Hierarchical Deep Hashing for Fast Large Scale Image Retrieval

Yongfei Zhang, Cheng Peng, Jingtao Zhang, Xianglong Liu, Shiliang Pu, Changhuai Chen

Beihang University, Beijing, China & Hikvision, Hangzhou, China

Jan.10-15, 2021, Milan, Italia

Introduction

Large-scale multimedia data has been pervasive in search engines and social networks. To efficiently find images in such huge amount of data, fast and accurate retrieval methods are necessary.

However, the efficiency of existing deep hashing schemes is relatively low, since only single level hash codes structure is used, which requires a linear traversal and distance computation of all hash codes of gallery images to a query image during retrieval. Thus, the complexity grows linearly with the scales of the image database, which is very time-consuming and even intractable for very large scale applications..

To address the above problems, we propose a hierarchical deep hashing scheme HDHash for large scale image retrieval. The contributions of the paper are four folds.

➢ We proposed a hierarchical deep hashing scheme HDHash, which can generate hierarchical deep hash codes of multiple levels end-to-end and index with tree structures rather than linear ones.

➢With the generated tree-structured hierarchical hash codes, coarse-to-fine search with irrelevant branches pruning can sharply decrease the retrieval time.

➢ The proposed HDHash scheme can also significantly reduce the memory to store the indexes of the gallery images.

➢ Extensive experiments on benchmark datasets demonstrate our proposed HDHash scheme achieves better or comparable accuracy with significantly improved efficiency and reduced memory as compared to ten state-of-the-art image retrieval schemes.

Proposed Hierarchical Deep Hashing Scheme HDHash

1. System Overview

The system architecture of the proposed hierarchical deep hashing-based image retrieval framework is composed of two main components.

➢ The first component is the supervised training of hierarchical deep Hashing Network HDHash, in which the state-of-the-art deep hashing networks HashNet is adopted as the baseline network and multiple HashNets are hierarchically organized and trained on the training dataset.

➢ Given an query image, the second component first generates the corresponding hierarchical hash codes through the trained HDHash and then retrieves images similar to the query from the tree-structured hash codes database via hierarchical coarse-to-fine searching.

2 Supervised Training of HDHash

As can be easily seen from Fig.1, HDHash is a multi-level hierarchically organized network of the baseline network HashNet, which are trained one by one. More specifically, the first HashNet, HashNet 1 at the first level of the HDHash, is first trained exactly the same way as HashNet, except for the difference of output sizes, until the Network converges.

Then, the second HashNet, HashNet 2 at the second level of the HDHash, is trained with fixed HashNet 1 as input and terminate when it converges. The network is designed in such a way to enforce the second HashNet learn the residual information lost in the first level HDHash. It can be proved using conditional probability theory that with sufficient level of HashNet, the residuals in previous level could be compensated. The step-by-step training of HashNet networks continues until the K-th, i.e., the last level HashNet is trained.

The loss function and training algorithm is similar to that of the HashNet. The only difference is that HDHash is trained in multiple stages, level-by-level, with each stage training a different level.

$$L_{HashNet_i} = \sum_{s_{jk}\in S} \alpha_{jk}(log(1+exp(\beta < h_{i,j}, h_{i,k} >)) - \beta s_{jk} < h_{i,j}, h_{i,k} >)$$

More generally, the loss function of the proposed HDHash scheme can also be formulated as a weighted sum of the losses of the HashNet in different levels. In this case, instead of level-by-level training, one can train the whole HDHash network in one stage for fast convergence.

$$L_{HDHash} = \sum_{i=1}^{K} w_i L_{HashNet_i}$$

$$w_i >= w_j, if\ i < j,\ \forall\, i, j = 1, 2, ..., K.$$

where wi denotes the weight of the hash codes of i-th hierarchical level. Since the hash codes in lower levels are usually more important, the weights of the hash codes wi should be non-increasing as the hierarchical level increases, as follows.

When the HDHash network are trained, all the gallery images are fed to generated the corresponding hierarchical hash codes, which are then stored in a tree-structured database.

3 HDHash-based Hierarchical Coarse-to-fine Retrieval

Given a query image, corresponding hierarchical hash codes are first generated by the trained HDHash. Then, hamming distance-based similarity are computed level-by-level. This can be regarded as a coarse-to-fine process. To accelerate the HDHash-based hierarchical search, a filtering mechanism is adopted in each level. More specifically, only a small portion of the closest hash codes are kept for similarity computation in the subsequent levels while most dissimilar hash codes are excluded to speedup the search process. In this paper, a constant filter coefficient p, characterizing the ratio of similar hashes to the total number left during each level, is adopted for simplicity. The smaller the filter coefficient p is, the more hash codes are excluded in the previous level of similarity checking and the less hash codes are left for the subsequent similarity calculation, the faster the retrieval would be. On the contrary, the larger the filter coefficient p is, the more hash codes are queried during filtering and the slower the speed. Extremely, when the filter coefficient p = 1, i.e., no hash codes are excluded in each level, the HDHash will degraded to the one level HashNet.

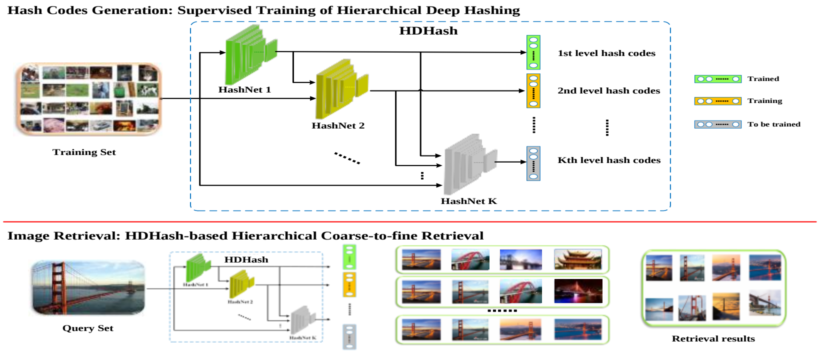

Fig. 1 System Architecture

Performance Evaluation and Comparison

➢ **Datasets:**

- ImageNet [24]
- NUSWIDE[25]
- MS COCO [26]

➢ **Comparable Schemes:**

- Deep hashing: CNNH [4],DNNH [5], DHN[3], HashNet [1]
- Supervised shallow hashing: KSH [10] and SDH [11]
- Classical unsupervised hashing: LSH [2], SH [9]
- Quantization-based retrieval: ITQ [14] and OPQ[21]

➢ **Comparison metrics:**

- Retrieval precision: Mean Average Precision (MAP)
- Retrieval efficiency: Search speed & Memory requirement

➢ **Experimental Results:**

TABLE I

Comparison of Map for HDHASH and Comparable Schemes on Three Benchmark datasets

Method		HDHash	HashNet [1]	DHN [3]	DNNH [5]	SDH [11]
ImageNet	16bits	0.4902	**0.5059**	0.3106	0.2903	0.2985
	32bits	0.6253	**0.6306**	0.4717	0.4605	0.4551
	48bits	0.6530	**0.6633**	0.5420	0.5301	0.5549
	64bits	0.6735	**0.6835**	0.5732	0.5645	0.5852
NUS-WIDE	16bits	**0.6626**	0.6623	0.6374	0.5976	0.4756
	32bits	0.6953	**0.6988**	0.6637	0.6158	0.5545
	48bits	0.7098	**0.7114**	0.6690	0.6345	0.5786
	64bits	**0.7186**	0.7163	0.6714	0.6388	0.5812
MS COCO	16bits	0.6831	**0.6873**	0.6774	0.5932	0.5545
	32bits	**0.7186**	0.7184	0.7013	0.6034	0.5642
	48bits	0.7291	**0.7301**	0.6948	0.6045	0.5723
	64bits	0.7301	**0.7362**	0.6944	0.6099	0.5799

Method		CNNH [4]	KSH [10]	SH [9]	LSH [2]	ITQ [14]	OPQ [21]
ImageNet	16bits	0.2812	0.1599	0.2066	0.1007	0.3255	–
	32bits	0.4498	0.2976	0.3280	0.2350	0.4620	0.41
	48bits	0.5245	0.3422	0.3951	0.3121	0.5170	–
	64bits	0.5538	0.3943	0.4191	0.3596	0.5520	0.48
NUS-WIDE	16bits	0.5696	0.3561	0.4058	0.3283	0.5086	–
	32bits	0.5827	0.3327	0.4209	0.4227	0.5425	0.52
	48bits	0.5926	0.3124	0.4211	0.4333	0.5580	–
	64bits	0.5996	0.3368	0.4104	0.5009	0.5611	0.60
MS COCO	16bits	0.5642	0.5212	0.4951	0.4592	0.5818	–
	32bits	0.5744	0.5343	0.5071	0.4856	0.6243	0.69
	48bits	0.5711	0.5343	0.5099	0.5440	0.6460	–
	64bits	0.5671	0.5361	0.5101	0.5849	0.6574	0.71

TABLE II

Comparison of Retrieval Efficiency and Memory Requirements

Method	search time (ms)	memory usage (MB)
HDHash(0.3)	**12.226**	**223**
HashNet [1]	60	1529
speedup (HashNet [1]/HDHash(0.3))	**4.91**	**6.86**

Conclusions

This paper proposed a novel hierarchical deep hashing scheme HDHash to speed up the state-of-the-art deep hashing methods for fast large scale image retrieval.

➢ Through careful design of the hierarchical HDHash network and the loss functions, multi-level tree-structured hash codes could be generated end-to-end, based on which the coarse-to-fine retrieval can be conducted.

➢ Extensive experimental results on three benchmark datasets have demonstrated that the proposed HDHash scheme achieves better or comparable accuracy with significantly improved efficiency and reduced memory as compared to state-of-the-art fast image retrieval schemes.

➢ These performance gain could be even enhanced with more hierarchical levels in large scale image retrieval scenarios and further optimization of the tree-based index structure.

➢ Besides, this proposed idea of hierarchical deep hashing could also be further applied to other feature extraction, indexing and similarity computations scenario, to further enhance the performance.

For Further Information

Please contact Dr. Yongfei Zhang (yfzhang@buaa.edu.cn, 0086-13811424077) for more information.