Deep Hashing for Compact Binary Codes Learning

Venice Erin Liong[1], Jiwen Lu[1], Gang Wang[1,2], Pierre Moulin[1,3], and Jie Zhou[4]
[1]ADSC, Singapore, [2]NTU, Singapore, [3]UIUC, USA, [4]Tsinghua University, China

Large scale visual search has attracted great attention in computer vision due to its wide potential applications [1]. Hashing is a powerful technique for large-scale visual search and a variety of hashing-based methods have been proposed in the literature [3, 4, 7]. The basic idea of hashing-based approach is to construct a series of hash functions to map each visual object into a binary feature vector so that visually similar samples are mapped into similar binary codes.

In this paper, we propose a new deep hashing (DH) method to learn compact binary codes for large scale visual search. Figure 1 illustrates the basic idea of the proposed approach. Different from most existing binary codes learning methods which usually seek a single linear projection to map each sample into a binary vector [2, 5, 6], we develop a deep neural network to seek multiple hierarchical non-linear transformations to learn these binary codes. For a given sample $\mathbf{x}_n$, we obtain a binary vector $\mathbf{b}_n$ by passing it to a network which contains multiple stacked layers of nonlinear transformations. Assume we have $M+1$ layers, the output for the mth layer is: $\mathbf{h}_n^m = s(\mathbf{W}^m\mathbf{h}_n^{m-1} + \mathbf{c}^m)$ where $\mathbf{W}^m$ and $\mathbf{c}^m$ is the projection matrix and bias vector, to be learned at the mth layer of the network respectively, and $s(\cdot)$ is a non-linear activation function. We perform hashing for the output $\mathbf{h}^M$ at the top layer of the network to obtain binary codes as follows $\mathbf{b}_n = \mathrm{sgn}(\mathbf{h}_n^M)$ which is parameterized by $\{\mathbf{W}^m, \mathbf{c}^m\}_{m=1}^M$. Our model is learned under three constraints at the top layer of the deep network. By using the matrix representation of the binary codes vectors and the output of the mth layer of the network, we formulate the following optimization problem to learn the parameters of our network:

$$\begin{aligned}\arg\min_{\mathbf{W},\mathbf{c}} J &= J_1 - \lambda_1 J_2 + \lambda_2 J_3 + \lambda_3 J_4 \\ &= \frac{1}{2}\|\mathbf{B}-\mathbf{H}^M\|_F^2 - \frac{\lambda_1}{2N}\mathrm{tr}((\mathbf{H}^M\mathbf{H}^M)^T) \\ &+ \frac{\lambda_2}{2}\sum_{m=1}^{M}\|\mathbf{W}^m(\mathbf{W}^m)^T - \mathbf{I}\|_F^2 + \frac{\lambda_3}{2}(\|\mathbf{W}^m\|_F^2 + \|\mathbf{c}^m\|_2^2)\end{aligned}$$

The first term J_1 aims to minimize the quantization loss between the learned binary vectors and the original real-valued vectors. The second term J_2 aims to maximize the variance of learned binary vectors to ensure balanced bits. The third term J_3 enforces a relaxed orthogonality constraint on those projection matrices so that the independence of each transform is maximized. The last term J_4 are regularizers to control the scales of the parameters. λ_1, λ_2, and λ_3 are three parameters to balance the effect of different terms.

We also propose a supervised deep hashing (SDH) method which extends DH into a supervised version to enhance the discriminative power of DH. For each pair of training samples $(\mathbf{x}_i, \mathbf{x}_j)$, we know whether they are from the same class or not. Hence, we can construct two sets $\mathcal{S}$ or $\mathcal{D}$ from the training set, which represents the positive samples pairs and the negative samples pairs in the training set, respectively. Then, we formulate the following optimization problem for our SDH method:

$$\begin{aligned}\arg\min_{\mathbf{W},\mathbf{c}} J &= \frac{1}{2}\|\mathbf{B}-\mathbf{H}^M\|_F^2 - \frac{\lambda_1}{2}(\mathrm{tr}(\frac{1}{N}\mathbf{H}^M(\mathbf{H}^M)^T) + \alpha\mathrm{tr}(\Sigma_B - \Sigma_W)) \\ &+ \frac{\lambda_2}{2}\sum_{m=1}^{M}\|\mathbf{W}^m(\mathbf{W}^m)^T - \mathbf{I}\|_F^2 + \frac{\lambda_3}{2}\sum_{m=1}^{M}(\|\mathbf{W}^m\|_F^2 + \|\mathbf{c}^m\|_2^2)\end{aligned}$$

where

$$\Sigma_W = \frac{1}{N_S}\mathrm{tr}((\mathbf{H}_{s1}^M - \mathbf{H}_{s2}^M)(\mathbf{H}_{s1}^M - \mathbf{H}_{s2}^M)^T) \quad (1)$$

$$\Sigma_B = \frac{1}{N_D}\mathrm{tr}((\mathbf{H}_{d1}^M - \mathbf{H}_{d2}^M)(\mathbf{H}_{d1}^M - \mathbf{H}_{d2}^M)^T) \quad (2)$$

This is an extended abstract. The full paper is available at the Computer Vision Foundation webpage.

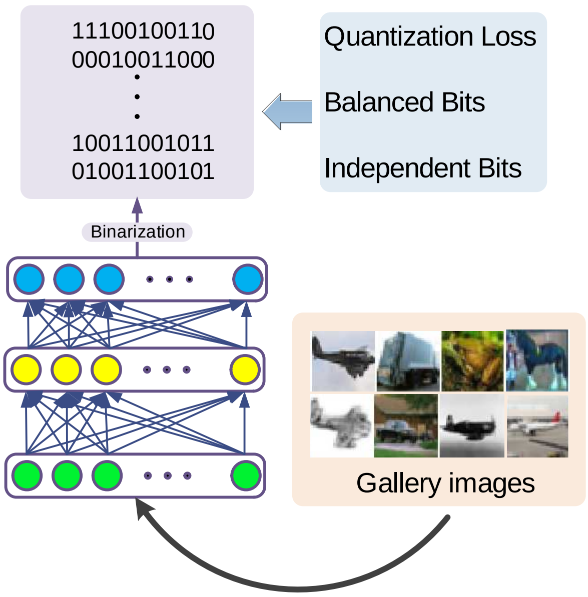

Figure 1: The basic idea of our proposed approach for compact binary codes learning. Given a gallery image set, we develop a deep neural network and learn the parameters of the network by using three criterions for the codes obtained at the top layer of the network: 1) minimizing loss between the original real-valued feature and the learned binary vector; 2) binary codes distribute evenly on each bit, and 3) each bit is as independent as possible. The parameters of the networks are updated by back-propagation based on the optimization objective function at the top layer.

N_S and N_D are the number of neighbor and non-neighbor pairs, $\{\mathbf{H}_{s1}^m, \mathbf{H}_{s2}^m\}_{m=1}^M$ are the hidden representation of sample pairs in $\mathcal{S}$ and $\{\mathbf{H}_{d1}^m, \mathbf{H}_{d2}^m\}_{m=1}^M$ in $\mathcal{D}$. The objective of J_2 is to minimize the intra-class variations and maximize the inter-class variations, α is the parameter to balance these two parts in this term. The aims of J_1, J_3, and J_4 are the same as those of the DH method.

Learning parameters $\{\mathbf{W}^m, \mathbf{c}^m\}_{m=1}^M$ by solving the optimization problems through spectral relaxation and stochastic gradient descent is described in our paper. Experimental results on three widely used datasets showed the effectiveness of the proposed methods.

[1] Ritendra Datta, Dhiraj Joshi, Jia Li, and James Z Wang. Image retrieval: Ideas, influences, and trends of the new age. *ACM CSUR*, 40(2):5, 2008.

[2] Yunchao Gong and Svetlana Lazebnik. Iterative quantization: A procrustean approach to learning binary codes. In *CVPR*, pages 817–824, 2011.

[3] Junfeng He, Wei Liu, and Shih-Fu Chang. Scalable similarity search with optimized kernel hashing. In *KDD*, pages 1129–1138, 2010.

[4] Xianglong Liu, Junfeng He, Bo Lang, and Shih-Fu Chang. Hash bit selection: a unified solution for selection problems in hashing. In *CVPR*, pages 1570–1577, 2013.

[5] Christoph Strecha, Alexander M Bronstein, Michael M Bronstein, and Pascal Fua. Ldahash: Improved matching with smaller descriptors. *PAMI*, 34(1):66–78, 2012.

[6] Jun Wang, Sanjiv Kumar, and Shih-Fu Chang. Semi-supervised hashing for scalable image retrieval. In *CVPR*, pages 3424–3431, 2010.

[7] Yair Weiss, Antonio Torralba, and Rob Fergus. Spectral hashing. In *NIPS*, pages 1753–1760, 2008.