IMAGE RETRIEVAL BASED ON DEEP CONVOLUTIONAL NEURAL NETWORKS AND BINARY HASHING LEARNING

Peng Tian-qiang, Li Fang

Henan Institute of Engineering
Department of Computer Science and Engineering
Zhengzhou,China

ABSTRACT

With the increasing amount of image data, the image retrieval methods have several drawbacks, such as the low expression ability of visual feature, high dimension of feature, low precision of image retrieval and so on. To solve these problems, a learning method of binary hashing based on deep convolutional neural networks is proposed. The basic idea is to add a hash layer into the deep learning framework and simultaneously learn image features and hash functions which should satisfy independence and quantization error minimized. First, convolutional neural network is employed to learn the intrinsic implications of training images so as to improve the distinguish ability and expression ability of visual feature. Second, the visual feature is putted into the hash layer, in which hash functions are learned. And the learned hash functions should satisfy the classification error and quantization error minimized and the independence constraint. Finally, given an input image, hash codes are generated by the output of the hash layer of the proposed framework and large scale image retrieval can be accomplished in low-dimensional hamming space. Experimental results on the three benchmark datasets show that the binary hash codes generated by the proposed method has superior performance gains over other state-of-the-art hashing methods.

Index Terms— Image retrieval, Deep convolutional neural networks, Binary hashing, Quantization error, Independence

1. INTRODUCTION

The early image retrieval technology is Text-based Image Retrieval (TBIR), and then the Content-based Image Retrieval (CBIR) technology gradually becomes the mainstream, which expresses the image content by the low-level features. The low-level features include local feature descriptor based on gradient, such as Scale-Invariant Feature Transform (SIFT) [1], Histogram of Orientated Gradients (HOG) [2] and so on. Convolutional Neural Networks (CNNS) can obtain the inherent features of an image compared with the hand-crafted features, and CNNS has shown good performance in object detection, image classification, and image segmentation. Krizhevsky [3] proposed the feature extraction architecture based on CNNS firstly, which achieve a good result on ImageNet.

Hashing methods are widely used in computer vision, machine learning, and information retrieval. The hashing methods can map high-dimensional features of the images into the compact binary hashing codes. Combined the advantage of CNNS with hash methods, we propose a method to learn the binary hash functions based on CNNS. The basic idea is to introduce hashing layer to CNNS, and learn the image features and hash functions simultaneously, and construct hash functions with the independence and quantization error minimized. Compared with other methods, our method has the following characteristics:

- A hashing layer is introduced to the existing CNNS architecture. The binary hash codes generated by the hashing layer are putted into the Softmax classifier, and the Softmax loss is one of the optimization objective.
- The hashing layer contains two parts. The first part includes slice layer, fully-connected layer, activation layer and concat layer. The first part is used to map the features to continuous codes and generate independent hash functions. The other part includes a threshold layer, which is used to binarize the continuous codes to generate binary hash codes, which can be used to calculate the quantization error.
- Considering the influence of quantization error, the error generated by binarizing the continuous values to binary hash codes is added into the optimization objective, and it helps to get more powerful hash codes.

2. RELATED WORK

Locality Sensitive Hashing (LSH) [4] can be split into two categories [5]: one is called "original LSH", which ranks the original data to speed up the search; the other is called "binary LSH", which embeds the high-dimensional data into hamming space and performs a bitwise-operation to find the similar objects. The binary hashing methods include unsupervised hashing, semi-supervised hashing and supervised hashing. The unsupervised hashing methods do not take account of the label information, including Isotropic hashing [6], spectral hashing (SH) [7], PCA-ITQ [8]. The semi-supervised hashing methods take account of the similarity information partly, including SSH [9]. The supervised hashing methods use the label information or the similar point pairs as the supervised information, including BRE [10], KSH [11].The objective of these hashing methods is to construct hash functions which can keep the similarity of the original data and generate the compact binary hashing codes. Three criterions to measure the hashing functions are proposed in [7]: 1) mapping the similar objects in the original space to similar binary codes in the hamming space; 2) encoding the whole dataset using less bits; 3) the binary codes can be calculated easily if a new data is given. The objective of the second condition is to generate compact binary codes, which means the constructed hashing functions should be independent. The PCA-ITQ [8] method minimizes the quantization error in the proceed of constructing hashing functions, and generates binary hashing codes with higher expression ability.

The methods based on CNNS [12][13][14] have shown good performance in target detection, image classification, and so on. The mainstream CNNS models, such as AlexNet [3], NIN[15], VGG [16] have verified the ability of learning image feature representation successfully.

Because of the superiority of CNNS in feature learning and the hashing methods in computing speed and memory space, the methods combined CNNS with hashing have been proposed in recent years. Rongkai Xia et al. [17] proposed a method with two steps, in which feature extracting and hash codes learning were separated. To improve the method, Hanjiang Lai et al.[18] proposed a method which can learn the features and hashing functions simultaneously using CNNS, in which the image triplet is used as supervised information, but the selection of triplet affects retrieval accuracy directly and needs a lot of work. Kevin Lin et al.[19] also proposed a method which can learn the features and hashing functions simultaneously, in which the label information is used as supervised information,but it does not consider quantization error and the independence between the hashing functions.

3. OUR METHOD

The CNNS architecture of our method is shown in Fig. 1. The inputs are image pixels and the corresponding label information. Our architecture includes three parts: 1) convolutional sub-network, the role of this part is to learn image features; 2) hashing layer, the role of this part is to construct hashing functions to get binary hash codes; 3) loss layer, this layer includes softmax classification loss and quantization error loss.

3.1. Convolutional sub-network

We adopt VGG [16] model with 16 layers as our basic framework, which includes five large convolutional layers, five pooling layers, two fully-connected layers. The architecture of VGG is simple and can extract pretty good feature for small images. We should adjust the output number of the convolutional layer according to the image size.

3.2. Hashing layer and optimization objective

The definition of binary hashing function is : given a feature $x \in R^m$, construct m-dimensional random vectors whose number is q to form a matrix $W \in R^{q\times m}$, then get the hash codes $(h_1, h_2, , h_q)^T = (sign(Wx))^T$ generated by the q hashing functions.

In this paper, the hashing layer includes slice layer, fully-connected layer, activation layer, concat layer and thresholding layer. The role of the first four layers is to construct independent hashing functions; the role of the last layer is to binarize the continuous values and calculate the quantization error.

After calculated by the convolutional sub-network, image feature x is obtained, which is used to imported into the hashing layer. First, x should be divided, if the dimensionality of x is m and the length of hashing codes is q, then we should divide x into q slices, denoted as $x^{(i)}(i = 1, 2, , q)$, the dimensionality of each slice is m/q.

After calculated by the slice layer, $x^{(i)}(i = 1, 2, , q)$ should be imported to fully-connected layer respectively, the output dimensionality of each fully-connected layer is one-dimensional, the corresponding formula is as follows

$$f_i(x_{(i)}) = W_i x^{(i)} \quad i = 1, 2, ..., q \tag{1}$$

Matrix $W_i \in R^{dim(x^{(i)})\times 1}$ indicates the weight matrix of the i-th fully-connected layer.

The activation layer adopts double-tangent function which can restrict the outputs in the range $[-1, 1]$, the corresponding formula is as follows

$$tanh(v^{(i)}) = \frac{1 - e^{\beta v^{(i)}}}{1 + e^{\beta v^{(i)}}} \quad i = 1, 2, ..., q \tag{2}$$

where, $v^{(i)} = f_i(x^{(i)})$, β is used to control the smoothness. The double-tangent function is used to approximate the sign function. The independent hash functions are constructed with slice layer, in which the weight matrix W_i is learned only related to the features of each sub-block.

Then, imported to the concat layer, which can merge one-dimensional output from the q sub-blocks as a q-dimensional vector, the formula is as follows

$$s = (tanh(v^{(1)}), tanh(v^{(2)}), ..., tanh(v^{(q)}))^T \tag{3}$$

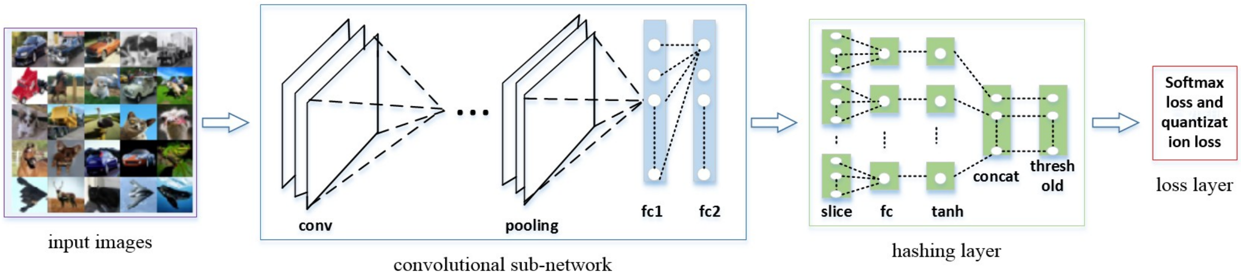

Fig. 1. The architecture of our model

The output of the concat layer is the approximation output of the hash functions, which is continuous value.

Finally, imported to the thresholding layer, which can binarize the q-dimensional continuous values generated by the concat layer into $\{-1, 1\}$. The formula is as follows

$$g(s^{(i)}) = \begin{cases} 1 & s^{(i)} > 0 \\ -1 & s^{(i)} < 0 \end{cases} \tag{4}$$

$s^{(i)}$ indicates the i-th component of the q-dimensional vector s generated by the concat layer. The output of the thresholding layer is binary hash codes.

The flow chart of the optimization objective based on C-NNs architecture is shown in Fig. 2, the loss layer includes Softmax loss and quantization error loss. When classify using the output of the activation layer, the Softmax classification error loss can be generated, denoted by $L_{softmaxloss}$. On the other hand, as the hash code is discrete, we should add the quantization error loss generated by binarizing the continuous value into discrete value to the objective loss function, the formula is as follows

$$L_q = \frac{1}{2}\|h - s\|_2^2 \tag{5}$$

where, $h = (g(s^{(1)}), g(s^{(2)}), ..., g(s^{(q)}))$ indicates the hash codes generated by the thresholding layer, s indicates the continuous values generated by the activation layer, the objective of the loss function is that the outputs of the activation layer are closed to -1 and 1 as far as possible.

The overall loss function of our architecture can be described as

$$L_{loss} = L_{softmaxloss} + \lambda L_q \tag{6}$$

λ indicates the weight factor, which decides the importance of quantization error loss.

After training, given an image as the input, passing into the convolutional sub-network and hashing layer successively, we can get the q-bits binary hash codes directly.

4. EXPERIMENT RESULTS AND ANALYSIS

4.1. Experiment settings

To verify the effectiveness of our method, we evaluate the performance on two image dataset. CIFAR-10 [20] includes 60,000 32×32 color images in 10 classes. NUS-WIDE [21] includes nearly 270,000 images. Each of these images is associated with one or multiple labels. We follow the settings in [22] to use the subset of images associated with the 21 most frequent labels, where each label associates with at least 5,000 images. We resize images of this subset into 256×256.

We compare the retrieval performance of our method with other hashing methods, including the unsupervised hashing methods, ITQ [8]; supervised hashing methods, KSH [11]; the methods based on deep CNNS and hashing, CNNH [17], improved CNNH [18], DCNNH [19].

In CIFAR-10, we select 10,000 images (1,000 images per class) as the testing dataset. For the unsupervised hashing methods, the rest are served as a training dataset. For the supervised hashing methods, we select 5000 images (500 images per class) as the testing dataset. In NUS-WIDE, we randomly select 100 images from each of the selected 21 classes to form a testimg dataset of 2,100 images. For the unsupervised hashing methods, the rest are served as a training dataset. For the supervised hashing methods, we select 500 images from each of the selected 21 classes to form a training dataset.

For the methods based on deep learning and hashing, we directly use the image pixels as input. For the rest of methods, we represent each image in CIFAR-10 by a 512-dimensional GIST vector; and we represent each image in NUS-WIDE by a 500-dimensional bag-of-words vector. To evaluate the retrieval performance, we use Mean Average Precision (MAP) as evaluation metrics.

We implement the experiment based on the open source Caffe framework. In all experiments, the weight factor of quantization loss λ is set as 0.2.

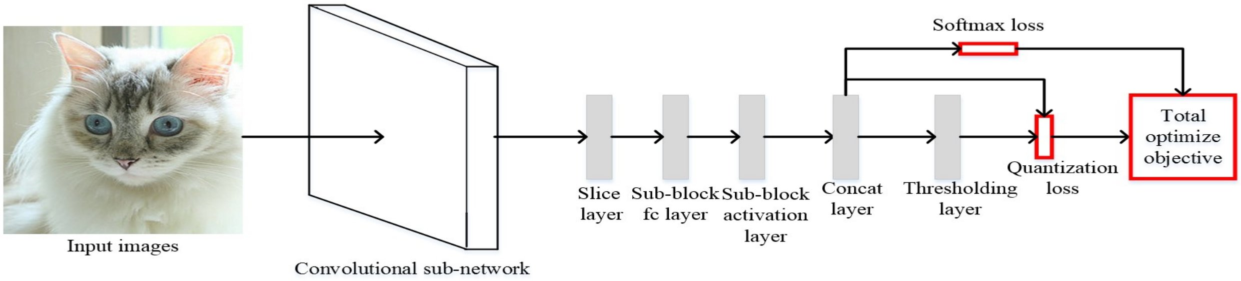

Fig. 2. The architecture of optimized objective of our model

4.2. Experiment analysis

Table 1-2 shows the comparison results of MAP on the two datasets. Compared with the methods based on hashing and traditional hand-crafted features, the MAP of our method is increased by 20%-50%. Compared with the methods based on CNNS and hashing, the MAP of our method is increased by 2%-35%. Compared with improved CNNH, our method uses the label information as the supervised information and considers the quantization error; compared with DCNNH, our method considers the quantization error and the independence of the hash functions. Therefore, the hash codes with stronger representation ability are learned and the MAP of our method is highest.

Table 1. MAP of Hamming ranking w.r.t different numbers of bits on CIFAR-10

Methods	12bits	24bits	32bits	48bits
Ours	0.838	0.845	0.851	0.855
DCNNH	0.816	0.826	0.829	0.832
Improved CNNH	0.552	0.566	0.558	0.581
CNNH	0.465	0.521	0.521	0.532
KSH	0.303	0.337	0.346	0.356
ITQ	0.162	0.169	0.172	0.175

Table 2. MAP of Hamming ranking w.r.t different numbers of bits on NUS-WIDE

Methods	12bits	24bits	32bits	48bits
Ours	0.761	0.767	0.771	0.773
DCNNH	0.741	0.748	0.752	0.752
Improved CNNH	0.674	0.697	0.713	0.715
CNNH	0.623	0.630	0.629	0.625
KSH	0.556	0.572	0.581	0.588
ITQ	0.452	0.468	0.472	0.477

In addition, we compare our method with two other architectures based on CNNS. The first architecture doesn't include the slice layer, thresholding layer and quantization error , which is similar to the model in [19] and construct hash functions without independence and the quantization error. The second architecture doesn't include the thresholding layer and quantization error, which is similar to the improved CNNH in [18] and construct hash functions only with independence.

Table 3-4 show the comparison results on the two image datasets. As can be seen from Table 3-4, the second model only with independence shows a relative increase of 1%-2% than the first model without independence and quantization error. Our method shows a relative increase of 1% than the second model. On the two datasets, the MAP of our method with 24-bits is higher than the first model with 48-bits. That means, in order to achieve the same accuracy, the bits number of our method is shortest. The shorter bits means less occupancy of memory space and faster computing speed, which is especially effective for large scale image retrieval.

Table 3. MAP of Hamming ranking w.r.t different numbers of bits on CIFAR-10

Methods	12bits	24bits	32bits	48bits
First architecture	0.8168	0.8266	0.8291	0.8328
Second architecture	0.8251	0.8308	0.8454	0.8488
Ours	0.8385	0.8450	0.8507	0.8556

Table 4. MAP of Hamming ranking w.r.t different numbers of bits on NUS-WIDE

Methods	12bits	24bits	32bits	48bits
First architecture	0.7413	0.7482	0.7528	0.7560
Second architecture	0.7540	0.7603	0.7644	0.7689
Ours	0.7608	0.7668	0.7710	0.7727

5. CONCLUSION

A method of learning binary hash codes based on CNNS is proposed in this paper, which can be used for large scale image retrieval. We use the label information instead of the image triple as the supervised information, which can reduce the workload of manual label. In addition, by introducing independence constraint between the hash functions and adding quantization error into the loss function, the better hash functions are gained.

6. REFERENCES

[1] D.G.Lowe, “Distinctive image features from scale-invariant keypoints”. International Journal of Computer Vision, 2004, vol.60, pp.91-110.

[2] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection”. Computer Vision and Pattern Recognition,2005, pp.886-893.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet classification with deep convolutional neural networks”. Advances in neural information processing systems, 2012, pp.1097-1105.

[4] Datar M, Immorlica N, Indyk P, et al, “Locality sensitive hashing scheme based on p-stable distributions”. In Proceedings of the ACM Symposium on Computational Geometry, 2004, pp.253-262.

[5] Zhang Lei, Zhang Yongdong, Zhang Dongming, and Tian Qi, “Distribution-Aware Locality Sensitive Hashing”. 19th International Conference On Multimedia Modeling, 2013, pp.395-406.

[6] Kong Weihao and Li Wujun, “Isotropic hashing”. Advances in neural information processing systems, 2012, pp.1646-1654.

[7] Yair Weiss, Antonio Torralba, and Rob Fergus, “Spectral Hashing”. Advances in neural information processing systems, 2009, pp.1753-1760.

[8] Gong Yunchao, S. Lazebnik, A. Gordo, and F. Perronnin, “Iterative quantization: A Procrustean approach to learning binary codes for large-scale image retrieval”. IEEE Transaction on Pattern Analysis and Machine Intelligence, 2012, vol.35, pp.2916-2929.

[9] Wang Jun, Kumar S, and Chang Shih-Fu, “Semi-Supervised hashing for large scale search”. IEEE Transaction on Pattern Analysis and Machine Intelligence, 2012, vol.34, pp.2393-2406.

[10] Brian Kulis and Trevor Darrell, “Learning to hash with binary reconstructive embeddings”. Advances in neural information processing systems, 2009, pp.1042-1052.

[11] Liu Wei, Wang Jun, Ji Rongrong, and Jiang Yugang, “Supervised hashing with kernels”. Computer Vision and Pattern Recognition, Providence, 2012, pp.2074-2081.

[12] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. “Rich feature hierarchies for accurate object detection and semantic segmentation”. Computer Vision and Pattern Recognition, Ohio, Columbus, 2014, pp.580-587.

[13] Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic, “Learning and transferring mid-level image representations using convolutional neural networks”. Computer Vision and Pattern Recognition, 2014, pp.1717-1724.

[14] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, and Stefan Carlsson, “CNN features off-the-shelf: an astounding baseline for recognition”. Computer Vision and Pattern Recognition, 2014, pp.806-813.

[15] Lin Min, Chen Qiang, and Yan Shuicheng, “Network in network”. http://arxiv.org/abs/1312.4400, 2013.

[16] Karen Simonyan, and Andrew Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition”. http://arxiv.org/abs/1409.1556, 2014.

[17] Xia Rongkai, Pan Yan, Lai Hanjiang, Liu Cong, and Yan Shuicheng, “Supervised hashing for image retrieval via image representation learning”. In Proceedings of the AAAI Conference on Artificial Intelligence, 2014, pp.2156-2162.

[18] Lai Hanjiang, Pan Yan, Liu Ye, and Yan Shuicheng, “Simultaneous Feature Learning and Hash Coding With Deep Neural Networks”. Computer Vision and Pattern Recognition, 2015, pp.3270-3278.

[19] Lin Kevin, Yang Huei-Fang, Hsiao Jen-Hao, and Chen Chu-Song, “Deep Learning of Binary Hash Codes for Fast Image Retrieval”. Computer Vision and Pattern Recognition, 2015, pp.27-35.

[20] A.Krizhevsky and G.Hinton, “Learning multiple layers of features from tiny images”. 2009.

[21] Tat-Seng Chua, Jinhui Tang, Richang Hong, Haojie Li, Zhiping Luo, and Yan-Tao Zheng, “NUS-WIDE: A Real-World Web Image Database from National University of Singapore”. Proceedings of the ACM international conference on image and video retrieval, 2009, pp.48.

[22] Liu Wei, Wang Jun, Kumar Sanjiv, and Chang Shih-Fu. “Hashing with graphs”. Proceedings of the 28th International Conference on Machine Learning, 2011, pp.1-8.