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
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


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

\[ L_{\text{HashNet}_i} = \sum_{s_{jk} \in S} \alpha_{jk} \left( \log(1 + \exp(\beta < h_{i,j}, h_{i,k} >)) - \beta s_{jk} < h_{i,j}, h_{i,k} > \right) \]

\[ L_{\text{HDHash}} = \sum_{i=1}^{K} w_i L_{\text{HashNet}_i} \]

\[ w_i \geq w_j, \text{if } i < j, \quad \forall i, j = 1, 2, \ldots, K. \]
2.3 HDHash-based Hierarchical Coarse-to-fine Retrieval

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

Hash Codes Generation: Supervised Training of Hierarchical Deep Hashing

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

Training Set

Query Set

Retrieval results

Hierarchical Hash codes of Query Image

Hierarchical Hash codes of Images in the Gallery

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**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ImageNet</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16bits</td>
<td>0.4902</td>
<td>0.5059</td>
<td>0.3106</td>
<td>0.2903</td>
<td>0.2812</td>
<td>0.1599</td>
<td>0.2066</td>
<td>0.1007</td>
<td>0.3255</td>
<td></td>
<td></td>
</tr>
<tr>
<td>32bits</td>
<td>0.6253</td>
<td>0.6306</td>
<td>0.4717</td>
<td>0.4605</td>
<td>0.4551</td>
<td>0.4498</td>
<td>0.2976</td>
<td>0.3280</td>
<td>0.2350</td>
<td>0.4620</td>
<td>0.41</td>
</tr>
<tr>
<td>48bits</td>
<td>0.6530</td>
<td>0.6633</td>
<td>0.5420</td>
<td>0.5301</td>
<td>0.5549</td>
<td>0.5245</td>
<td>0.3422</td>
<td>0.3951</td>
<td>0.3121</td>
<td>0.5170</td>
<td></td>
</tr>
<tr>
<td>64bits</td>
<td>0.6735</td>
<td>0.6835</td>
<td>0.5732</td>
<td>0.5645</td>
<td>0.5852</td>
<td>0.5538</td>
<td>0.3943</td>
<td>0.4191</td>
<td>0.3596</td>
<td>0.5520</td>
<td>0.48</td>
</tr>
<tr>
<td><strong>NUS-WIDE</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16bits</td>
<td>0.6626</td>
<td>0.6623</td>
<td>0.6374</td>
<td>0.5976</td>
<td>0.4756</td>
<td>0.5696</td>
<td>0.3561</td>
<td>0.4058</td>
<td>0.3283</td>
<td>0.5086</td>
<td></td>
</tr>
<tr>
<td>32bits</td>
<td>0.6953</td>
<td>0.6988</td>
<td>0.6637</td>
<td>0.6158</td>
<td>0.5545</td>
<td>0.5827</td>
<td>0.3327</td>
<td>0.4209</td>
<td>0.4227</td>
<td>0.5425</td>
<td>0.52</td>
</tr>
<tr>
<td>48bits</td>
<td>0.7098</td>
<td>0.7114</td>
<td>0.6690</td>
<td>0.6345</td>
<td>0.5786</td>
<td>0.5926</td>
<td>0.3124</td>
<td>0.4211</td>
<td>0.4333</td>
<td>0.5580</td>
<td></td>
</tr>
<tr>
<td>64bits</td>
<td>0.7186</td>
<td>0.7163</td>
<td>0.6714</td>
<td>0.6388</td>
<td>0.5812</td>
<td>0.5996</td>
<td>0.3368</td>
<td>0.4104</td>
<td>0.5009</td>
<td>0.5611</td>
<td>0.60</td>
</tr>
<tr>
<td><strong>MS COCO</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16bits</td>
<td>0.6831</td>
<td>0.6873</td>
<td>0.6774</td>
<td>0.5932</td>
<td>0.5545</td>
<td>0.5642</td>
<td>0.5212</td>
<td>0.4951</td>
<td>0.4592</td>
<td>0.5818</td>
<td></td>
</tr>
<tr>
<td>32bits</td>
<td>0.7186</td>
<td>0.7184</td>
<td>0.7013</td>
<td>0.6034</td>
<td>0.5642</td>
<td>0.5744</td>
<td>0.5343</td>
<td>0.5071</td>
<td>0.4856</td>
<td>0.6243</td>
<td>0.69</td>
</tr>
<tr>
<td>48bits</td>
<td>0.7291</td>
<td>0.7301</td>
<td>0.6948</td>
<td>0.6045</td>
<td>0.5723</td>
<td>0.5711</td>
<td>0.5343</td>
<td>0.5099</td>
<td>0.5440</td>
<td>0.6460</td>
<td></td>
</tr>
<tr>
<td>64bits</td>
<td>0.7301</td>
<td>0.7362</td>
<td>0.6944</td>
<td>0.6099</td>
<td>0.5799</td>
<td>0.5671</td>
<td>0.5361</td>
<td>0.5101</td>
<td>0.5849</td>
<td>0.6574</td>
<td>0.71</td>
</tr>
</tbody>
</table>
3. Performance Evaluation

• Search speed & Memory requirement

<table>
<thead>
<tr>
<th>Method</th>
<th>search time (ms)</th>
<th>memory usage (MB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>HDHash(0.3)</td>
<td>12.226</td>
<td>223</td>
</tr>
<tr>
<td>HashNet [1]</td>
<td>60</td>
<td>1529</td>
</tr>
<tr>
<td>speedup (HashNet [1]/HDHash(0.3))</td>
<td>4.91</td>
<td>6.86</td>
</tr>
</tbody>
</table>

• **Note:** 3-level hash codes of 8+16+40 bits 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