CLUSTERING AND SEPARATING SIMILARITIES FOR DEEP UNSUPERVISED HASHING

Wanqian Zhang[1] *Dayan Wu*[1*] *Chule Yang*[2] *Bo Li*[1] *Weiping Wang*[1]

[1]Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
[2]National Innovation Institute of Defense Technology, Academy of Military Science, Beijing, China
{zhangwanqian,wudayan,libo,wangweiping}@iie.ac.cn, yangchule@126.com

ABSTRACT

The lack of supervised information is the pivotal problem in unsupervised hashing. Most methods leverage deep features extracted from pre-trained models to generate semantic similarities as supervised information. These fixed features are, however, neither designed originally for retrieval nor updated adaptively during training. In this paper, we propose a novel deep Unsupervised Cluster and Separate Hashing (UCSH) to address these issues. Specifically, we introduce a fully end-to-end deep hashing network with a binary latent Variational AutoEncoder (VAE), which enables hash codes capable of reconstructing deep features as well as preserving semantic relations. Moreover, a '*Cluster* and *Separate*' scheme is proposed to jointly *cluster* deep features and *separate* semantic similarities. Both the implicit feature clustering and the explicit similarity separating loss encourage the separation of similar and dissimilar pairs, enabling the iteratively updated similarities to better excavate semantic relations. Experiments conducted on three benchmarks show the superiority of UCSH.

Index Terms— deep hashing, image retrieval, unsupervised method

1. INTRODUCTION

To cope with the tremendous volume of visual data, many image retrieval systems resort to hashing to embed high dimensional features in the original space into binary codes in the Hamming space, whilst preserving the semantic similarities. Generally, hashing can be divided into supervised hashing [1, 2, 3, 4, 5, 6] and unsupervised hashing [7, 8, 9]. Unsupervised hashing methods train the model in a fully label-free manner, which is also the interest of this paper.

Traditional unsupervised hashing [8, 10] usually achieves far below satisfactory performance, since the hand-crafted features are extracted from shallow models. Recent advances in convolutional neural networks (CNNs) usher great progress in deep unsupervised hashing. DSTGeH [11] tries to capture the second-order proximity on the graph and maps them into pseudo labels. Despite the improvements they have made, there still exist some long-standing and significant challenges in the unsupervised hashing paradigm.

First, how to define the semantic relations when the label information can't be accessed. SSDH [12] designs a pairwise loss to preserve the similarities in the original space by constructing a semantic structure based on feature distribution. However, the introduced prior knowledge of data relevance is biased. These features are not originally designed for image retrieval, which means trivially combining pre-trained features with hash code learning often leads to degenerate solutions.

Second, previous methods usually make the constructed similarities fixed during the whole training procedure, which obviously enlarges the discrepancy between fixed features and updated hash codes. SADH [13] alternatingly proceeds over deep hash training, similarity updating and binary code optimization in a unified framework. TBH [14] introduces an efficient and adaptive code-driven graph, which is updated by decoding in the context of an auto-encoder. While fairly successful, these methods pay little attention to the updated similarity (the supervised information) itself.

Bearing in mind the above issues, we propose a novel deep Unsupervised Cluster and Separate Hashing (UCSH), of which the schematic diagram is illustrated in Fig. 1. First, the fully end-to-end (from images to codes) generative framework integrates the binary latent VAE into the deep hashing network, and hash codes are thus capable of reconstructing deep features as well as preserving semantic similarities. Second, a novel '*Cluster* and *Separate*' scheme is proposed to jointly cluster deep features and separate semantic similarities. Deep image features are clustered into centroids assigned with binary codes, which implicitly encourages the separation of similar and dissimilar image pairs. Meanwhile, a plug-and-play similarity separating loss is further devised to explicitly achieve the desired separation effect, which also allows for end-to-end training with a simple and elegant implementation. All in all, by alternating between updating model parameters, learning hash codes, clustering deep features and separating semantic similarities, UCSH can achieve satisfactory retrieval performance.

The main contributions are outlined as follows: (1) The proposed deep Unsupervised Cluster and Separate Hashing

[*] Corresponding author

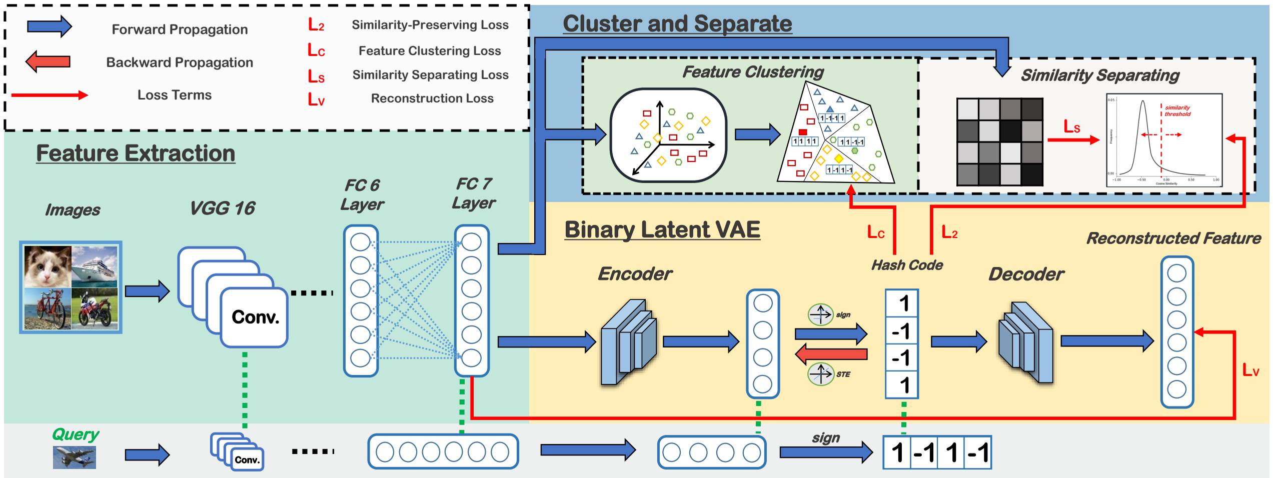

Fig. 1. Schematic diagram of UCSH framework (best viewed in color). Images are fed into VGG16 to obtain deep features. The binary latent VAE enables hash codes to reconstruct image features and to preserve semantic relations. Moreover, the implicit feature clustering and explicit similarity separating encourage the separation of similar and dissimilar pairs iteratively, leading to discriminative hash codes.

(UCSH) incorporates a binary latent VAE into an end-to-end hashing network, making hash codes capable of reconstructing deep features as well as preserving semantic similarities. (2) A '*Cluster* and *Separate*' scheme is proposed to jointly cluster deep features and separate semantic similarities. Similarities can thus be updated iteratively with the guidance of the implicit feature clustering and the explicit similarity separating loss. (3) Competitive results on three benchmark datasets, i.e., CIFAR-10, NUS-WIDE and MIRFLICKR-25K, demonstrate the superiority of UCSH.

2. PROPOSED METHOD

2.1. Notations and Framework Overview

Assume that there are a set of n images $\mathbf{A}$ and the corresponding d-dimensional image features $\mathbf{X} = [x_1, x_2, ..., x_n] \in \mathbb{R}^{n\times d}$. The goal is to learn the hash function $h : \mathbf{X} \rightarrow \mathbf{B}$, where $\mathbf{B} = [b_1, b_2, ..., b_n] \in \{-1, +1\}^{n\times k}$, and k is the code length. Here, image feature x_i is obtained through CNN $F(a_i; \Theta)$, where Θ is the model parameters to be learned. In particular, we use the VGG16 network pre-trained on ImageNet dataset as the initialization of our model, which means $\mathbf{X}$ can be denoted as the features of last but one layer (i.e., FC7) and d is 4096. We further replace the last FC layer with the proposed hash layer, of which the output can be applied with element-wise $sign$ operation to generate desired binary codes $\mathbf{B}$. We use $||\cdot||$ to denote the l_2 norm of vectors and the Frobenius norm of matrices.

2.2. Binary Latent VAE

We first train the VAE network to embed the deep features $\mathbf{X}$ into a binary latent space. Under the VAE framework [15], a generative (decoder) model reconstructs the inputs $\mathbf{X}$ from the binary latent variables (hash codes), while an inference (encoder) model infers the codes $\mathbf{B}$ from the inputs $\mathbf{X}$. In particular, given an image feature x, in order to get the binary code b, we need to learn a posteriori probability distribution $p(b|x)$, which is considered infeasible to compute. We thereafter resort to learning a proxy posteriori $q_{\theta^v}(b|x)$, parameterized by θ^v of VAE, which can approximate $p(b|x)$. To estimate parameters of the encoder and the decoder, the VAE framework optimizes evidence lower bound (ELBO) with the Kullback–Leibler (KL) regularization defined as:

$$\mathcal{L}_V = \mathbb{E}_{(x,b)\sim p_\phi(B|X)}[\log q_{\theta^v}(x|b)] - KL[p(b|x)||q_{\theta^v}(b)]. \tag{1}$$

2.3. Pairwise Similarity Hashing

The core idea of similarity-preserving hashing is to minimize/maximize the distance between binary codes of similar/dissimilar images. To that end, we first analyze the semantic relations of images by calculating the cosine distance d_{ij} for each image pair (x_i, x_j) based on the extracted deep features. The widely adopted cosine distance can be defined as: $d_{ij} = 1 - \langle x_i, x_j\rangle/||x_i||||x_j||$. Here, the range of cosine distance d_{ij} is $[0, 1)$, due to the $ReLU$ activation on x_i in the FC7 layer. Then, with a simple transformation, we can obtain the desired semantic similarities, which is parameter-free and

Method	CIFAR-10			NUS-WIDE			MIRFLICK-25K		
	16bit	32bit	64bit	16bit	32bit	64bit	16bit	32bit	64bit
SH	0.1905	0.2083	0.2109	0.4350	0.4129	0.4062	0.6091	0.6105	0.6033
ITQ	0.2042	0.2286	0.2251	0.5270	0.5241	0.5334	0.6492	0.6518	0.6546
LSH	0.1235	0.1478	0.1675	0.4632	0.5243	0.6012	0.6031	0.6285	0.6333
SpH	0.1671	0.1825	0.1903	0.4458	0.4537	0.4926	0.6119	0.6315	0.6381
DeepBit	0.2543	0.2672	0.2714	0.3844	0.4341	0.4461	0.5934	0.5933	0.6199
SSDH	0.3003	0.3153	0.3228	0.6374	0.6768	0.6829	0.7240	0.7276	0.7377
DistillHash*	0.2844	0.2853	0.2867	0.6667	0.6752	0.6769	0.6964	0.7056	0.7075
UCSH	**0.4953**	**0.4988**	**0.5047**	**0.7795**	**0.7831**	**0.8146**	**0.8093**	**0.8166**	**0.8312**

Table 1. MAP@5k on CIFAR-10, NUS-WIDE and MIRFLICKR-25K. The best is shown in boldface.

can be defined as:

$$S_{ij} = 1 - 2d_{ij}. \tag{2}$$

Since there exists a nice linear relationship between the Hamming distance $distH(b_i, b_j)$ and the inner product $\langle b_i, b_j \rangle$: $distH(b_i, b_j) = (r - \langle b_i, b_j \rangle)/2$, we can thereafter adopt the L_2 loss to guide the hash code learning:

$$\min \mathcal{L}_2 = \sum_{i=1}^{n} \sum_{j=1}^{n} ||kS_{ij} - b_i b_j^T||^2. \tag{3}$$

2.4. *Cluster* and *Separate* Scheme

Feature Clustering. Retrieval bears some similarities to clustering - if all relevant images are '*closer*' than non-relevant images in the original space, perfect retrieval is achieved. Consequently, we hypothesize that by the clustering of deep image features we can achieve an analogous effect, encouraging the separation of similar and dissimilar image pairs. Thus, we first acquire classes $\mathbf{C} = [C_1, C_2, ..., C_N]$ after a few iterations of K-means clustering, where N is a hyper parameter denoting the number of centroids. We then introduce the Hadamard matrix [16] into our method, which is $\mathbf{H}_{2^k}$ of size $2^k \times 2^k$ and can be constructed as:

$$\mathbf{H}_2 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \quad \mathbf{H}_{2^k} = \begin{bmatrix} \mathbf{H}_{2^{k-1}} & \mathbf{H}_{2^{k-1}} \\ \mathbf{H}_{2^{k-1}} & -\mathbf{H}_{2^{k-1}} \end{bmatrix} \quad (k \geq 2). \tag{4}$$

Both row and column vectors of $\mathbf{H}_{2^k}$ are binary and pairwise orthogonal, which also satisfies the desired properties of independence and balance for hash codes. Hence, we assign each image x_i a column vector $h_{C_{x_i}}$ and introduce the feature clustering loss:

$$\min \mathcal{L}_C = \sum_{i=1}^{n} ||b_i - h_{C_{x_i}}||^2. \tag{5}$$

By clustering deep features during each iteration, the semantic similarities can be guided implicitly and updated adaptively.

Similarity Separating. Borrowing ideas from clustering, we wonder whether it is possible to explicitly guide the updating of semantic similarities, thus images with higher similarity scores move closer together, while those with lower scores get pushed further apart. Here, we propose a novel similarity separating loss:

$$\min \mathcal{L}_S = -\sum_{i=1}^{n} ||s_{ij} - s_t||^2, \tag{6}$$

where s_t is a similarity threshold. Following SSDH [12], the distribution of feature distances can be approximately estimated by two half Gaussian distributions, denoted as $\mathcal{N}(m_1, \sigma_1^2)$ and $\mathcal{N}(m_2, \sigma_2^2)$ respectively, where m and σ^2 are corresponding means and variances. We consider the image pair (x_i, x_j), of which the distance is smaller than $d_t = (m_1 - 2\sigma_1)$, as a similar one. Thus, we set $s_t = 1 - 2d_t$ as the similarity threshold and update it adaptively during each iteration.

In a nutshell, we have the overall loss function which is formulated as:

$$\min \mathcal{L} = \mathcal{L}_2 + \alpha \mathcal{L}_V + \beta \mathcal{L}_C + \gamma \mathcal{L}_S, \tag{7}$$

where α, β and γ are hyper parameters for balance. With the straight through gradient estimator (STE) [17], the gradient can be back propagated through the whole network.

3. EXPERIMENTS

3.1. Datasets and Implementations

We conduct the experiments on three benchmarks for image retrieval: CIFAR-10, NUS-WIDE and MIRFLICKR-25K. CIFAR-10 consists of 60k color images of size 32×32 selected from the 80M tiny image collection. Images are divided into 10 classes evenly. NUS-WIDE consists of 269,648 images collected from Flickr. It contains 81 classes, and each image is tagged with multiple semantic labels. As in [12], we use the 10 most frequent labels for evaluation, which results in a total of 186,577 images. MIRFLICKR-25K is also collected from Flickr, which consists of 25k multi-label images.

Method	CIFAR-10			NUS-WIDE			MIRFLICK-25K		
	16bit	32bit	64bit	16bit	32bit	64bit	16bit	32bit	64bit
UCSH	**0.4953**	**0.4988**	**0.5047**	**0.7795**	**0.7831**	**0.8146**	**0.8093**	**0.8166**	**0.8312**
UCSH-V	0.4805	0.4843	0.4922	0.7690	0.7721	0.7902	0.7921	0.8075	0.8233
UCSH-C	0.4628	0.4691	0.4688	0.7472	0.7511	0.7714	0.7652	0.7728	0.7841
UCSH-S	0.4775	0.4833	0.4845	0.7552	0.7593	0.7782	0.7737	0.7795	0.7993
UCSH-R	0.4831	0.4865	0.4881	0.7658	0.7717	0.7986	0.7898	0.7981	0.8106

Table 2. MAP@5k results of UCSH and its variants on three datasets with different code lengths.

Each image is associated with at least one of the 24 labels. We adopt the same data splits as in [12].

We compare our method with several state-of-the-art unsupervised hashing methods, including ITQ [18], SH [19], LSH [20], SpH [21], DeepBit [22], SSDH [12], DistillHash [23] and TBH [14]. The batch size is 24, and the step-decayed learning rate is initialized with 1e-5. Besides, α is set to 1e-3, β is set to 10, γ is set to 1e-2 and N is set to 10 for CIFAR-10, 30 for NUS-WIDE and MIRFLICK-25K. We conduct experiments on a NVIDIA M40 GPU server with the deep learning framework Pytorch.

3.2. Result Analysis

The MAP@5k results of our method and other baselines on three datasets are shown in Table 1. On all datasets, UCSH consistently outperforms other baselines by a large margin. To be specific, UCSH shows improvements of 28.04%, 25.39% and 15.06% in average on CIFAR-10, NUS-WIDE and MIRFLICKR-25K respectively. Compared with traditional methods, deep hashing methods can always show superior MAP performances, which indicates the great power of CNN models in image feature representations. When compared with state-of-the-art deep hashing methods SSDH [12] and DistillHash* [23] (MAP@all of DistillHash is reported from the original paper since the codes are unavailable), our method can still achieve the significant results in all situations.

3.3. Ablation and Parameter Study

Table 2 shows the MAP@5k results of UCSH and several variants on three datasets. UCSH outperforms UCSH-V (w/o VAE) by a large margin, indicating the importance of reconstruction ability of hash codes. The performance degradation of UCSH-C (w/o clustering) shows that deep clustering exerts an implicit effort to make similarities better fit the semantic relations of images. Lower results of UCSH-S (w/o separating) proves the effectiveness of the '*Cluster* and *Separate*' scheme. Compared with UCSH-R (replacing STE by continuous relaxation), UCSH shows better retrieval performance, indicating the excellent approximation of STE.

Fig.2 shows the MAP@5k results of UCSH with respect to different values of the hyper parameters α, β, γ and N on three datasets. When α varies in $[10^{-4}, 10^{0}]$, β varies in $[10^{-2}, 10^{1}]$, γ varies in $[10^{-4}, 10^{-1}]$ and N varies in $[10, 40]$, our method can always obtain the satisfying performance on three datasets. As can be seen, UCSH is not sensitive to hyper parameters, and achieves great results within a wide range.

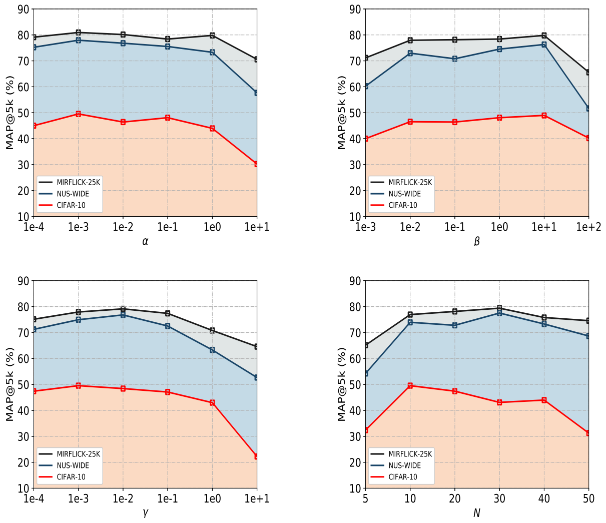

Fig. 2. MAP@5k results *w.r.t.* α, β and N on three datasets with 16 bits (best viewed in color).

4. CONCLUSION

In this paper, we propose a novel deep Unsupervised Cluster and Separate Hashing (UCSH). The proposed binary latent VAE enables hash codes to reconstruct deep features as well as preserve semantic similarities. Moreover, a novel '*Cluster* and *Separate*' scheme is proposed to jointly *cluster* deep features and *separate* semantic similarities. Extensive experiments conducted on three benchmark datasets show the superiority of UCSH. In future work, we will explore the unsupervised hashing through the lens of data augmentation and category alignment across different scales.

5. ACKNOWLEDGEMENT

This work was supported by the National Natural Science Foundation of China under Grants 62106258 and 62006242.

6. REFERENCES

[1] Zhan Yang, Jun Long, Lei Zhu, and Wenti Huang, "Nonlinear robust discrete hashing for cross-modal retrieval," in *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020.

[2] Feng Zheng, Cheng Deng, and Heng Huang, "Binarized neural networks for resource-efficient hashing with minimizing quantization loss," in *Proceedings of the International Joint Conference on Artificial Intelligence*, 2019.

[3] Yuming Shen, Li Liu, Fumin Shen, and Ling Shao, "Zero-shot sketch-image hashing," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[4] Dayan Wu, Qi Dai, Jing Liu, Bo Li, and Weiping Wang, "Deep incremental hashing network for efficient image retrieval," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[5] Wanqian Zhang, Dayan Wu, Yu Zhou, Bo Li, Weiping Wang, and Dan Meng, "Binary neural network hashing for image retrieval," in *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020.

[6] Shu Zhao, Dayan Wu, Yucan Zhou, Bo Li, and Weiping Wang, "Rescuing deep hashing from dead bits problem," in *Proceedings of the International Joint Conference on Artificial Intelligence*, 2021.

[7] Xuesong Gu, Guohua Dong, Xiang Zhang, Long Lan, and Zhigang Luo, "Towards making unsupervised graph hashing robust," in *IEEE International Conference on Multimedia and Expo*, 2020.

[8] Zhihui Lai, Yudong Chen, Jian Wu, Wai Keung Wong, and Fumin Shen, "Jointly sparse hashing for image retrieval," *IEEE Transactions on Image Processing*, 2018.

[9] Wanqian Zhang, Dayan Wu, Yu Zhou, Bo Li, Weiping Wang, and Dan Meng, "Deep unsupervised hybrid-similarity hadamard hashing," in *Proceedings of the ACM International Conference on Multimedia*, 2020.

[10] Tian-yi Chen, Lan Zhang, Shi-cong Zhang, Zi-long Li, and Bai-chuan Huang, "Extensible cross-modal hashing," in *Proceedings of the International Joint Conference on Artificial Intelligence*, 2019.

[11] Yu Liu, Yangtao Wang, Jingkuan Song, Chan Guo, Ke Zhou, and Zhili Xiao, "Deep self-taught graph embedding hashing with pseudo labels for image retrieval," in *IEEE International Conference on Multimedia and Expo*, 2020.

[12] Erkun Yang, Cheng Deng, Tongliang Liu, Wei Liu, and Dacheng Tao, "Semantic structure-based unsupervised deep hashing," in *Proceedings of the International Joint Conference on Artificial Intelligence*, 2018.

[13] Fumin Shen, Yan Xu, Li Liu, Yang Yang, Zi Huang, and Heng Tao Shen, "Unsupervised deep hashing with similarity-adaptive and discrete optimization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[14] Yuming Shen, Jie Qin, Jiaxin Chen, Mengyang Yu, Li Liu, Fan Zhu, Fumin Shen, and Ling Shao, "Auto-encoding twin-bottleneck hashing," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[15] Diederik P Kingma and Max Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.

[16] Mingbao Lin, Rongrong Ji, Hong Liu, and Yongjian Wu, "Supervised online hashing via hadamard codebook learning," in *Proceedings of the ACM International Conference on Multimedia*, 2018.

[17] Yoshua Bengio, Nicholas Léonard, and Aaron Courville, "Estimating or propagating gradients through stochastic neurons for conditional computation," *arXiv preprint arXiv:1308.3432*, 2013.

[18] Yunchao Gong, Svetlana Lazebnik, Albert Gordo, and Florent Perronnin, "Iterative quantization: A procrustean approach to learning binary codes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2011.

[19] Yair Weiss, Antonio Torralba, and Rob Fergus, "Spectral hashing," in *Proceedings of the International Conference on Neural Information Processing Systems*, 2009.

[20] Aristides Gionis, Piotr Indyk, and Rajeev Motwani, "Similarity search in high dimensions via hashing," in *Proceedings of the International Conference on Very Large Data Bases*, 1999.

[21] Jae-Pil Heo, Youngwoon Lee, Junfeng He, Shih-Fu Chang, and Sung-Eui Yoon, "Spherical hashing," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2012.

[22] Kevin Lin, Jiwen Lu, Chu-Song Chen, and Jie Zhou, "Learning compact binary descriptors with unsupervised deep neural networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2016.

[23] Erkun Yang, Tongliang Liu, Cheng Deng, Wei Liu, and Dacheng Tao, "Distillhash: Unsupervised deep hashing by distilling data pairs," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019.