Learning Compact Binary Descriptors with Unsupervised Deep Neural Networks

Kevin Lin[†], Jiwen Lu[‡], Chu-Song Chen[†], Jie Zhou[‡]
[†]Institute of Information Science, Academia Sinica, Taipei, Taiwan
[‡]Department of Automation, Tsinghua University, Beijing, China
kevinlin311.tw@iis.sinica.edu.tw; lujiwen@tsinghua.edu.cn;
song@iis.sinica.edu.tw; jzhou@tsinghua.edu.cn

Abstract

In this paper, we propose a new unsupervised deep learning approach called DeepBit to learn compact binary descriptor for efficient visual object matching. Unlike most existing binary descriptors which were designed with random projections or linear hash functions, we develop a deep neural network to learn binary descriptors in an unsupervised manner. We enforce three criterions on binary codes which are learned at the top layer of our network: 1) minimal loss quantization, 2) evenly distributed codes and 3) uncorrelated bits. Then, we learn the parameters of the networks with a back-propagation technique. Experimental results on three different visual analysis tasks including image matching, image retrieval, and object recognition clearly demonstrate the effectiveness of the proposed approach.

1. Introduction

Feature descriptor plays an important role in computer vision [28], which has been widely used in numerous computer vision tasks such as object recognition [10, 26, 42], image classification [15, 52] and panorama stitching [5]. A desirable feature descriptor should fulfill two essential properties: (1) high quality representations, and (2) low computational cost. A feature descriptor is desired to capture important and distinctive information in an image [26, 28] and also to be robust to various image transformations [26, 27]. On the other hand, highly efficient descriptor enables machines to run in real-time, which is also important for retrieving image in a large corpus [37], or detecting objects with mobile devices [43, 50].

Over the past decade, high quality descriptors such as the rich features learned from the deep Convolutional Neural Networks (CNN) [20, 32], and the representative SIFT descriptor [26], have been widely explored. These descriptors demonstrate superior discriminability, and bridge the gap between low-level pixels and high-level semantic information [44]. However, they are high-dimensional real-valued descriptors, and usually require high computational cost.

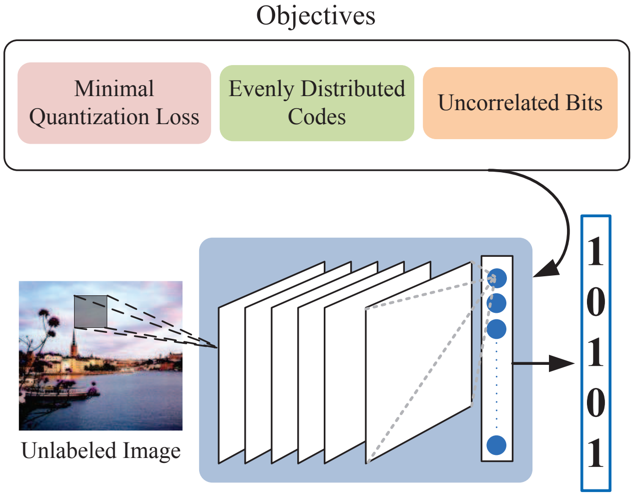


Figure 1: The basic idea of our proposed method. We enforce three criterions on the binary descriptors, and optimize the parameters of the network with back-propagation. Our approach dose not require labeled training data and is more practical to real-world applications in comparison to supervised binary descriptors.

In order to reduce the computational complexity, several lightweight binary descriptors have been recently proposed such as BRIEF [6], ORB [33], BRISK [22], and FREAK [1]. These binary descriptors are highly efficient to storing and matching. Given compact binary descriptors, one can rapidly measure the similarity of the images by computing the Hamming distance between binary descriptors via XOR bitwise operations. Since these early binary descriptors are computed by simple intensity comparisons, they are usually unstable and sensitive to scales, rotations, and noises. Some previous works [9, 40, 48, 53, 54] improved the binary descriptors by encoding the similarity relationship during optimization. However, the success of these methods is mainly attributed to pair-wised similarity labels. In other words, their methods is unfavourable in the case when training data do not have label annotations.

In this work, we raise a question - can we learn binary descriptor from data without labels? Inspiring from the recent advancement of deep learning, we propose an effective deep learning approach, dubbed DeepBit, to learn compact binary descriptors. We enforce three important criterions on the learned binary descriptor, and optimize the parameters of the network with back-propagation. We employ our approach on three different visual analysis tasks including image matching, image retrieval and object recognition. Experimental results clearly demonstrate that our proposed method outperforms state-of-the-arts.

2. Related Work

Binary Descriptors: Earlier works related to binary descriptors can be traced back to BRIEF [6], ORB [33], BRISK [22], and FREAK [1]. These binary descriptors are built upon hand-crafted sampling patterns, and a set of pairwise intensity comparisons. While these descriptors are efficient, their performance is limited because pairwise intensity comparison is sensitive to the scale and geometric transformation. To address these limitations, several supervised approaches [3, 9, 38, 39, 41, 50, 53] have been proposed to learn binary descriptors. D-BRIEF [41] encodes the desired similarity relationships and learns a project matrix to compute discriminative binary features. On the other hand, Local Difference Binary (LDB) [50, 51] applies Adaboost to select optimal sampling pairs. Linear Discriminat Analysis (LDA) is also applied to learn binary descriptors [14, 38]. Recently proposed BinBoost [39, 40] learns a set of projection matrix using the boosting algorithm, and achieves state-of-the-art performance on patches matching. While these approaches have achieved impressive performance, their success is mainly attributed to pair-wise learning with similarity labels, and is unfavorable for the case when transferring the binary descriptor to a new task.

Unsupervised hashing algorithms learn compact binary descriptors whose distance is correlated to the similarity relationship of the original input data [2, 14, 34, 46]. Locality Sensitive Hashing (LSH) [2] applies random projections to map original data into a low-dimensional feature space, and then performs a binarization. Semantic hashing (SH) [34] builds a multi-layers Restricted Boltzmann Machines (RBM) to learn compact binary codes for text and documents. Spectral hashing (SpeH) [46] generates efficient binary codes by spectral graph partitioning. Iterative qauntization (ITQ) [14] uses iterative optimization strategy to find projections with minimal binarization loss. Even if these approaches have been proved effective, the binary codes are still not as accurate as the real-valued equivalents.

Deep Learning: Deep Learning has drawn increasing attention in visual analysis since Krizhevsky *et al.* [20] demonstrated the outstanding performance of the deep CNN on the $1,000$ class image classification. Their success is attributed to training a deep CNN to learn rich mid-level image representations on millions of images. Oquab *et al.* [31] showed that transferring the mid-level image representations to a new domain can be achieved with a few amount of training data. Chatfield *et al.* [7] showed that the fine-tuned domain-specific deep features yield better performance than the non-finetuned ones. Several visual analysis tasks have been greatly improved via pre-trained deep CNN and deep transfer learning, such as object detection [12], image segmentation [25], and image search [23]. Among the recent studies of deep learning and binary codes learning, Xia *et al.* [47] and Lai *et al.* [21] take deep CNN to learn a set of hash functions, but they require pair-wised similarity labels or triplets training data. SSDH [49] constructs hash functions as a latent layer in the deep CNN and achieves state-of-the-art image retrieval performance, but their method belongs to supervised learning. Deep Hashing (DH) [24] builds three layers hierarchical neural networks to learn discriminative projection matrix, but their method does not take the advantage of deep transfer learning, thus makes the binary codes less effective.

In contrast, the proposed DeepBit not only transfers the mid-level image representations pre-trained from ImageNet to the target domain, but also learns compact yet discriminative binary descriptor without label information. We will show that our method achieves better or comparable performance than state-of-the-art descriptors on three public datasets.

3. Approach

Figure 2 shows the learning framework of our proposed method. We introduce an unsupervised deep learning approach, dubbed DeepBit, to learn compact yet discriminative binary descriptors. Unlike previous works [9, 39–41] that optimize the projection matrix with hand-crafted features and pair-wised similarity information, DeepBit learns a set of non-linear projection functions to compute compact binary descriptors. We enforce three important objectives on the binary descriptors, and optimize the parameters of the proposed network with the stochastic gradient descent technique. Note that our method does not require labeled training data, and is more practical than the supervised approaches. In this section, we first give an overview of our approach, and then describe the proposed learning objectives in the following sections.

3.1. Overall Learning Objectives

The proposed DeepBit computes the binary descriptor by applying the projections to the input image and then bi-

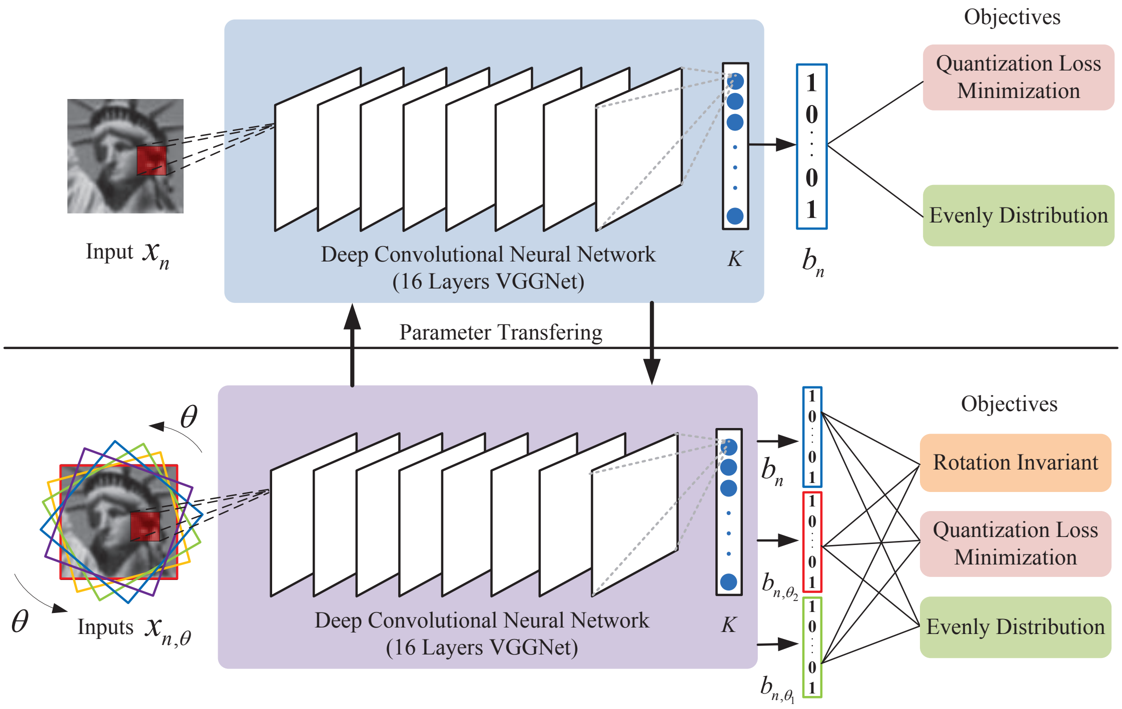


Figure 2: We enforce three objectives on the neurons at the top layer of the network to learn compact yet discriminative binary descriptor. The training procedure includes two alternative stages. The top row shows the first stage; We optimize the parameters of the network by minimizing the quantization error and enforcing binary codes to be evenly distributed. The bottom row shows the second stage; We augment the training data with different rotations, and update the parameters of the network by minimizing the distance between binary descriptors that describe the reference image and the rotated one. The alternative stages will be repeated until the stopping criterion is satisfied.

narizes the results:

$$b = 0.5 \times (\operatorname{sign}(\mathcal{F}(x;\mathcal{W})) + 1), \qquad (1)$$

where x represents the input image, and b is the resulting binary descriptor in the vector form. $\operatorname{sign}(k) = 1$ if $k > 0$ and -1 otherwise. $\mathcal{F}(x;\mathcal{W})$ is a composition of number of non-linear projection functions which can be written as:

$$\mathcal{F}(x;\mathcal{W}) = f_k(\cdots f_2(f_1(x;w_1);w_2)\cdots;w_k), \qquad (2)$$

where f_i takes the data x_i and parameter w_i as inputs, and produces the projection result x_{i+1}.

The proposed approach aims to learn a set of non-linear projection parameters $\mathcal{W} = (w_1, w_2, ..., w_k)$ that quantizes the input image x into a compact binary vector b while preserving the information from the input. In order to learn compact yet discriminative binary descriptor, we enforce three important criterions to learn $\mathcal{W}$. First, the learned compact binary descriptor should preserve the local data structure of the activations of the last layer. The quantization loss should be as less as possible after projection. Second, we encourage the binary descriptor to be evenly distributed, so that the binary string will convey more discriminative messages. The third is to make the descriptor invariant to rotations and noises, and thus the binary descriptor will tend to capture more uncorrelated information from the input image. To achieve these objectives, we formulate the following optimization problem to learn a set of non-linear projection parameters $\mathcal{W}$ using the proposed deep neural networks:

$$\begin{aligned}\min_{\mathcal{W}} L(\mathcal{W}) &= \alpha L_1(\mathcal{W}) + \beta L_2(\mathcal{W}) + \gamma L_3(\mathcal{W}) \\ &= \alpha \sum_{n=1}^{N} ||(b_n - 0.5) - \mathcal{F}(x_n;\mathcal{W})||^2 \\ &+ \beta \sum_{m=1}^{M} ||(\mu_m - 0.5)||^2 \\ &+ \gamma \sum_{n=1}^{N} \sum_{\theta=-R}^{R} \mathcal{C}(\theta)||b_{n,\theta} - b_n||^2,\end{aligned} \qquad (3)$$

where N is the number of training data for each mini-batch, M is the bit length of the binary codes, and R represents the image rotation angle. $b_{n,\theta}$ is the binary descriptor projected from image x_n with rotation angle θ, and $\mathcal{C}(\theta)$ is the cost function which penalizes the training data according to its rotation degree. Moreover, α, β, and γ are three parameters to balance different objectives.

To give a better understanding of the proposed objectives, we describe the physical meaning of (3) as below. First, L_1 minimizes the quantization loss between the binary descriptor and the original input image. Then, L_2 encourages the binary descriptor to be evenly distributed to maximize the information capacity of the binary descriptor. Finally, L_3 tolerates the rotation transformations by minimizing the Hamming distance between the descriptors that describe the reference image and the rotated ones. We elaborate the details of each proposed objective as follows.

3.2. Learning Discriminative Binary Descriptors

The proposed DeepBit seeks to learn the projections that maps the input image into a binary string while preserving the discriminative information of the original input. The soul idea to keep the binary descriptors informative is to minimize the quantization loss by rewriting (1) as follows:

$$(b - 0.5) = \mathcal{F}(x; \mathcal{W}), \tag{4}$$

the smaller the quantization loss is, the better the binary descriptor will preserve the original data information. Different from the previous work [13] that addresses this problem by iteratively updating $\mathcal{W}$ and b with two alternating steps, we formulate this optimization problem as the neural networks training objective. Since then, the goal of the proposed network becomes learning the $\mathcal{W}$ that minimizes the quantization loss between the binary descriptor and the original input image. To this end, we optimize the parameters $\mathcal{W}$ of the proposed network through back-propagation and stochastic gradient descent (SGD) using the following loss function:

$$\min_{\mathcal{W}} L(\mathcal{W}) = \sum_{n=1}^{N} ||(b_n - 0.5) - \mathcal{F}(x_n; \mathcal{W})||^2. \tag{5}$$

3.3. Learning Efficient Binary Descriptors

To increase the information capacity of the binary descriptors, we maximize the usage of each bin in the binary string. Considering the variance for each bin, the higher the entropy is, the more information the binary codes express. Accordingly, we enhance the binary descriptor by making each bit has 50% probability of being one or zero. In other words, there is no preference for each bit to be one or zero, and the resulting binary string will convey the information as much as possible. To achieve this goal, we keep the binary descriptors to be evenly distributed by formulating the following objective, and minimizing the loss computed by the forward pass of the network:

$$\min_{\mathcal{W}} L(\mathcal{W}) = \sum_{m=1}^{M} ||(\mu_m - 0.5)||^2, \tag{6}$$

where M represents the bit length of the binary string. For each bin we compute the average response μ_m using:

$$\forall_{m \in 1,\ldots,M} \;\; \mu_m = \frac{1}{N} \sum_{n=1}^{N} b_n(m), \tag{7}$$

where N is the number of training data, and function $b(m)$ produces the binary value at m-th bin.

3.4. Learning Rotation Invariant Binary Descriptors

Since rotation invariant is essential for a local descriptor, we hope to enhance this property during optimization. We address this issue by minimizing the difference between binary descriptors that describe the reference image and the rotated one. Considering the estimation error between images, the estimation error may become larger when increasing the rotation degree. Hence, we mitigate the estimation error by penalizing the training loss of the network according to the rotation degree. We formulate the proposed objective as a cost-sensitive optimization problem as follow:

$$\min_{\mathcal{W}} L(\mathcal{W}) = \sum_{n=1}^{N} \sum_{\theta=-R}^{R} \mathcal{C}(\theta) ||b_{n,\theta} - b_n||^2, \tag{8}$$

where $\theta \in (-R, R)$ is the rotation angle. $b_{n,\theta}$ denotes the descriptor mapping from input x_n with rotation θ. $\mathcal{C}(\theta)$ provides the cost information to reflect the relationship of binary descriptors between different rotation transformations. In this paper, we mitigate the estimation error by setting:

$$\mathcal{C}(\theta) = \exp\left(-\frac{(\theta - \mu)^2}{2\sigma^2}\right), \tag{9}$$

where $\mathcal{C}(\theta)$ is the Gaussian distribution, and $\mu = 0$, $\sigma = 1$ in our experiments.

We implement our approach using the open source Caffe [18], and **Algorithm 1** summarizes the detail procedure of the proposed DeepBit. The proposed approach includes two main components. The first is network initialization. Second is the optimization step. We initialize our network with the pre-trained weights from the 16 layers VGGNet [36], which is trained on the ImageNet large scale dataset. Then, we replace the classification layer of the VGGNet with a new fully connected layer, and enforce the neurons in this layer to learn binary descriptor. To this end, we use stochastic gradient descent (SGD) method and back-propagation to train our network, and optimize $\mathcal{W}$ using the proposed objectives (see (3)). Other settings are listed below. $\alpha = 1.0$, $\beta = 1.0$, $\gamma = 0.01$. We rotate the image by 10, 5, 0, -5, -10 degrees, respectively. Mini-batch size is 32, and the bit-length of our binary descriptor is 256. Images are normalized to 256×256 and then center-cropped to 224×224 as the network input.

Algorithm 1: DeepBit

```
Input: Training set $X = [x_1, x_2, ..., x_n]$
Output: A set of non-linear projection parameters $\mathcal{W}$
Step 1 (Initialization):
Initialize $\mathcal{W}$ with pre-trained weights from ImageNet;
Step 2 (Optimization):
while $iter < max\_iter$ do
    Fix $\mathcal{W}$ update $b_n$ using (1);
    while $iter1 < max\_iter1$ do
        Fix $b_n$ update $\mathcal{W}$ by minimizing the sum of (5) and (6);
    Fix $\mathcal{W}$ update $b_n$ using (1);
    while $iter2 < max\_iter2$ do
        Fix $b_n$ update $\mathcal{W}$ using (8);
return $\mathcal{W}$;
```

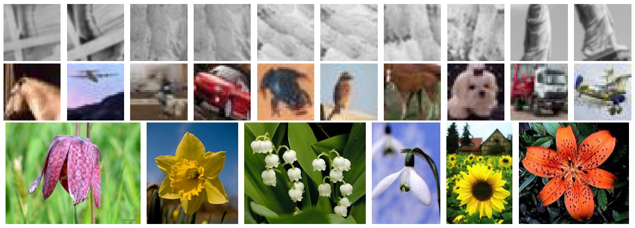


Figure 3: Sample images from the Brown dataset, CIFAR10 dataset, and Oxford flower dataset, respectively. We test our approach on a wide range of image types, including gray-scale local patches, color category images, and flowers in the wild.

4. Experimental Results

We conduct experiments on three challenging datasets, the Brown gray-scale patches [4], the CIFAR-10 color images [19], and the Oxford 17 category flowers [29]. We provide extensive evaluations of the proposed binary descriptor, and demonstrate its performance on various tasks, including image matching, image retrieval, and image classification. We start with introducing the datasets and then present our experimental results as well as the comparative evaluations with other state-of-the-arts.

4.1. Datasets

- **Brown Dataset** [4] consists of three datasets, namely Liberty, Notredame, Yosemite dataset. Each of them includes more than $400,000$ gray-scale patches, resulting in a total of $1,200,000$ patches. Each dataset is split into training and test sets, with $20,000$ training pairs ($10,000$ matched and $10,000$ non-matched pairs) and $10,000$ test pairs ($5,000$ matched, and $5,000$ non-matched pairs), respectively.

- **CIFAR-10 Dataset** [19] contains 10 object categories and each class consists of $6,000$ images, resulting in a total of $60,000$ images. The dataset is split into training and test sets, with $50,000$ and $10,000$ images respectively.

- **The Oxford 17 Category Flower Dataset** [29] contains 17 categories and each class consists of 80 images, resulting in a total of $1,360$ images. The dataset is split into the training (40 images per class), validation (20 images per class), and test (20 images per class) sets.

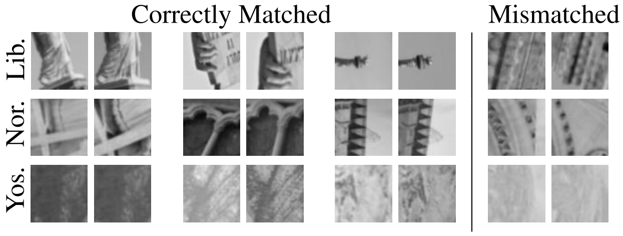


Figure 5: Correctly matched patches and mismatched ones from the Brown dataset. Top row shows the patches from Liberty classified as matched pairs; the first three are correctly classified, but the fourth is mismatched, which describes different architectures. Middle row shows the image pairs from Notredame classified as the matched pairs; the fourth is mismatched although both of them share similar pattern. Bottom row shows the patches from Yosemite classified as matched pairs; the last one is mismatched, which are visually similar but belong to different locations.

4.2. Results on Image Matching

To evaluate the performance of local descriptors, we compare the proposed DeepBit with several state-of-the-art binary descriptors, including unsupervised (BRIEF [6], ORB [33], BRISK [22], and Boosted SSC [35]), and supervised methods (D-BRIEF [41], LDAHash [38]).

Following the settings in [40], Figure 4 shows the ROC curves for DeepBit and the compared methods, and Table 1 summarizes the 95 percent error rates for the Brown dataset. As can be seen, the overall performance of the proposed method achieves 40.67% error rate when recall rate is 95%, which outperforms BRIEF, ORB, BRISK, Boosted SSC with $15.56\% (= 56.23\% - 40.67\%)$, $15.56\% (= 56.23\% - 40.67\%)$, $35.14\% (= 75.81\% - 40.67\%)$, $32.84\% (= 73.51\% - 40.67\%)$ lower error rate over the different training and testing configurations of the Brown dataset, respectively. It is important to point out that unlike several previous works [3, 9, 38–41, 50, 53] that em-

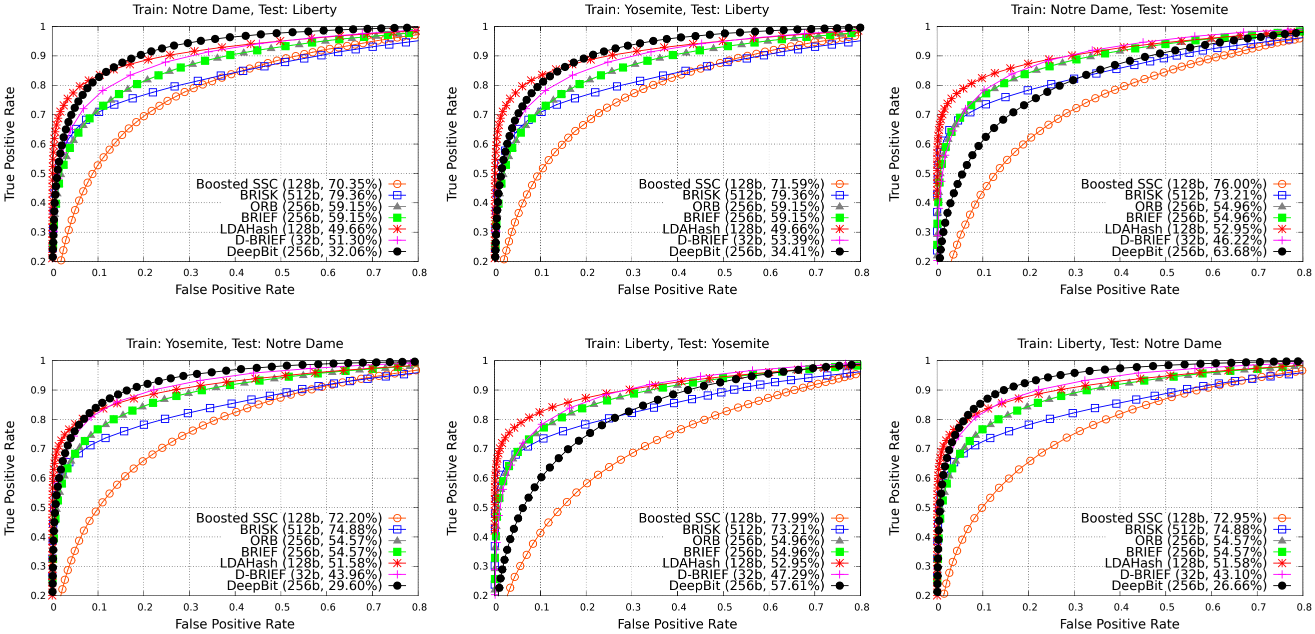


Figure 4: ROC curves of the proposed DeepBit descriptors and the compared binary descriptors, across all the splits of training and testing configurations on the Brown datasets. In parentheses: the bit length of the binary descriptor (b), and the 95% error rates.

Table 1: Comparison of the proposed binary descriptor to the state-of-the-art binary descriptors, in terms of 95% error rates (ERR) across all the splits of training and testing configurations. For reference, we also provide the results of real-valued descriptor SIFT [26]. The proposed method achieves better performance than the unsupervised binary descriptors in most cases, while remaining competitive to supervised approaches (D-BRIEF and LDAHash).

		Real-valued	Binary						
Train	Test	SIFT [26] 128 bytes	Boosted SSC [35] 16 bytes	BRISK [22] 64 bytes	ORB [33] 32 bytes	BRIEF [6] 32 bytes	LDAHash [38] 16 bytes	D-BRIEF [41] 4 bytes	DeepBit 32 bytes
Yosemite	Notredame	28.09	72.20	74.88	54.57	54.57	51.58	43.96	**29.60**
Yosemite	Liberty	36.27	71.59	79.36	59.15	59.15	49.66	53.39	**34.41**
Notredame	Yosemite	29.15	76.00	73.21	54.96	54.96	52.95	**46.22**	63.68
Notredame	Liberty	36.27	70.35	79.36	59.15	59.15	49.66	51.30	**32.06**
Liberty	Notredame	28.09	72.95	74.88	54.57	54.57	51.58	43.10	**26.66**
Liberty	Yosemite	29.15	77.99	73.21	54.96	54.96	52.95	**47.29**	57.61
Average 95% ERR		31.17	73.51	75.81	56.23	56.23	51.40	47.54	**40.67**

ploy similarity information (matched and non-matched labels) to optimize the projection matrix, our learning process does not require the training labels and still performs more favorably against the supervised ones such as D-BRIEF and LDAHash. We further visualize the image matching results on the Brown dataset in Figure 5. As can be seen, the proposed method successfully matches pairs of patches when they are visually similar, as shown in the first three columns of Figure 5. Our method could also mismatch some patches as shown in the fourth column of Figure 5. It is worth noting that the mismatched patches are still visually similar although they are from different scenes or locations. More specifically, the patches from Liberty and Notredame describe the local structure of the statue and architecture, where the visual similarity between different patches is usually weak. Our approach achieves more favorable performance in these two datasets. However, the patches from Yosemite depict the surface of a mountain. Different local patches (such as snow and forest) could generate visually similar patterns, making them difficult to be distinguished.

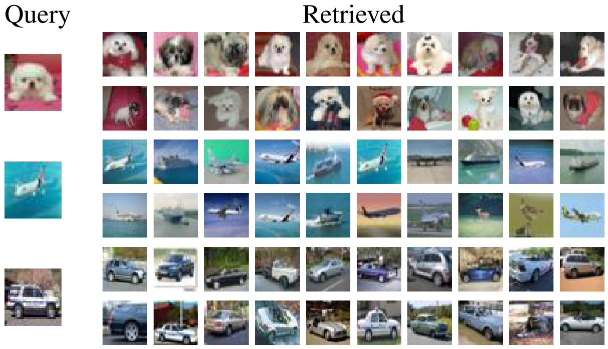


Figure 7: Top 20 retrieved images from CIFAR10 dataset by DeepBit with 32 bit length.

This could be the reason why our approach, which tends to match patterns that are visually similar, performs less favorable than some methods for the Yosemite dataset.

4.3. Results on Image Retrieval

To evaluate the discriminability of the proposed binary descriptor, we further test our method on the task of image retrieval. We compare DeepBit with several unsupervised hashing methods, including LSH [2], ITQ [14], PCAH [45], Semantic Hashing (SH) [34], Spectral hashing (SpeH) [46]), Spherical hashing (SphH) [17], KMH [16], and Deep Hashing (DH) [24] on the CIFAR-10 dataset. Among these eight unsupervised approaches, Deep Hashing (DH), like our approach, takes advantage of deep neural networks for learning compact binary codes.

Following the settings in [24], Table 2 shows the CIFAR-10 retrieval results based on the mean Average Precision (mAP) of the top $1,000$ returned images with respect to different bit lengths. DeepBit improves previous best retrieval performance by 3.26%, 8.24%, and 10.77% mAP with respect to 16, 32, and 64 hash bits, respectively. According to the results, we found that the longer the hash bits, the better performance DeepBit achieves. Moreover, Figure 6 shows the Precision/Recall curves of different unsupervised hashing methods with 16, 32, 64 hash bits, respectively. As can be seen, DeepBit constantly outperforms previous unsupervised methods. This indicates the proposed method is effective to learn binary descriptors. It is worth to note that DH [24] takes three layers hierarchical neural networks to learn binary hash codes; however, DH dose not take advantage of the deep transfer learning during training. In contrast, the proposed DeepBit not only transfers the mid-level image representations pre-trained from ImageNet to the target domain, but also learns binary descriptor with desirable criterions. The experiments reveal that deep transfer learning with the proposed objectives can improve the unsupervised hashing performance.

Table 2: Performance comparison (mAP, %) of different unsupervised hashing algorithms on the CIFAR-10 dataset. This table shows the mean Average Precision (mAP) of top $1,000$ returned images with respect to different number of hash bits.

Method	16 bit	32 bit	64 bit
KMH [16]	13.59	13.93	14.46
SphH [17]	13.98	14.58	15.38
SpeH [46]	12.55	12.42	12.56
SH [34]	12.95	14.09	13.89
PCAH [45]	12.91	12.60	12.10
LSH [2]	12.55	13.76	15.07
PCA-ITQ [13]	15.67	16.20	16.64
DH [24]	16.17	16.62	16.96
DeepBit	**19.43**	**24.86**	**27.73**

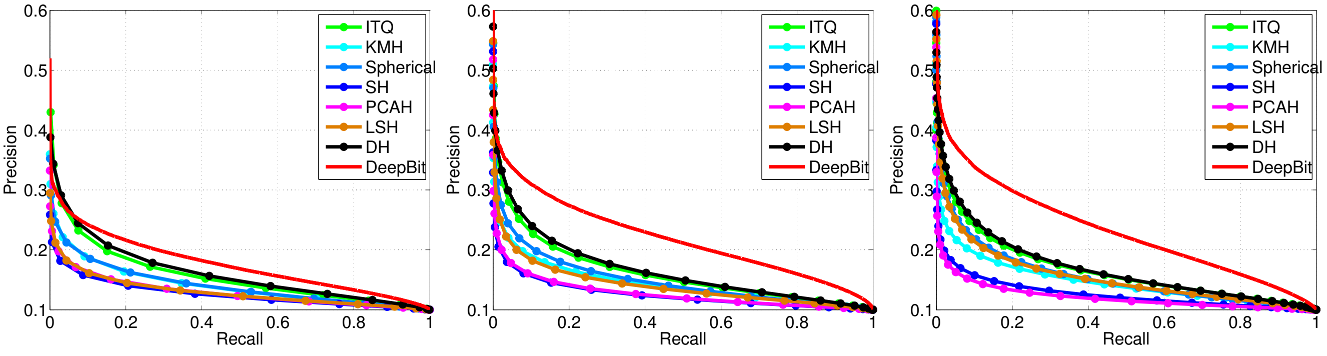
Table 3: The categorization accuracy (mean±std%) for different features on the Oxford 17 Category Flower Dataset [11].

Descriptors	Accuracy	Training Time (sec)
Colour [29]	$60.9 \pm 2.1\%$	3
Shape [29]	$70.2 \pm 1.3\%$	4
Texture [29]	$63.7 \pm 2.7\%$	3
HOG [8]	$58.5 \pm 4.5\%$	4
HSV [30]	$61.3 \pm 0.7\%$	3
SIFT-Boundary [30]	$59.4 \pm 3.3\%$	5
SIFT-Internal [30]	$70.6 \pm 1.6\%$	4
DeepBit	$\mathbf{75.1 \pm 2.5\%}$	**0.07**

4.4. Results on Object Recognition

Unlike previous binary descriptors that require matched/non-matched labels during training, the proposed DeepBit learns compact binary descriptors in an unsupervised manner; thus, DeepBit is practical and flexible for various applications. In this section, we extend the evaluation to object recognition and show that the proposed binary descriptor performs more favorably against several real-valued descriptors such as HOG [8], and SIFT [26].

Flower classification is a classic visual analysis task, and it is challenging due to the variation of shapes, color distributions, and pose deformations. Besides, the computation cost becomes demanding while one wants to recognize the flowers in the wild using mobile devices. We test our binary descriptors on the flower recognition. Following the setting in [29], we train the multi-class SVM classifier with the proposed binary descriptor. Table 3 compares the classification accuracy of the 17 categories flowers using different descriptors proposed in [29, 30], in-



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

Figure 6: Precision/Recall curves of different unsupervised hashing methods on the CIFAR-10 dataset with respect to 16, 32 and 64 bits, respectively.

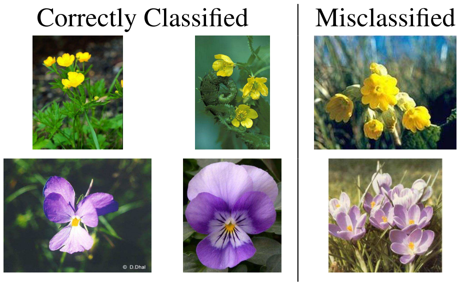


Figure 8: Correctly classified test images and misclassified ones. The top row shows images classified as Cowslip; the first two are correctly classified but the correct category of the third is Buttercup. The bottom row shows images classified as Pansy; the third is misclassified, which belongs to Crocus.

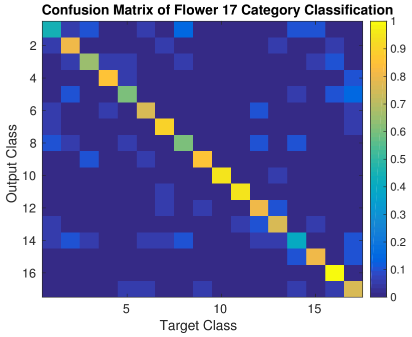


Figure 9: Confusion matrix of Oxford 17 flower classification using the proposed DeepBit. Classification results indicate that the proposed learning method is effective to learn compact but informative binary descriptor.

cluding low-level (Colour, Shape, Texture), and high level (SIFT, and HOG) features. The proposed binary descriptor improves previous best recognition accuracy by around 4.5% (75.1% $vs.$ 70.6%). In addition, DeepBit greatly reduces the computational complexity during SVM classifier training. Our training process is 71.42x faster than the one trained with SIFT because the dimension of DeepBit is lower than that of SIFT. Figure 8 and Figure 9 shows some visualization results. DeepBit demonstrates its efficiency and efficacy, and performs more favourably against various existing descriptors including Colour [29], Shape [29], Texture [29], HOG [8], HSV [30], and SIFT [26, 30]. This indicates the proposed method is effective to learn discriminative and compact binary codes.

5. Conclusions

In this paper, we have presented an unsupervised deep learning framework to learn compact binary descriptor. We employ three criterions to learn the binary codes and estimate the parameters of the deep neural network to obtain binary descriptor. Our approach does not require labeled data during learning, and is more practical to real-world applications compared to supervised binary descriptors. Experiments on three benchmark databases include gray-scale local patches, color images, and flowers in the wild demonstrate that our method achieves better performance than the state-of-the-art feature descriptors in most cases.

Acknowledgement: This work is supported in part by the Ministry of Science and Technology of Taiwan under Contract MOST 104-2221-E-001-023-MY2. This work is also supported in part by the National Natural Science Foundation of China under Grants 61225008, 61572271, 61527808, 61373074 and 61373090, 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] A. Alahi, R. Ortiz, and P. Vandergheynst. Freak: Fast retina keypoint. In *Proc. CVPR*, 2012. 1, 2

[2] A. Andoni and P. Indyk. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In *Proc. FOCS*, 2006. 2, 7

[3] V. Balntas, L. Tang, and K. Mikolajczyk. Bold-binary online learned descriptor for efficient image matching. In *Proc. CVPR*, 2015. 2, 5

[4] M. Brown, G. Hua, and S. Winder. Discriminative learning of local image descriptors. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(1):43–57, 2011. 5

[5] M. Brown and D. G. Lowe. Automatic panoramic image stitching using invariant features. *Int'l J. Computer Vision*, 74(1):59–73, 2007. 1

[6] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. Brief: Binary robust independent elementary features. In *Proc. ECCV*, 2010. 1, 2, 5, 6

[7] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *Proc. BMVC*, 2014. 2

[8] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. CVPR*, 2005. 7, 8

[9] B. Fan, Q. Kong, T. Trzcinski, Z. Wang, C. Pan, and P. Fua. Receptive fields selection for binary feature description. *IEEE Trans. Image Proc.*, 23(6):2583–2595, 2014. 1, 2, 5

[10] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. In *Proc. CVPR*, 2003. 1

[11] P. Gehler and S. Nowozin. On feature combination for multiclass object classification. In *Proc. ICCV*, 2009. 7

[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proc. CVPR*, 2014. 2

[13] Y. Gong and S. Lazebnik. Iterative quantization: A procrustean approach to learning binary codes. In *Proc. CVPR*, 2011. 4, 7

[14] 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.*, 35(12):2916–2929, 2013. 2, 7

[15] K. Grauman and T. Darrell. The pyramid match kernel: Discriminative classification with sets of image features. In *Proc. ICCV*, 2005. 1

[16] K. He, F. Wen, and J. Sun. K-means hashing: An affinity-preserving quantization method for learning binary compact codes. In *Proc. CVPR*, 2013. 7

[17] J.-P. Heo, Y. Lee, J. He, S.-F. Chang, and S.-E. Yoon. Spherical hashing. In *Proc. CVPR*, 2012. 7

[18] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *Proc. ACM MM*, 2014. 4

[19] A. Krizhevsky. Learning multiple layers of features from tiny images, 2009. 5

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *Proc. NIPS*, 2012. 1, 2

[21] H. Lai, Y. Pan, Y. Liu, and S. Yan. Simultaneous feature learning and hash coding with deep neural networks. In *Proc. CVPR*, 2015. 2

[22] S. Leutenegger, M. Chli, and R. Y. Siegwart. Brisk: Binary robust invariant scalable keypoints. In *Proc. ICCV*, 2011. 1, 2, 5, 6

[23] K. Lin, H.-F. Yang, J.-H. Hsiao, and C.-S. Chen. Deep learning of binary hash codes for fast image retrieval. In *Proc. CVPR Workshops*, 2015. 2

[24] V. E. Liong, J. Lu, G. Wang, P. Moulin, and J. Zhou. Deep hashing for compact binary codes learning. In *Proc. CVPR*, 2015. 2, 7

[25] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proc. CVPR*, 2014. 2

[26] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *Int'l J. Computer Vision*, 60(2):91–110, 2004. 1, 6, 7, 8

[27] K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. In *Proc. ICCV*, 2001. 1

[28] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(10):1615–1630, 2005. 1

[29] M.-E. Nilsback and A. Zisserman. A visual vocabulary for flower classification. In *Proc. CVPR*, 2006. 5, 7, 8

[30] M.-E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *Proc. ICVGIP*, 2008. 7, 8

[31] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *Proc. CVPR*, 2014. 2

[32] A. S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. CNN features off-the-shelf: An astounding baseline for recognition. In *Proc. CVPR Workshops*, 2014. 1

[33] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb: an efficient alternative to sift or surf. In *Proc. ICCV*, 2011. 1, 2, 5, 6

[34] R. Salakhutdinov and G. E. Hinton. Semantic hashing. *Int. J. Approx. Reasoning*, 50(7):969–978, 2009. 2, 7

[35] G. Shakhnarovich. *Learning task-specific similarity*. PhD thesis, Massachusetts Institute of Technology, 2005. 5, 6

[36] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proc. ICLR*, 2015. 4

[37] A. W. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain. Content-based image retrieval at the end of the early years. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(12):1349–1380, 2000. 1

[38] C. Strecha, A. M. Bronstein, M. M. Bronstein, and P. Fua. Ldahash: Improved matching with smaller descriptors. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(1):66–78, 2012. 2, 5, 6

[39] T. Trzcinski, M. Christoudias, P. Fua, and V. Lepetit. Boosting binary keypoint descriptors. In *Proc. CVPR*, 2013. 2

[40] T. Trzcinski, M. Christoudias, and V. Lepetit. Learning image descriptors with boosting. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(3):597–610, 2015. 1, 2, 5

[41] T. Trzcinski and V. Lepetit. Efficient discriminative projections for compact binary descriptors. In *Proc. ECCV*, 2012. 2, 5, 6

[42] M. Vidal-Naquet and S. Ullman. Object recognition with informative features and linear classification. In *Proc. ICCV*, 2003. 1

[43] P. Viola and M. Jones. Robust real-time object detection. *Int'l J. Computer Vision*, 4:51–52, 2001. 1

[44] J. Wan, D. Wang, S. C. H. Hoi, P. Wu, J. Zhu, Y. Zhang, and J. Li. Deep learning for content-based image retrieval: A comprehensive study. In *Proc. ACM MM*, 2014. 1

[45] J. Wang, S. Kumar, and S. Chang. Semi-supervised hashing for scalable image retrieval. In *Proc. CVPR*, 2010. 7

[46] Y. Weiss, A. Torralba, and R. Fergus. Spectral hashing. In *Proc. NIPS*, 2008. 2, 7

[47] R. Xia, Y. Pan, H. Lai, C. Liu, and S. Yan. Supervised hashing for image retreieval via image representation learning. In *Proc. AAAI*, 2014. 2

[48] X. Xu, L. Tian, J. Feng, and J. Zhou. Osri: A rotationally invariant binary descriptor. *IEEE Trans. Image Proc.*, 23(7):2983–2995, 2014. 1

[49] H.-F. Yang, K. Lin, and C.-S. Chen. Supervised learning of semantics-preserving hashing via deep neural networks for large-scale image search. *arXiv preprint arXiv:1507.00101*, 2015. 2

[50] X. Yang and K.-T. Cheng. Ldb: An ultra-fast feature for scalable augmented reality on mobile devices. In *Proc. ISMAR*, 2012. 1, 2, 5

[51] X. Yang and K.-T. Cheng. Local difference binary for ultrafast and distinctive feature description. *IEEE Trans. Pattern Anal. Mach. Intell.*, 36(1):188–194, 2014. 2

[52] J. Zhang, M. Marszałek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. *Int'l J. Computer Vision*, 73(2):213–238, 2007. 1

[53] S. Zhang, Q. Tian, Q. Huang, W. Gao, and Y. Rui. Usb: ultrashort binary descriptor for fast visual matching and retrieval. *IEEE Trans. Image Proc.*, 23(8):3671–3683, 2014. 1, 2, 5

[54] L. Zheng, S. Wang, and Q. Tian. Coupled binary embedding for large-scale image retrieval. *IEEE Trans. Image Proc.*, 23(8):3368–3380, 2014. 1