O1-2-4
17th International Conference on Machine Vision Applications (MVA)
Fully Online, July 25–27, 2021.

Semantic Hierarchy Preserving Deep Hashing for Large-Scale Image Retrieval

Ming Zhang
City University of Hong Kong
Hong Kong, China
mzhang367-c@my.cityu.edu.hk

Xuefei Zhe
Tencent AI Lab
Shenzhen, China
elizhe@tencent.com

Le Ou-Yang
Shenzhen University
Shenzhen, China
leouyang@szu.edu.cn

Shifeng Chen
Shenzhen Institutes of Advanced Technology, CAS
Shenzhen, China
shifeng.chen@siat.ac.cn

Hong Yan
City University of Hong Kong
Hong Kong, China
h.yan@cityu.edu.hk

Abstract

Deep hashing models have been proposed as an efficient method for large-scale similarity search. However, most existing deep hashing methods only utilize fine-level labels for training while ignoring the natural semantic hierarchy structure. This paper presents an effective method that preserves the classwise similarity of full-level semantic hierarchy for large-scale image retrieval. Experiments on two benchmark datasets show that our method helps improve the fine-level retrieval performance. Moreover, with the help of the semantic hierarchy, it can produce significantly better binary codes for hierarchical retrieval, which indicates its potential of providing more user-desired retrieval results. The codes are available at `https://github.com/mzhang367/hpdh.git`.

1 Introduction

The past few years have witnessed a significant improvement in the quality of content-based image retrieval (CBIR) [1, 2, 3] due to the emergence of deep learning. With the explosive growth of online visual data, there is an urgent need to develop more efficient deep learning models. Recently, deep hashing has been proposed as a promising method for large-scale image retrieval. It directly projects images to binary codes for approximate nearest neighbor search, which considerably reduces the storage and computation cost.

In the real world, the classification and description of things often follow a hierarchy structure. One typical example is taxonomy, as shown in Fig. 1. However, most existing deep hashing models [4, 5, 6, 7] only utilize single-level semantic labels for training. Their training processes are either supervised with the fine-level labels or similar/dissimilar pairs of labels that are converted from fine-level labels. With such supervision, the deep hashing model can only learn partly class similarity within the lowest hierarchy while the class similarity between the upper-level labels is not well preserved as the semantic hierarchy structure. Consequently, it hinders the deep hashing model from learning a better semantic hashing space for hierarchical retrieval. Taking Fig. 1 as an example. Without hierarchy, bears' feature embeddings are not necessarily closer to giant pandas belonging to the same super-class (Ursidae) than that to other species belonging to different super-class.

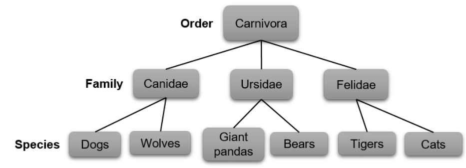

Figure 1: A tree visualization of three-level hierarchy comprising six kinds of mammals. Note that we refer the Species level to fine-level and the Order level to the highest level, which is analogous to the leaf node and the root node in the hierarchy tree, respectively.

In this work, we propose a novel deep hashing method called Hierarchy Preserving Deep Hashing (HPDH). Fig. 2 illustrates a comparison of the fine-level labels-based Deep Class-Wise Hashing (DCWH) [10] and the proposed method. It is clear that the hashing codes generated by our method fall into an obvious hierarchy structure. The main contribution of HPDH can be summarized in three folds:

- We propose a hierarchical loss function that directly uses class labels for hashing learning. The proposed loss function preserves intra-class compactness and inter-class separability in each hierarchy level.

- To better leverage the label information's hierarchy structure, we design a simple yet efficient

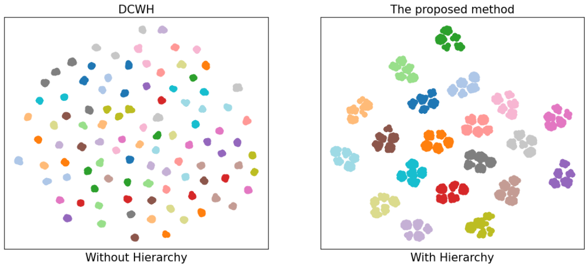

Figure 2: Visualization of learned hashing codes using t-SNE [8]. The samples in CIFAR-100 [9] are labeled with both fine-level and coarse-level labels. We illustrate samples by their coarse-level labels. Each color indicates one coarse-level class. Note each coarse-level class contains exactly five fine-level classes.

scheme to update the class centers per level in a periodical and recursive manner.

- Experiments on two benchmark datasets: CIFAR-100 and NABirds show that our method consistently outperforms other state-of-the-art baselines with a distinct margin on both general fine-level retrieval and hierarchical retrieval.

2 Related Works

Deep hashing methods [6, 10, 11, 12] have shown their great superiority to traditional hashing methods in image retrieval with the advantages of simultaneously feature representations learning and hashing learning. Most existing deep hashing methods adopt the similar/dissimilar pairs of samples for training that are constructed from class labels. Following this direction, DPSH [4] firstly utilizes the pairwise label information to train an end-to-end deep hashing model. HashNet [13] defines a weighted maximum likelihood to balance the similar and dissimilar pairs within the dataset. To make the most of the semantic information, recently, several works propose directly relying on class labels for similarity supervision. SSDH [14] introduces a softmax classifier to unify the classification and retrieval in a single learning process. DCWH [10] designs a normalized Gaussian-based loss, clustering intra-class samples to the corresponding class center.

The aforementioned deep hashing methods propose different learning metrics to conduct similarity learning. However, most of them cannot handle the multi-level semantic labels. Meanwhile, there appear more and more hierarchical labeled datasets. For example, ImageNet [15] is a large-scale database organized by WordNet [16] hierarchy, and each node of the hierarchy is illustrated by hundreds, even thousands of images. Therefore, just considering the similarity within a single level of labels inevitably leads to a loss of the rich semantic information stored in the hierarchical data.

It has been verified that providing the hierarchy information relating to semantic labels during training can boost image retrieval performance. In [17], researchers prove that the image classification performance can be improved by combining the coarse and fine-level labels. A similar idea is shared in [18], where a hierarchical training strategy is applied to handle the face recognition task. Recently, SHDH [19] is the first deep hashing work that tackles the hierarchy similarity by weighting the Hamming distance at each level. However, SHDH applies a pairwise labels relation, which is inferior to class-wise labels-based methods [14, 10, 20]. Motivated by these issues, we introduce the hierarchy -preserving method HPDH based on the class-wise label information in each hierarchy, suitable for learning more discriminative binary codes.

3 Proposed Approach

Suppose a hierarchical labeled dataset, and each image x_i is annotated with a K-level semantic label, denoted with a K-dimensional vector y_i. The label vector consists of the class labels from the lowest hierarchy level to the highest hierarchy level in a tree-like structure. For example, a dolphin image is labeled with $y = \{\text{Dolphin, Aquatic Mammals}\}$ in a two-level hierarchy. We aim to apply deep convolutional neural network (CNN) with learnable parameters Θ to project images to a particular Hamming space. In this space, for any hierarchy level, the Hamming distance between intra-class samples to the corresponding class center is smaller than that to other class centers.

We propose a multi-level normalized Gaussian model to keep a hierarchical semantic structure in the Hamming space. Denote the hashing output as $r_i = f(x_i, \Theta)$ and $r_i \in \{-1, 1\}^L$, where L is the binary code length. The proposed objective function is formulated as following:

$$\begin{aligned} \min_{\Theta, M} \mathcal{L} = -\sum_{i=1}^{N}\sum_{k=1}^{K} \log \frac{\exp\{-\frac{1}{2\sigma_k^2} d(r_i, \mu_{k y_{ik}})\}}{\sum_{j=1}^{C_k} \exp\{-\frac{1}{2\sigma_k^2} d(r_i, \mu_{kj})\}} \\ s.t. \quad r_i = f(x_i, \Theta) \in \{-1,1\}^L, \quad \mu_{kj} \in \{-1,1\}^L \end{aligned} \tag{1}$$

where $M = \{\mu_k\}_{k=1}^K$ and $\mu_k = \{\mu_{kj}\}_{j=1}^{C_k}$. Here, μ_{kj} represents the j-th class centers in the k-th level of the semantic hierarchy, C_k is the total number of classes at the k-th level and σ_k is a parameter to control the intra-class variance at the level k. $d(\cdot, \cdot)$ is the Hamming distance function. To solve this discrete optimization problem, we follow the optimization strategy in DCWH [10]. We first relax the r_i to $[-\alpha, \alpha]$ where α is empirically set to 1.1 in [10]. Then, the original distance $d(\cdot, \cdot)$ can be replaced with a Euclidean distance.

And the loss function becomes the following:

$$\mathcal{L} = -\sum_{i=1}^{N}\sum_{k=1}^{K}\log\frac{\exp\{-\frac{1}{2\sigma_k^2}\|r_i - \mu_{ky_{ik}}\|\}}{\sum_{j=1}^{C_k}\exp\{-\frac{1}{2\sigma_k^2}\|r_i - \mu_{kj}\|\}} + \eta_1\{ReLU(-\alpha - r_n) + ReLU(r_n - \alpha)\} \quad (2)$$

where η_1 is the regularization weight. $ReLU$ is the rectified linear unit defined as $ReLU(x) = \max(0, x)$. The above loss function is differentiable and the classical back-propagation can be applied to optimize the network parameters Θ.

We update the fine-level class centers $\{\mu_{1j}\}$ with the training data as following:

$$\mu_{1j} = \frac{1}{N_{1j}}\sum_{n=1}^{N_{1j}} f(x_n, \Theta) \quad (3)$$

where N_{1j} is the number of images that belong to the j-th class in the lowest hierarchy level in the whole dataset. Based on $\{\mu_{1j}\}$, the upper-level class centers $\{\mu_{kj}\}_{k=2}^{K}$ can be calculated from their own child level class centers as following:

$$\mu_{kj} = \frac{1}{C_{kj}}\sum_{c=1}^{C_{kj}} \mu_{(k-1),c} \quad (4)$$

where C_{kj} is the number of level-(k-1) classes with the same parent class, i.e., j-th class at level k. By such a recursive calculation, not only can we save the computation cost but also eliminate the influence of imbalanced training data between classes. The hyper parameters $\{\sigma_k^2\}$ are chosen to satisfy the criterion $\sigma_{k-1}^2 \leq \sigma_k^2$ so that the variances within parent classes are larger than the child ones. Finally, we introduce a quantization term following [6] to encourage the relaxed real-valued hashing outputs to be binary. The finalized objective function is shown as:

$$\min_{\Theta, M} \mathcal{J} = \mathcal{L} + \eta_2\sum_{i=1}^{N}\|b_i - r_i\|_2^2 \quad (5)$$

where $b_i = \mathbf{sgn}(r_i)$ and η_2 is a hyper parameter controlling the weight of the quantization term. The whole training procedure is summarized in Algorithm 1.

4 Experiments

4.1 Datasets

We conduct experiments on two hierarchical datasets including CIFAR-100 [9] and NABirds [21]. CIFAR-100 is a dataset collecting $60,000$ tinny images with a two-level semantic hierarchy, i.e., $K = 2$. Specifically, the total 100 fine-level classes are grouped into 20 coarse categories, and each coarse category contains exactly 5 fine-level classes. We follow the official split with $50,000$ images for training and $10,000$ images for testing. NABirds [21] dataset comprises more than $48,000$ images from 555 visual categories of North America birds species. These categories are organized taxonomically into a four-level hierarchy (excluding the root node "bird" that all images belong to, which does not provide any information gain). We use the official split that has $23,929$ images and $24,633$ images for training and testing, respectively. In the training procedure, we directly resize the original images in two datasets to 240×240 and then randomly crop them to 224×224 as inputs of the network. For both datasets, the training set serves as the database, and the testing images are used as queries during the testing phase.

Algorithm 1 The training procedure of HPDH

Initialize CNN parameters Θ, class centers M
repeat
1. Compute features $r_i = f(x_i, \Theta)$;
2. Update fine-level centers $\{\mu_{1j}\}$ by Eq. (3);
3. Update upper level centers by Eq. (4);
4. Compute the loss $\mathcal{J}$ according to Eq. (5);
5. Calculate derivatives of $\mathcal{J}$ w.r.t. r_i and update Θ by back propagation

until Converge

4.2 Setups and Evaluation Metrics

We compare our method with a series of pairwise and triplet labels-based deep hashing methods, including DPSH [4], DTSH [22], SHDH [19], and class-wise labels-based methods, including DCWH [10], IDCWH [20], and CSQ [23]. For a fair comparison, we apply ResNet-50 [24] pre-trained on the ImageNet [15] in all the methods. For our HPDH, we fine-tune the backbone with a newly added fully-connected hashing layer. We train the whole network for 150 epochs using mini-batch stochastic gradient descent (SGD) with momentum 0.9 and weight decay 5e-4. The initial learning rate is set to 5e-3 and decayed by 0.1 every 50 epochs. The hyper parameters are fixed to $\eta_1 = 10$ and $\eta_2 = 0.1$. We set $\{\sigma_k\} = \{1, 2\}$ for CIFAR-100 and $\{1, 1.5, 2, 4\}$ for NABirds by cross-validation. All the experiments are run on two Nvidia RTX-2080 GPU cards with PyTorch.

We present results under both fine-level and hierarchical-level evaluation metrics. The fine-level retrieval performance is evaluated with the mean average precision (mAP@all). While the hierarchical retrieval performance is reported by mean average Hierarchical Precision (mAHP) [19, 25] and the normalized Discounted Cumulative Gain (nDCG) [26]. Specifically, we adopt mAHP@2,500 for CIFAR-100 which contains exactly $2,500$ images for each coarse category, while mAHP@250 for NABirds dataset following [25].

Table 1: Results on CIFAR-100 dataset

Methods	mAP@all			mAHP@2.5k			nDCG@100		
	32-bit	48-bit	64-bit	32-bit	48-bit	64-bit	32-bit	48-bit	64-bit
DPSH	0.2861	0.3571	0.3952	0.4715	0.5048	0.5212	0.5409	0.6089	0.6442
DTSH	0.6950	0.7269	0.7366	0.6744	0.7019	0.7051	0.7468	0.7625	0.7820
SHDH	-	-	-	-	-	-	0.6141	0.6281	0.6406
DCWH	0.7680	0.8023	0.8178	0.6598	0.6608	0.6698	0.7894	0.8306	0.8439
CSQ	0.7991	0.8032	0.8093	0.5660	0.5677	0.5792	0.8348	0.8376	0.8387
HPDH	**0.8292**	**0.8347**	**0.8534**	**0.8802**	**0.8846**	**0.8974**	**0.8520**	**0.8558**	**0.8718**

Table 2: Results on NABirds dataset

Methods	mAP@all			mAHP@250			nDCG@100		
	32-bit	48-bit	64-bit	32-bit	48-bit	64-bit	32-bit	48-bit	64-bit
DTSH	0.3511	0.3614	0.3764	0.6514	0.6566	0.6634	0.5702	0.5798	0.5902
DCWH	0.4132	0.4882	0.5132	0.5715	0.6400	0.6637	0.5954	0.6551	0.6754
CSQ	0.4419	0.4733	0.5121	0.4557	0.4998	0.5390	0.5445	0.5981	0.6396
IDCWH	0.6811	0.7158	0.7249	0.5857	0.6389	0.6843	0.7228	0.7602	0.7878
HPDH	**0.7014**	**0.7372**	**0.7366**	**0.8512**	**0.8661**	**0.8560**	**0.7987**	**0.8160**	**0.8040**

We provide results nDCG@100 for easy comparison with [19]. We refer readers to [27] for more details about hierarchical precision (HP).

4.3 Results and Analysis

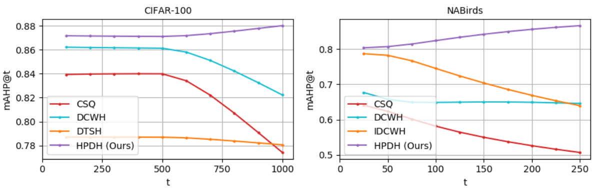

Figure 3: The comparison on mAHP results w.r.t. different cutoff points t on two datasets.

We present results on CIFAR-100 in Table 1. We can find, for general retrieval performance, the proposed HPDH performs better than two state-of-the-art, i.e., DCWH and CSQ. The mAP of HPDH is 3.0% and 3.6% higher than that of the second place at 32-bit and 64-bit, respectively. When it comes to hierarchical retrieval, our method surpasses the previous best one with a large margin: it obtains 88.74% mAHP scores on average, which is 19.4% higher than DTSH's 69.38%. Note the distinct performance drop from mAP to mAHP for DCWH and CSQ. In contrast, our HPDH utilizing the all-level hierarchy labels performs even better on mAHP than mAP. It indicates that the semantic hierarchical Hamming space is hard to learn if only use the fine-level class labels. From the comparison with SHDH, which also utilizes hierarchy labels but pairwise label similarity, our method significantly outperforms SHDH by 23.2% under nDCG metric in average. The performance on NABirds dataset is presented in Table 2. We add the comparison with latest IDCWH. We can observe that the proposed HPDH performs the best on NABirds in general retrieval and hierarchical retrieval tasks. Specifically, it achieves 73.72% under mAP and 81.60% under nDCG at 48-bit, with a superiority of 2.1% and 5.6% to that of IDCWH. While for mAHP, HPDH obtains the best result 86.61% at 48-bit, surpasses the second place DTSH by a nearly 21% margin.

In Fig. 3, we plot the mAHP@t results w.r.t. different cutoff points t in CIFAR-100 and NABirds datasets, respectively. From Fig. 3, one phenomenon worth noting is, there is a declining trend of mAHP with the increase on t for all the compared methods, while only our method improves mAHP with the growth on t. Specifically, all the methods have a turning point at $t = 500$ in the curves of CIFAR-100. The reason is there are exactly 500 images per fine-level class, and it is no longer sufficient to retrieve the images with exactly the same labels after this point. However, the proposed HPDH, which benefits from learning the hierarchical similarity information, is the only method capable of retrieving similar images with the same parent class at later positions. Thus, it proves the effectiveness of the proposed method for discriminative hierarchical retrieval.

5 Conclusion

We propose a novel deep hashing model towards fully utilizing the hierarchy structure of the semantic information. Our method is based on a multi-level Gaussian loss function, and it takes the advantages of class-level similarity learning and full-level hierarchy labels in training. Experiments on two hierarchical datasets show that our method not only helps improve the fine-level retrieval performance but also results in state-of-the-art results regarding hierarchical retrieval.

References

[1] B. Kulis, P. Jain, and K. Grauman, "Fast similarity search for learned metrics," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 12, pp. 2143–2157, 2009.

[2] Y. Gong, S. Lazebnik, A. Gordo, and F. Perronnin, "Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 12, pp. 2916–2929, 2013.

[3] Y. Li, C. Chen, W. Liu, and J. Huang, "Sub-selective quantization for large-scale image search." in *AAAI*, 2014, pp. 2803–2809.

[4] W.-J. Li, S. Wang, and W.-C. Kang, "Feature learning based deep supervised hashing with pairwise labels," in *IJCAI*, 2016, pp. 1711–1717.

[5] Y. Cao, M. Long, J. Wang, H. Zhu, and Q. Wen, "Deep quantization network for efficient image retrieval." in *AAAI*, 2016, pp. 3457–3463.

[6] Q. Li, Z. Sun, R. He, and T. Tan, "Deep supervised discrete hashing," in *NIPS*, 2017, pp. 2482–2491.

[7] X. C. R. B. Sen Su, Gang Chen, "Deep supervised hashing with nonlinear projections," in *IJCAI*, 2017, pp. 2786–2792.

[8] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne." *Journal of Machine Learning Research*, vol. 9, no. 11, 2008.

[9] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," 2009, University of Toronto Technical Report.

[10] X. Zhe, S. Chen, and H. Yan, "Deep class-wise hashing: Semantics-preserving hashing via class-wise loss," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 5, pp. 1681–1695, 2019.

[11] M. Zhang, X. Zhe, S. Chen, and H. Yan, "Deep center-based dual-constrained hashing for discriminative face image retrieval," *Pattern Recognition*, p. 107976, 2021.

[12] X. Luo, C. Chen, H. Zhong, H. Zhang, M. Deng, J. Huang, and X. Hua, "A survey on deep hashing methods," *arXiv preprint arXiv:2003.03369*, 2020.

[13] Z. Cao, M. Long, J. Wang, and P. S. Yu, "Hashnet: Deep learning to hash by continuation," in *ICCV*, 2017, pp. 5608–5617.

[14] H.-F. Yang, K. Lin, and C.-S. Chen, "Supervised learning of semantics-preserving hash via deep convolutional neural networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 2, pp. 437–451, 2017.

[15] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[16] C. Fellbaum, "Wordnet," in *Theory and applications of ontology: computer applications.* Springer, 2010, pp. 231–243.

[17] A. Dutt, D. Pellerin, and G. Quénot, "Improving image classification using coarse and fine labels," in *ICMR*. ACM, 2017, pp. 438–442.

[18] Y. Ma, M. Kan, S. Shan, and X. Chen, "Hierarchical training for large scale face recognition with few samples per subject," in *ICIP*. IEEE, 2018, pp. 2401–2405.

[19] D. Wang, H. Huang, C. Lu, B.-S. Feng, G. Wen, L. Nie, and X.-L. Mao, "Supervised deep hashing for hierarchical labeled data," in *AAAI*, vol. 32, no. 1, 2018.

[20] M. Zhang and H. Yan, "Improved deep classwise hashing with centers similarity learning for image retrieval," in *ICPR*. IEEE, 2021, pp. 10 516–10 523.

[21] C. L. of Ornithology, "Nabirds dataset," accessed:2019-01-25. [Online]. Available: http://dl.allaboutbirds.org/nabirds

[22] X. Wang, Y. Shi, and K. M. Kitani, "Deep supervised hashing with triplet labels," in *ACCV*, 2016, pp. 70–84.

[23] L. Yuan, T. Wang, X. Zhang, F. E. Tay, Z. Jie, W. Liu, and J. Feng, "Central similarity quantization for efficient image and video retrieval," in *CVPR*, 2020, pp. 3083–3092.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[25] B. Barz and J. Denzler, "Hierarchy-based image embeddings for semantic image retrieval," in *WACV*, 2019, pp. 638–647.

[26] K. Järvelin and J. Kekäläinen, "Cumulated gain-based evaluation of IR techniques," *ACM Transactions on Information Systems*, vol. 20, no. 4, pp. 422–446, 2002.

[27] J. Deng, A. C. Berg, and L. Fei-Fei, "Hierarchical semantic indexing for large scale image retrieval," in *CVPR*. IEEE, 2011, pp. 785–792.