A Deep Hashing Method for Image Retrieval

Ling Gan, Tianzhen Zhang[a]

School of Chongqing University of Posts and Telecommunications Chongqing 400000, China

[a]tianzhen_z@126.com

Keywords: Image retrieval, Hashing, Deep quantization learning

Abstract: All Deep quantization learning based hashing methods have been proven to be effective in the field of image retrieval recently. How to improve the description ability of hash codes is still a challenging problem. In this paper, we propose a new deep quantization network architecture for supervised hashing called Supervised Deep Quantized Hashing (SDQH)for image retrieval. The main contribution is to reduce the redundancy between the hash code and the network parameters, Moreover, a block coding module is proposed and the similarity is learned through a specific loss function and joint quantization method to represent the image features as compact binary codes. Experimental results on three benchmark data sets show that the image retrieval performance of the hash code obtained by the proposed method outperforms the current main methods.

1. Introduction

In recent years, for the retrieval of large-scale data, the approximate nearest neighbor (ANN) retrieval of images has high computational efficiency and search quality, which has attracted widespread attention. Hashing is the mainstream method for solving approximate nearest neighbor problems. Generally speaking, hash methods are divided into unsupervised and supervised methods and semi-supervised methods Unsupervised hashing is the application of unlabeled data to the learning hash encoding process. Representative unsupervised learning methods are Kernelized LSH[1], Quantization-based Hashing(QBH)[2], Spectral Hashing(SH)[5], stochastic generative hashing (SGH) etc. Supervised hashing method is to learn hash codes with label data. Supervised Discrete Hashing (SDH), Minimal Loss Hashing(MLH)[4], Kernel-based Supervised Hashing (KSH)[3] are the popular example of supervised hashing approaches. The semi-supervised method uses part of the image's label information to learn the hash function. Among them, it mainly includes Parameter-Sensitive Hashing (PSH), Semi-Supervised Hashing (SSH).

In recent years, deep hashing methods have been proposed to show superior performance. Most of the existing supervised deep hashing methods directly extract image features from the original image and learn hash functions. Furthermore, the quantization method [6,7,8,9] forms a short code to represent each point through the index of the nearest center, which has been proven to be able to achieve an approximate nearest neighbor search more than the hash method. For example, DQN[6] jointly learns deep representations via a pairwise cosine loss and a product quantization loss for generating compact binary codes. Deep Visual-Semantic Quantization (DVSQ)[7] proposes a point-by-point adaptive edge and joint semantic quantization method, but did not consider the independence between hash functions.

There are several key differences between our work and previous work Deep learning quantification methods. 1) Our work introduces an idea of block coding, which can improve the independence between hash functions, reduce the redundancy between hash codes and network parameters, and perform effective similarity retrieval. 2) A new approximate loss function is proposed which adds constraints and can implement effective image retrieval in implementing an end-to-end learning network framework.

2. Proposed Method

Our proposed architecture is shown in Figure 1. It mainly contains two parts: 1) a feature

extraction component provided in a multi-tasking manner. 2) a quantization method that converts image features into binary codes. In the rest of this section, we will first describe them and then explain the optimization and search.

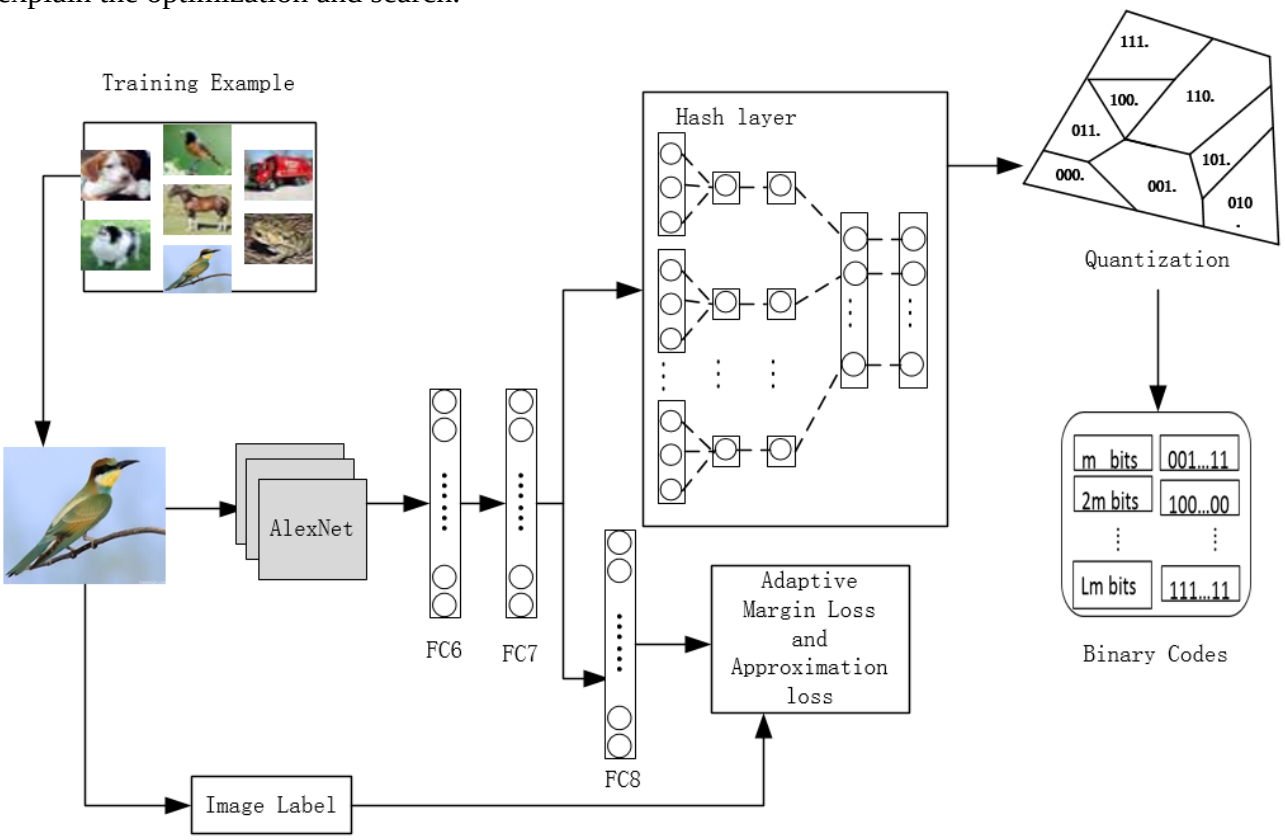

Fig. 1 Illustration of the proposed SDQH framework for fast image retrieval.

2.1 Feature extraction using multi-task learning

For the feature extraction phase, we use the Alexnet network as the feature extraction architecture, and the first to seventh layers use pre-trained models. To make the most of regulatory information, and we introduced two tag-based losses.

The first loss is an adaptive margin loss,Follow DVSQ[7],to embed the projects with the semantic label. We first embed each label into a 300-dimensional semantic vector, predict the semantic label corresponding to each picture, and then combine the image feature Z_n output by the eighth fully connected layer to define the adaptive edge loss function as:

$$L_S(x) = \sum_{i \epsilon y_n} \sum_{j \notin y_n} (0, \delta_{ij} - \frac{v_i^T Z_n}{||v_i|| ||Z_n||} + \frac{v_j^T Z_n}{||v_j|| ||Z_n||}) \quad (1)$$

$$\delta_{ij} = 1 - \frac{v_i^T v_j}{||v_i|| ||v_j||} \quad (2)$$

Here v_i is the word embedding of correct text label of image X_n,while v_j is the word embedding of false text label of image X_n, δ_{ij} is in order to adjust the distance between correct text and incorrect text.

As can be seen from Figure 1, we limit the output of image features to {-1,1} by adding a tanh activation function, we impose a regularizer on the continuous values to approach the desired discrete values (+1/-1). We proposed approximation loss as:

$$L_H(x) = \sum_M^N ||Z_n| - 1|_1 \quad (3)$$

Here |.|is the element-wise absolute value operation,1 is a vector of one and $||.||_1$is the l_1-norm of one vector.

2.2 Hashing layer

Three criteria for measuring the quality of a hash function are given in the spectral hash. The goal of the second criterion requires the generation of compact binary codes, that is, the different Hashing functions should be independent. Reduce the number of network parameters while having redundancy between required hash codes. Recently, a differentiated divide-and-encode module was introduced, similar to NINH [10], it evenly divides intermediate features to obtain K groups from image features. Then we map each group to a 1-dimensional feature through a fully connected layer. Finally, we enter all K sub-blocks into the activation layer separately. The activation layer uses the tanh activation function. The random assigned weight of each sub-block makes each bit of the hash code Only the part of the feature is relevant, thus achieving the independence of the hash function construction. Finally, it is merged into a layer, and its output is an approximate value of the output value of the hash function

$$B \in \{-1, +1\}^{K*N}.$$

$$B=g(X) \in \{-1, +1\}^{K*N} \tag{4}$$

Here X is a data matrix containing N images and g(·) representing the transformation of the deep network. Deep hash networks use an end-to-end learning framework to learn good features and compact binary hash codes together.

2.3 Inner-Product Quantization

In order to map a low-dimensional vector to a binary hash code, and keep the original similarity as much as possible, the proposed quantization model uses a set of M codebooks C = [C_1; ...; C_M], where each codebook C_M contains K codewords C_M=[C_{M1}, …. , C_{Mk}] ,and codeword $C_{Mk}$is a D-dimensional cluster-centroid vector as in K-means. According to the above process, the binary codeword assignment vector b_n is divided into M codebooks, defined as b_n = [b_{1n}; ...; b_{Mn}], and each indication vector b_{Mn} represents the K codeword in the mth codebook. A codeword (also unique) used to approximate the nth data point. Each embedded image Z_n is the sum of M codebooks, each codebook has only one codeword, each codeword is represented by a binary assignment vector b_n, then the embedded image Z_n is represented :

$$Z_n \approx \sum_{m=1}^{M} C_m b_{mn} \tag{5}$$

To reduce the difference between before and after reconstruction and regulatory information, we use the method of least square difference, which is represented:

$$Q_n = \sum_{i=1}^{|y|} (v_i^T Z_n - v_i^T \sum_{m=1}^{M} C_m b_{mn})^2 \tag{6}$$

Which is subject to the discrete constraints $||b_{mn}||_0 = 1$ and $b_{mn} \epsilon \{0,1\}^K$, v_i is the word embedding of correct text label of image X_n, with $||.||_0$ being the l_0-norm that simply counts the number of the vector's nonzero elements.

2.4 Overall Objective and Implementation

The entire objective function aiming for constructing similarity preserving (Eq. (1) and Eq.(3)) and binarization properties (Eq. (6)) is given as

$$\min_{\theta,C,B} \sum_{n-1}^{N} (L_s + \alpha L_H + \beta Q_n) \tag{7}$$

where α and β are the weights of each term, θ denotes the set of learn able parameters of the deep network. B denotes binary codes, C is codebooks.

We use Asymmetric Quantizer Distance(AQD) as the metric that computes the inner-product similarity between a given query q and database point X_n in the semanic space.

$$AQD(q, X_n) = Z_q^T (\sum_{m=1}^{M} C_m b_{mn}) \tag{8}$$

Z_q represents the output value of the query image processed by the convolutional neural network,

and the AQD point between the query and all databases is calculated by Equation (8).

3. Experimental data and analysis

3.1 Setup

We evaluate retrieval quality based on the standard evaluation metrics: Mean Average Precision (MAP) and conduct the experiments on three widely used image retrieval benchmark datasets: ImageNet,NUS-WIDE and CIFAR-10. We follow DVSQ[7] and adopt MAP@5000 for NUS-WIDE dataset, MAP@5000 for ImageNet dataset, and MAP@54000 for CIFAR-10 dataset.

Our implementation of SDQH is based on **TensorFlow**. This experimental environment is the Intel i7-8700 processor, GeForceRT2080x2, GPU memory 7GB. We adopt Alexnet network, fine-tuned conv1-fc7. We adopt the adamoptimizer optimizer as the gradient descent method, first-order moment estimation and second-order moment estimation of the gradient are used to dynamically adjust the learning rate of each parameter. In order to compare with other methods using the same bit-length, we construct the codebook with K=256=2^8 , so that each codebook can provide a piece of 8 bits binary code. We tune the learning rate from 10^{-5} to 10^{-2} . As for α, β in loss function Eq. 7, we empirically set them as α = 0.0001, β = 0.0001.

3.2 Results

We compare the MAP result of all method are list Table1, five deep supervised methods, **DSH**[11], **DQN**[6], **DVSQ**[7],which shows that the proposed DVSQ method substantially outperforms all the comparison methods. showing that the proposed SDQH outperforms all the comparison method. In CIFAR-10, the improvement of SDQH over the other methods is more significant, compared with that in NUS-WIDE and ImageNet datasets. Specifically, it outperforms the best counterpart (DVSQ) by **0.8%**, **0.5%**, **1.4%** and **0.7%** for 8, 16, 24 and 32-bits hash codes.

SDQH improves the state-of-the-art by **1.4%**, **1.0%**, **1.8%** and **1.4%** in NUS-WIDE dataset, and **1.6%**, **1.7%**, **1.7%**, **0.8%** in ImageNet dataset. The proposed SDQH method can learn discriminative and compact hash codes by integrating the feature extraction, the divide-and-encode module, the hashing learning and prediction learning into one unified deep network.

Table 1 Mean Average Precision (MAP) Results for Different Number of Bits on the Three Benchmark Image Datasets

Method	CIFAR-10				NUS-WIDE				ImageNet			
	8bits	16bits	24bits	32bits	8bits	16bits	24bits	32bits	8bits	16bits	24bits	32bits
DSH[11]	0.592	0.625	0.651	0.659	0.653	0.688	0.695	0.699	0.332	0.398	0.487	0.537
DQN[6]	0.527	0.551	0.558	0.564	0.721	0.735	0.747	0.752	0.488	0.552	0.598	0.625
DVSQ[7]	0.745	0.745	0.746	0.746	0.781	0.780	0.780	0.782	0.563	0.567	0.572	0.574
SDQH	0.753	0.750	0.760	0.753	0.795	0.790	0.798	0.796	0.579	0.584	0.589	0.582

4. Conclusions

In this work, we propose a deep hash quantization model. For large-scale image retrieval, SDQH can improve the independence of the hash function and combine label information to implement an end-to-end learning method to learn compact binary code. Experimental results on benchmark datasets show that our model is significantly better than current image retrieval.

References

[1] Kulis B, Grauman K.Kernelized Locality-Sensitive Hashing[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2012, 34(6): 1092-1104.

[2] Song, Jingkuan, Gao, Lianli, Liu, Li, Zhu, Xiaofeng and Sebe, Nicu (2018) Quantization-based hashing: a general framework for scalable image and video retrieval. Pattern Recognition, 75. pp. 175-187. ISSN 0031-3203.

[3] Liu W, Wang J, Ji R, Jiang Y-G, Chang S-F (2012) Supervised hashing with kernels. In: Computer vision and pattern recognition (CVPR), pp 2074–2081.

[4] Norouzi M, Fleet DJ (2011) Minimal loss hashing for compact binary codes. In: ICML, pp 353–360.

[5] Li P, Wang M, Cheng J, Xu C (2013) Spectral hashing with semantically consistent graph for image indexing. IEEE Trans Multimed 15:141–152.

[6] Yue Cao, Mingsheng Long, Jianmin Wang, Han Zhu, and Qingfu Wen. 2016. Deep Quantization Network for Efficient Image Retrieval. AAAI.

[7] Yue Cao, Mingsheng Long, Jianmin Wang, and Shichen Liu. 2017. Deep visual semantic quantization for efficient image retrieval. In CVPR.

[8] Gao L, Zhu X, Song J, et al. Beyond Product Quantization: Deep Progressive Quantization for Image Retrieval[J]. arXiv: Computer Vision and Pattern Recognition, 2019.

[9] Song J, Zhu X, Gao L, et al. Deep Recurrent Quantization for Generating Sequential Binary Codes[C]. international joint conference on artificial intelligence, 2019: 912-918.

[10] Lai H, Pan Y, Liu Y, et al. Simultaneous feature learning and hash coding with deep neural networks[C]. computer vision and pattern recognition, 2015: 3270-3278.

[11] H. Liu, R. Wang, S. Shan, and X. Chen. Deep supervised hashing for fast image retrieval. In CVPR, 2016.