Deep Multi-level Hashing Codes for Image Retrieval

Zhenjiang Dong[1,2], Ge Song[3,4], Xia Jia[1], and Xiaoyang Tan[3,4(✉)]

[1] Shanghai Jiaotong University, Shanghai 200240, China
[2] ZTE Corporation, Nanjing 210012, China
[3] Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China
x.tan@nuaa.edu.cn
[4] Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing 211106, China

Abstract. In this paper, we propose a deep siamese convolutional neutral network (DSCNN) to learn semantic-preserved global-level and local-level hashing codes simultaneously for effective image retrieval. Particularly, we analyze the visual attention characteristic inside hash bits by activation map of deep convolutional feature and propose a novel approach of bit selecting to reinforce the pertinence of local-level code. Finally, unlike most existing retrieval methods which use global or unsupervised local descriptors separately, leading to unexpected precision, we present a multi-level hash search method, taking advantage of both local and global properties of deep features. The experimental results show that our method outperforms several state-of-the-art on the Oxford 5k/105k and Paris 6k datasets.

1 Introduction

Due to the explosive growth of the Internet, massive images have flooded our daily lives. Image retrieval, i.e. finding images containing the same object or scene as in a query image, has attracted more attention from researchers.

Recently, most studies have reported that deep Convolutional Neural Networks (CNNs) achieved the state of the art performance in many computer vision tasks [1–3]. Notably, many works [4,5] have demonstrated the suitability of features from fully-connected layers for image retrieval. While several works [6–8] focused on features from deep convolutional layers and showed that these features have the natural interpretation as descriptors of local image regions. However, most CNN features for image retrieval are directly extracted from classification model, and subjected to low precision. Furthermore, the features with rich semantic information distract the target sense of query. Early work by Zhou et al. [9] revealed that the convolutional units of CNNs actually behave as object detectors, and proposed a method to generate Class Activation Map (CAM) [10] for localizing the discriminative image regions, which make it available to use deep localizable representations for visual tasks.

Z. Zhang and K. Huang (Eds.): IVS 2016, CCIS 664, pp. 91–98, 2016.
DOI: 10.1007/978-981-10-3476-3_11

Besides, traditional nearest neighbor search methods are faced with the computational cost of similarity calculation of high-dimension features, are not appropriate for rapid retrieval, especially under the circumstances of big data age. A practical alternative is to use the hashing based methods [11–13]. Hash method designs a group function which project images into binary codes so that similar images are mapped into similar code. Therefore, the retrieval problem can be done efficiently by computing Hamming distance. Benefiting from deep learning, several researchers [13–17] combined image representations learning with hash learning into one CNN architecture to learn semantic-preserved binary code. Although these methods achieved outstanding performance, have not shed light on the relation between each bit and semantic concept.

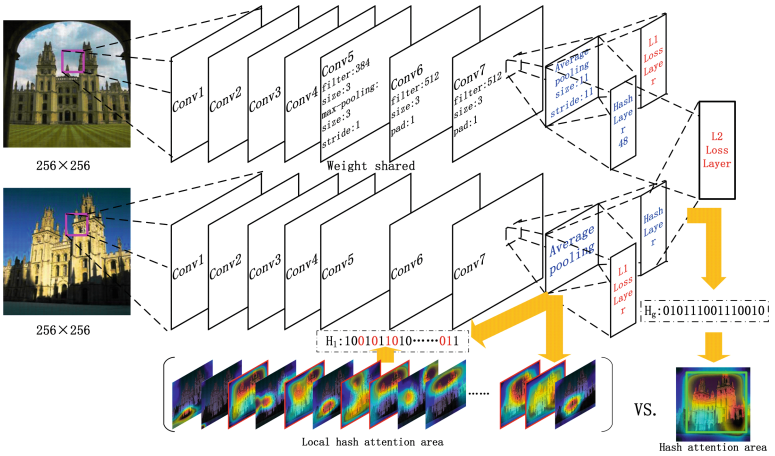

Fig. 1. The DSCNN framework is proposed. Firstly, The semantic-preserved global-level H_g and local-level hash codes H_l are learned. Secondly, we obtain CAMs of each bits of H_g and average these CAMs to acquire 'Hash attention area', and get 'Local hash attention area' by activation maps corresponding to each bits of H_l. Then visually highlight bits (red colored) are selected as compact hash code. Finally, we retrieval similar images by the presented multi-level search strategies. (Color figure online)

In this paper, we propose a deep siamese CNN (DSCNN) framework to learn semantic-preserved hash code, and design the last convolutional layer of DSCNN to obtain local-level hash codes, which is essentially different from other methods [13–15]. Above all, we propose a novel method to obtain compact bits with salient local-semantic. Finally, we present a multi-level hash search method for retrieval.

2 Our Method

Learning Semantic-Preserved Hash Code. It is feasible to embed a latent layer in high-level of a network to output global binary code [13–15]. We follow

it and use both label and pair information to guide hash leaning. Otherwise, inspired by discovery [4], we propose to hashing convolutional activations. As Fig. 1 shows, the activation of hash layer and conv7 are both tanh function. And we impose constraints on these layers to embed semantic information. Assuming that the feature maps of conv7 are $\{I_i\}_{i=1}^{C} \in (-1,1)^{W\times H}$, W, H is weight and height, C is the number of filters, the output of Hash Layer are $a \in (-1,1)^{H}$, H is the length of hash code. $\hat{y}$ is output of softmax layer, y is expected output. And we minimize the loss function defined following to learn parameters of our network. For local-level hash:

$$L_1 = -\sum_{j=1}^{N} y_i \log(\hat{y_j}) \tag{1}$$

For global-level hash:

$$\begin{aligned} L_2 &= -L_1 + \alpha J_{11} + \alpha J_{12} + \beta J_2 + \gamma J_3 \\ &= -\sum_{j=1}^{N} y_i \log(\hat{y_j}) + \alpha \sum_{j=1}^{N}\sum_{i=1}^{N} \delta(y_j = y_i)\|a_j - a_i\|_2^2 \\ &+ \alpha \sum_{j=1}^{N}\sum_{i=1}^{N} \delta(y_j \neq y_i)\max(0, c - \|a_j - a_i\|_2^2) \\ &+ \beta \sum_{j=1}^{N}(\||a_j| - 1\|^2) + \gamma \sum_{j=1}^{N}(\|avg(a_j) - 0\|^2) \end{aligned} \tag{2}$$

where δ is indicator function, avg is the mean function, c is a constant, N is the number of images. The terms L_1 and J_{1*} aim to embed semantic consistency and similarity to hash code respectively. The term J_2 aims to minimize the quantization loss between the learned binary code and the original feature. The last term J_3 enforces evenly distribution of -1 and 1 in hash code. α, β, γ are parameters to balance the effect of different terms.

Finally, the global-level hash code H_g and local-level hash codeH_l are defined:

$$\begin{aligned} &H_g = \delta(a > 0), H_l = \delta(f > 0) \\ &where f \in (-1,1)^C, f_k = \frac{1}{W \times H}\sum_{i=1}^{W}\sum_{j=1}^{H} I_k(i,j) \end{aligned} \tag{3}$$

Selecting Compact Bits. The deep convolutional feature maps are activated with different regions [18,19]. And through careful observation we found that some feature maps are not related to the salient area, it may be possible to boost feature discrimination by discarding unrelated feature maps. Therefore we propose to select compact bit to enforce retrieval performance.

The first stage is to catch the attention region of H_g. We compute CAMs of H_g. Then we average these maps to M_{avg} and binarize by $B_{avg} = \delta(M_{avg} > \theta)$,

where θ is a threshold. And we get attention region by finding the largest connected subgraph of B_{avg}. As Fig. 1 shows.

The second stage is selecting local feature maps. We convert all feature maps of Conv7 into activation maps $\{AM_i\}_{i=1}^{C}$ by up-sampling, and obtain corresponding binary maps $\{B_i\}_{i=1}^{C}$ as the first stage done. We definite the score of relevant to salient area of feature maps as follows:

$$S(B_i, B_{avg}) = sum(B_i \wedge B_{avg}) \tag{4}$$

where $\wedge$ is AND operation bit-by-bit, *sum* represents sum all elements of matrix.

In the last stage, Ranking $I_1, I_2, \ldots, I_C$ by their scores S and selecting top L filters as informative local features. Then we choose associated L bits of H_l as $H_l^{'}$ for efficient retrieval. In our experiment, we only compared the local-level hash code of query's L positions with corresponding position bits of others.

$$H_q^{'} = \Psi_q(H_q), d_H(H_q^{'}, H_i) = d_H(H_q^{'}, \Psi_q(H_i)) \tag{5}$$

where $\Psi_q(*)$ indicates obtain the bits of $*$ as the same positions as H_q.

Searching via Multi-level Hashing. The original data space could be mapped to Hamming space by several group hash functions with similarity structure preserved separately. We proposed a multi-level search method of hashing, using several sets of function with different properties to reinforce positive neighborhoods retrieval and develop two strategies.

Rerank-Based Strategy#1. Firstly, we use global-level hash code to retrieval and select top K as candidates. Then, we use local-level hash code to rerank these candidates.

Hamming Distance Weighted Strategy#2. Assuming that query image x_q and N images $\{x_i\}_{i=1}^{N}$ and corresponding global-level hash code $H_{gq}, \{H_{gi}\}_{i=1}^{N}$ and local-hash code $H_{lq}^{'}, \{H_{li}\}_{i=1}^{N}$. Fusing distance as:

$$Sim(x_q, x_i) = \lambda d_H(H_{gq}, H_{gi}) + (1 - \lambda) d_H(H_{lq}^{'}, H_{li}) \tag{6}$$

In experiments, we firstly retrieval use the global-level code, then rerank by proposed weighted strategy.

3 Experiments

Datasets. We evaluate performance on three standard datasets with mean average precision (MAP). Oxford Buildings [20] (Oxford5k) contains 5063 images, including 55 queries corresponding 11 landmark buildings. Oxford Buildings+100K [20] (Oxford105k) includes Oxford5k and extra 100K images as distractor. Paris Buildings [21] (Paris6k) contains 6412 images, 55 queries corresponding to 11 Paris landmarks.

Experimental Details. We implement the proposed DSCNN by Caffe [22] package. We design DSCNN based on the AlexNet architecture, details as

Fig. 1 shows. All images are resized to 256×256 before passing through the network. For training model, we randomly select positive and negative pairs from dataset exclude queries and initial weights of Conv1-Conv4 with pre-trained AlexNet.

Fig. 2. Examples of the compact of using local-level code to reranking. For each query image, the first line represents the rankings with global-level hash code, and the next line is the retrieval result by using proposed multi-level hash search method.

Results of Local Features. We compare local-level code from DSCNN with other state local descriptors. Firstly, we compare with the sophisticated local descriptors aggregation methods Fisher vectors [6], VLAD [23] and Tri. embedding [24]. Table 1 summaries the results. We attain the best performance on three datasets. Compared with deep feature, we can see that our average-pooling strategy (local-level hash) outperforms max-pooling [25] and SPoC [6] on Oxford dataset. Then, the result on Paris demonstrates that the local-level is superior to global-level hash code. And multi-level improve the performance of global-level code by 12 and 14 on Oxford and Paris, respectively. Some qualitative examples of retrieval using multi-level hash are shown in Fig. 2, local-level hash enhances the ranking of relevant results and decrease the irrelevant images, as expected. Finally, our method is different from PCA and performs better.

Table 1. mAP comparison with local descriptors. Local-level hash perform better.

Method	D	Oxford5k	Oxford105k	Paris6k
Fisher Vector [6]	256	54.0	-	-
Trian. embedding [24]	1024	56.0	50.2	-
VLAD [6]	128	44.8	37.4	55.5
CNN+VLAD [8]	128	55.8	-	58.3
CNN+Max pooling [25]	256	53.3	48.9	67.0
SPoC [6]	256	58.9	57.8	-
Conv7+PCA	256	58.6	55.7	68.6
global-level hash code	48	59.3	58.2	69.2
local-level hash code	256	**69.7**	**63.9**	**85.2**
multi-level hash code	256	67.1	63.4	83.7

Comparison with State-of-the-Art. Approaches based on deep model in the literature. We set length of H_l to 256 impartially. As Table 2 reveals that our method produced better or competitive results. For strategy #1, we use global-level hash code to retrieval 50 candidates and rerank by local-level hash code, achieving mAP 67.1% on Oxford5k and 83.7% on Paris6k. Then, we adopt strategy #2 to retrieval with setting λ to 0.5 empirically, obtaining slightly different performance with strategy#1. We conjectured that the fusion weaken some discriminant of local-level code caused the gap in performance.

For deep convolutional features, CNN+fine-tuning [26] gains mAP 55.7% on Oxford by retraining deep models with additional landmarks dataset collected by themselves, while we obtain 67.2% only with limited training samples provide by datasets. Although we did not promote performance by spatial reranking or query expansion strategies as Tolias et al. [7] done, our method achieve competitive results. Compared with R-CNN+CS-SR+QE [26], our method is more simple and effective (83.7 vs 78.4), exploring the inside property of deep convolutional descriptor to select compact local feature for retrieval, while R-CNN+CS-SR+QE locates objects by RPN. Mention that our method can carry out fast image retrieval via Hamming distance measurement, which is obviously superior to others based on Euclidean or Cosine distance.

Table 2. mAP comparison with state-of-the-art methods CNN-based.

Method	Oxford5k	Oxford105k	Paris6k
SPoC [6]	58.9	57.8	-
Razavian et al. [5]	55.6	-	69.7
Kalantidis et al. [27]	65.4	59.3	77.9
Tolias et al. [7]	66.8	61.6	83.0
CNN+fine-tuning et al. [4]	55.7	52.4	-
R-CNN+CS-SR+QE [26]	**67.8**	-	78.4
Ours(#1)	67.1	**63.4**	**83.7**
Ours(#2)	67.2	62.8	83.4

4 Conclusion

This paper has presented a deep siamese CNN to produce global and local levels hash codes for image retrieval with the proposed multi-level search method. And we firstly propose to select region-related bits by activation maps. Finally, we demonstrate the efficacy and applicability of the proposed approach on retrieval benchmarks. Experimental results show that our method improves the previous performance on Oxford and Paris datasets, respectively.

Acknowledgements. This work is supported by National Science Foundation of China (61373060,61672280), Qing Lan Project and the Research Foundation of ZTE Corporation.

References

1. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, vol. 2012 (2012)
2. Szegedy, C., Toshev, A., Erhan, D.: Deep neural networks for object detection. In: Advances in Neural Information Processing Systems, pp. 2553–2561 (2013)
3. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. Computer Science, pp. 1337–1342 (2015)
4. Babenko, A., Slesarev, A., Chigorin, A., Lempitsky, V.: Neural codes for image retrieval. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8689, pp. 584–599. Springer, Heidelberg (2014). doi:10.1007/978-3-319-10590-1_38
5. Razavian, A.S., Azizpour, H., Sullivan, J., Carlsson, S.: CNN features off-the-shelf: an astounding baseline for recognition. In: IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 512–519 (2014)
6. Babenko, A., Lempitsky, V.: Aggregating deep convolutional features for image retrieval. Computer Science (2015)
7. Tolias, G., Sicre, R., Jgou, H.: Particular object retrieval with integral max-pooling of CNN activations. Computer Science (2015)
8. Ng, Y.H., Yang, F.: Davis, L.S.: Exploiting local features from deep networks for image retrieval. Computer Science, pp. 53–61 (2015)
9. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Object detectors emerge in deep scene CNNs. Computer Science (2014)
10. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. Computer Science (2015)
11. Andoni, A., Indyk, P.: Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In: Annual Symposium on Foundations of Computer Science, pp. 117–122 (2006)
12. Liong, V.E., Lu, J., Wang, G., Moulin, P., Zhou, J.: Deep hashing for compact binary codes learning. In: Computer Vision and Pattern Recognition (2015)
13. Zhao, F., Huang, Y., Wang, L., Tan, T.: Deep semantic ranking based hashing for multi-label image retrieval. In: Computer Vision and Pattern Recognition (2015)
14. Lin, K., Yang, H.F., Hsiao, J.H., Chen, C.S.: Deep learning of binary hash codes for fast image retrieval. In: Computer Vision and Pattern Recognition Workshops, pp. 27–35 (2015)
15. Li, W.J., Wang, S., Kang, W.C.: Feature learning based deep supervised hashing with pairwise labels. Computer Science (2015)
16. Lai, H., Pan, Y., Liu, Y., Yan, S.: Simultaneous feature learning and hash coding with deep neural networks. In: Computer Vision and Pattern Recognition (2015)
17. Liu, H., Wang, R., Shan, S., Chen, X.: Deep supervised hashing for fast image retrieval. In: Computer Vision and Pattern Recognition (2016)
18. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8689, pp. 818–833. Springer, Heidelberg (2014). doi:10.1007/978-3-319-10590-1_53

19. Mahendran, A., Vedaldi, A.: Understanding deep image representations by inverting them. In: Computer Vision and Pattern Recognition (2015)
20. Philbin, J., Chum, O., Isard, M., Sivic, J.: Object retrieval with large vocabularies and fast spatial matching. In: Computer Vision and Pattern Recognition (2007)
21. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Lost in quantization: improving particular object retrieval in large scale image databases. In: Computer Vision and Pattern Recognition (2008)
22. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: convolutional architecture for fast feature embedding. Eprint Arxiv, pp. 675–678 (2014)
23. Arandjelovic, R., Zisserman, A.: All about VLAD. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1578–1585 (2013)
24. Jegou, H., Zisserman, A.: Triangulation embedding and democratic aggregation for image search. In: Computer Vision and Pattern Recognition, pp. 3310–3317 (2014)
25. Razavian, A.S., Sullivan, J., Maki, A., Carlsson, S.: A baseline for visual instance retrieval with deep convolutional networks. Computer Science (2015)
26. Salvador, A., Giro-I-Nieto, X., Marques, F., Satoh, S.: Faster R-CNN features for instance search. Eprint Arxiv (2016)
27. Kalantidis, Y., Mellina, C., Osindero, S.: Cross-dimensional weighting for aggregated deep convolutional features. Eprint Arxiv (2015)