Prompt Image Search with Deep Convolutional Neural Network via Efficient Hashing Code and Addictive Latent Semantic Layer

Jun-yi Li, Jian-hua Li

Shanghai Jiaotong University Electrical and Electronic Engineering College, China
Leejy2006@163.com, lijh888@sjtu.edu.cn

Abstract

As we know that the nearest neighbor search is a good and effective method for good-sized image search. This paper indicates a vision learning framework to generate compact binary hash codes for quick vision search after knowing the recent benefits of convolution neural networks (CNN). Our concept is that binary codes can be obtained using a hidden layer to present some latent concepts dominating the class labels with usable data labels. CNN also can be used to learn image representations. Binary code learning is required for other supervised methods. However, our method is effective in obtaining hash codes and image representations and we use pretrained model from googlenet for incremental learning so it is suitable for good-sized dataset. It is demonstrated in our experiment that this method is better than some most advanced hashing algorithms in MINIST, NUS-WIDE and CIFAR-10 dataset. The scalability and efficiency still needs to be further investigated in a good-sized dataset.

Keywords: Convolutional neural networks, Nearest neighbor search, hidden layer, LSH, Supervised learning

1 Introduction

Image retrieval based on content aims at searching similar images through image content analysis, so for such a task, it is quite important for image representations and similarity measure. There are many provocative issues in the research and one of it is about the relationship between the pixel-level information and high-level semantic [7, 14, 18, 25, 27]. Some manual features proposed to represent the listed image [2, 7, 19, 22], but such visual descriptors are not well-performed before the recent discovery in deep learning. The deep CNN greatly shows such results in different vision tasks like object detection of objects, and classification of images. Segmentation is a fact that has been shown by recent studies [21, 23]. Deep neural network obtains ability to study the rich mid-level image representations, and because of that we have got the accomplishment.

Feature vectors on the 7th layer in search of image was used by Krizhevsky et al. [14, 34], which showed excellent result on ImageNet, during learning of mid-level image descriptors. But, since the features of CNN are high-dimensional and it is not good enough to directly calculate similarities between such two vectors, Babenko [1] suggests using PCA and distinctive dimensionality reduction to make the features of CNN compact, and finally they have achieved a brilliant result.

In CBIR, image representations as well as computational cost are important. Because of the recent growth of visual contents, people need fast search in a large database. Many studies aim at efficiently retrieving the relevant data from the good-sized database. Because of this high-computational cost, common linear search is inappropriate for searching in a large database. The best way for hashing-based is nearest neiborhood search [6, 15, 20, 28-30]. The high-dimensional features are reflected to a lower dimensional room by these methods, which will produce those compact binary codes. Thanks to this generated binary codes, quick image search is workable through binary pattern matching or Hamming distance method. The result is the dramatic reduction of the computational cost as well as further optimization of the efficiency of the searches. Part of the methods is contained using similarity matrix to describe this relationship between images and the similarity information is used to study functions of hash. But, to build the matrix and produce the codes while handling a good-sized dataset is not easy.

In recent years, deep hashing techniques [3, 11, 13, 33] have developed quickly, and have achieved good results, though it has made harsh requirement in hardware condition and training data scale. For example, the core idea proposed by paper [30] is deep hash algorithm by CNN, which take the value of whether two image samples are similar for the value of similarity matrix element, by decomposing the matrix and then it get the binary hash code of the sample, and finally use it to predict the result. The lost function in this paper used the cross-entropy function. This

*Corresponding Author: Jun-yi Li; E-mail: Leejy2006@163.com
DOI: 10.3966/160792642018051903030

method has made much improvement in performance by artificial hand-craft feature. However, this method has not been designed by end to end method, therefore it could not update the binary code and play to our strength. Professor Yan Shui-cheng and Pan Yan proposed NIN (Network in network), which has advantage over CNHH [30]. Paper [37] proposed DSRH (Deep Semantic Ranking Hashing) method, directly focus on network ranking results. The authors directly adopt the convex upper bound for optimization. Paper [38] proposed DRSCH algorithm (Deep Regularized Similarity Comparison Hashing DRSCH), which use weighted hamming distance instead of hamming distance and use tuple lost function, taking account of the location of pair-wise images. In 2016, CPVR paper [39] adopt regularization item to constraint the network output, which make it closer to two valued encoding. Based on the development of deep learning, whether we can use the benefit of deep CNN to reach hashing is still a question. Is it possible for us to produce this binary compact codes directly from this deep CNN rather than use the pair-wised learning method? To answer these questions, a deep CNN model which could simultaneously study image representations and binary codes has been proposed. This data are labeled is the premise, which mean it is specially made for supervised learning. Furthermore, it is debated that when a powerful studying model like deep CNN is adopted when this data labels are available, by using some hidden layer which are utilized to represent these potential image concepts with binary activation functions like Relu (Rectified linear unit) and sigmoid, which dominate these image category labels in buildings. Our method is different from traditional supervised methods of other kind (e.g [30]). And if we think about data label but request pairs of input to this intended studying procedure, the binary codes can be achieved. That is, by making use of the incremental studying character (through stochastic gradient descent) of deep CNN, this method studies binary hashing codes in the manner of point wise. Efficient-retrieval learning feature is brought by employment of deep architecture. Comparing with conventional methods, this method is appropriate for good-sized database.

The characteristics of the method are as follows:

A simple and effective supervised learning framework is introduced for fast image retrieval.

Deep CNN can immediately obtain domain specific image representation and hashing-like function for prompt image search with small modifications to the network model.

The proposed method is better than all of the most advanced works stored in public database MNIST and CIFAR-10. 12% precision of CIFAR10 dataset and 0.75% precision of MNIST dataset have been improved from the previous best retrieval performance.

Comparing with conventional pair-wised methods, this method adopts the point wise manner to learn binary hashing codes. Regarding the data size, it is easily measurable.

From the Figure 1, we can see that our model is composed of three convolution-pooling layers, and the added latent semantic layer is set between the orange signed layer (full-connected layer) and the red signed layer (output layer).

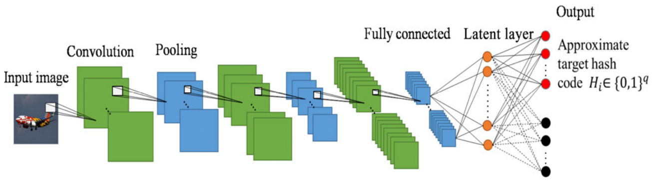

Figure 1. The framework of image retrieval framework through classified deep search

Below is the organization of this paper: We elaborate on the method in part 2, with experimental results shown in part 3, and conclusion in the final part.

Method

As shown in Figure 1, the devised framework contains three major elements, among which the first one is the supervision-based pre-training on this good-sized ImageNet dataset [14]. Adjusting this network with the latent layer for simultaneous learning of feature representation in specific domain and hash-like function is the second one. The third one retrieves images similar to this query one through this suggested classified search. Zhong et al. [34] proposed this pre-trained CNN model in the google net library, and we used the three layers for pre-training. Below we'll describe the method for learning binary codes.

1.1 Studying Hash-like Binary Codes

That the feature activations of layers F4 induced by this input image could be regarded as this visual signatures has been showed by recent studies [1, 5, 7, 14]. It shows great improvement in various aspects, such as image classification, and search based on mid-level image representation. But, the signatures are vectors with high dimension, which are not efficient for image search in large corpus. In order to promote good quality image search, it needs to convert this feature vectors into binary codes to reduce computational cost. Employing hashing or hamming distance, we can compare the binary compact codes in a fast way.

This paper aims to explore domain specific image representation and hash-like (or binary coded) function. It is assumed that the classification layer of F5 is dependent on a batch of hidden attributes. The status of these attribute is on or off. According to other viewpoints, any image generating similar or same binary activation should possess likewise labels. We embed the latent layer H between F4 and F5 as shown in the middle row of Table 1 to implement this idea. As a completely connected layer, neuron activities of latent layer H are subject to the regulation of the subsequent layer F4, which is classification of achieves and encoding of semantics. The latent hash layer H has to provide an abstraction of rich features from F4, and connect the learnt mid-level features with image high-end semantics information. In our method, we activate the neurons from the latent layer H using relu function, thus activation is almost equal to {-1, 1}.

We adjust the intended network on this target-domain database through reverse transferring to realize domain adaptation. Besides, we also set the original weight of deep CNN, which is the same with those weights from the pretraining model. Weights of classification layer F5 and latent layer H are set at random. The initial random weight of latent layer H is the same with that of LSH [6], which is obtained based on random predictions of setting up hashing bits. The codes in this case are obtained from LSH and others appropriate for supervision-based deep network studying. With striking changes in a deep CNN model, this target model studies visual descriptors with domain specific and hashing-like function with efficient image search.

1.2 Image Search with Hierachical Deep Search

Zeiler [32] made an analysis on this deep CNN. It is indicted that we learn the semantic information from the deep layers and the local image descriptors from the shallow layers. We applied two-step search method to achieve prompt and effective image search.

One-step Search: if given one image, we firstly select the output of hidden layer for image hash sign, and then we can get binary hash coding by threshold binary. This we can set the hash threshold value 0.5, if the hash(x) is larger than 0.5, we can define it 1, otherwise 0. See formula 1. Therefore, if the inquiry image of hash value is H_v, it can be compared with existing image hash values. Then, we could obtain the hamming distance between H_v and $H_i \in K_H$ $K=\{I_1,I_2,...I_n\}$, $K_H=\{H_1,H_2,...H_n\}$, then compute the candidate dataset P.

$$H^k = \begin{cases} 1 & \text{Code}^k(H) \geq 0.5 \\ 0 & \text{else.} \end{cases} \quad \textbf{(1)}$$

Two-step Search: In the first step, we have obtained the candidate P pooling dataset P_1,P_K, and we defined similarity distance of the inquiry image I and our P candidate dataset by cosine distance. The threshold is between [-1,1]. Here the value of cosine value larger, the similarity of two images is higher. And finally we can compute the ranking Top K images.

2 Experimental Results

2.1 Datasets

MNIST Dataset [16] is composed of ten sets of handwritten digits from 0 to 9, of which we can find 10,000 test images and 60,000 training images, with all normalized to gray scale image in size 28×28.

CIFAR-10 Dataset [12] is composed of ten objects, each containing up to 6,000 images. That is to say, there are 60,000 images in total. With 50,000 and 10,000 images respectively, the dataset is divided into training and test set.

NUS-WIDE Dataset contains 270000 images in total and 81 categories. By catching the images from the websites, we collected the dataset. Each image has been labeled with a category. Each sample image is readjusted to 64 * 64 to diminish the impact of complexity, as shown in Figure 2.

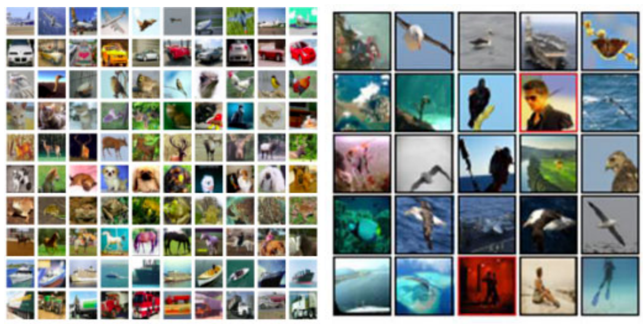

Figure 2. Examples of dataset CIFAR-10 (left) and NUS-WIDE (right)

For the experiments of MNIST and CIFAR-10, those similar images through the learned binary codes are searched so that we can compare them with others. For evaluation, a standard [4] based on ranking is used.

2.2 Results on MNIST Dataset

Image classification performance. In order to accommodate our deep learning framework to MNIST data domain, we modified the layer F5 to 10 classes for the prediction of ten digit classes. The quantity of neurons h in this latent layer is set to 48. The neurons are used to investigate the influence of latent semantic layer in this DCNN before stochastic gradient descent (SGD) is set to train CNN in MNIST database. We train this network for 10,000 repetitions with the learning rate of 0.001.

Several most advanced methods [9, 17, 31] in Table 1 are compared with our result. This method with 48 latent nodes at trains 0.50% error rate and is better than most of the other methods. We'd like to mention that this model is made specially for image search while others are optimized for one classification task via modification of a network. To give you an example, the [31] proposed activation function is used to show the accuracy of dropout in model averaging technique. And there's some other famous work, one is Network in Network (NIN) [17], strengthening this identification of local patches through multi-layer perception and avoids over fitting by using the international average pooling, so that the fully connected layers are left behind.

Table 1. Comparison (Error %) of classification in MNIST dataset

Method	Test Error (%)
2-CNN [31]	0.53
Stochastic Pooling	0.47
NIN [17]	0.47
Conv. maxout[9]	0.45
Our method (48 bit code)	0.485

Images retrieval performance. In the experiment, the retrieve evaluation is used to retrieves these similar images with 48 bits binary code. A total of 1,000 query images are randomly selected from this testing dataset to search similar ones in training set. In order to assess the retrieval performance of our method, our results are compared with some advanced hashing methods, including supervised [35] and unsupervised methods [10, 36]. It is shown in Figure 3 and Figure 4 that the retrieval precision of various methods is correlated with the total number of searched images. Despite the number of images retrieved, our method works quite well (98.2% retrieval precision). In addition, this method improves the precision from 97.5% achieved by CNNH+ to 98.2%. And this learns the hashing functions through dissolution of this similarity information in pairs. And this indicates that this point-wised method only asking for class labels is effective.

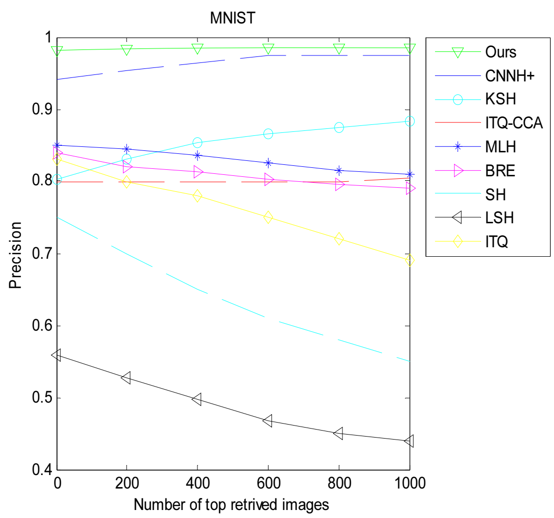

Figure 3. Performance of image retrieval on MINIST

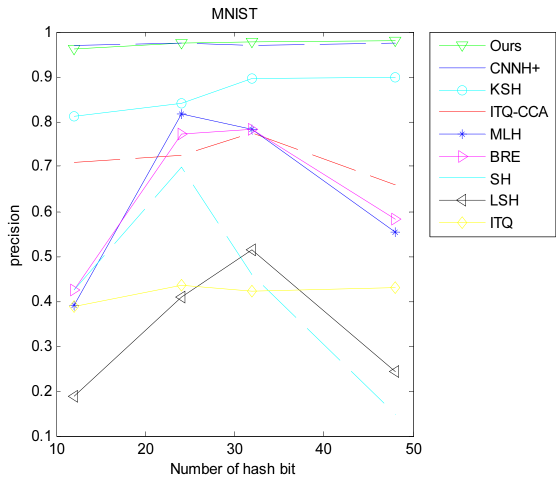

Figure 4. Precision of different hash bit on MNIST

In this experiment, our method has made improvement on the precision, and it can be effective only considering the class label. So we further analyze and compare the learned hash code with different hash bit code and return the most similar images.

From figure 5, we can see that with dataset MNIST, the performance of ours is better than other three hand craft features such as KSH with LLC feature (Locality-constrained linear coding), KSH with 1024-d gist in different hash bit.

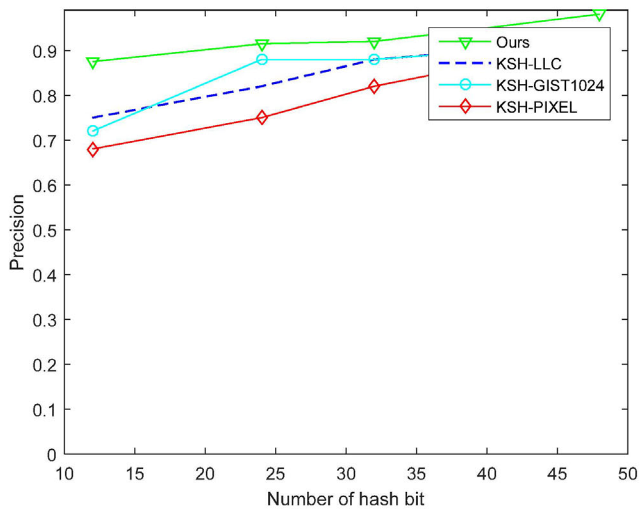

Figure 5. Comparison result with different features on MNIST

Finally, with this dataset, we compare the method with the liner search in time efficiency. From Figure 6 it can be seen that with 32 bit hash code, we compute the feature learning with 29 seconds and finally finish one search task has cost 64 seconds while linear search will cost 1435 seconds, which shorten 20 times than before, which has increased in efficiency obviously.

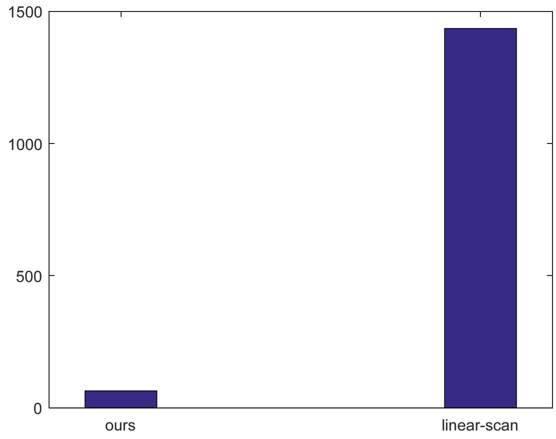

Figure 6. CPU time cost comparison with different search method on MNIST

2.3 Result from CIFAR 10 Dataset

Image classification performance. We modify F5 to 10-way to predict ten object categories for the purpose of transferring the deep CNN to the domain of CIFAR-10, and h is also set as 48. In this case the network model on this CIFAR-10 database is adjusted with about 87.5% testing accuracy after 50, 000 repeated training. As can be seen in Table 2, the method has better performance than [19, 26, 30-31], which include KSH (kernel sensitive hashing), CNNH+ (Convolutional Neural Network Hashing+), ITQ-CCA ((Iterative quantization-canonical correlation analysis) [8], MLH (Minimal loss hashing), BRE (Binary Reconstruction Embeddings), SH (Spectral Hashing) [24], LSH (Locality-Sensitive Hashing), that also demonstrate that the deep CNN with binary embedded will not change the effect.

Table 2. Comparison (mAP,%) of classification accuracy on the CIFAR-10 dataset

Method	Accuracy (%)
Stochastic Pooling [31]	84.87
CNN [26]	85.02
CNNH[30]	53.2
AlexNet + Fine-tuning [14]	89
KSH[19]	35.6
NIN [17]	91.2
Ours 48 nodes	87.5

Image retrieval performance. Compared with other hashing algorithms, the evaluation method is unified, and it searches the related images based on Hamming distance and 48 bits binary codes. The precision curves are shown in Figure 7, and they are related to various number of this top retrieved samples. Performance of other unsupervised and supervised methods are all inferior than this method. What's more, the precision it gets is 71%, and it varies the number of retrieved images and improves the performance by more than 12% compared to CNNH+. According to the outcome, it is a practical method to use a latent layer to represent these invisible concepts for studying of efficient binary codes.

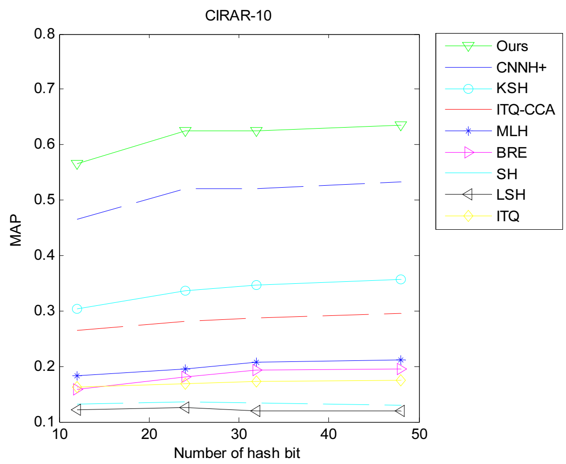

Figure 7. Performance comparison of different hash bit on CIRAR-10

As we can see in Figure 6, our method is better than most traditional methods including supervised and unsupervised hashing methods (such as KSH, MLH, ITQ-CCA [8], SH [4], BRE), indicating that this performance is not effected largely even with buried binary latent layer in the deep CNN.

The searching result is shown in Figure 8. The proposed latent binary codes search images with relevant category, relevant appearance. We retrieve more appearance-relevant images by increasing this bit numbers from h = 48 according to our sight checking based on experience.

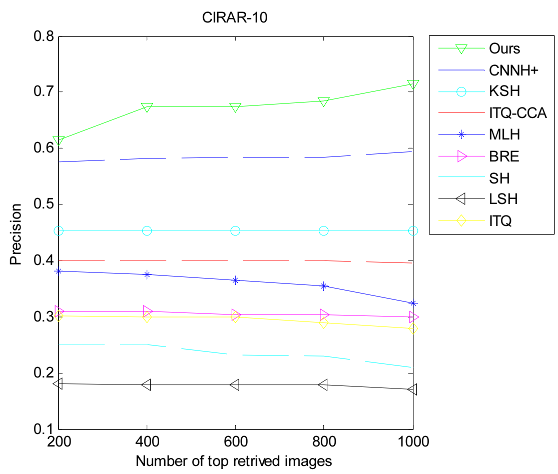

Figure 8. Performance comparison of different top retrieval image numbers on CIFAR-10

From Figure 9 we can see that with dataset CIFAR-10, the performance of ours is better than other three manual design features such as KSH with LLC feature, KSH with 1024-d gist in different hash bit.

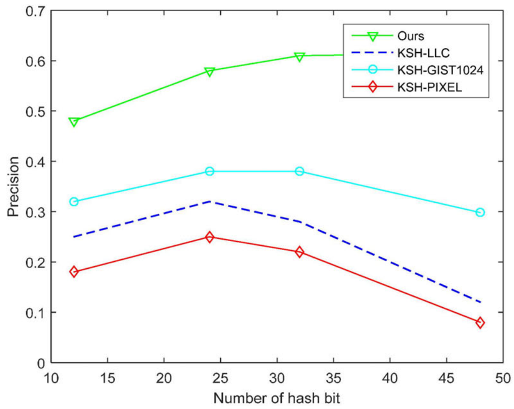

Figure 9. Comparison result with different features on CIFAR-10

Finally, with this dataset, we compare the method with the liner search in time efficiency. From figure 10it can be seen that with 32 bit hash code, we compute the feature learning with 33 seconds and finally finish one search task has cost 78 seconds while linear search will cost 1544 seconds, which shorten 20 times than before, which has increased in efficiency obviously.

2.4 Results on NUS-WIDE Dataset

Image classification performance. We further test it on the good-sized NUS-WIDE dataset to show the scalability and efficacy of our method. This dataset is composed of efficient product images that are uneven and the background is noisy with different person posing in it.

In the classification layer, we set *h* in this latent layer as 48, and the neuron number as 30. Under such condition, we adjust our network with the whole NUS-WIDE dataset. After 100000 repeated trainings; the accuracy of our method reaches to 78.75% on this task of 30 classes

Images retrieval performance. This experiment shows that the method can be used to study efficient deep binary codes for this dataset of million data. It is difficult to achieve this by using original pair-wised-data methods because of the complexity of large time and storage. Figure 10 shows our searching results.

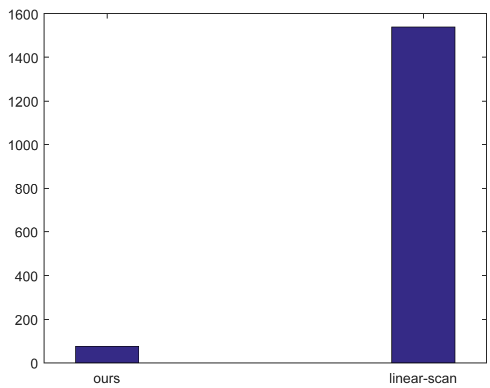

Figure 10. CPU time cost comparison with different search method on CIFAR-10

To illustrate the flexibility and effect, we will further test retrieval on the NUS-WIDE. The Figure 11 show that the precision with the top retrieved images from NUS-WIDE dataset, in our experiment, we still use 32 bit hash code value and hamming distance to measure the related image retrieval performance. The retrieval has selected randomly 3000 inquiry images from the test dataset (30 categories, each category for 100 images) to retrieve the related image from the training set. To evaluate the retrieval effect, we will compare ours' method with several advanced hash method and other unsupervised method ([7, 19, 30) for comparison, the Figure 10 shows the retrieval precision with different retrieval number and methods. We can see that our algorithm has stability performance (62%-71.75%) under different retrieval image numbers.

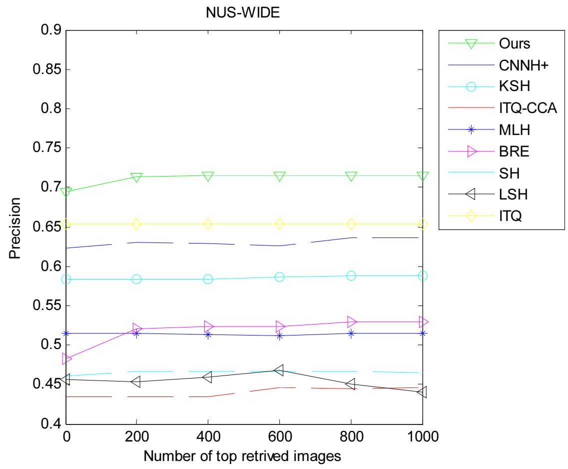

Figure 11. Precision comparison with different retrieval images on NUS-WIDE

To illustrate the flexibility and effect, we will further test retrieval on the NUS-WIDE. The Figure 12 show that the MAP with the number of hash bit from NUS-WIDE dataset. The MAP performance has stable performance with the hash bit increase.

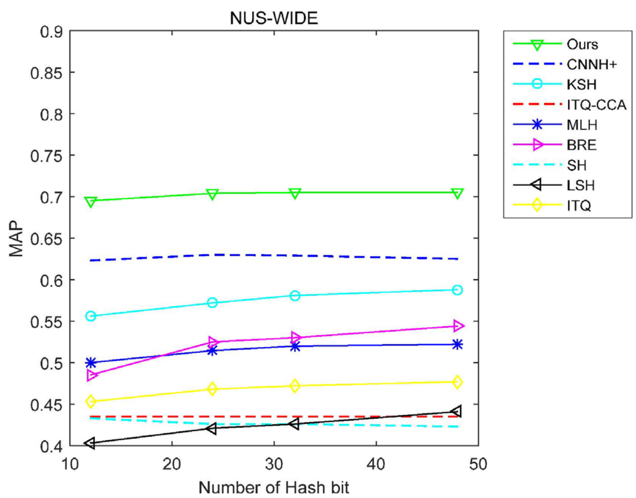

Figure 12. MAP comparison with different hash bit on NUS-WIDE

In search precision task, we can see from Figure 13 the trend of the precision with different hash bit is consistent with the Figure 11 and the algorithm in our paper is higher than CNNH+ and other methods and we can demonstrate that it has improvement in effect.

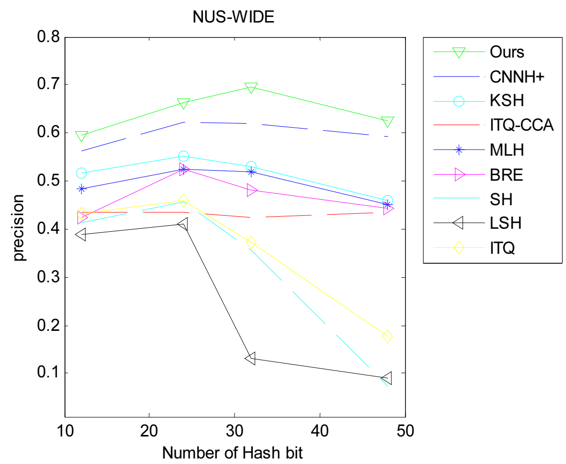

Figure 13. Precision comparison with different hash bit on NUS-WIDE

Finally, with this dataset, we compare the method with the liner search in time efficiency. From Figure 14, it can be seen that with 32 bit hash code, we compute the feature learning with 59 seconds and finally finish one search task has cost 124.3 seconds while linear search will cost 9682 seconds, which shorten 77.8 times than before, which has increased in efficiency obviously.

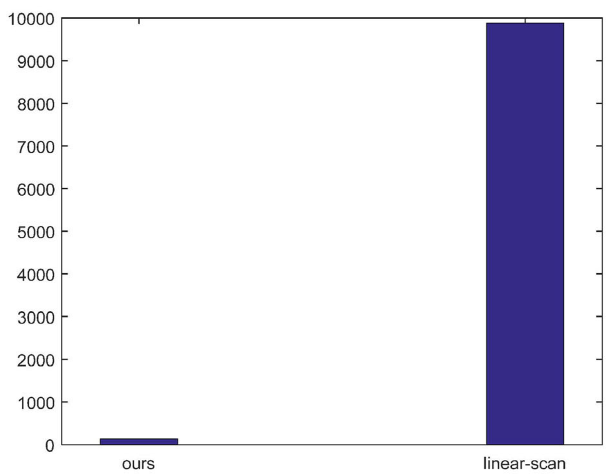

Figure 14. CPU time cost comparison with different search method on NUS-WIDE

3 Conclusions

This paper presents an easy and effective deep studying framework, and creates hash-based binary learning codes for quick image classification and search. Besides, it adds a latent feature layer in this deep network for studying image representations and a set of hash-like functions. The framework adopts googlenet for pretraining and fine-tuned for incremental learning, which improves the learning speed and efficiency. It is featured as scalable to the size of dataset. It is shown from experimental results that this method improves the original search results with 0.75% and 12% search precision in the datasets of CIFAR-10 and MNIST, in which there's just one easy change of the deep CNN. The scalability and efficacy of our method on this good-sized dataset with up to 1 million shopping images is further proved.

Acknowledgments

This work has received financial support by the 973 National Fund Project (2013CB329603) and the National Natural Science Fund (61272441, 61171173).

References

[1] A. Babenko, A. Slesarev, A. Chigorin. V. Lempitsky, Neural Codes for Image Retrieval, *European Conference on Computer Vision*, Zurich, Switzerland, 2014, pp. 584-599.

[2] H. Bay, T. Tuytelaars, L. Van Gool, Surf: Speeded up Robust Features, *European Conference on Computer Vision*, Graz, Austria, 2006, pp. 404-417.

[3] D. Ciresan, U. Meier, J. Schmidhuber, Multi-column Deep Neural Networks for Image Classification, *2012 IEEE Conference on Computer Vision & Pattern Recognition*, Providence, RI, 2012, pp. 3642-3649.

[4] J. Deng, A. C. Berg, F. F. Li, Hierarchical Semantic Indexing for Large Scale Image Retrieval, *2011 IEEE Conference on Computer Vision & Pattern Recognition*, Colorado Springs, CO, 2011, pp. 785-792.

[5] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, T. Darrell, DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition, *Proceedings of the 31st International Conference on Machine Learning*, Beijing, China, 2014, pp. 647-655.

[6] A. Gionis, P. Indyk, R. Motwani, Similarity Search in High Dimensions via Hashing, *International Conference on Very Large Data Bases*, Edinburgh, Scotland, 1999, pp. 518-529.

[7] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, 2014, pp. 580-587.

[8] Y. Gong, S. Lazebnik, A. Gordo, 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, December, 2013.

[9] I. J. Goodfellow, D. Warde-Farley, M. Mirza, A. Courville, Y. Bengio, Maxout Networks, *Proceedings of the 30th International Conference on International Conference on Machine Learning - Volume 28*, Atlanta, GA, 2013, pp. 1319-1327.

[10] P. Jain, B. Kulis, K. Grauman, Fast Image Search for Learned Metrics, *IEEE Conference on Computer Vision and Pattern Recognition*, Anchorage, AK, 2008, pp. 1-8.

[11] Y. Jia, E. Shellhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: Convolutional Architecture for Fast Feature Embedding, *Proceedings of the 22nd ACM International Conference on Multimedia*, Orlando, Florida, 2014, pp. 675-678.

[12] A. Krizhevsky, *Learning Multiple Layers of Features from Tiny Images*, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.222.9220&rep=rep1&type=pdf.

[13] A. Krizhevsky, G. E. Hinton, Using Very Deep Autoencoders for Content-Based Image Retrieval, *European Symposium on Artificial Neural Networks*, Bruges, Belgium, 2011, pp. 489-494.

[14] A. Krizhevsky, I. Sutskever, G. E. Hinton, ImageNet Classification with Deep Convolutional Neural Networks, *Advances in Neural Information Processing Systems 25 (NIPS 2012)*, Lake Tahoe, NV, 2012, pp. 1097-1105.

[15] B. Kulis, T. Darrell, Learning to Hash with Binary Reconstructive Embeddings, *Advances in Neural Information Processing Systems 22 (NIPS 2009)*, Vancouver, British Columbia, Canada, 2009, pp. 1042-1050.

[16] Y. LeCun, *The MNIST Database of Handwritten Digits*, http://yann.lecun.com/exdb/mnist/.

[17] M. Lin, Q. Chen, S. Yan, *Network In Network*, arXiv preprint arXiv:1312.4400, December, 2013.

[18] W. Liu, J. Wang, R. Ji, Y. G. Jiang, S. F. Chang, Supervised Hashing with Kernels, *2012 IEEE Conference on Computer Vision and Pattern Recognition*, Providence, RI, 2012, pp. 2074-2081.

[19] D. G. Lowe, Distinctive Image Features from Scale-Invariant Keypoints, *International Journal of Computer Vision*, Vol. 60, No. 2, pp. 90-110, November, 2004.

[20] M. Norouzi, D. J. Fleet, Minimal Loss Hashing for Compact Binary Codes, *Proceedings of the 28th International Conference on Machine Learning*, Bellevue, WA, 2011, pp. 353-360.

[21] M. Oquab, L. Bottou, I. Laptev, J. Sivic, Learning and Transferring Mid-level Image Representations Using Convolutional Neural Networks, *2014 IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, 2014, pp. 1717-1724.

[22] G. Qiu, Indexing Chromatic and Achromatic Patterns for Content-based Colour Image Retrieval, *Pattern Recognition*, Vol. 35, No. 8, pp. 1675-1686, August, 2002.

[23] A. S. Razavian, H. Azizpour, J. Sullivan, S. Carlsson, CNN Features Off-the-shelf: An Astounding Baseline for Recognition, *2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, Columbus, OH, 2014, pp. 512-519.

[24] R. Salakhutdinov, G. Hinton, S. Hashing, *International Journal of Approximate Reasoning*, Vol. 50, No. 7, pp. 969-978, July, 2009.

[25] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, R. Jain, Content-based Image Retrieval at the End of the Early Years, *IEEE Transactions on Pattern Analysis & Machine Intelligence*, Vol. 22, No. 12, pp. 1349-1380, December, 2000.

[26] J. Snoek, H. Larochelle, R. P. Adams, Practical Bayesian Optimization of Machine Learning Algorithms, *Advances in Neural Information Processing Systems*, Lake Tahoe, NV, 2012, pp. 2951-2959.

[27] J. Wan, D. Wang, S. C. H. Hoi, P. Wu, J. Zhu, Y. Zhang, J. Li, Deep Learning for Content-Based Image Retrieval: A Comprehensive Study, *Proceedings of the 22nd ACM International Conference on Multimedia*, Orlando, Florida, 2014, pp. 157-166.

[28] J. Wang, S. Kumar, S. F. Chang, Semi-supervised Hashing for Scalable Image Retrieval, *2010 IEEE Conference on Computer Vision & Pattern Recognition*, San Francisco, CA, 2010, pp. 3424-3431.

[29] Y. Weiss, A. Torralba, R. Fergus, Spectral Hashing, *Advances on Neural Information Processing Systems 21 (NIPS 2008)*, Vancouver, British Columbia, Canada, 2008, pp. 1753-1760.

[30] R. Xia, Y. Pan, H. Lai, C. Liu, S. Yan, Supervised Hashing for Image Retrieval via Image Representation Learning, *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, Québec, QC, 2014, pp. 2156-2162.

[31] M. D. Zeiler, R. Fergus, *Stochastic Pooling for Regularization of Deep Convolutional Neural Networks*, arXiv preprint arXiv:1301.3557, December, 2013.

[32] M. D. Zeiler, R. Fergus, Visualizing and Understanding Convolutional Networks, *European Conference on Computer Vision*, Zurich, Switzerland, 2014, pp. 818-833.

[33] K. Lin, H. F. Yang, J. H. Hsiao, C. S. Chen, Deep Learning of Binary Hash Codes for Fast Image Retrieval, *2015 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, Boston, MA, 2015, pp. 27-35.

[34] Z. Zhong, L. Jin, Z. Xie, High Performance Offline Handwritten Chinese Character Recognition Using GoogLeNet and Directional Feature Maps, *13th International Conference on Document Analysis and Recognition*, Tunis, Tunisia, 2015, pp. 846-850.

[35] D. Tian, D. Tao, Global Hashing System for Fast Image Search, *IEEE Transactions on Image Processing*, Vol. 26, No. 1, pp. 79-89, January, 2017.

[36] M. Niu, L. Wu, J. Zeng, Locality Preserving Hashing for Fast Image Search: Theory and Applications, *Journal of Experimental & Theoretical Artificial Intelligence*, Vol. 29, No. 2, pp. 349-359, 2017.

[37] F. Zhao, Y. Huang, L. Wang, T. Tan, Deep Semantic Ranking based Hashing for Multi-label Image Retrieval, *2015 IEEE Conference on Computer Vision and Pattern Recognition*, Boston, MA, 2015, pp. 1556-1564.

[38] R. Zhang, L. Lin, R. Zhang, W. Zuo, L. Zhang, Bit-Scalable Deep Hashing With Regularized Similarity Learning for Image Retrieval and Person Re-Identification, *IEEE Transactions on Image Processing*, Vol. 24, No. 12, pp. 4766-4779, December, 2015.

[39] H. Liu, R. Wang, S. Shan, X. Chen, Deep Supervised Hashing for Fast Image Retrieval, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, NV, 2016, pp. 2064-2072.

Biographies

Jun-yi Li finished his Ph.D. study at Shanghai Jiaotong University Cyber Space Security Institute in 2015 and he completed a MSc in telecommunication Engineering in 2009, a BSc in electronic engineering in 2006. His research interest is focused on image search, image processing, machine learning and distributed system design.

Jian-hua Li is a full professor in the Cyber Space Security Institute of Shanghai Jiaotong University, China. His research interest is focused on internet and space security, network communication. He is the lead of the college of Information and Security.