MBDH: A Multi-scale Balanced Deep Hashing Method for Image Retrieval (Postprint)

Authors: Yichao Zhang, Huang Zhangcan, Yaxiong Chen

Date: 2018-05-18T00:00:00+00:00

Abstract

Hashing has been widely utilized for large-scale multimedia retrieval owing to its advantages in storage and retrieval efficiency. Supervised hashing, which exploits semantic similarity of data to enhance hash code quality, has recently garnered significant attention. Conventional supervised hashing methods decouple the learning of hand-crafted or machine learning features from images and the separate quantization step into binary codes, inadequately control quantization errors, and fail to guarantee the balance of generated hash codes. To remedy this limitation, a novel multi-scale balanced deep hashing approach is proposed. This method employs multi-scale input, which effectively improves the network' s efficacy in learning image features. Moreover, a new loss function is introduced that, while preserving semantic similarity effectively, considers quantization errors and hash code balance to generate superior hash codes. The optimal retrieval results of this method on the CIFAR-10 and Flickr datasets improve retrieval accuracy by 5.5% and 3.1%, respectively, compared to state-of-the-art methods.

Full Text

Preamble

MBDH: A Multi-Scale Balanced Deep Hashing Method for Image Retrieval

Zhang Yichao[1], Huang Zhangcan[1], Chen Yaxiong[2,3]
(1. Department of Mathematics, School of Science, Wuhan University of Technology, Wuhan 430070, China;
2. Xi' an Institute of Optics & Precision Mechanics, Chinese Academy of Sciences, Xi' an 710048, China;
3. University of Chinese Academy of Sciences, Beijing 100049, China)

Abstract: Hashing has been widely used for large-scale multimedia retrieval due to its advantages in storage and retrieval efficiency. Supervised hashing, which leverages semantic similarity to improve hash coding quality, has recently attracted more extensive attention. Traditional supervised hashing methods separate the learning of hand-crafted features or machine learning features from the quantization step that generates binary codes, which neither effectively controls quantization error nor guarantees the balance of generated hash codes. To address this problem, this paper proposes a novel multi-scale balanced deep hashing method. This method employs multi-scale input, which effectively enhances the network' s ability to learn image features. Additionally, a new loss function is proposed that, while preserving semantic similarity, considers both quantization error and hash code balance to generate higher-quality hash codes. Experimental results on the CIFAR-10 and Flickr datasets show that the proposed method improves retrieval accuracy by 5.5% and 3.1% respectively compared to state-of-the-art methods.

Keywords: multi-scale; balance; deep hashing; convolutional neural network; image retrieval

0 Introduction

Nearest neighbor search (NNS) [1] has become fundamental to many machine learning, data mining, and image retrieval problems. Given a query point, NNS attempts to find the point in the database closest to the query. NNS has attracted significant attention from academia and industry for its potential applications in big data. However, dimensionality curse, storage cost, and query speed pose major challenges when applying NNS to large-scale problems.

Hashing represents an important and effective approach for NNS, offering excellent performance and ideal time complexity. Hashing methods can be categorized into data-independent and data-dependent approaches, differing in how hash functions are generated. Shift invariant kernel hashing (SIKH) [2] and minimal loss hashing (MLH) [3] are representative data-independent methods where hash functions are constructed through manual or random projections. The limitations of data-independent methods are obvious: excessive manual intervention may lead to deficiencies in adaptability and accuracy.

Consequently, data-dependent hashing methods that learn hash functions from given databases have been proposed. Such methods can generate more compact binary hash codes. Data-dependent hashing methods are further divided into supervised and unsupervised approaches based on whether training data has labels. Unsupervised hash functions achieve locality-sensitive effects based on probability theory by learning from unlabeled data. Popular unsupervised methods include locality-sensitive hashing (LSH) [2], iterative quantization (ITQ) [4], and spectral hashing (SH) [5]. While unsupervised hashing algorithms are fast, they fail to adequately utilize the rich semantics contained in images.

To avoid losing valuable semantic information, supervised hashing methods have been proposed. In scenarios where accuracy is prioritized over speed, supervised hashing methods become more appropriate. Examples include semi-supervised hashing (SSH) [6], minimal loss hashing (MLH) [3], linear discriminant analysis based hashing (LDA hash) [7], kernel-based supervised hashing (KSH) [8], and latent factor models for supervised hashing (LFH) [9].

Despite numerous supervised hashing methods, several deficiencies remain: (a) they often adopt GIST global features, which may cause semantic information loss; (b) most image retrieval methods can only learn shallow features, causing the correlation structure between image information to be ignored; and (c) previous methods generally fail to consider quantization error.

To address these issues, this paper proposes a multi-scale balanced deep hashing method for image retrieval that: (a) uses multi-scale features as input to obtain more robust semantic information—unlike local feature descriptors, features extracted from deep convolutional networks after multi-scale input partitioning are global descriptors that encode holistic information; (b) introduces convolutional neural networks (CNN) as deep mapping to better consider correlation structures between image information; and (c) proposes a novel loss function composed of cross-entropy, L2 norm, and a balance term that preserves semantic information while effectively solving quantization error problems to generate balanced and compact hash codes.

1 Convolutional Neural Network Hashing

Since their emergence, convolutional neural networks have rapidly attracted widespread attention in the computer vision community, narrowing the gap between machine and human visual perception capabilities. Their remarkable achievements in object recognition, detection, image parsing, and video classification have propelled the field of artificial intelligence forward.

CNN [10] is a constrained multi-layer neural network with inputs on a two-dimensional plane. Inspired by the human visual system, neurons in CNN hidden layers receive input from local regions of the previous layer and tile across two-dimensional feature maps relative to their input regions. A typical CNN consists of three structural components: convolutional layers, pooling layers, and fully connected layers. Neurons in convolutional feature maps share weights. Pooling layers are placed after convolutional layers and can be categorized as max pooling or average pooling based on the operation employed. Both convolutional and pooling layers can overlap by adjusting stride and filter size.

CNNH [11] and its later improvement CNNH* [12] are two-stage frameworks that take raw image data as input. In the first stage, the similarity matrix S is decomposed into the product of an approximate hash code matrix H where each row represents a K-dimensional hash code. In the second stage, raw image pixels and pre-generated binary codes H (CNNH* and their binary labels Y) are fed into a CNN, whose objective is to minimize the error between outputs and connect target binary vectors to H and Y.

During the hash function learning stage, CNNH uses deep networks to learn image feature representations and hash functions. Specifically, CNNH adopts a common deep framework as its base network and designs an output layer with softmax activation to generate q-dimensional hash codes. CNNH trains the designed deep network in a supervised manner, using the hash codes learned during the hash coding learning stage as ground truth. Additionally, if discrete class labels for training images are available, CNNH incorporates these image labels to learn hash functions. Based on deep networks, CNNH simultaneously learns deep features and hash functions. However, CNNH is a two-stage framework where deep features learned in the second stage cannot help improve approximate hash code learning in the first stage, which significantly limits hashing learning performance.

Typically, logistic regression with negative log-likelihood is used as the loss function for single-label classification. Other loss functions such as Euclidean distance and cross-entropy are also commonly employed. This paper proposes a new loss function composed of cross-entropy, L2 norm, and a balance term, which will be detailed later.

Many recent methods utilize convolutional neural networks to learn more effective image representations, achieving better performance than conventional hashing methods. However, these methods still face challenges: image information does not receive sufficiently diversified learning, and uncontrollable quantization errors exist during deep hashing algorithm learning, which cannot optimally accommodate the conversion of continuous hash codes to discrete binary codes, ultimately affecting binary code quality.

2 Multi-Scale Balanced Deep Hashing

In similarity retrieval, given a training set of N points, each represented as a D-dimensional feature vector, some points are associated with similarity labels where similarity between points is indicated. The goal is to learn a non-linear hash function that encodes each point into a compact K-bit hash code while preserving similarity between given pairs.

This paper proposes a novel multi-scale balanced deep hashing method for image retrieval. The method accepts input images at multiple scales and processes them through convolutional neural network hash channels: (a) multi-scaling images as input; (b) a fully connected hash layer for generating compact hash codes; (c) a loss layer using cross-entropy functions to preserve semantic similarity information; and (d) a quantization loss layer using L2 norm to control hash code quality.

2.1 Multi-Scale Balanced Deep Hashing Method

Lazebnik et al. proposed the spatial pyramid matching (SPM) method [13] that encodes spatial information using a bag-of-features (BoF) approach, representing images using pyramids with several levels or scales. Features from different scales are combined to form image representations, where coarser features receive smaller weights while finer features receive larger weights. This paper argues that matches found at coarser levels may involve increasingly diverse image features. This work similarly uses convolutional feature maps as local descriptors to explore multi-scale scenarios. Experiments reveal that deep features from convolutional feature maps differ from traditional descriptors: weighted sums of features at different levels do not demonstrate superior performance compared to simple summation.

Kaiming et al. designed a method called spatial pyramid pooling (SPP) [14], where feature maps from the last convolutional layer are divided into pyramids of 3 or 4 scales. First, region features from each scale are concatenated, then scale-level features are concatenated into fixed-length vectors to be forwarded to the next fully connected layer. However, it has been proven in [15] that this strategy does not achieve good results for unsupervised retrieval, leading to poor performance compared to other simple combination methods.

The proposed multi-scale balanced deep hashing method re-projects linearly between raw image regions and regions in feature maps at certain layers, enabling efficient computation of region feature vectors without re-feeding corresponding image regions, inspired by the work of Girshick and Tolias et al. [16,17].

The flowchart of the proposed multi-scale balanced deep hashing method is shown in Figure 1 [Figure 1: see original paper]. After receiving multi-scale input, the network first enters a sub-convolutional neural network component consisting of convolutional layers, pooling layers, and fully connected layers. This paper employs three convolutional layers: the first with 32 kernels of size 3×3, and the second and third each with 64 kernels of size 3×3. Two max pooling layers of size 2×2 are placed between convolutional layers. Two fully connected layers follow, with the second serving as the hash layer. Finally, the network enters a loss function module composed of three parts: softmax loss, quantization loss, and balance term components.

2.2 Loss Function

Let the training set with N samples be denoted as , where is the nth sample. The ultimate goal of hashing is to map and quantize it into binary codes. This paper feeds training samples as input into a multi-layer neural network for nonlinear transformation to obtain binary codes as output. Assuming the network is an L+1 layer network with m scale inputs, the output at layer l under scale i can be expressed as:

where represents the output at layer l under scale i, represents the weight at layer l under scale i, and represents the bias at that layer.

The fused output at the highest layer across all scales can be obtained through:

where is the fused output at the highest layer in the form of a set of K-bit hash codes, and are the weight and bias at the highest layer respectively, and is the tanh function.

The method proposed in this paper maps convolutional features to , where hash codes are continuous real values. To obtain binary hash codes , a threshold function is applied:

where represents a set of K-bit binary hash codes, is the sign function, returns 1 if , and returns -1 otherwise.

Assuming the binary code of an image is used as input to the softmax layer, the probability of predicting label is:

where is the jth weight parameter of the softmax layer, is the jth bias parameter of the softmax layer, and M is the number of training image categories.

By considering the negative log-likelihood of label , the following optimization problem can be obtained:

where is the indicator function that equals 1 if and 0 otherwise.

Let be the binary code of one image and be the binary code of another image, where images and are similar, while images and are dissimilar. represents the Hamming distance between binary codes and . Optimizing this problem minimizes the Hamming distance between similar images and while maximizing the Hamming distance between dissimilar images and [19].

Since binary constraint optimization is challenging, this paper proposes a novel strategy using continuous relaxation to replace binary constraints, a technique widely adopted by existing hashing methods [18]. However, continuous relaxation leads to two important issues widely ignored in hash function learning: uncontrollable quantization error from replacing binary constraints with continuous relaxation, and approximation error from using inner products between continuous relaxations as substitutes for Hamming distance between binary codes. To control quantization error and bridge the gap between Hamming distance and its substitute for learning high-quality hash codes, this paper designs a new loss function composed of cross-entropy, L2 norm, and a balance term. This loss not only preserves semantic information but also effectively solves quantization error problems.

Assuming is the binary code of an image, the optimization problem can be redefined as:

where is the cross-entropy loss function.

However, continuous relaxation leads to uncontrollable quantization error [20]. This paper introduces a regularization term to control quantization error, using the L2 norm between continuous hash codes and discrete binary codes as the regularization term . However, optimizing alone may result in binary codes consisting entirely of 1s, as optimizing the L2 norm term affects hash code balance. To maintain hash code balance, the squared mean value of hash codes is used as a balance criterion. This balance standard encourages each bit of hash codes to be mapped to -1 or 1 as uniformly as possible [21]. The optimization problem for generating good binary codes becomes:

where is a weight parameter controlling regularization strength, is a parameter controlling the relative importance of the balance criterion, is the averaging operator, is the L2 norm, and |.| represents absolute value. The regularization term controls uncontrollable quantization error from replacing binary constraints with continuous relaxation, while the balance standard ensures each bit of hash codes has equal probability of being 1 or -1, making the occurrence of 0 and 1 in binary hash codes as equal as possible.

3 Experiments

3.1 Experimental Algorithm Flow

Algorithm 1: Multi-Scale Balanced Deep Hashing Algorithm

Input: Training samples and their corresponding label vectors
Output: All weight parameters ; all bias parameters .

Initialization: Weights are initialized with Gaussian distribution.

1. Compute through forward propagation;
2. Calculate hash codes using equations (2) and (3);
3. Calculate predicted output using equation (4);
4. Compute using equation (8) with ;
5. Update parameters using stochastic gradient descent (SGD) until fixed iteration count is reached.

3.2 Experimental Datasets

The proposed method is evaluated using two benchmark datasets: CIFAR-10 and Flickr.

a) CIFAR-10 Image Database: This dataset is a subset of the 80 million Tiny Images dataset, containing 60,000 color images across 10 object categories (6,000 images per class). Each image is 32×32 pixels. The categories include airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck. For CIFAR-10, 1,000 images are randomly selected from each category to form the test query set, with the remaining 50,000 images used as the training set. This paper uses 512-dimensional GIST features as traditional feature representation and 4,096-dimensional CNN features as deep semantic features.

b) Flickr Image Database: This dataset consists of 25,000 images collected from Flickr, each labeled with one of 38 semantic concepts. In this paper, images from this subset are resized to 32×32 pixels.

3.3 Experimental Environment Configuration

The experimental environment consists of: GeForce GTX Titan X GPU, Intel(R) Core i7-5930K 3.50 GHz CPU, 64 GB RAM, and Ubuntu 14.04 operating system. The proposed model is implemented using the open-source library KERAS. The overall objective function is optimized via stochastic gradient descent (SGD) with a learning rate of 10 , decay value of 10 after each update, and Nesterov momentum of 0.9. The batch size is 32. Initial weights between input and hidden layers are set to normal distribution. The initial recurrent weight matrix is set to identity matrix, with remaining weights using Gaussian distribution. In this experiment, parameters and are set to 0.001 and 0.001 respectively. Subsequent experiments demonstrate the rationale and reasonableness of these parameter choices.

3.4 Experimental Evaluation Metrics

To evaluate hashing method effectiveness, several commonly used quantitative performance comparison metrics are considered, with the following three measures ultimately adopted:

a) Mean Average Precision (MAP): The average of average precision values for each query sample.

b) Precision@top-n at 48-bit hash code length: Images in the database are ranked according to Hamming distance from query samples, and precision of the top-n returned results is calculated.

c) Precision within Hamming radius 2: Precision is computed for query samples where Hamming distance to dataset images is less than 2.

In experiments, class labels are used as the ground truth. All metrics are calculated by checking whether query and returned images share the same label. Higher values indicate better performance.

3.5 Experimental Results and Analysis

1) Results on CIFAR-10 Database

The proposed multi-scale convolutional neural network hashing method is evaluated using the three metrics and compared with classical and state-of-the-art methods, including three unsupervised methods (LSH [2], SH [5], ITQ [4]) and eight supervised methods (DHN [22], CNNH [11] and its variant CNNH* [12], DNNH [12], KSH [8], MLH [3], BRE [23], and ITQ-CCA [24]). LSH, SH, ITQ, KSH, MLH, BRE, and ITQ-CCA are traditional hashing methods using 512-dimensional vectors as input to learn hash functions. The other four methods (DHN, DNNH, CNNH*, and CNNH) use 4,096-dimensional CNN features as input for hash function learning.

Key observations include: (a) Comparing methods using hand-crafted features versus 4,096-dimensional CNN features reveals that CNN features improve traditional method performance; (b) The proposed MBDH achieves superior performance compared to other existing hashing techniques, improving average MAP from 42.9% (CNNH), 48.4% (CNNH*), 55.2% (DNNH), and 55.5% (DHN) to 67.6%. This improvement stems from multi-scale image input and a novel loss function that preserves semantic similarity and balance while considering quantization error from continuous-to-discrete conversion.

Table 1 shows MAP results on CIFAR-10 database. Figure 2 [Figure 2: see original paper] displays precision curves: the left plot shows precision within Hamming distance 2 across different bit lengths, with MBDH achieving best retrieval accuracy; the middle plot shows MAP across all bit lengths; the right plot displays precision@top-1000 at 48 bits, where MBDH again achieves best performance. Figure 3 [Figure 3: see original paper] shows top-10 retrieval results for 10 classes, with incorrectly matched images marked by red boxes.

2) Results on Flickr Database

Using the same experimental settings, Table 2 presents MAP results with different hash code lengths on Flickr dataset. The proposed MBDH improves average MAP from DHN' s 84.1% to 87.2%, demonstrating superior performance over eight other supervised hashing methods (DHN, DNNH, CNNH, CNNH*, KSH, MLH, BRE, and ITQ-CCA) due to multi-scale input and the new loss function.

3.6 Experimental Analysis of Parameter Configuration Impact

Parameters and are set to different values from 10 to 1 (with equal values for simplicity), and MAP is computed for 48-bit hash codes on CIFAR-10. Table 3 shows that MAP reaches its maximum value of 0.676 when and are set to 10 ^{3}, validating the parameter selection rationale.

3.7 Experimental Analysis of Multi-Scale Effectiveness

To further validate the proposed method, experiments are conducted on single-scale and dual-scale versions with identical network architecture, computing three evaluation metrics at 48-bit hash code length. Table 4 shows that the proposed method significantly outperforms both single-scale and dual-scale methods across all metrics, with MAP improving by 4.4% over the single-scale method. This demonstrates the effectiveness of the multi-scale balanced deep hashing approach.

4 Conclusion

This paper proposes a simple yet effective multi-scale balanced deep hashing model (MBDH) for learning binary hash codes for fast image retrieval. The method does not rely on symmetric data similarity. The proposed deep hashing network architecture replaces single-scale input with multi-scale input, simultaneously optimizing cross-entropy loss for semantic similarity and quantization loss for compact hash code generation while considering hash code balance. Extensive experiments provide MBDH results on two benchmark databases and comparative evaluations with multiple state-of-the-art hashing methods. Experimental results demonstrate that by adopting multi-scale input and optimizing the loss function, MBDH improves retrieval accuracy by 5.5% and 3.1% on CIFAR-10 and Flickr datasets respectively compared to advanced methods. The proposed method also demonstrates scalability and effectiveness on large-scale datasets exceeding one million images.

References

[1] Yi T T, Huang L H. An improved relevance feedback algorithm based on nearest neighbor in CBIR [J//OL]. *Computer Application Research*, 2015, 32(08): 2326-2330.
[2] Raginsky M, Lazebnik S. Locality-sensitive binary codes from shift-invariant kernels [C]// *Advances in Neural Information Processing Systems*. 2009: 1509-1517.
[3] Norouzi M, Blei D M. Minimal loss hashing for compact binary codes [C]// *Proc of International Conference on Machine Learning*. 2011: 353-360.
[4] Gong Y, Lazebnik S, Gordo A, et al. Iterative quantization: a procrustean approach to learning binary codes for large-scale image retrieval [J]. *IEEE Trans on Pattern Analysis and Machine Intelligence*, 2013, 35(12): 2916-2929.
[5] Weiss Y, Torralba A, Fergus R. Spectral hashing [C]// *Advances in Neural Information Processing Systems*. 2009: 1753-1760.
[6] Wang J, Kumar S, Chang S F. Semi-supervised hashing for scalable image retrieval [C]// *Proc of IEEE Conference on Computer Vision and Pattern Recognition*. 2010: 3424-3431.
[7] Strecha C, Bronstein A, Bronstein M, et al. LDAHash: improved matching with smaller descriptors [J]. *IEEE Trans on Pattern Analysis and Machine Intelligence*, 2012, 34(1): 66-78.
[8] Liu W, Wang J, Ji R, et al. Supervised hashing with kernels [C]// *Proc of IEEE Conference on Computer Vision and Pattern Recognition*. 2012: 2074-2081.
[9] Zhang P, Zhang W, Li W J, et al. Supervised hashing with latent factor models [C]// *Proc of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval*. 2014: 173-182.
[10] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks [C]// *Advances in Neural Information Processing Systems*. 2012: 1097-1105.
[11] Xia R, Pan Y, Lai H, et al. Supervised hashing for image retrieval via image representation learning [C]// *Proc of the 28th AAAI Conference on Artificial Intelligence*. 2014: 2156-2162.
[12] Lai H, Pan Y, Liu Y, et al. Simultaneous feature learning and hash coding with deep neural networks [C]// *Proc of International Conference on Computer Vision and Pattern Recognition*. 2015: 3270-3278.
[13] Lazebnik S, Schmid C, Ponce J. Beyond bags of features: Spatial pyramid

matching for recognizing natural scene categories [C]// *Proc of International Conference on Computer Vision and Pattern Recognition*. 2006: 2169-2178.
[14] He K, Zhang X, Ren S, et al. Spatial pyramid pooling in deep convolutional networks for visual recognition [J]. *IEEE Trans on Pattern Analysis and Machine Intelligence*, 2015, 37(9): 1904-1916.
[15] Hao J, Dong J, Wang W, et al. What Is the Best Practice for CNNs Applied to Visual Instance Retrieval? [J]. *arXiv preprint arXiv: 1611.01640*, 2016.
[16] Girshick R. Fast r-CNN [C]// *Proc of International Conference on Computer Vision and Pattern Recognition*. 2015: 1440-1448.
[17] Tolias G, Sicre R, Jégou H. Particular object retrieval with integral max-pooling of CNN activations [J]. *arXiv preprint arXiv: 1511.05879*, 2015.
[18] Wang J, Shen H T, Song J, et al. Hashing for similarity search: A survey [J]. *arXiv preprint arXiv: 1408.2927*, 2014.
[19] Li W J, Wang S, Kang W C. Feature learning based deep supervised hashing with pairwise labels [J]. *arXiv preprint arXiv: 1511.03855*, 2015.
[20] Kang W C, Li W J, Zhou Z H. Column Sampling Based Discrete Supervised Hashing [C]// *Proc of the 30th AAAI Conference on Artificial Intelligence*. 2016: 1230-1236.
[21] Do T T, Doan A Z, Cheung N M. Discrete hashing with deep neural network [J]. *arXiv preprint arXiv: 1508.07148*, 2015.
[22] Zhu H, Long M, Wang J, et al. Deep Hashing Network for Efficient Similarity Retrieval [C]// *Proc of the 30th AAAI Conference on Artificial Intelligence*. 2016: 2415-2421.
[23] Kulis B, Darrell T. Learning to hash with binary reconstructive embeddings [C]// *Advances in Neural Information Processing Systems*. 2009: 1042-1050.
[24] Gong Y, Lazebnik S, Gordo A, et al. Iterative quantization: a procrustean approach to learning binary codes for large-scale image retrieval [J]. *IEEE Trans on Pattern Analysis and Machine Intelligence*, 2013, 35(12): 2916-2929.