ATLANTIS PRESS

Advances in Engineering Research, volume 149

2nd International Conference on Mechanical, Electronic, Control and Automation Engineering (MECAE 2018)

Image retrieval based on ResNet and ITQ

Guijun Wang[1], Baohua Qiang[2], Xianchun Zou[1], Jinyun Lu[3,*]

[1]Southwest University, Chongqing, China

[2]Guangxi Cooperative Innovation Center of cloud computing and Big Data, University of Electronic Technology, Chongqing, Guilin, China

[3]Chongqing Educational Evaluation Institute, Chongqing, China

Keywords: Learning to Hash, Deep Residual Network, Iterative Quantization.

Abstract. In recent years, more and more hash learning methods have been applied to solve large-scale vision problems. It has been shown that learning hash function by using supervised information can boost hashing quality. The state-of-the-art image retrieval hashing methods based on visual features lacks of learning ability, the image expression ability is weak and the efficiency of large-scale image retrieval is low. In this paper, we propose a new supervised hashing framework based on deep Residual Networks and Iterative Quantization hashing. Firstly, we exploit the learning abilities of deep residual network to mine the inherent hidden relationship of image content, extract deep feature descriptors, and increase the visual expression of images Secondly, Iterative Quantization Hashing is applied to learn from the high-dimensional image feature and map into low-dimensional hamming space and achieve compact Hash codes. Finally, image retrieval is accomplished in low-dimensional hamming space. Experimental results of MNIST, CIFAR-10, CIFAR-100 and Caltech 256 show that the expression ability of visual feature is effectively improved and the image retrieval performance is substantially boosted compared with other related methods.

1. Introduction

With the big data era coming, the Internet images, video, audio, text and other heterogeneous data explosive growth. For these rich visual information Picture, how to query and retrieve the images needed by the users in the vast image library conveniently, quickly and accurately, and become a research hot spots in the field of multimedia information retrieval [1]. The image retrieval technology is gradually developed into text-based image retrieval technology based on text-based image retrieval. Content-based image retrieval methods give full play to the advantages of the computer over repetitive tasks, and liberate people from artificial annotations that require a lot of manpower, material and financial resources. After decades of development, content-based image retrieval technology has been widely used in search engines, e-commerce, medicine, and other aspects of life.

Large-scale image retrieval problems, there are high-dimensional, massive data, computing time-consuming and other issues. In order to achieve efficient retrieval of large-scale images, an approximate nearest neighbor (ANN) is proposed [1]. A classic example of solving this problem is based on trees such as kd-tree. However for high-dimensional data, most tree-based methods are significantly affected and their performance is usually reduced to linear search.

In recent years, hash learning has been widely used in information retrieval and other related fields [2]. In the field of large-scale image retrieval, the hash learning maps the high-dimensional features of the image to a compact binary hash codes. Due to the computational efficiency and storage space advantages of Hamming distance, Hashing can solve the problem of storage space, computational complexity and communication overhead on large-scale image retrieval.

In this paper, we investigate the recently widely-applied technique, deep learning and Iterative Quantization (ITQ) [3], to learn hash. For this investigation, we apply deep Residual Networks (ResNet)[4] to extract deep feature description and ITQ to learn hash function for fast image retrieval. Experimental results on the MNIST, CIFAR-10, CIFAR-100 and Caltech 256, The method achieves better performance than other related methods. The empirical and comparative study shows ResNet hash achieves better results than other related methods.

2. Related Work

Hash learning can be divided into unsupervised hash, and supervised hash.

The unsupervised hash method does not take into account the data of the oversight information, including Locality Sensitive Hashing (LSH)[2], spectral hash (SH)[5], Iterative Quantization (ITQ) [3] and so on; LSH generates a set of random linear projection as hash functions. SH first employs PCA on the original data, then calculate the analytical Laplacian eigefunctions along the principal directions. Hash codes are generated according to the projection based on these eigenfunctions.ITQ is also a PCA-based hashing method which first conducts PCA on the original data and then finds an orthogonal matrix to make the variance of each bit maximized and hash bits pairwise uncorrelated. PCA-Random Rotation (PCA-RR)[3] is the basic version of ITQ, which adopts the random orthogonal matrix instead of learning based orthogonal matrix proposed in ITQ.

The supervised hash method utilizes the tag information of the dataset or the similarity point to the information as supervisory information. including the supervised nuclear hash(KSH)[6], BRE[7], and so on. KSH is a kernel based method which maps the data to binary hash codes by maximizing the separability of code inner products between similar and dissimilar pairs. BRE does not require any assumptions on data distribution, and directly learns the hash functions by minimizing the reconstruction error between the distances in the original feature space and the Hamming distances in the embedded binary space. Minimal Loss Hashing (MLH)[8]: By treating the hash code as the latent variables, MLH adopts the structured prediction formulation for hash learning.

Deep Learning: Deep learning aims to learn hierarchical feature representations by building high-level features from raw data. In recent years, a variety of deep learning algorithms have been proposed in computer vision and machine learning [9,10], and some of them have successfully applied to many visual analysis applications image classification, object detection, action recognition, face verification, and visual tracking.

Deep learning has been used for image retrieval[11-14]. Very recently, Xia *et al* [15]proposed CNNH, a supervised hashing method in which the learning process is decomposed into a stage of learning approximate hash codes from the supervised information, and a stage, followed by, of simultaneously learning hash functions and image representations based on the learned approximate hash codes. Lai *et al* [14]developed a “one-stage” supervised hashing method for image retrieval, which generates bitwise hash codes for images via a carefully designed deep architecture. The proposed deep architecture uses a triplet ranking loss designed to preserve relative similarities. However, using triples as supervised information in his paper, the quality of triples directly influence the precision of retrieval and large work to select. Lin *et al* [9]present a simple yet effective deep learning framework to create the hash-like binary codes for fast image retrieval. It simultaneously learn domain specific image representations and a set of hash-like functions However, this method did not consider the quantization error while transforming the sequent value into binary and the independence of hash functions.

ResNet have been successfully applied to a wide range of problems, such as image recognition[4,10,16],object detection[4]and so on. ResNet can rather fast decelerate the training of neural network, and descriptors send layer by layer, the descriptors r representing ability of output is guaranteed. The using of batch normalization and global pool can lead to better generalize network. Inspired by these successful applications. ITQ can get a better projection matrix by approximating the smallest error between real data and hash codes. In this work we in virtue of the advantage of ResNet and hash learning, proposed ResNet network and ITQ to learn binary hash codes applied to large-scale image retrieval.

Based on the above research, this paper proposes an image retrieval method based on ResNet and ITQ,this method referred to as RITQ. The basic idea is to introduce a deep residual network to learn the training data, and to use its special network, the hash function needs to satisfy the constraints of independence and least quantization error. This paper presents a binary hash function learning algorithm that takes into account the independence between hash functions and the quantization error caused by thresholding

3. Method

In this sectionwe solve the problem of learning binary codes . Firstly we will apply the improved deep residual network extract deep feature description, and then perform binary quantization in the resulting space.

3.1 Based deep residual network extract deep feature description.

Deep residual network was proposed in 2015 and showed good performance in computer target detection, image classification and image segmentation. Compared with other networks, Resnet can decelerate neural network training rapidly. Passing, more to ensure that the output of the characteristics of expression, the use of normalization and global avg_pool more generalization of the network. The size of the input image of the neural network is 224 * 224. The output is size of 256 deep features description.

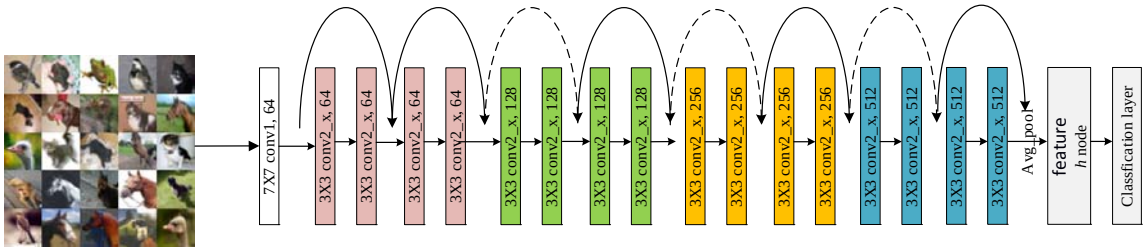

Figure1 Extracting features description through ResNet training

Table I. Architectures for ResNet framework. Building blocks are shown in brackets, with the numbers of blocks stacked. Down-sampling is performed by conv2_1, conv3_1, and conv4_1 with a stride of 2.

Layer name	Con1	Con2_x	Con3_x	Con4_x	Avg_pool	Feature layer	Classification layer
Configuration	$[3\times3,16]$	$\begin{bmatrix}3\times3,16\\3\times3,16\end{bmatrix}\times5$	$\begin{bmatrix}3\times3,32\\3\times3,32\end{bmatrix}\times5$	$\begin{bmatrix}3\times3,64\\3\times3,64\end{bmatrix}\times5$			

3.2 Iterative Quantization Learning Hash Codes

ITQ method before constructing the objective function, firstly reduce the number of features, extract the main information and reduce the training time. In this paper, the principal component analysis method is used to reduce the dimensionality of the extracted features K_n , suppose the matrix of eigenvectors of dimensionality reduction $W \in R^{m\times r}$ is reduced after dimensionality K_nW .

By looking for the optimal orthogonal matrix R, ITQ can make a sample with a small Euclidean distance quantized into a hash code with a small Hamming distance. Quantifying these samples, the samples with small Euclidean distance are quantized to different binary hash codes. By multiplying the samples with the random orthogonal matrix, we can find that the samples are rotated to find the most suitable orthogonal matrix , It is possible to quantify the sample with a small Euclidean distance to the same hash code. In order to get the least loss of quantization error, the hash code is to minimize the error between the real sample data and the hash code. Therefore build the objective function

$$\min J(B,R) = \left\|B - K_nWR\right\|_F^2 \tag{1}$$

Where $\left\|\bullet\right\|_F$ denotes the F-norm.

In order to minimize the objective function and obtain a suitable orthogonal matrix R. Using iterative quantization method, firstly using the symbol function to get B, then using the optimization method to get R, the iterative quantization steps are as follows:

Step 1: Initialization R is a random orthogonal matrix.

Step 2: Fixed R, Update B. The extracted eigenvalues are: $B = \mathrm{sgn}(K_nR)$.

Step 3: Fix B and update R. Minimize the objective function belongs to the Orthogonal Procrustes Problem, seeking a Singular Value Decomposition (SVD)[17] for the matrix $B^T K_n$: $SVD\left[B^T K_n \right] = U\Sigma V^T$,where $SVD[\bullet]$ represents the singular value decomposition, get $R = VU^T$.

Step 4: Loop through steps 2 and 3.The objective function J is minimized by iteratively updating B and R.

4. Experimental results

In this section, we validate our RITQ hashing learning framework on several public datasets of image retrieval, including MNIST, CIFAR-10, CIFAR-100 and Caltech 256. For each dataset, the images are split into a training set and a query set. We use the training set to learn the network parameters and use the query set to compare the competing methods. Note that, in all of the experiments, the query image is searched within the query set itself by applying the leave-one-out procedure. We compare our methods with eight state-of-the-art approaches:

4.1 Data Sets

We evaluate our method on two standard large image datasets with semantic labels: MNIST, CIFAR-10, CIFAR-100 and Caltech 256.

MNIST Dataset[39]contains 10 categories of the handwritten digits form 0 to 9. There are 60k training images, and 10k test images. All the digits are normalized to gray-scale images with size 28×28 . We use 10K images as the query set and the other 60K as the training samples.

CIFAR-10 dataset [40]consists of 60,k 32×32 color images which are categorized into 10 classes (6k images per class). It is a single-label dataset in which each image belongs to one of the ten classes. We use 10K images as the query set and the other 50K as the training samples.

CIFAR-100 Dataset[40]t contains 100 object categories and each class consists of 6,00 images, resulting in a total of 60,000 images. The dataset is split into training and query sets, with 50,000 and 10,000 images respectively.

Caltech 256 Dataset[41] contains a total of 30,607 images, split between 256 distinct object categories and a background category. We randomly selected 70% as training sets, the rest as the query sets.

For hashing methods which use hand-crafted features, we represent each image in datasets by a 512-dimensional GIST vector. For deep hashing methods, we first resize all images to be 224×224 pixels and then directly use the raw image pixels as input. For data pre-processing, we follow the standard way of feature normalization by making each dimension of the feature vectors to have zero mean and equal variance.

4.2 Experimental Baselines

We compare our method with several state-of-the-art hashing methods. These methods can be categorized into five classes:

Data-dependent hashing methods with hand-crafted features, including locality-sensitive hash(LSH)[2], Locality-Sensitive Binary Codes from Shift-Invariant Kernels(SKLSH)[18].

• Unsupervised hashing methods with hand-crafted features, including principal component analysis(PCA), spectral hashing(SH)[5], spherical hashing (SpH)[19], Iterative Quantization(ITQ)[3], principal component analysis -random rotation(PCA-RR)[3], Density sensitive hashing(DSH)[20].

• Supervised hashing methods with Deep hashing methods, including Deep Learning of Binary Hash Codes (DLBH)[9].

To evaluate the quality of hashing, we use evalution metrics: Mean Average Precision (MAP).

4.3 Results of Search Accuracies

In table 2 and table 3, we report the Map results with different code lengths on MNIST, CIFAR10, CIFAR100, Caltech 256. Our RITQ hashing is abbreviated as RNH. In these tables, the best results are in boldface. From these tables, we can see that RITQ hashing achieves better results than data-dependent hashing method, unsupervised hashing methods and deep hashing in most cases. For example, with respect to MAP, compared to the corresponding second best competitor(DLBH), the proposed method shows a relative increase of 5.3 % ~ 14.7 % ,9% ~ 19.2% , 42% ~ 50% on CIFAR 10 , CIFAR 100, Caltech 256,respectively. RITQ hashing shows slightly but consistently better search accuracies than

other methods, which verifies that incorporating both the approximate hash codes and image tags in training helps to learn a better shared image representation and enhance the hashing performance. These results verify that using Residual Networks simultaneously learning useful representation of images and ITQ get hash codes of preserving similarities can benefit each other.

Table II. Image retrieval results (Map) with various number of bits on the MNIST dataset and the CIFAR 10. The scale of test query set is 1k (1k per class). The proposed method outperforms the state-of-the methods.

Method	MNIST					CIFAR 10				
Bits	8	16	48	64	128	8	16	48	64	128
DLBH	0.974	0.983	0.973	0.993	0.988	0.684	0.715	0.715	0.741	0.748
RITQ	0.972	0.974	0.989	0.985	0.990	0.741	0.761	0.862	0.859	0.811
ITQ	0.649	0.779	0.874	0.890	0.908	0.209	0.240	0.280	0.288	0.302
LSH	0.345	0.504	0.751	0.768	0.827	0.154	0.182	0.220	0.238	0.262
PCAH	0.606	0.723	0.737	0.732	0.703	0.194	0.214	0.215	0.210	0.194
SH	0.105	0.105	0.105	0.105	0.105	0.106	0.106	0.106	0.106	0.106
SKLSH	0.237	0.370	0.512	0.539	0.636	0.139	0.147	0.158	0.165	0.192
PCA-	0.513	0.682	0.796	0.819	0.859	0.205	0.230	0.265	0.273	0.290
DSH	0.365	0.565	0.603	0.598	0.660	0.184	0.214	0.243	0.250	0.268
SpH	0.381	0.523	0.545	0.595	0.690	0.1 Table 2	0.181	0.191	0.195	0.216

Table III. Image retrieval results (Map) with various number of bits on the CIFAR 100 dataset and the Caltech 256. The scale of test query set are 1k (1k per class), 10(10 per class), respectively. The proposed method outperforms the state-of-the methods.

Method	CIFAR 100					Caltech 256				
Bits	8	16	48	64	128	8	16	48	64	128
DLBH	0.029	0.047	0.384	0.365	0.398	0.076	0.094	0.109	0.112	0.114
RITQ	0.224	0.283	0.462	0.512	0.583	0.443	0.615	0.634	0.688	0.606
ITQ	0.064	0.112	0.196	0.208	0.245	0.037	0.066	0.104	0.113	0.128
LSH	0.044	0.074	0.151	0.176	0.221	0.020	0.035	0.080	0.089	0.116
PCAH	0.066	0.121	0.182	0.189	0.205	0.031	0.059	0.091	0.097	0.108
SH	0.029	0.029	0.029	0.029	0.029	0.009	0.009	0.009	0.009	0.009
SKLSH	0.040	0.052	0.072	0.087	0.120	0.006	0.006	0.006	0.007	0.007
PCA-RR	0.065	0.121	0.199	0.214	0.245	0.030	0.059	0.101	0.108	0.129
DSH	0.054	0.086	0.129	0.151	0.188	0.022	0.041	0.070	0.076	0.087
SpH	0.057	0.094	0.131	0.148	0.182	0.006	0.006	0.006	0.005	0.005

5. Conclusion

In this paper, we have proposed a novel supervised hashing method for image retrieval based on ResNet and ITQ, called RITQ, Hashing generates bitwise hash codes for images via a carefully designed deep architecture. RITQ hashing can learn better codes than other methods without end-to-end architecture. Experiments on real datasets show that RITQ hashing can outperform other methods to achieve the state-of-the-art performance in image retrieval applications.

Acknowledgements

Correspondence should be addressed to Jingyun Lu; jimmloo@qq.cn.This work is Supported by Guangxi Cooperative Innovation Center of cloud computing and Big Data (No YD16E04). Guangxi Colleges and Universities Key Laboratory of cloud computing and complex systems

References

[1] Andoni, A., & Indyk, P. 2008. “Near-optimal hashing algorithms for approximate nearest neighbor

in high dimensions." Communications of the Acm, 511, 459-468.
[2] Gionis, A., Indyk, P., & Motwani, R. 1999. "Similarity Search in High Dimensions via Hashing." International Conference on Very Large Data Bases Vol.8, pp.518--529. Morgan Kaufmann Publishers Inc.
[3] Gong, Y., Lazebnik, S., Gordo, A., & Perronnin, F. 2013. Iterative quantization: a procrustean approach to learning binary codes for large-scale image retrieval." IEEE Transactions on Pattern Analysis & Machine Intelligence, 3512, 2916.
[4] He, K., Zhang, X., Ren, S., & Sun, J. 2015. "Deep residual learning for image recognition." 770-778.
[5] Weiss, Y., Torralba, A., & Fergus, R. 2008. "Spectral hashing." International Conference on Neural Information Processing Systems Vol.282, pp.1753-1760. Curran Associates Inc.
[6] Liu, W., Wang, J., Ji, R., & Jiang, Y. G. 2012. "Supervised hashing with kernels." IEEE Conference on Computer Vision and Pattern Recognition Vol.157, pp.2074-2081. IEEE Computer Society.
[7] Kulis, B., & Darrell, T. 2009. "Learning to hash with binary reconstructive embeddings." International Conference on Neural Information Processing Systems pp.1042-1050. Curran Associates Inc.
[8] Norouzi, M., & Fleet, D. J. 2011. "Minimal loss hashing for compact binary codes." International Conference on International Conference on Machine Learning pp.353-360. Omnipress.
[9] Lin, K., Yang, H. F., Hsiao, J. H., & Chen, C. S. 2015. "Deep learning of binary hash codes for fast image retrieval." Computer Vision and Pattern Recognition Workshops pp.27-35. IEEE.
[10] He, K., Zhang, X., Ren, S., & Sun, J. 2016. "Identity mappings in deep residual networks." 630-645.
[11] Liong, V. E., Lu, J., Wang, G., Moulin, P., & Zhou, J. 2015. "Deep hashing for compact binary codes learning." Computer Vision and Pattern Recognition pp.2475-2483. IEEE.
[12] Li, W. J., Wang, S., & Kang, W. C. 2016. "Feature learning based deep supervised hashing with pairwise labels." International Joint Conference on Artificial Intelligence pp.1711-1717. AAAI Press.
[13] Zhang, R., Lin, L., Zhang, R., Zuo, W., & Zhang, L. 2015. "Bit-scalable deep hashing with regularized similarity learning for image retrieval and person re-identification." IEEE Transactions on Image Processing, 2412, 4766-4779.
[14] Lai, H., Pan, Y., Liu, Y., & Yan, S. 2015. "Simultaneous feature learning and hash coding with deep neural networks." Computer Vision and Pattern Recognition pp.3270-3278. IEEE.
[15] Xia, R., Pan, Y., Lai, H., Liu, C., & Yan, S. 2014. "Supervised hashing for image retrieval via image representation learning." AAAI Conference on Artificial Intelligence.
[16] Zagoruyko, S., & Komodakis, N. 2016. "Wide residual networks."
[17] Alter, O., Brown, P. O., & Botstein, D. 2000. "Singular value decomposition for genome-wide expression data processing and modeling". Proceedings of the National Academy of Sciences of the United States of America, 9718, 10101.
[18] Raginsky, M., & Lazebnik, S. 2009. "Locality-sensitive binary codes from shift-invariant kernels". Advances in Neural Information Processing Systems 22:, Conference on Neural Information Processing Systems 2009. Proceedings of A Meeting Held 7-10 December 2009, Vancouver, British Columbia, Canada pp.1509-1517. DBLP.
[19] Heo, J. P., Lee, Y., He, J., & Chang, S. F. 2012. "Spherical hashing". Computer Vision and Pattern Recognition Vol.157, pp.2957-2964. IEEE.
[20] Jin, Z., Li, C., Lin, Y., & Cai, D. 2014. "Density sensitive hashing". IEEE Transactions on Cybernetics, 448, 1362.