北京航空航天大學
BEIHANG UNIVERSITY

ICPR 2020

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

Outlines

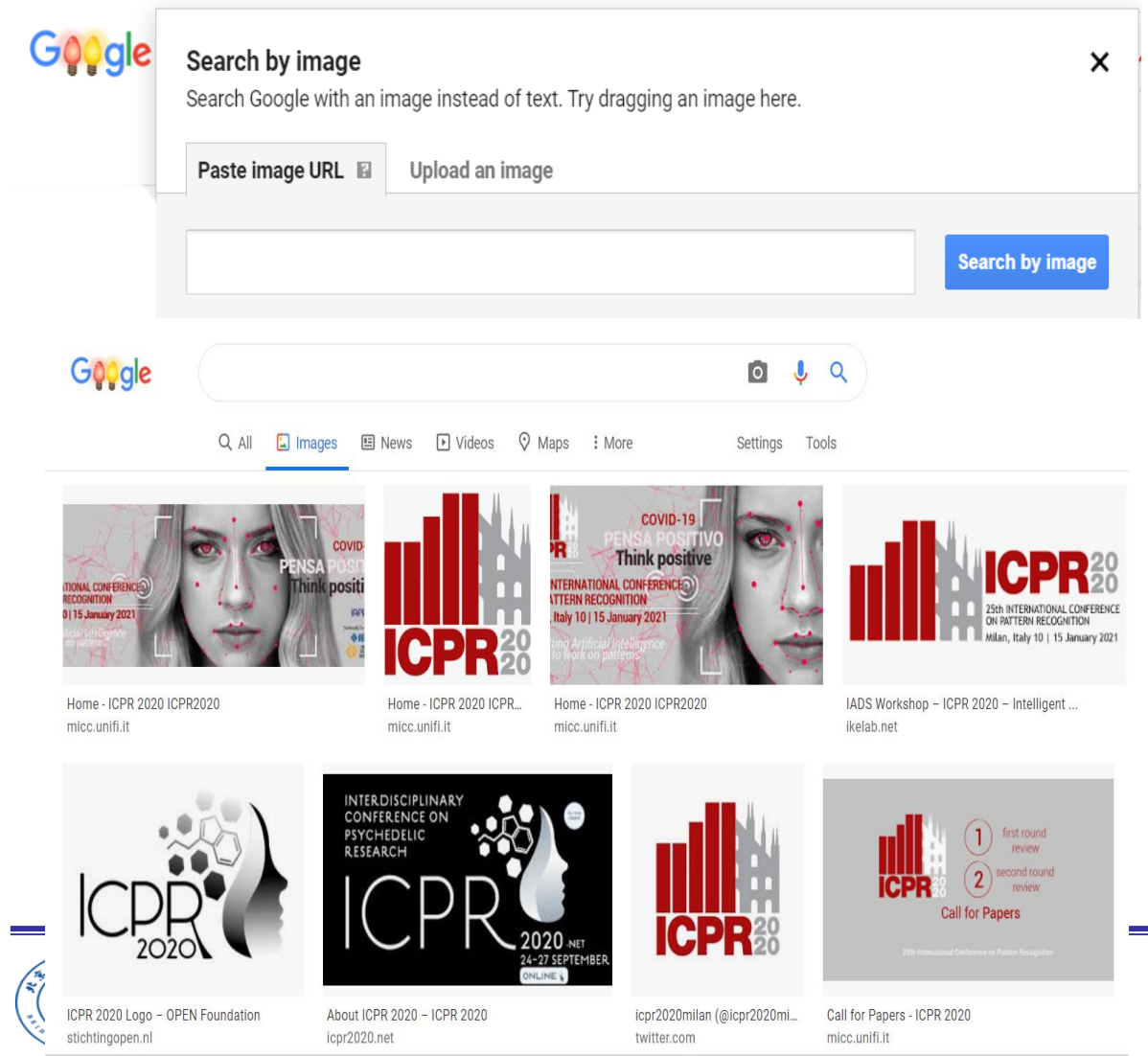
- **Background & Motivations**
- **Proposed Hierarchical Deep Hashing Scheme HDHash**
  - **System Overview**
  - **Supervised Training of Hierarchical Deep Hashing**
  - **HDHash-based Hierarchical Coarse-to-fine Retrieval**
- **Performance Evaluation**
- **Conclusions and Remarks**

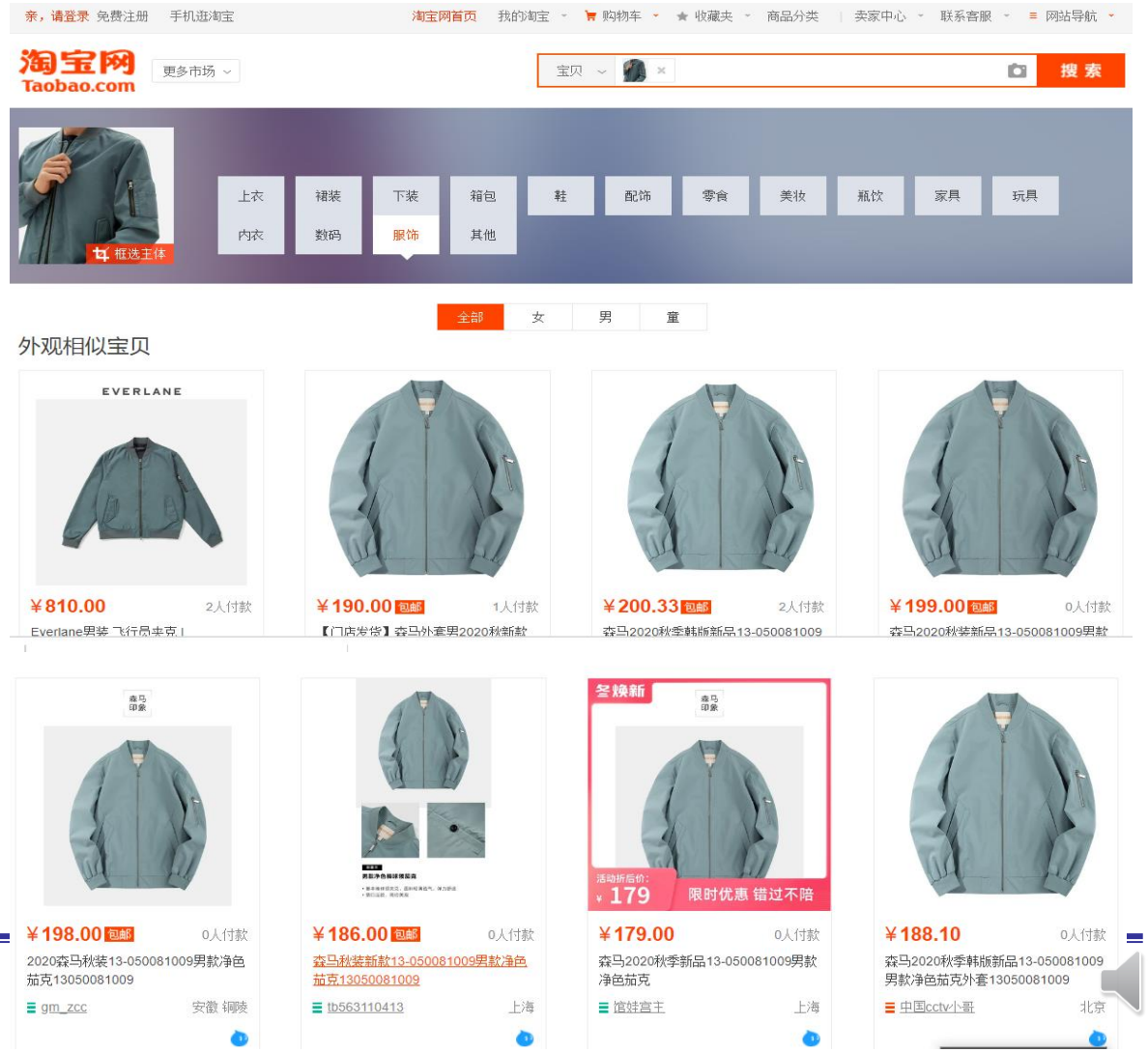
Outlines

- **Background & Motivations**
- **Proposed Hierarchical Deep Hashing Scheme HDHash**
  - **System Overview**
  - **Supervised Training of Hierarchical Deep Hashing**
  - **HDHash-based Hierarchical Coarse-to-fine Retrieval**
- **Performance Evaluation**
- **Conclusions and Remarks**

1. Background & Motivations

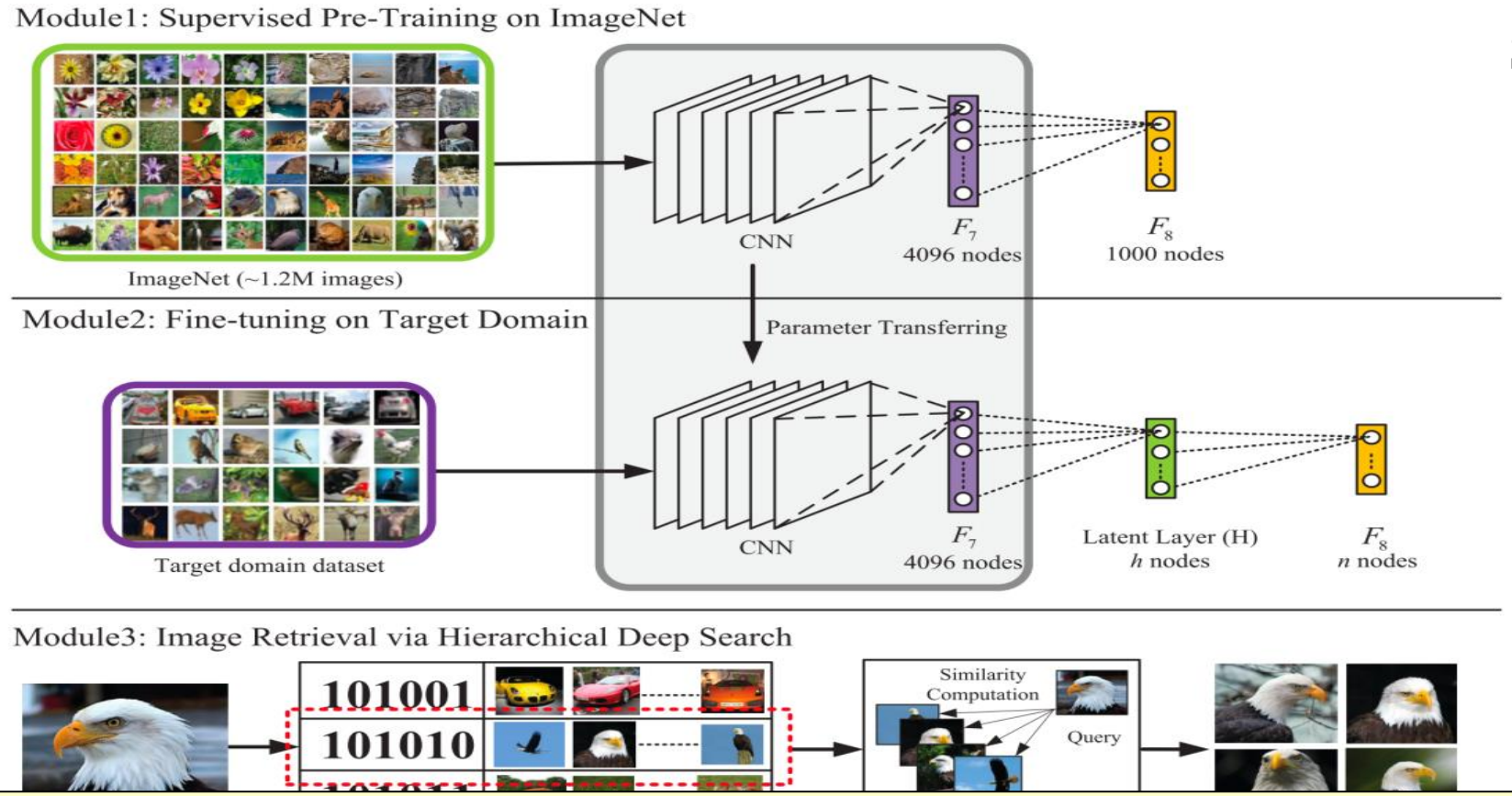
- **Image Retrieval: supporting techniques for various applications**





1. Background & Motivations

- **Related Works: Deep Hashing-HashNet[1]**



Single-level hash codes

linear traversal and distance computation

Low retrieval efficiency

Not good for large

Idea: Hierarchical Deep Hashing + Course-to-fine Retrieval

Outlines

- Background & Motivations
- Proposed Hierarchical Deep Hashing Scheme HDHash
  - System Overview
  - Supervised Training of Hierarchical Deep Hashing
  - HDHash-based Hierarchical Coarse-to-fine Retrieval
- Performance Evaluation
- Conclusions and Remarks

2.1 System Overview

Hash Codes Generation: Supervised Training of Hierarchical Deep Hashing

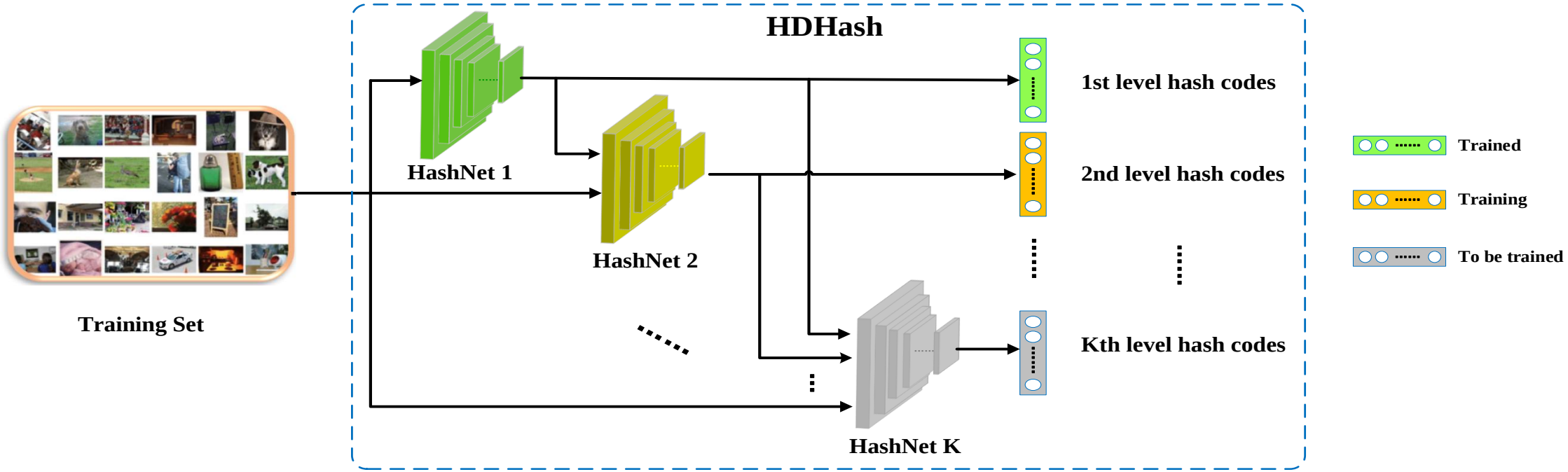
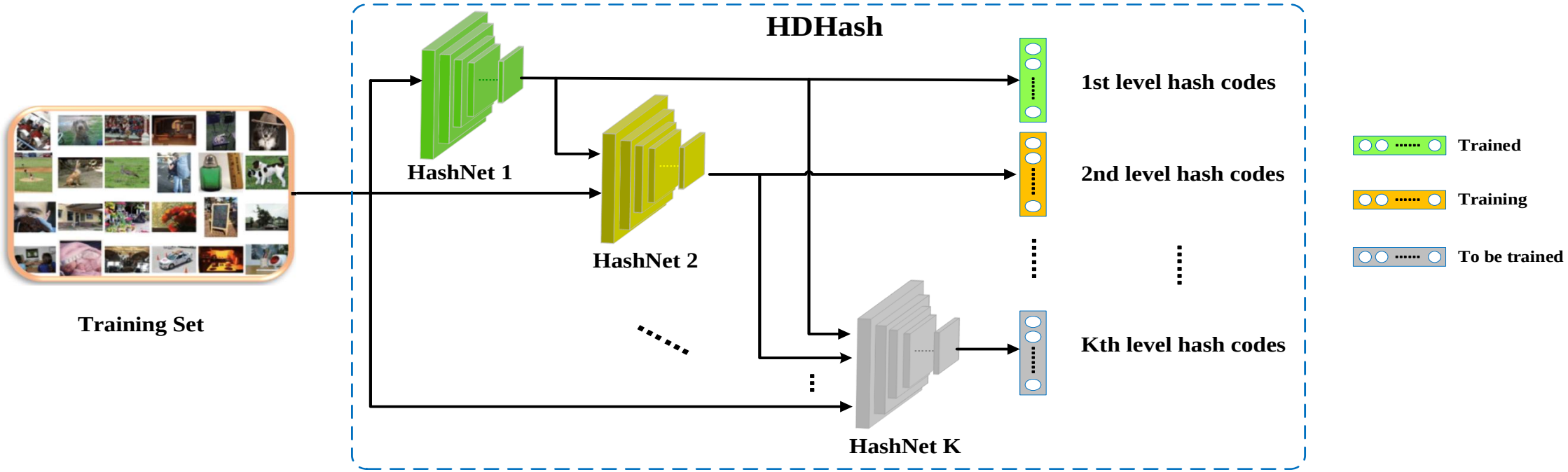


Image Retrieval: HDHash-based Hierarchical Coarse-to-fine Retrieval

2.2 Supervised Training of Hierarchical Deep Hashing

- **Hash Codes Generation: Supervised Training of Hierarchical Deep Hashing**



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

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

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

2.3 HDHash-based Hierarchical Coarse-to-fine Retrieval

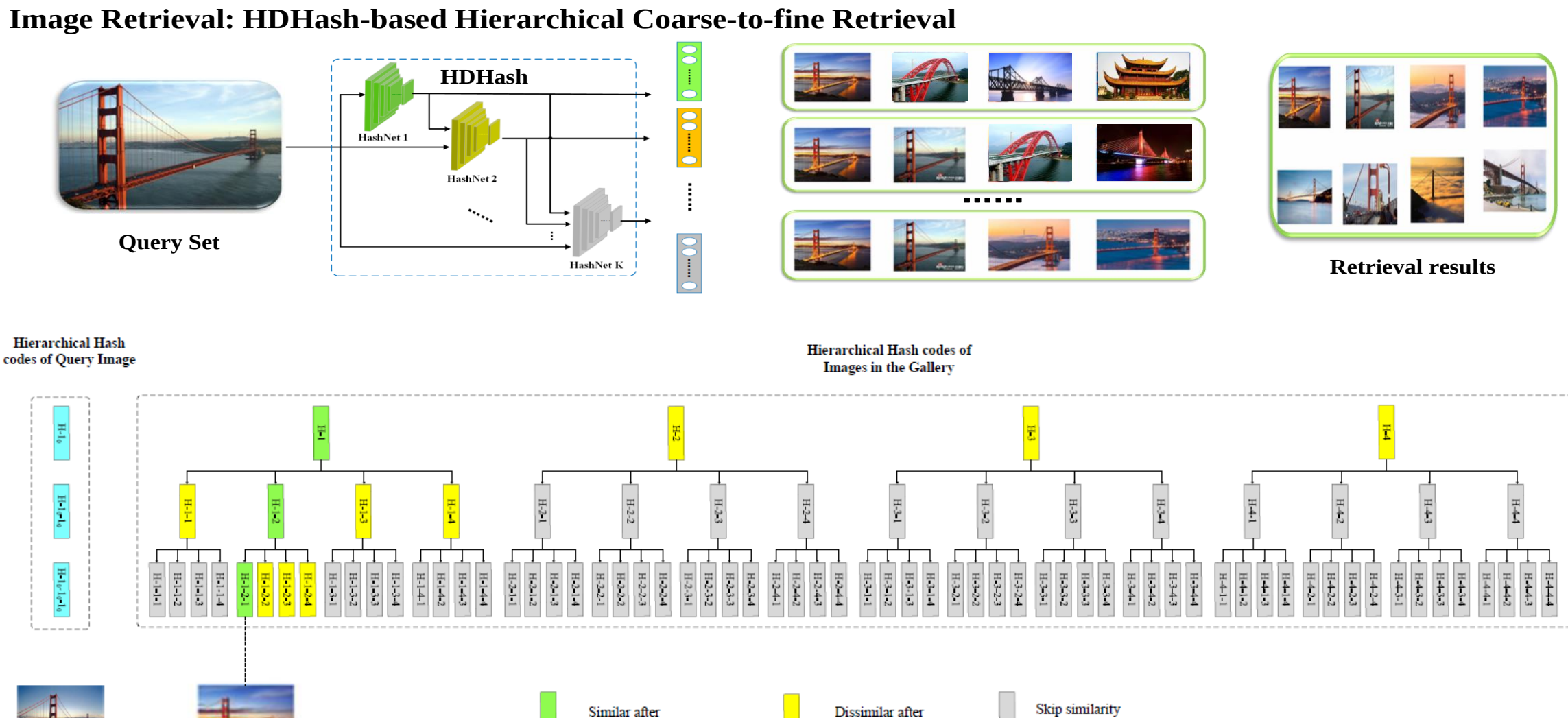


Fig. 2. Example of Tree-structured Indexing and Hierarchical Image Retrieval of the Proposed HDHash Scheme.

Outlines

- **Background & Motivations**
- **Proposed Hierarchical Deep Hashing Scheme HDHash**
  - **System Overview**
  - **Supervised Training of Hierarchical Deep Hashing**
  - **HDHash-based Hierarchical Coarse-to-fine Retrieval**
- **Performance Evaluation**
- **Conclusions and Remarks**

3. Performance Evaluation

- **Datasets:**
  - **ImageNet [24],**
  - **NUSWIDE[25]**
  - **MS COCO [26]**
- **Comparable Schemes:**
  - **SOTA deep hashing: CNNH [4], DNNH[5], DHN [3], HashNet [1]**
  - **Supervised shallow hashing schemes: KSH [10] and SDH [11]**
  - **Classical unsupervised hashing schemes: LSH [2], SH [9]**
  - **Quantization-based retrieval algorithms: ITQ [14] and OPQ[21]**

3. Performance Evaluation

- **Comparison metrics**
  - **Retrieval precision**
    - **Mean Average Precision (MAP)**
  - **Retrieval efficiency**
    - **Search speed**
    - **Memory requirement**

3. Performance Evaluation

- **Mean Average Precision (MAP)**

TABLE I
COMPARISON OF MAP FOR HDHASH AND COMPARABLE SCHEMES ON THREE DATASETS

Method		HDHash	HashNet [1]	DHN [3]	DNNH [5]	SDH [11]	CNNH [4]	KSH [10]	SH [9]	LSH [2]	ITQ [14]	OPQ [21]
ImageNet	16bits	0.4902	**0.5059**	0.3106	0.2903	0.2985	0.2812	0.1599	0.2066	0.1007	0.3255	–
	32bits	0.6253	**0.6306**	0.4717	0.4605	0.4551	0.4498	0.2976	0.3280	0.2350	0.4620	0.41
	48bits	0.6530	**0.6633**	0.5420	0.5301	0.5549	0.5245	0.3422	0.3951	0.3121	0.5170	–
	64bits	0.6735	**0.6835**	0.5732	0.5645	0.5852	0.5538	0.3943	0.4191	0.3596	0.5520	0.48
NUS-WIDE	16bits	**0.6626**	0.6623	0.6374	0.5976	0.4756	0.5696	0.3561	0.4058	0.3283	0.5086	–
	32bits	0.6953	**0.6988**	0.6637	0.6158	0.5545	0.5827	0.3327	0.4209	0.4227	0.5425	0.52
	48bits	0.7098	**0.7114**	0.6690	0.6345	0.5786	0.5926	0.3124	0.4211	0.4333	0.5580	–
	64bits	**0.7186**	0.7163	0.6714	0.6388	0.5812	0.5996	0.3368	0.4104	0.5009	0.5611	0.60
MS COCO	16bits	0.6831	**0.6873**	0.6774	0.5932	0.5545	0.5642	0.5212	0.4951	0.4592	0.5818	–
	32bits	**0.7186**	0.7184	0.7013	0.6034	0.5642	0.5744	0.5343	0.5071	0.4856	0.6243	0.69
	48bits	0.7291	**0.7301**	0.6948	0.6045	0.5723	0.5711	0.5343	0.5099	0.5440	0.6460	–
	64bits	0.7301	**0.7362**	0.6944	0.6099	0.5799	0.5671	0.5361	0.5101	0.5849	0.6574	0.71

3. Performance Evaluation

- **Search speed & Memory requirement**

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**

- **Note: 3-level hash codes of 8+16+40bits for HDHash and 1-level codes of 64 bits for HashNet**

Outlines

- **Background & Motivations**
- **Proposed Hierarchical Deep Hashing Scheme HDHash**
  - **System Overview**
  - **Supervised Training of Hierarchical Deep Hashing**
  - **HDHash-based Hierarchical Coarse-to-fine Retrieval**
- **Performance Evaluation**
- **Conclusions and Remarks**

Conclusions and Remarks

- **A novel hierarchical deep hashing scheme HDHash to speed up the state-of-the-art deep hashing methods for fast large scale image retrieval**
- **Multi-level tree-structured hash codes could be generated end-to-end, based on which the coarse-to-fine retrieval can be conducted.**
- **HDHash achieves better or comparable accuracy with significantly improved efficiency and reduced memory as compared to SOTA fast image retrieval schemes.**
- **Could be enhanced with more hierarchical levels and further optimization of the tree-based index structure**
- **Could also be further applied to other feature extraction, indexing and similarity computations scenario, to further enhance the performance**

THANK YOU!

Questions/Comments?

Yongfei Zhang, Associate Professor

Tel: +86-13811424077(Beijing, China)/(213)284-4430 (Los Angeles, CA, US)

Email: yfzhang@buaa.edu.cn / yz_515@usc.edu