Target Code Guided Binary Hashing Representations with Deep Neural Network

Yunbo Wang, Dong Cao, Zhenan Sun
Center for Research on Intelligent Perception and Computing
Institute of Automation, Chinese Academy of Sciences
University of Chinese Academy of Sciences
yunbo.wang@cripac.ia.ac.ac,dong.cao@nlpr.ia.ac.cn,znsun@nlpr.ia.ac.cn

Abstract

Most existing hashing approaches usually impose some artificial constraints (e.g. uncorrelated and balanced) on hash functions to learn high-quality binary codes, and apply an optimization process which is typically compatible with these hash functions. However, these tight constraints potentially restrict the flexibility of hash functions to fit training data, and result in complicated optimization problem. In this paper, we propose a learning-based hashing method called "deep supervised hashing with target code"(DSHT) to distill the desirable properties in the target coding into hash functions to generate high-quality binary codes. Benefiting from recent advances in deep learning, our framework constructs hash functions as a latent layer in a deep neural network in which binary hashing representations are learned with the guide of target code and semantic information. Experiments on two two large-scale image dataset (MNIST, CIFAR-10) demonstrate that the proposed framework is available, flexible and show comparable performance against other state-of-the-art hashing methods.

1. Introduction

With the rapidly growing amount of image data on the web or video surveillance, nearest neighbor search in large scale database via hashing approach [2, 3, 18] has attracted much more attention. Generally, hash coding can both improve the efficiency in storage and search speed of images, especially when dealing with millions or billions images. Given a query image in a recognition or retrieval task, the similarity between the query image and dataset can be rapidly computed by XOR operations in the Hamming distance space.

In the past few years, many learning-based hashing approaches have been introduced e.g.[11, 14, 21]. According to whether supervised information is available, the learning-based hashing approach can be categorized into unsupervised and supervised approaches. [4, 8]adopt unsupervised hash coding which only uses unlabeled data to learn hash functions. Comparatively, [14, 18] perform the hash learning process with the help of supervised information to generate more discriminative binary codes. In addtion, most of the existing hashing methods use some hand-crafted visual descriptors(e.g., LBP[16], GIST[17]) to represent each input image. However, such hand-crafted features can not guarantee the compatibility with the followed coding process. Very recently, benefiting from the great advances of deep Learning[5, 7, 19, 20, 25] in various visual tasks, deep learning based hashing methods[11, 22, 26]have been proposed to simultaneously learn the image representations and hash coding, which have shown superior performance over the traditional hashing methods with hand-crafted visual descriptors. In this paper, we focus on deep learning based hashing for image coding. Although the above learning-based hashing methods have made promising results, two critical problems are seldom mentioned.

First, recently theoretical and empirical evidences suggest that the balanced and uncorrelated binary constraint (which favor a large information entropy) can yield high-quality codes to well facilitate retrieval tasks. To do this, one common fashion is to reformulate hash coding as a constrained optimization problem. For example, [13, 15, 21] enforce the learned binary codes have 50% chance of being one or zero to maximize the information from each code, and the orthogonal item constrains the independence of different codes to minimize the redundancy. However, these discrete constraints are artificially imposed on the large training samples, making the optimization process difficult. Even worse, these constraints may result in adverse effects when excessively pursuing the desired structure.

Second, in real-world settings, any two images from the same category have more or less disparities due to various pose, lighting, background and rotation. However, most of the exiting hashing methods give the same supervised information for images belonging to the same category, while ignore the disparity of the intra-class.

To overcome these problems above, we propose a deep

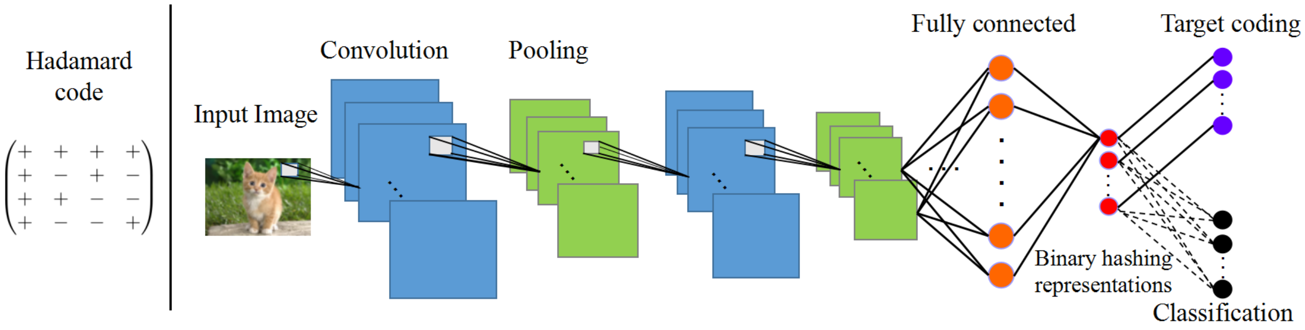


Figure 1: An overview of our proposed Deep Supervised Hashing with Target code(DSHT) consisting of 5 convolutional layers, 2 fully connected layers and a fully-connected hashing layer. We construct a set of hash functions between the second fully-connected layer and our fully-connected hashing layer. DSHT is trained by jointly the target coding and classification, where we introduce Hadamard code for target coding and use Softmax function to reduce classification error.

supervised hashing with target code(DSHT)method to address the two problems simultaneously. Figure 1 shows the basic idea of our DSHT model, and the key components of the DSHT model include:

- A fully-connected hashing layer is used to generate binary codes. With the design, a set of effective hashing functions are learned between the penultimate fully layer and our fully-connected hashing layer.

- In order to obtaining high-quality binary codes, we embed the Hadamard code(target code) into the hash coding to guide binary hashing representations, facilitating the learned binary codes meeting these information-theoretic favored requirements (i.e., balanced and uncorrelated constraints).

- A set of additional bias variables are introduced into corresponding target code to mitigate the disparity of the intra-class.

- Our DSHT model is trained by jointly the target coding and classification with stochastic gradient descent, where Hadamard code for target coding and Softmax-loss for clasification.

2. Methodology

We first give a general introduction of Hadamard code, which is different from 1-of-K code.

2.1. Hadamard code

The Hadamard code can be generated from the Hadamard matrix[10]. Specifically, the binary matrix $H \in \{+1,-1\}^{m\times m}$ is Hadamard matrix if $HH^T = mI$, where I is an identity matrix. This definition implies that any two distinct rows or columns are orthogonal. Generally, one can use the Sylvester's method [10] to generate Hadamard matrix, where a new matrix can be produced from the old one by the Kronecker product. For example, given a Hadamard matrix $H^2 = [++;+-]$, H^4 and H^8 can be generated as following formula (1) and formula (2).

$$H^4 = \begin{pmatrix} + & + \\ + & - \end{pmatrix} \bigotimes \begin{pmatrix} + & + \\ + & - \end{pmatrix} = \begin{pmatrix} + & + & + & + \\ + & - & + & - \\ + & + & - & - \\ + & - & - & + \end{pmatrix} \quad (1)$$

$$H^8 = \begin{pmatrix} + & + & + & + & + & + & + & + \\ + & - & + & - & + & - & + & - \\ + & + & - & - & + & + & - & - \\ + & - & - & + & + & - & - & + \\ + & + & + & + & - & - & - & - \\ + & - & + & - & - & + & - & + \\ + & + & - & - & - & - & + & + \\ + & - & - & + & - & + & + & - \end{pmatrix} \quad (2)$$

It is clearly that the size of the Hadamard matrix H is a power of 2, and each element of the first row or column is 1. The Hadamard code can be obtained by removing the first row and the column of H. According to the definition of H, some properties of the Hadamard code can be obtained. First, each row or column of a Hadamard code has $m/2$ symbols that equal one. Second, the dot of any two rows or columns is -1. Third, the Hamming distance between any two rows or columns is $m/2$. The first two properties show that each row or column codes meet balanced and orthogonal property, and the third property implies that it has own unique code distance equally away from other row or column codes. These properties are beneficial to guide hash functions to learn effective binary hashing representations.

2.2. Target coding

Inspired by the favourable code property and error-correcting capacity of Hadamard code, as well as its superior performances having been demonstrated on image classification[23], we use Hadamard code as the target code to guide our DSHT model binary hashing representations. Tabel 1 shows the basic configuration of our model. Unlike previous works that suffer from the limitations of hand-crafted features and linear projections, or enforce the learned binary codes approximately having 50% chance of being one or zero to maximize the information. Our DSHT can not only break out the limitations of both hand-crafted features or linear models, but also can learn highly balanced codes and uncorrelated bits.

Layer	Configuration
conv1	filter 96x11x11, stride 4x4, pad 0, LRN, pool 2x2
conv2	filter 256x5x5, stride 1x1, pad 2, LRN, pool 2x2
conv3	filter 384x3x3, stride 1x1, pad 1
conv4	filter 384x3x3, stride 1x1, pad 1
conv5	filter 256x3x3, stride 1x1, pad 1
full6	4096
full7	4096
full8	code length

Table 1: The CNN configuration used in our DSHT model.

Denoting $X = [X_1, X_2, \cdots, X_n] \in \mathbb{R}^{d \times n}$ as the training data of n images collected from c classes. In this paper, our purpose is to learn a projection matrix $W_h \in \mathbb{R}^{d \times b}$ that generates binary codes by $B_i = sign(W_h X_i) \in \{-1, 1\}^{b \times 1}$. Specifically, we construct a set of hash functions W_h that mapping the raw image into binary codes with function sign(x), while: a) satisfying the balanced property, b) keeping the uncorrelated property, c) minimizing the quantization loss. More formally, the constraint minimization problem is defined as:

$$\begin{aligned} L_h(W_h) &= ||sign({W_h}^T X) - B||_F^2 + \lambda_1 ||W_h||_F^2 \\ s.t.\ &\forall i \in c, \sum_j B_{i,j} = 0, B^T B = nI \end{aligned} \tag{3}$$

where $B_{i,j}$ denoting j-th bit of i-th image B_i, $||\cdot||_F$ denoting the Frobenius norm, and I being identity matrix as well as λ_1 being a scalar. The constraint $\sum_j B_{i,j} = 0$ enforces each bit to fire 50% of the time, and the constraint $B^T B = nI$ enforces these bits uncorrelated. Obviously, the formula is non-convex, we adopt the following relaxation:

$$\begin{aligned} L_h(W_h) &= ||{W_h}^T X - B||_F^2 + \lambda_1 ||W_h||_F^2 \\ s.t.\ &\forall i \in c, \sum_j B_{i,j} = 0, B^T B = nI \end{aligned} \tag{4}$$

where ${W_h}^T X$ is the real-value output, we can undate the W_h by stochastic gradient descent method. However, the Eq(4) is still difficult to solve due to the discrete constraint variable B. To make the problem tractable, we further propose to simply enforce the binary codes to learn from the predefined Hadamard code, whose potential information-theoretic properties can promote the hash coding to yield high-quality codes without constraint. Thus, the final binary codes can successfully distill the desired properties of Hadamard code. The Eq(4) further relaxtion is given by replacing B with $HY + B^{'}$:

$$L_h(W_h, B^{'}) = ||W_h^T X - (HY + B^{'})||_F^2 + \lambda_1 ||W_h||_F^2 \tag{5}$$

where $H \in \{-1, 1\}^{b \times c}$ is the predefined Hadamard code corresponding to c classes with code length k. The matrix $Y \in \{0, 1\}^{c \times n}$ represents the label ground-truth where each column $y_i = [0 \cdots 1 \cdots 0]^T$ is a one-hot vector and the position of 1 indicates the specific class information. Thus, each column of matrix HY represents the Hadamard code of the corresponding sample. Namely, each sample of the specific class has the identical Hadamard code as target code. Apart from HY representing target code, another encouraging property of HY can facilitate the learning of projection matrix $W_h \in \mathbb{R}^{d \times b}$ to obtain effective binary hashing representations.

In addition, the sample disparity of the intra-class is extensively existed due to various pose, lighting, background and rotation. Obviously, it isn't feasible choice that we only use the identical Hadamard code for each sample of the intra-class to learn binary hashing representations, because it could not reflect the disparity learning of the intra-class. Therefore, we introduce the noise component $B^{'}$ into B to mitigate the intra-class disparity. Finally, the desired solution B comprises two parts HY and $B^{'}$, which not only represents the class information, but also show the disparity learning of the intra-class.

According to the back-propagation algorithm, the gradients of L_n with respect to X_i and W_h are computed as follows respectively:

$$\frac{\partial L_h}{\partial W_h} = (W_h^T X_i - (HY + B_i^{'})) \odot X_i + \lambda_1 ||W_h||_F \tag{6}$$

$$\frac{\partial L_h}{\partial X_i} = (W_h^T X_i - (HY + B_i^{'})) \odot W_h \tag{7}$$

here $\odot$ denotes the multiplication operation by element-wise. Then we can update these parameters by the following stochastic gradient descent with a certain learning rate η until convergence:

$$W_h^{t+1} = W_h^t - \eta \frac{\partial L_h}{\partial W_h} \tag{8}$$

Method	MNIST(bits)				CIFAR-10(bits)			
	12	24	32	48	12	24	32	48
LSH	0.189	0.209	0.235	0.243	0.121	0.126	0.120	0.120
SH	0.265	0.267	0.259	0.250	0.131	0.135	0.133	0.130
ITQ	0.388	0.436	0.422	0.429	0.162	0.169	0.172	0.175
BRE	0.515	0.593	0.613	0.634	0.159	0.181	0.193	0.196
CCA-ITQ	0.659	0.694	0.714	0.726	0.264	0.282	0.288	0.295
KSH	0.872	0.891	0.897	0.900	0.303	0.337	0.346	0.356
SDH	0.896	0.921	0.924	0.929	0.203	0.340	0.354	0.351
FashtHash	0.905	0.916	0.934	0.936	0.293	0.345	0.365	0.391
CNNH	0.969	0.975	0.971	0.975	0.465	0.521	0.521	0.532
DNNH	0.970	0.971	0.974	0.975	0.552	0.566	0.558	0.581
BOH	0.970	0.975	0.978	0.982	0.620	0.633	0.644	0.657
DSHT(Ours)	**0.972**	**0.980**	**0.982**	**0.982**	**0.653**	**0.657**	**0.647**	**0.659**

Table 2: Mean Average Precision (MAP) of Hamming Ranking for different number of bits on MNIST, CIFAR-10

$$X_i^{t+1} = X_i^t - \eta \frac{\partial L_h}{\partial X_i} \tag{9}$$

The gradients of L_h with respect to the variable $B_i^{'}$ are computed as:

$$\Delta B_i^{'} = \frac{\sum_{i=1}^{m}(\delta y_i = k)\cdot(HY + B_i^{'} - W_h X_i)}{1 + \sum_{i=1}^{m}\delta(y_i = k)} \tag{10}$$

here, the indicator function $\delta(y_i = k) = 1$ if the label y_i of image X_i is k; otherwise $\delta(y_i = k) = 0$. Then, we can update those parameters by stochastic gradient descent until convergence:

$$B_{i,t+1}^{'} = B_{i,t}^{'} - \Delta B_i^{'} \tag{11}$$

2.3. Classification

The image label provides supervised information for mining semantic structures in images, and previous researches[24] have made use of image label under a two stream multi-task learning framework to learn effective binary hashing representations. In this paper, in order to make the learned binary codes have more discriminative power, we introduce a co-training mechanism in the way of jointing target coding and classification stream, noting that the classification stream is linked to the hashing layer. In the classification stream, a classification error is measured by the Softmax loss[6]. The Softmax loss function formulation is presented as follows:

$$L_s(\theta) = -\sum_{i=1}^{m} log \frac{e^{\theta_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^{c} e^{\theta_j^T x_i + b_j}} \tag{12}$$

where θ denotes the parameters of network architecture, $y_i \in \{1, 2, \cdots, c\}$ represents image label and $b \in \mathbb{R}^c$ is the bias term. Then, the overall formulation is listed as follows:

$$\begin{aligned} L(\theta, W_h) = & -\sum_{i=1}^{m} log \frac{e^{\theta_{y_i}^T X_i + b_{y_i}}}{\sum_{j=1}^{c} e^{\theta_j^T X_i + b_j}} \\ & + \frac{1}{2}\sum_{i=1}^{m} ||W_h X_i - (HY + B_i{'})||^2 + \lambda_1 ||W_h||_F^2 \end{aligned} \tag{13}$$

We adopt stochastic gradient descent algorithm to update all these above parameters until convergence.

3. Experiments

We conducted evaluations of our proposed method on two extensively used image datasets, i.e., MNIST, CIFAR-10.

3.1. Datasets

The MNIST dataset consists of 700,000 greyscale images of handwritten digits from '0' to '9' with size 28 x 28. The CIFAR-10 dataset consists of 60,000 color images in 10 classes, and each class has 6,000 images with size 32 x 32.

Following the same setting[9, 22], we randomly select 1,000 images (100 images per class) from the whole set as test queries. For the unsupervised methods, the whole rest images are used for training. While for the supervised methods, we randomly select 5,000 images (500 images per class) from the rest images as the training set.

3.2. Experimental Settings

We implement the proposed method based on the open source Caffe [6] framework. In all experiments, our networks are trained by stochastic gradient descent with 0.9 momentum. The weight decay parameter is 0.0005. The initiate based learning rate is 0.0001 and decrease it by 20%

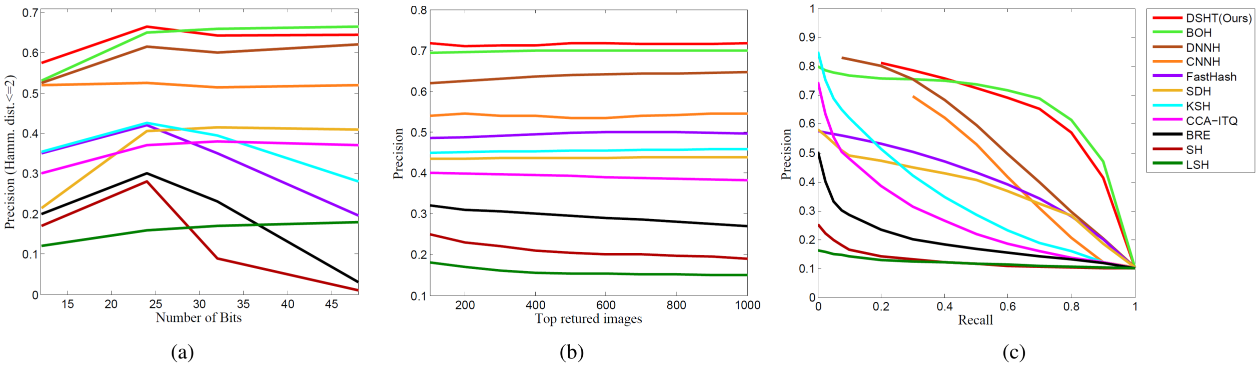
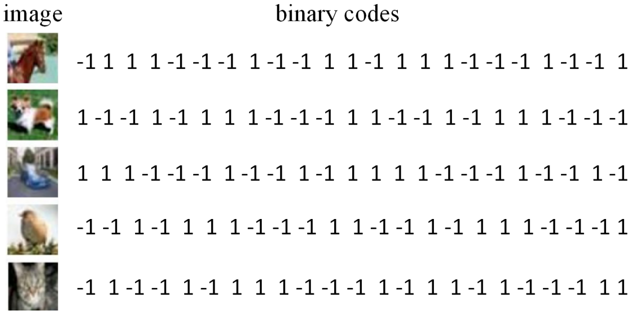


Figure 2: Comparative evaluation of different hashing algorithms on CIFAR-10. (a)Precision within Hamming radius 2 curves with respect to different number of hash bits. (b) Precision curves with respect to different number of top retrieved samples when the 48-bit hash codes are used. (c)Precision-recall curves with 48 bits.



-1 1 1 1 -1 -1 -1 1 -1 -1 1 1 -1 1 1 1 -1 -1 -1 1 -1 -1 1

1 -1 -1 1 -1 1 1 1 -1 -1 -1 1 1 -1 -1 1 -1 1 1 1 -1 -1 -1

1 1 1 -1 -1 -1 1 -1 -1 1 -1 1 1 1 1 -1 -1 -1 1 -1 -1 1 -1

-1 -1 1 -1 1 1 1 -1 -1 -1 1 1 -1 -1 1 -1 1 1 1 -1 -1 -1 1

-1 1 -1 -1 1 -1 1 1 1 -1 -1 -1 1 -1 1 1 -1 1 -1 -1 -1 1 1

Figure 3: Binary codes of selected query from 5 classes.

after every 3,000 iterations. The mini-batch size of images is 128. The size of input image is resized to 256 x 256.

3.3. Results and Analysis

We compare our method with LSH[3], SH[21], ITQ[4], CCA-ITQ[4], BRE[8], KSH[14], FashterHash[12], SDH[18], CNNH[22], DNNH[9] and BOH[1]. These methods were all implemented using source codes provided by the author, the experiments result of CIFAR-10 refer to the result of paper[1].

Table 2 shows the comparative results based on the MAP. As can be seen, those CNN-based methods outperform the conventional hash learning methods on both datasets by a large margin. Our method provides the best performance for different code lengths.

Figure 2 shows Precision curves with hamming radius 2 w.r.t the different code of length, Precision w.r.t. top returned samples curves with 48 bits and Precision-recall curves with 48 bits on CIFAR-10, respectively. From these curves, we can see clearly that our method outperforms other methods by certain margins.

Figure 3 shows binary codes of 5 query images from different classes with 23 bits. Specifically, the left column is raw images, and the right column is the corresponding binary codes. Obiviously, the sum of each row binary codes is -1(equaling to the reality -1), meeting the balanced constraints, and the dot of binary codes from any two is -1(equaling to the reality -1), meeting to the unrelated properties. Those results demonstrate that our proposed method can get unrelated and balanced binary codes .

Method	12 bits	24 bits	32 bits	48 bits
DSHT	0.620	0.633	0.644	0.657
DSHT-B	0.608	0.629	0.641	0.645

Table 3: Mean Average Precision (MAP) results of DSHT and DSHT-B on CIFAR-10

Method	12 bits	24 bits	32 bits	48 bits
DSHT	0.620	0.633	0.644	0.657
DSHT-S	0.598	0.616	0.635	0.639

Table 4: Mean Average Precision (MAP) results of DSHT and DSHT-S on CIFAR-10

We investigate another variant of DSHT:DSHT-B is the DSHT variant without the bias B(not considering the difference of intra-class), and comparative results are shown in Table 3. We can observe that, the MAP of DSHT is slightly superior to DSHT-B with different length of binary codes. These results validate that the bias B is flexiabel and helpful.

The learning of binary hashing representations is implemented by jointly target coding and classification stream in our model. In order to evaluate whether the classification stream is conducive to learn binary hashing representations, we consider another model DSHT-S: DSHT-S is the DSHT variant, which only implement target coding without classification stream. Tabel 4 shows the MAP of two individual model on CIAFR-10 with different length of binary codes. Obviously, the classification stream is conducive to learn binary codes.

4. Conclusions

In this paper, we present a deep supervised hashing with target code method for learning effective binary hashing representations. The desirable properties of final binary codes are obtained by distilling the knowledge of the predefined target code. Specifically, we constructs hash functions as a latent layer in a deep neural network, and the model is trained in the way of jointing target coding and classification. Moreover, we introduce an additional variable to mitigate the disparity of the intra-class. The experimental results validate the effectiveness of our model.

5. Acknowledgement

This work is funded by Beijing Municipal Science and Technology Commission (No.Z161100000216144)

References

[1] Q. Dai, J. Li, J. Wang, and Y. Jiang. Binary optimized hashing. In *ACM Multimedia*, pages 1247–1256. ACM, 2016. 5

[2] T. Do, D. L. Tan, T. T. Pham, and N. Cheung. Simultaneous feature aggregating and hashing for large-scale image search. *CoRR*, abs/1704.00860, 2017. 1

[3] A. Gionis, P. Indyk, and R. Motwani. Similarity search in high dimensions via hashing. In *VLDB*, pages 518–529. Morgan Kaufmann, 1999. 1, 5

[4] Y. Gong and S. Lazebnik. Iterative quantization: A procrustean approach to learning binary codes. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 817–824. IEEE, 2011. 1, 5

[5] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778. IEEE Computer Society, 2016. 1

[6] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. B. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *ACM Multimedia*, pages 675–678. ACM, 2014. 4

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1106–1114, 2012. 1

[8] B. Kulis and T. Darrell. Learning to hash with binary reconstructive embeddings. In *NIPS*, pages 1042–1050. Curran Associates, Inc., 2009. 1, 5

[9] H. Lai, Y. Pan, Y. Liu, and S. Yan. Simultaneous feature learning and hash coding with deep neural networks. In *CVPR*, pages 3270–3278. IEEE Computer Society, 2015. 4, 5

[10] J. Langford and A. Beygelzimer. Sensitive error correcting output codes. In *COLT*, volume 3559, pages 158–172. Springer, 2005. 2

[11] W. Li, S. Wang, and W. Kang. Feature learning based deep supervised hashing with pairwise labels. In *IJCAI*, pages 1711–1717. IJCAI/AAAI Press, 2016. 1

[12] G. Lin, C. Shen, Q. Shi, A. van den Hengel, and D. Suter. Fast supervised hashing with decision trees for high-dimensional data. In *CVPR*, pages 1971–1978. IEEE Computer Society, 2014. 5

[13] K. Lin, J. Lu, C. Chen, and J. Zhou. Learning compact binary descriptors with unsupervised deep neural networks. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 1183–1192, 2016. 1

[14] W. Liu, J. Wang, R. Ji, Y. Jiang, and S. Chang. Supervised hashing with kernels. In *CVPR*, pages 2074–2081. IEEE Computer Society, 2012. 1, 5

[15] J. Lu, V. E. Liong, X. Zhou, and J. Zhou. Learning compact binary face descriptor for face recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 37(10):2041–2056, 2015. 1

[16] T. Ojala, M. Pietikäinen, and T. Mäenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(7):971–987, 2002. 1

[17] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International Journal of Computer Vision*, 42(3):145–175, 2001. 1

[18] F. Shen, C. Shen, W. Liu, and H. T. Shen. Supervised discrete hashing. In *CVPR*, pages 37–45. IEEE Computer Society, 2015. 1, 5

[19] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. 1

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

[21] Y. Weiss, A. Torralba, and R. Fergus. Spectral hashing. In *NIPS*, pages 1753–1760. Curran Associates, Inc., 2008. 1, 5

[22] R. Xia, Y. Pan, H. Lai, C. Liu, and S. Yan. Supervised hashing for image retrieval via image representation learning. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada.*, pages 2156–2162, 2014. 1, 4, 5

[23] S. Yang, P. Luo, C. C. Loy, K. W. Shum, and X. Tang. Deep representation learning with target coding. In *AAAI*, pages 3848–3854. AAAI Press, 2015. 3

[24] T. Yao, F. Long, T. Mei, and Y. Rui. Deep semantic-preserving and ranking-based hashing for image retrieval. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 3931–3937, 2016. 4

[25] F. Yu, V. Koltun, and T. Funkhouser. Dilated residual networks. *arXiv preprint arXiv:1705.09914*, 2017. 1

[26] H. Zhu, M. Long, J. Wang, and Y. Cao. Deep hashing network for efficient similarity retrieval. In *AAAI*, pages 2415–2421. AAAI Press, 2016. 1