\mathbf{C}} J = \alpha \sum_{n=1}^{N} \left(\sum_{i=1}^{d} \varepsilon^{(i)}_{nk_n^{(i)}} \right)^2 \\ + \lambda_3 \left(\sum_{k=1}^{K} \left(\frac{||\mathbf{C}||_F}{K} - \sum_{i=1}^{d} C_{ik} \right)^2 - \beta \sum_{i=1}^{d} r_i^2 \right) \\ \text{subject to} \quad \sum_{i=1}^{d} c_i = d.\end{aligned} \tag{10}$$

With the constraint of the binary length, we learn an optimal allocation of Autoencoders to minimize (10), so that more discriminative feature dimensions gain more bits for representation. As optimizing (10) is a combinatorial problem, we initialize $\mathbf{C}$ by utilizing two Autoencoders for each dimension, and apply dynamic-programming to learn the optimal allocation of Autoencoders.

Learning $\mathbf{X}$ *fixing* $\mathbf{W}_k$ *and* $\mathbf{C}$: when KAEs and their elementwise allocation are fixed, we can rewrite the objective function (7) as follows:

$$\begin{aligned}\min_{\mathbf{X}} J = \sum_{n=1}^{N} \left(\varepsilon^2_{nk_n} + \alpha \left(\sum_{i=1}^{d} \varepsilon^{(i)}_{nk_n^{(i)}} \right)^2 \right) \\ - \lambda_2 \mathrm{tr}((\mathbf{X}-\mathbf{U})^T(\mathbf{X}-\mathbf{U})),\end{aligned} \tag{11}$$

and we update $\mathbf{X}$ with the SGD algorithm.

4.2 Similarity-Aware Binary Encoding

When a real-valued feature dimension gains more than two Autoencoders ($K_i > 2$) for quantization, it would be binarized into multiple bits. However, the binary encoding for Autoencoders is uncertain in this case, where different pairs of Autoencoders may have varying Hamming distances. For example, in the third dimension of Fig. 4b, the Hamming distance between *AE 0* and *AE 3* is 2, while the distance between *AE 0* and *AE 1* is 1. In order to obtain a similarity-aware binary encoding strategy, the key is to measure the distances between pairs of Autoencoders.

In this paper, we propose an earth mover's distance (EMD) [57] based similarity measurement for Autoencoders. For a pair of Autoencoders k_1 and k_2, each sample $\mathbf{x}_n$ would have two reconstruction results $\mathbf{x}_n^{(k_1)}$ and $\mathbf{x}_n^{(k_2)}$, respectively. We employ the pointwise distance $||\Delta\mathbf{x}_n^{(k_1,k_2)}||_2 = ||\mathbf{x}_n^{(k_1)} - \mathbf{x}_n^{(k_2)}||_2$ to describe the pointwise distance of the same sample between a pair of Autoencoders. However, as the reconstructed subspace of each Autoencoder suffers from highly nonlinearity, the pointwise distance of the same sample may not be optimal to fully describe the distance between Autoencoders. To this end, we consider all the pointwise distances between the Autoencoders $||\Delta\mathbf{x}_{n_1,n_2}^{(k_1,k_2)}||_2 = ||\mathbf{x}_{n_1}^{(k_1)} - \mathbf{x}_{n_2}^{(k_2)}||_2$ in a more general manner as shown in Fig. 5 rather than only using the ones reconstructed from the same original samples. We convert the distance between a pair of Autoencoders to the integrated pointwise distances:

$$\begin{aligned}\mathbf{D}(k_1,k_2) = \sum_{n_1=1}^{N}\sum_{n_2=1}^{N} \hat{p}_{n_1,n_2}^{(k_1,k_2)} ||\Delta\mathbf{x}_{n_1,n_2}^{(k_1,k_2)}||_2 \\ \text{subject to} \quad \sum_{n_1=1}^{N}\sum_{n_2=1}^{N} \hat{p}_{n_1,n_2}^{(k_1,k_2)} = 1,\ \hat{p}_{n_1,n_2}^{(k_1,k_2)} \geq 0,\end{aligned} \tag{12}$$

where the distance between Autoencoders is represented as a weighted average of pointwise distances, and we should choose proper $\hat{p}$ to determine $\mathbf{D}(k_1,k_2)$ in (12). In the following, we omit the superscript of $\hat{p}_{n_1,n_2}^{(k_1,k_2)}$ for simplicity.

We exploit EMD to compute the weights in (12). We consider $\mathbf{x}_{n_1}^{(k_1)}$ as N suppliers, where the total supply of each supplier is $\omega_{n_1}^{(k_1)}$. Similarly, we consider $\mathbf{x}_{n_2}^{(k_2)}$ as N consumers, where the total capacity of each consumer is $\omega_{n_2}^{(k_2)}$. We set the default value of $\omega_n^{(k)}$ as $\frac{1}{||\Delta\mathbf{x}_{nk}||_2}$, so that the points with less reconstruction losses gain larger supply or capacity. The task is to deliver products from suppliers to consumers, and the weight p_{n_1,n_2} represents the quantity of the $\mathbf{x}_{n_1}^{(k_1)} \rightarrow \mathbf{x}_{n_2}^{(k_2)}$ delivery. Fig. 5 shows an example to illustrate the physical meanings of the variables. We compute the EMD between Autoencoders by:

$$\mathbf{D}(k_1,k_2) = \frac{\sum_{n_1=1}^{N}\sum_{n_2=1}^{N} p_{n_1,n_2} ||\Delta\mathbf{x}_{n_1,n_2}^{(k_1,k_2)}||_2}{\sum_{n_1=1}^{N}\sum_{n_2=1}^{N} p_{n_1,n_2}}, \tag{13}$$

where $||\Delta\mathbf{x}_{n_1,n_2}^{(k_1,k_2)}||_2$ is the ground distance, and we obtain the optimal flow p_{n_1,n_2} by solving the linear programming problem as follows:

$$\begin{aligned}p_{n_1,n_2} = \arg\min_{p'_{n_1,n_2}} \sum_{n_1=1}^{N}\sum_{n_2=1}^{N} p'_{n_1,n_2} ||\Delta\mathbf{x}_{n_1,n_2}^{(k_1,k_2)}||_2 \\ \text{subject to} \quad \sum_{n_2=1}^{N} p'_{n_1,n_2} \leq \omega_{n_1}^{(k_1)},\ \sum_{n_1=1}^{N} p'_{n_1,n_2} \leq \omega_{n_2}^{(k_2)} \\ \sum_{n_1=1}^{N}\sum_{n_2=1}^{N} p'_{n_1,n_2} = \min\left(\sum_{n_1=1}^{N} \omega_{n_1}^{(k_1)}, \sum_{n_2=1}^{N} \omega_{n_2}^{(k_2)} \right) \\ p'_{n_1,n_2} \geq 0.\end{aligned} \tag{14}$$

TABLE 2
Summarization of the Benchmark Datasets Used in the Experiments

Dataset	Input type	Task	Category	Training samples	Test samples
CIFAR-10	Image patch	Patch retrieval	10 classes	50,000	10,000
Brown	Image patch	Patch matching	Pairwise	200,000	100,000
HPatches	Image patch	Patch verification, matching, retrieval	76 classes	$\sim 10^6$	$\sim 10^6$
Paris	Landscape image	Image retrieval	N/A	N/A	55
Oxford	Landscape image	Image retrieval	N/A	N/A	55
INRIA Holidays	Natural image	Image retrieval	N/A	N/A	500

In (14), the first two constraints limit the total amount of supply and capacity for each point. The third constraint aims at the total flow by encouraging maximum supplies. The last constraint allows a directional flow. With the learned weights p_{n_1,n_2}, we measure the distances between Autoencoders according to (13). As each real-valued feature dimension would obtain only a few bits for representation at most, we can encode the selected Autoencoders for each element to maintain the relative distances through exhaustive search. Algorithm 2 summarizes the detailed approach of the proposed DCBD-MQ.

Algorithm 2. DCBD-MQ

Input: Training image set, parameters λ_1 and λ_2, and iteration number T.
Output: Projection parameters of CNN $\mathbf{W}$, parameters of KAEs $\mathbf{W}_k$, and allocation of KAEs $\mathbf{C}$.
Initialize pre-trained CNN features $\mathbf{X}$, parameters of KAEs $\mathbf{W}_k$, and allocation of KAEs $\mathbf{C}$.
for $iter = 1, 2, \ldots, T$ **do**
  loop
    Cluster each $\mathbf{x}_n$ into an Autoencoder using (1).
    Quantize each $x_n^{(i)}$ into an Autoencoder with (8).
    Update $\mathbf{W}_k$ with corresponding $\mathbf{x}_n$ and $x_n^{(i)}$ using (9).
  end loop until convergence
  Allocate KAEs to feature dimensions with others fixed using (10).
  Update CNN fixing others with (11).
end for
Encode the Autoencoders of each element according to (13).
return $\mathbf{W}$, $\mathbf{W}_k$ and $\mathbf{C}$.

5 Experiments

We evaluated the proposed DBD-MQ and DCBD-MQ methods on six challenging datasets including the CIFAR-10 [36], Brown [8], HPatches [6], Paris [52], Oxford [51] and INRIA Holidays [33] datasets. We conducted experiments on four different visual analysis tasks, which contain patch retrieval, patch matching, patch verification and image retrieval. We compared the proposed methods with several state-of-the-art unsupervised binary descriptors to demonstrate their effectiveness. Table 2 summarizes the benchmark datasets used in the experiments.

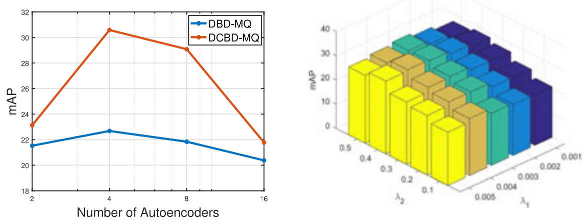
5.1 Results on Patch Retrieval

The CIFAR-10 dataset [36] 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 [36], and tested the proposed DBD-MQ and DCBD-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 and DCBD-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.

Then, we tested the mean average precision (mAP) under different number of Autoencoders K, with the structure of Autoencoders fixed as $[16 \rightarrow 12 \rightarrow 8 \rightarrow 12 \rightarrow 16]$. For DBD-MQ, Fig. 6a shows that the best result was obtained when K is equal to 4. Although the binary lengths are 16, 32, 48 and 64 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 dimension, 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. For DCBD-MQ, it is worth noticing that the binary length is fixed to 16 despite of varying numbers of Autoencoders, and the only difference between DCBD-MQ and DBD-MQ would be the first term J_1 and the parameters in the objective function if K is equal to 2. As each feature dimension only selects some of the Autoencoders rather than using all of them, the description would suffer from severe locality when K is too large.

In our experiments, we fix $K = 2$ for DBD-MQ and $K = 4$ for DCBD-MQ. For DBD-MQ, $K = 2$ leads to 1-bit encoding per dimension and $K = 4$ results in 2-bit encoding. In general, there are mainly three reasons that we set K to 2:



(a) Autoencoders (b) λ_1 and λ_2

Fig. 6. The mean average precision (mAP) performance (%) of (a) DBD-MQ and DCBD-MQ under varying number of Autoencoders, and (b) 16-bit DCBD-MQ under different λ_1 and λ_2.

TABLE 3
The Mean Average Precision (mAP) Performance (%) of Top 1,000 Returned Images for DCBD-MQ without Specific Terms on CIFAR-10

Method	16 bits	32 bits	64 bits
DCBD-MQ ($\lambda_1 = 0$)	30.46	32.95	36.52
DCBD-MQ ($\lambda_3 = 0$)	22.06	27.41	32.73
DCBD-MQ ($\beta = 0$)	30.14	32.09	35.64
DCBD-MQ	**30.58**	**33.01**	**36.59**

(1) One of the key advantages for binary representation learning is the high efficiency. In Fig. 6a, the improvement is relatively small (by 1.15 percent mAP) from $K = 2$ to $K = 4$ at the cost of doubling the dimension of the final representations, where we consider the setting of $K = 2$ to be more applicable in most cases.

(2) The mAP is 22.68 percent on CIFAR-10 to binarize 16-dimensional real-valued features with $K = 4$, while the performance is 26.50 percent for 32-dimensional real-valued features with $K = 2$ according to the experimental results on CIFAR-10 in Table 4. As both methods share the same binary length of 32, it is more effective to increase the dimension of real-valued features for longer binary codes.

(3) As most existing binary representations employ 1-bit encoding strategies [16], [27], [40], we set K to 2 for fair comparisons.

For DCBD-MQ, we directly select $K = 4$ with the best result as the binary length is fixed with different K in DCBD-MQ.

We also studied the influence of different terms. More specifically, we examined the mAP of 16-bit DCBD-MQ versus different values of λ_1 and λ_2 by fixing other parameters. Fig. 6b shows that the best performance was obtained when the parameters λ_1 and λ_2 were selected as 0.004 and 0.4, respectively.

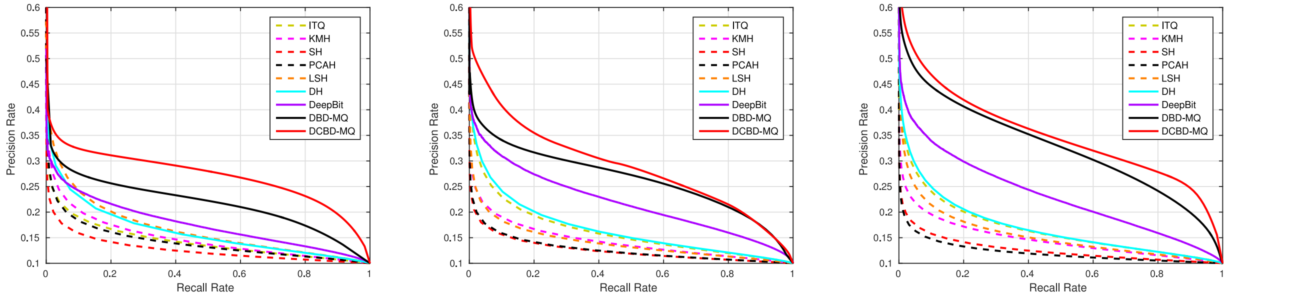
There are five parameters including λ_1, λ_2, λ_3, α and β in (7), and we designed an ablation study with some parameters set to 0 to demonstrate the impact of each term. As λ_2 and α are the bases of feature learning and bitwise allocation which cannot be removed, we tested the performance of DCBD-MQ by fixing λ_1, λ_3 and β to 0 on CIFAR-10, respectively. Table 3 show the experimental results. For $\lambda_1 = 0$, the performance drops slightly and the training process of KAEs may suffer from overfitting. For $\lambda_3 = 0$, DCBD-MQ is more likely to degenerate to DBD-MQ (with the modified J_1) as feature dimensions may tend to select the same well-trained Autoencoders. For $\beta = 0$, redundant Autoencoders would be selected with the same binary length. For example, four Autoencoders will always be used instead of three for 2-bit encoding.

Comparison with the State-of-the-Art Unsupervised Binary Descriptors: We compared the proposed DBD-MQ and DCBD-MQ with several state-of-the-art unsupervised binary descriptors on this image retrieval task, where deep hashing (DH) and DeepBit are two latest deep binary representation learning methods. Table 4 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 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. While DBD-MQ allocates the same number of bits to each dimension despite of the diversity in informativeness (1 bit per dimension under $K = 2$), the proposed DCBD-MQ learns a more optimal allocation of bits in a competitive manner. As the discriminative feature dimensions gain more bits for fully representation, it further boosts the average mAP by 6.77 percent. We also evaluated the performance of only modifying J_1 according to (7) for DBD-MQ, as shown in DBD-MQ + J_1 of Table 4. We observe that the modified J_1 term slightly boosts the performance of DBD-MQ. Fig. 7 illustrates the Precision/Recall curves of the proposed methods and the state-of-the-art unsupervised binary descriptors. We observe that the proposed DBD-MQ and DCBD-MQ consistently outperform other approaches.

TABLE 4
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 [25]	13.59	13.93	14.46
SphH [27]	13.98	14.58	15.38
SpeH [73]	12.55	12.42	12.56
SH [58]	12.95	14.09	13.89
PCAH [71]	12.91	12.60	12.10
LSH [3]	12.55	13.76	15.07
PCA-ITQ [22]	15.67	16.20	16.64
DH [16]	16.17	16.62	16.96
DeepBit [40]	19.43	24.86	27.73
DBD-MQ [15]	21.53	26.50	31.85
DBD-MQ + J_1	21.71	26.84	32.15
DCBD-MQ	**30.58**	**33.01**	**36.59**

Evaluation of Different Binarization Strategies: One of the most significant contributions of the proposed DBD-MQ and DCBD-MQ is the application of KAEs for fine-grained binarization. In the previous experiments, we obtained state-of-the-art performance compared with existing unsupervised binary descriptors, yet it could not directly show the effectiveness of 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 5 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



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

Fig. 7. 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.

improvement of KAEs becomes more significant. On one hand, KAEs minimizes 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 feature dimension.

As other quantization methods can also be used in the proposed framework, we conducted another experiment on CIFAR-10 to compare our KAEs with the K-Means method, where the same real-valued descriptors were used. Table 5 shows that KAEs achieves higher mAP and suffers from less mean quantization loss. The main reason is that KAEs performs quantization by learning K subspace projections rather than K centroids, which presents stronger descriptive power and robustness.

Moreover, we evaluated the proposed similarity-aware binary encoding strategy on the CIFAR-10 dataset. As aforementioned, the discriminative dimensions in DCBD-MQ may gain more than two Autoencoders for representation, which leads to confusing binary encoding. The proposed similarity-aware binary encoding strategy is designed to minimize the Hamming distance between similar Autoencoders. Table 6 shows the experimental results of the similarity-aware binary encoding strategy compared with the random encoding. We observe that the proposed encoding strategy has a more precise similarity measurement in Hamming space, which achieves better performance on the CIFAR-10 dataset.

TABLE 5
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 (1.43)**	**26.50 (1.92)**	**31.85 (2.84)**
Sign	19.16 (-)	23.89 (-)	26.90 (-)
K-Means	20.59 (1.56)	24.94 (2.20)	30.92 (3.18)

Numbers in parentheses represent the mean quantization loss for the quantization based methods.

TABLE 6
The Mean Average Precision (mAP) Performance (%) of Different Encoding Strategies on the CIFAR-10 Dataset under Different Binary Code Length

Encoding	16 bits	32 bits	64 bits
Similarity-aware	**30.58**	**33.01**	**36.59**
Random	30.20	31.81	34.97
ΔmAP	0.38	1.20	1.62

Learning with Light-Weight CNN Models: While the proposed binary descriptors are efficient for storage and matching, a natural question is raised: can we use a light-weight CNN model to further accelerate the procedure of feature extraction? As we have evaluated the effectiveness of the proposed methods with a very deep network structure VGG, we tested the performance of DCBD-MQ with simplified CNN models in this subsection. We employed SqueezeNet [30] and MobileNet [28] to initialize the network, where SqueezeNet replaces 3×3 filters with 1×1 to reduce the number of parameters and MobileNet utilizes depth-wise separable convolutions. In order to train DCBD-MQ, we replace the softmax layer of SqueezeNet and MobileNet with a fully connected layer, which is initialized with random Gaussian. Table 7 shows the mAP of DCBD-MQ with varying CNN models under different binary code length and the total number of parameters on the CIFAR-10 dataset. We observe that DCBD-MQ also achieves encouraging performance with much less parameters.

Image Clustering Results. As KAEs quantizes the input image patches into K classes, we show the cluster samples in Fig. 8 under $K = 4$ on the CIFAR-10 dataset. We observe that patches with similar semantic contents are usually clustered together by the same Autoencoder, where vehicles and ships are quantized into the first group, quadrupeds for the second, birds for the third and aircrafts for the last group.

TABLE 7
The Mean Average Precision (mAP) Performance (%) and Total Parameters of DCBD-MQ with Varying CNN Models on the CIFAR-10 Dataset under Different Binary Code Length

Encoding	16 bits	32 bits	64 bits	Parameters
VGG	**30.58**	**33.01**	**36.59**	134M
SqueezeNet	22.32	24.20	27.81	1.2M
MobileNet	27.13	29.97	32.42	4.2M

Fig. 8. Samples of the clustered images under $K = 4$, where each group of images is clustered with the same Autoencoder.

TABLE 8
95 Percent 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 Features and LDAHash, D-BRIEF, BinBoost, RFD, Binary L2-Net and Binary DOAP Are Supervised

Train Test	Yosemite Notre Dame	Yosemite Liberty	Notre Dame Yosemite	Notre Dame Liberty	Liberty Notre Dame	Liberty Yosemite	Average ERR
SIFT [43] (128 bytes)	28.09	36.27	29.15	36.27	28.09	29.15	31.17
Boosted SSC [59] (16 bytes)	72.20	71.59	76.00	70.35	72.95	77.99	73.51
BRISK [39] (64 bytes)	74.88	79.36	73.21	79.36	74.88	73.21	75.81
BRIEF [9] (32 bytes)	54.57	59.15	54.96	59.15	54.57	54.96	56.23
DeepBit [40] (32 bytes)	29.60	34.41	63.68	32.06	26.66	57.61	40.67
LDAHash [64] (16 bytes)	51.58	49.66	52.95	49.66	51.58	52.95	51.40
D-BRIEF [70] (4 bytes)	43.96	53.39	46.22	51.30	43.10	47.29	47.54
BinBoost [68] (8 bytes)	14.54	21.67	18.96	20.49	16.90	22.88	19.24
RFD [17] (50-70 bytes)	11.68	19.40	14.50	19.35	13.23	16.99	15.86
Binary L2-Net [67] (32 bytes)	2.51	6.65	4.04	4.01	1.90	5.61	4.12
Binary DOAP [24] (32 bytes)	1.76	4.17	3.64	2.87	0.96	3.93	2.89
DBD-MQ [15] (32 bytes)	27.20	33.11	57.24	31.10	25.78	57.15	38.59
DBD-MQ + J_1 (32 bytes)	26.94	32.67	56.37	30.78	25.40	56.83	38.17
DCBD-MQ (32 bytes)	**20.13**	**25.77**	**50.99**	**22.92**	**18.95**	**50.36**	**31.52**

The real-valued feature SIFT is provided for reference.

Moreover, the mean quantization loss of KAEs is less than the widely-used K-Means as shown in Table 5, because KAEs quantizes each vector to a subspace rather than a centroid in a nonlinear manner. Besides minimizing the binarization loss in binary code learning, the proposed KAEs can also be used as an unsupervised deep cluster, which present stronger discriminative power than K-Means.

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 GTX 1080 Ti GPU for acceleration. We tested the total computational time of extracting one probe feature and retrieving from 50,000 gallery features in CIFAR-10. It took 0.022s for a 32 bit DCBD-MQ, while HOG [11] and SIFT [43] took 0.030s and 0.054s, respectively. For the storage cost, a 32-bit DCBD-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 DCBD-MQ is more suitable for scalable visual matching and search in practical applications.

5.2 Results on Patch Matching

We evaluated the proposed DBD-MQ and DCBD-MQ on the Brown dataset [8], 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 and the others mismatched. In the experiments, we followed the settings in [69] where all six training and test combinations were used. We fixed the binary length as 256, applying the KAEs with the structure of $[256 \rightarrow 160 \rightarrow 100 \rightarrow 60 \rightarrow 100 \rightarrow 160 \rightarrow 256]$.

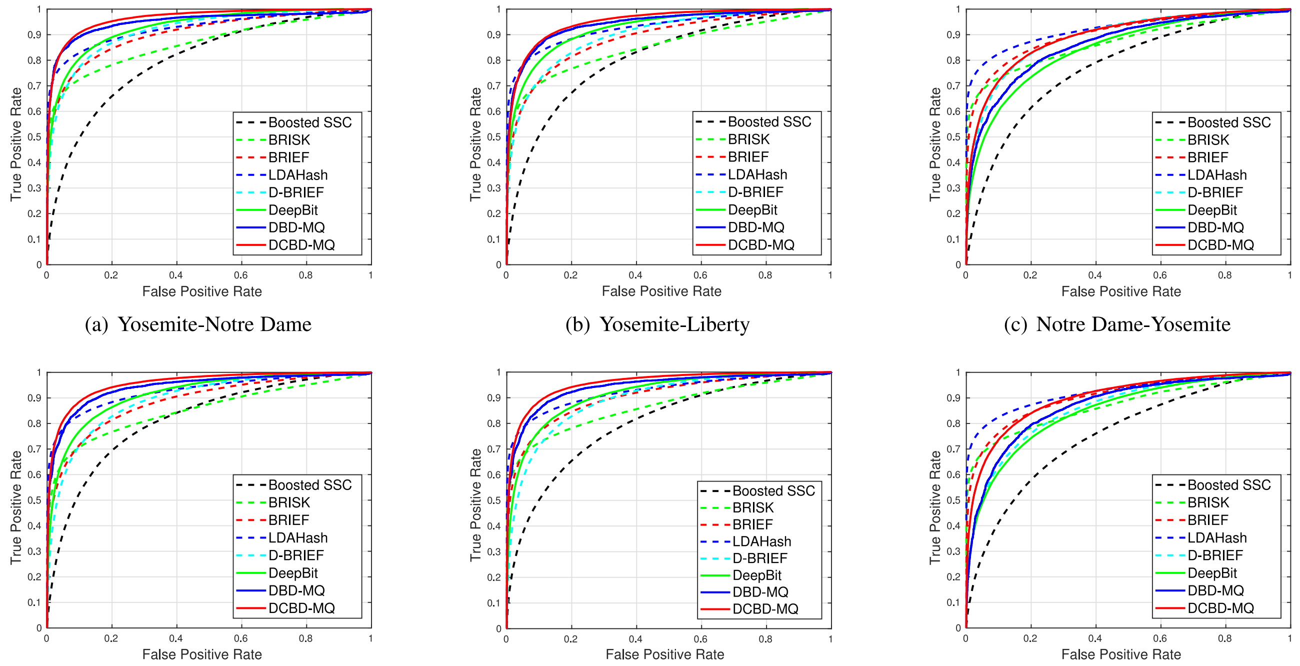
Comparison with the State-of-the-Arts: Table 8 shows the 95 percent error rates (ERR) of DBD-MQ and DCBD-MQ compared with several state-of-the-art descriptors, and Fig. 9 shows the ROC curves. Among the existing unsupervised binary descriptors, DeepBit [40] obtains outstanding results due to its strong discriminative power. However, DeepBit employs the hand-crafted sign function for binarization, while the proposed DBD-MQ learns data-dependent KAEs to minimize the quantization loss. DCBD-MQ further boosts the performance by encouraging elementwise competition for bits to obtain a more optimal allocation, which leads to better performances on all six experiments. Our DBD-MQ and DCBD-MQ also achieve better average 95 percent error rates than some of the supervised approaches without using any label information. As unsupervised methods, DBD-MQ and DCBD-MQ fit for the applications where it is difficult to collect labels, while supervised approaches fail to work in such scenarios. Moreover, DCBD-MQ achieves comparable average error rate than the widely-used real-valued descriptor SIFT. As label information is unused for both DCBD-MQ and SIFT, DCBD-MQ obtains encouraging performance with 4 times less storage costs, which demonstrates the effectiveness of DCBD-MQ. We also observe that DBD-MQ + J_1 obtains lower error rates than DBD-MQ on the Brown dataset, which demonstrates the effectiveness of the modification in J_1.

Evaluation of Different Binarization Strategies: We conducted an additional experiment to evaluate the effectiveness of the proposed multi-quantization based binarization. Table 9 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.

5.3 Results on HPatches

The HPatches dataset [6] is a recent benchmark for local descriptors. The dataset provides three visual analysis tasks for comprehensive evaluation, which includes patch verification, patch matching and patch retrieval. The HPatches dataset contains 116 sequences with 57 under photometric changes and 59 under significant geometric deformations.

We followed the standard evaluation protocol [6] to test the mean average precision (mAP) on the patch verification, patch matching and patch retrieval tasks, respectively. We provided the results of BinBoost [68], SIFT [43] and RSIFT [4]



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

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

Fig. 9. 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.

TABLE 9
95 Percent Error Rates (ERR) of Different Binarization Strategies on the Brown Dataset (%)

Train Test	Yosemite Notre Dame	Yosemite Liberty	Notre Dame Yosemite	Notre Dame Liberty	Liberty Notre Dame	Liberty Yosemite	Average 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

as important references, and compared the proposed DCBD-MQ with the unsupervised binary descriptors including BRIEF [9], ORB [56] and DeepBit [40]. Table 10 shows that the proposed methods outperform other unsupervised binary descriptors due to the data-dependent binarization, and they also achieve comparable performance with the real-valued and the supervised binary descriptors.

5.4 Results on Image Retrieval

The Paris dataset [52] is a standard benchmark for image retrieval, which consists of 6,412 images of Paris landmarks. We need to retrieve all the image of the same place with the 55 queries. The Oxford dataset [51] 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. 55 queries are employed for evaluation. The INRIA Holidays dataset [33] has 1,491 images from 500 groups, with varying rotations and scales. We evaluate on 500 queries in the dataset. We followed the experimental settings in [5] by training on the Landmark dataset [5] and testing on Paris, Oxford and Holidays, respectively. We set the length of the binary codes as 512, applying the KAEs of [$512 \rightarrow 400 \rightarrow 256 \rightarrow 400 \rightarrow 512$].

Table 11 shows the image retrieval results on the three baseline datasets. The SIFT descriptor [43] based methods BoW 200k-D [34] and IFV [34] are listed as baselines. Among the compared methods, only Neural codes [5] is 512-bit binary descriptor, while others are real-valued descriptors. Our DCBD-MQ obtains encouraging result on the Oxford dataset. CKN [50] extracts patch-level descriptors using an unsupervised CNN, while the proposed DCBD-MQ learns energy-saving and evenly-distributive binary descriptors, which presents stronger discriminative power. Moreover, as a binary descriptor learning method, the proposed DCBD-MQ has higher efficiency for storage and computation on image retrieval tasks compared with real-valued descriptors.

TABLE 10
Comparison of Mean Average Precision (mAP) (%) with Baseline Methods under Various Tasks on HPatches

Method	Verification	Matching	Retrieval
BinBoost [68] (32 bytes)	66.67	14.77	22.45
SIFT [43] (128 bytes)	65.12	25.47	31.98
RSIFT [4] (128 bytes)	58.53	27.22	33.56
BRIEF [9] (32 bytes)	58.07	10.50	16.03
ORB [56] (32 bytes)	60.15	**15.32**	18.85
DeepBit [40] (32 bytes)	61.27	13.05	20.61
DCBD-MQ (32 bytes)	**64.78**	14.01	**24.41**

TABLE 11
The Mean Average Precision (mAP) Performance (%) of Different Approaches on the Paris, Oxford and INRIA Holidays Dataset

Methods	Paris	Oxford	Holidays
BoW 200k-D [34]	46.0	36.4	54.0
IFV [34]	-	41.8	62.6
AlexNet [37]	-	33.4	75.3
PhilippNet[19]	-	38.3	74.1
CKN [50]	-	56.5	79.3
Neural codes [5]	-	55.7	78.9
VLAD-CNN [46]	58.3	55.8	83.6
CNN+aug+ss [60]	79.5	**68.0**	84.3
DCBD-MQ	**83.9**	64.1	**84.6**

5.5 Analysis

The above experiments suggest the following key observations:

(1) Our DBD-MQ achieves encouraging performance on the widely-used datasets. Unlike existing binary descriptors which utilize the hand-crafted sign function for binarization, DBD-MQ performs a data-dependent binarization by simultaneously learning the parameters of KAEs and the CNN model to minimize the quantization loss.

(2) KAEs achieves better performance than the commonly-used sign function, because the fine-grained multi-quantization minimizes the quantization loss and enables the holistic descriptors to provide prior knowledge for the elementwise binarization.

(3) Based on DBD-MQ, the proposed DCBD-MQ further learns an optimal allocation of bits in a competitive manner, so that informative dimensions gain more bits for complete description to achieve better results.

(4) The proposed similarity-aware binary encoding strategy ensures relatively small Hamming distances for the elements which are quantized into similar Autoencoders, and improves the discriminative power of the learned binary codes compared with random encoding.

(5) The evaluation of different numbers of Autoencoders shows that, the mean average precision (mAP) increases with K at first, and then descents when K is relatively too large. The reason is that while binary codes preserve more information of a real-valued element with a larger K, it enlarges the searching space and the locality of each Autoencoder, which leads to a lower mAP.

6 Conclusion

In this paper, we have proposed a deep binary descriptor with multi-quantization (DBD-MQ) learning method. Unlike most existing binary representation learning methods which utilize the hand-crafted sign function for binarization, our DBD-MQ simultaneously learns the parameters of CNN and KAEs, replacing the sign function with the data-dependent multi-quantization to minimize the quantization loss. While DBD-MQ evenly allocates bits to the real-valued feature dimensions despite of the diversity of informativeness, we have further proposed a deep competitive binary descriptor with multi-quantization (DCBD-MQ) and a similarity-aware binary encoding strategy to learn an optimal allocation of bits in a competitive manner. In the elementwise contest, the discriminative dimensions grasp more bits from the uninformative ones for complete description. The proposed DBD-MQ and DCBD-MQ outperform most state-of-the-art unsupervised binary descriptors on six widely-used datasets.

Acknowledgments

This work was supported in part by the National Key Research and Development Program of China under Grant 2017YFA0700802, in part by the National Natural Science Foundation of China under Grant U1713214, Grant 61672306, Grant 61572271, and Grant 61527808, in part by the National 1000 Young Talents Plan Program, and in part by the Shenzhen Fundamental Research Fund (Subject Arrangement) under Grant JCYJ20170412170602564.

References

[1] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: Application to face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 12, pp. 2037–2041, Dec. 2006.

[2] A. Alahi, R. Ortiz, and P. Vandergheynst, "FREAK: Fast retina keypoint," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 510–517.

[3] A. Andoni and P. Indyk, "Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions," in *Proc. 47th Annu. IEEE Symp. Foundations Comput. Sci.*, 2006, pp. 459–468.

[4] R. Arandjelovic and A. Zisserman, "Three things everyone should know to improve object retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 2911–2918.

[5] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky, "Neural codes for image retrieval," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 584–599.

[6] V. Balntas, K. Lenc, A. Vedaldi, and K. Mikolajczyk, "HPatches: A benchmark and evaluation of handcrafted and learned local descriptors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 5173–5182.

[7] V. Balntas, L. Tang, and K. Mikolajczyk, "Binary online learned descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 3, pp. 555–567, Mar. 2018.

[8] M. Brown, G. Hua, and S. Winder, "Discriminative learning of local image descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 1, pp. 43–57, Jan. 2011.

[9] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "BRIEF: Binary robust independent elementary features," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 778–792.

[10] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," in *Proc. British Mach. Vis. Conf.*, 2015, pp. 1–12.

[11] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2005, pp. 886–893.

[12] T.-T. Do, A.-D. Doan, and N.-M. Cheung, "Learning to hash with binary deep neural network," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 219–234.

[13] Y. Duan, J. Lu, J. Feng, and J. Zhou, "Learning rotation-invariant local binary descriptor," *IEEE Trans. Image Process.*, vol. 26, no. 8, pp. 3636–3651, Aug. 2017.

[14] Y. Duan, J. Lu, J. Feng, and J. Zhou, "Context-aware local binary feature learning for face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 5, pp. 1139–1153, May 2018.

[15] Y. Duan, J. Lu, Z. Wang, J. Feng, and J. Zhou, "Learning deep binary descriptor with multi-quantization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1183–1192.

[16] V. Erin Liong, J. Lu, G. Wang, P. Moulin, and J. Zhou, "Deep hashing for compact binary codes learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 2475–2483.

[17] B. Fan, Q. Kong, T. Trzcinski, Z. Wang, C. Pan, and P. Fua, "Receptive fields selection for binary feature description," *IEEE Trans. Image Process.*, vol. 23, no. 6, pp. 2583–2595, Jun. 2014.

[18] R. Fergus, P. Perona, and A. Zisserman, "Object class recognition by unsupervised scale-invariant learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2003, pp. 264–271.

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

[20] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1440–1448.

[21] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 580–587.

[22] Y. Gong, S. Lazebnik, A. Gordo, and F. Perronnin, "Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 12, pp. 2916–2929, Dec. 2013.

[23] K. Grauman and T. Darrell, "The pyramid match kernel: Discriminative classification with sets of image features," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2005, pp. 1458–1465.

[24] K. He, Y. Lu, and S. Sclaroff, "Local descriptors optimized for average precision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 596–605.

[25] K. He, F. Wen, and J. Sun, "K-means hashing: An affinity-preserving quantization method for learning binary compact codes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 2938–2945.

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[27] J.-P. Heo, Y. Lee, J. He, S.-F. Chang, and S.-E. Yoon, "Spherical hashing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2012, pp. 2957–2964.

[28] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," *CoRR*, vol. abs/1704.04861, 2017, http://arxiv.org/abs/1704.04861

[29] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4700–4708.

[30] F. N. Iandola, M. W. Moskewicz, K. Ashraf, S. Han, W. J. Dally, and K. Keutzer, "SqueezeNet: Alexnet-level accuracy with 50x fewer parameters and 0.5 mb model size," *CoRR*, vol. abs/1602.07360, 2016, http://arxiv.org/abs/1602.07360

[31] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern Recognit. Lett.*, vol. 31, no. 8, pp. 651–666, 2010.

[32] H. Jain, J. Zepeda, P. Perez, and R. Gribonval, "SUBIC: A supervised, structured binary code for image search," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 833–842.

[33] H. Jegou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large scale image search," *Proc. Eur. Conf. Comput. Vis.*, 2008, pp. 304–317.

[34] H. Jegou, F. Perronnin, M. Douze, J. Sánchez, P. Perez, and C. Schmid, "Aggregating local image descriptors into compact codes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 9, pp. 1704–1716, Sep. 2012.

[35] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 221–231, Jan. 2013.

[36] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Department of Computer Science, Master Thesis, Univ. Toronto, Toronto, Ontario, 2009.

[37] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. 25th Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[38] H. Lai, Y. Pan, Y. Liu, and S. Yan, "Simultaneous feature learning and hash coding with deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3270–3278.

[39] S. Leutenegger, M. Chli, and R. Siegwart, "BRISK: Binary robust invariant scalable keypoints," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 2548–2555.

[40] K. Lin, J. Lu, C.-S. Chen, and J. Zhou, "Learning compact binary descriptors with unsupervised deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1183–1192.

[41] H. Liu, R. Wang, S. Shan, and X. Chen, "Deep supervised hashing for fast image retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2064–2072.

[42] H. Liu, R. Wang, S. Shan, and X. Chen, "Learning multifunctional binary codes for both category and attribute oriented retrieval tasks, "in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 3901–3910.

[43] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[44] J. Lu, V. E. Liong, and J. Zhou, "Simultaneous local binary feature learning and encoding for homogeneous and heterogeneous face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 8, pp. 1979–1993, Aug. 2018.

[45] J. Lu, V. E. Liong, X. Zhou, and J. Zhou, "Learning compact binary face descriptor for face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 10, pp. 2041–2056, Oct. 2015.

[46] J. Y.-H. Ng, F. Yang, and L. S. Davis, "Exploiting local features from deep networks for image retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2015, pp. 53–61.

[47] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 971–987, Jul. 2002.

[48] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1717–1724.

[49] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," in *Proc. British Mach. Vis. Conf.*, 2015, vol. 1, pp. 1–12.

[50] M. Paulin, M. Douze, Z. Harchaoui, J. Mairal, F. Perronin, and C. Schmid, "Local convolutional features with unsupervised training for image retrieval," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 91–99.

[51] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2007, pp. 1–8.

[52] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Lost in quantization: Improving particular object retrieval in large scale image databases," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2008, pp. 1–8.

[53] X. Qi, R. Xiao, C. Li, Y. Qiao, J. Guo, and X. Tang, "Pairwise rotation invariant co-occurrence local binary pattern," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 11, pp. 2199–2213, Nov. 2014.

[54] X. Qian, X. Hua, P. Chen, and L. Ke, "PLBP: An effective local binary patterns texture descriptor with pyramid representation," *Pattern Recognit.*, vol. 44, no. 10, pp. 2502–2515, 2011.

[55] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 91–99.

[56] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 2564–2571.

[57] Y. Rubner, C. Tomasi, and L. J. Guibas, "The earth mover's distance as a metric for image retrieval," *Int. J. Comput. Vis.*, vol. 40, no. 2, pp. 99–121, 2000.

[58] R. Salakhutdinov and G. Hinton, "Semantic hashing," *Int. J. Approximate Reasoning*, vol. 50, no. 7, pp. 969–978, 2009.

[59] G. Shakhnarovich, "Learning Task-Specific Similarity," Department of Electrical Engineering and Computer Science, PhD thesis, Massachusetts Institute Technol., Cambridge, MA, 2005.

[60] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: an astounding baseline for recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2014, pp. 806–813.

[61] Y. Shen, L. Liu, L. Shao, and J. Song, "Deep Binaries: Encoding semantic-rich cues for efficient textual-visual cross retrieval," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 4097–4106.

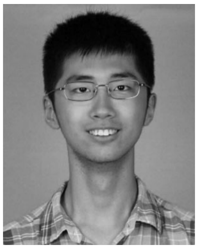
[62] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Representations*, 2015, pp. 1–14.

[63] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 568–576.

[64] C. Strecha, A. Bronstein, M. Bronstein, and P. Fua, "LDAHash: Improved matching with smaller descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 1, pp. 66–78, Jan. 2012.

[65] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation from predicting 10,000 classes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1891–1898.

[66] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1–9.
[67] Y. Tian, B. Fan, and F. Wu, "L2-Net: Deep learning of discriminative patch descriptor in Euclidean space," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 661–669.
[68] T. Trzcinski, M. Christoudias, P. Fua, and V. Lepetit, "Boosting binary keypoint descriptors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 2874–2881.
[69] T. Trzcinski, M. Christoudias, and V. Lepetit, "Learning image descriptors with boosting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 3, pp. 597–610, Mar. 2015.
[70] T. Trzcinski and V. Lepetit, "Efficient discriminative projections for compact binary descriptors," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 228–242.
[71] J. Wang, S. Kumar, and S.-F. Chang, "Semi-supervised hashing for scalable image retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 3424–3431.
[72] J. Wang, T. Zhang, N. Sebe, and H. T. Shen, "A survey on learning to hash," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 769–790, Apr. 2018.
[73] Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing," in *Proc. 21st Int. Conf. Neural Inf. Process. Syst.*, 2009, pp. 1753–1760.
[74] R. Xia, Y. Pan, H. Lai, C. Liu, and S. Yan, "Supervised hashing for image retrieval via image representation learning," in *Proc. 28th AAAI Conf. Artif. Intell.*, 2014, pp. 2156–2162.
[75] S. Zhang, Q. Tian, Q. Huang, W. Gao, and Y. Rui, "USB: Ultrashort binary descriptor for fast visual matching and retrieval," *IEEE Trans. Image Process.*, vol. 23, no. 8, pp. 3671–3683, Aug. 2014.
[76] Z. Zhang, Y. Chen, and V. Saligrama, "Efficient training of very deep neural networks for supervised hashing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 1487–1495.



Yueqi Duan received the BS degree from the Department of Automation, Tsinghua University, China, in 2014. He is currently working toward the PhD degree in the Department of Automation, Tsinghua University, China. His current research interests include unsupervised learning, metric learning, and binary representation learning. He has authored 9 scientific papers in these areas, where 6 papers are published in top journals and conferences including the *IEEE Transactions on Pattern Analysis and Machine Intelligence*, the *IEEE Transactions on Signal Processing* and CVPR. He serves as a regular reviewer member for a number of journals and conferences, e.g., the *IEEE Transactions on Image Processing*, *the IEEE Transactions on Information Forensics and Security*, the *IEEE Transactions on Circuits and Systems for Video Technology*, the *IEEE Access*, Pattern Recognition, the *Journal of Visual Communication and Image Representation*, Neurocomputing, ICME and ICIP. He has obtained the National Scholarship of Tsinghua in 2017.

Jiwen Lu received the BEng degree in mechanical engineering and MEng degree in electrical engineering from the Xi'an University of Technology, Xi'an, China, and the PhD degree in electrical engineering from the Nanyang Technological University, Singapore. He is currently an associate professor with the Department of Automation, Tsinghua University, China. His research interests include computer vision, pattern recognition, machine learning, and deep learning, where he has authored/co-authored more than 200 scientific papers in these areas. He serves an associate editor for several international journals including the *IEEE Transactions on Circuits and Systems for Video Technology*, the *IEEE Transactions on Biometrics, Behavior, and Identity Science, Pattern Recognition*, and the *Journal of Visual Communication and Image Representation*. He was a recipient of the National 1000 Young Talents Plan Program in China. He is a senior member of the IEEE.

Ziwei Wang received the BS degree from the Department of Physics, Tsinghua University, China, in 2018. He is currently working toward the PhD degree in the Department of Automation, Tsinghua University, China. His research interests include computer vision, deep learning, and binary representation.

Jianjiang Feng received the BS and PhD degrees from the School of Telecommunication Engineering, Beijing University of Posts and Telecommunications, China, in 2000 and 2007, respectively. He is an associate professor with the Department of Automation, Tsinghua University, Beijing. From 2008 to 2009, he was a post doctoral researcher with the PRIP lab at Michigan State University. He is an associate editor of Image and Vision Computing. His research interests include fingerprint recognition and computer vision. He is a member of the IEEE.

Jie Zhou received the BS and MS degrees from the Department of Mathematics, Nankai University, Tianjin, China, in 1990 and 1992, respectively, and the PhD degree from the Institute of Pattern Recognition and Artificial Intelligence, Huazhong University of Science and Technology (HUST), Wuhan, China, in 1995. From then to 1997, he served as a postdoctoral fellow with the Department of Automation, Tsinghua University, Beijing, China. Since 2003, he has been a full professor in the Department of Automation, Tsinghua University. His research interests include computer vision, pattern recognition, and image processing. In recent years, he has authored more than 100 papers in peer-reviewed journals and conferences. Among them, more than 30 papers have been published in top journals and conferences such as the *IEEE Transactions on Pattern Analysis and Machine Intelligence*, the *IEEE Transactions on Image Processing*, and CVPR. He is an associate editor for the *IEEE Transactions on Pattern Analysis and Machine Intelligence* and two other journals. He received the National Outstanding Youth Foundation of China Award. He is a senior member of the IEEE.