Deep Residual Net Based Compact Feature Representation for Image Retrieval

Cong Bai[1(✉)], Jian Chen[1], Qing Ma[1,2], Zhi Liu[1], and Shengyong Chen[1]

[1] College of Computer Science, Zhejiang University of Technology, Hangzhou, China
congbai@zjut.edu.cn

[2] College of Science, Zhejiang University of Technology, Hangzhou, China

Abstract. Deep learning technology has been introduced into many multimedia processing tasks, including multimedia retrieval. In this paper, we propose a deep residual net (ResNet) based compact feature representation improve the content-based image retrieval (CBIR) performance. The proposed method integrates ResNet and hashing networks to convert the raw images into binary codes. The binary codes of images in query set and that of the database are compared using Hamming distance for retrieval. Comprehensive experiments are executed on three public databases. The results show that the proposed method outperforms state-of-the-art methods. Furthermore, the impact of the deep convolutional network (DCNN)'s depth on the performance is investigated.

Keywords: Content-based image retrieval · Residual Nets · Hashing
Depth of deep convolutional neural network

1 Introduction

With the huge accumulation of digital images and videos in the society, the demand of searching such kind of data is also increasing [8]. Traditional multimedia search engines usually use the surrounding meta data, such as titles and tags or manually annotated keywords as the index to retrieval the multimedia data, named keyword-based retrieval [18]. The main drawback of such kind of retrieval technology is the inconsistence between the textural information and visual content of multimedia data. So the content-based multimedia retrieval is proposed and makes great progress in the past decades [13]. In content-based multimedia retrieval, semantic gap is a challenging problem, which refers to the difference between the low level representations of images and the higher level concepts used by human beings to describe the images. To narrow this gap, extensive efforts have been made both from the academic and industry communities [9,10,17].

Over the past few years, deep learning has been witnessed as one of the most promising technology in computer vision as their outstanding performance in

R. Hong et al. (Eds.): PCM 2018, LNCS 11165, pp. 737–747, 2018.
https://doi.org/10.1007/978-3-030-00767-6_68

a series of vision related tasks, such as image classification [19], face recognition [29], image segmentation [27] and so on. Since the successful application of AlexNet [12] in computer vision, the instinct idea to get better feature representation is to use deep convolution neural networks (DCNN). For example, AlexNet contains 8 learned layers, VGGNet [21] has 19 learned layers and GoogLeNet [22] consists of 22 learned layers. However, going deeper means that training such network will become more difficult. Thus deeper but easy to be trained DCNN is proposed, namely, ResNet [6]. Inspired by these successful DCNN, deep learning has already been introduced into content-based image retrieval (CBIR) [2]. However, it is instinct to ask: whether using very deep DCNN will improve the performance of image retrieval, especially for large scale image database?

In order to answer the above question, we use a very deep DCNN, ResNet for image retrieval. However, features extracted by ResNet are high-dimensional, thus they are not compatible with large scale image database if they are used directly for retrieval. So the proposed method constitutes a framework for converting the raw images into binary codes for effectively large scale image retrieval. To do so, the raw images are firstly input into the ResNet to get the deep features. And then the deep features are converted into binary codes by a DCNN based hashing network. In summary, the contributions of this work are twofold:

(1) We investigate a new framework that could convert the raw images into binary codes for large scale image retrieval. This framework integrates ResNet and DCNN based hashing network.

(2) Extensive experiments are conducted for comprehensive evaluations of the proposed framework, especially with different depth of the DCNN.

The reminder of the manuscript is structured as follows: Sect. 2 describes the proposed framework, followed by the experiment results in Sect. 3. Finally, conclusion and perspectives are given in Sect. 4.

2 Proposal

The framework of our method is shown in Fig. 1. The inputs are the pixels of the raw image and the corresponding label information of the image (for training only) and the output are the binary codes of the images. Such codes could be used for image retrieval by comparing the Hamming distances between the query's codes and the codes of the gallery images. The proposed framework includes two kinds of DCNN. Deep Residual Network (ResNet) [6] is used to convert the raw images into deep features and hashing neural network (HNN) is used to convert the deep feature into binary codes. We name our proposal as ResHNN. Details will be explained in this section.

2.1 ResNet

ResNet won the 1st place on the ILSVRC 2015 classification task and is proved to be easily optimized. And it gains improvement from increased depth [6]. ResNet

is composed by many stacked "Residual Units" and each unit could be expressed in the following form:

$$y_i = h(x_i) + F(x_i, W_l)$$
$$x_{i+1} = f(y_i) \tag{1}$$

where x_i and $x_i + 1$ are input and output of the i-th unit, and F is residual function. $h(x_i)$ is an identity mapping and f is a ReLU function [7]. The core idea of ResNet is to learn the additive residual function F with respect to $h(x_i)$, with a key choice of using an identity mapping $h(x_i) = x_i$. This is realized by attaching an shortcut connection that performs identity mapping and their outputs are added to the outputs of the stacked layers. The residual unit we used is as shown in Fig. 1. More details could be found in [6,7].

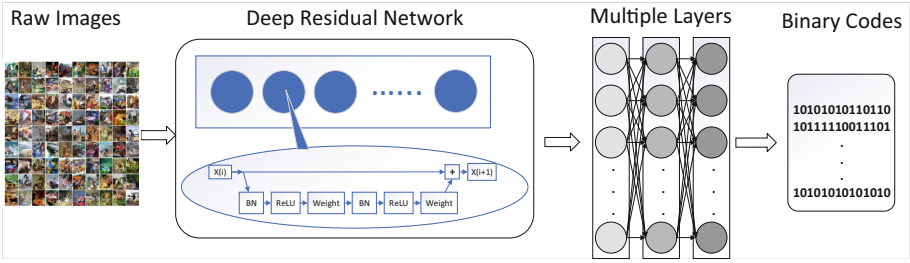

Fig. 1. The framework of transferring images into compact feature representation

2.2 HNN

The hashing layer is based on Nonlinear Discrete Hashing [3], which uses a multi-layer neural network to obtain the compact binary codes through nonlinear transformations. Let $X = [x_1, x_2, \ldots, x_n] \in R^{n \times d}$ denote the training set with n samples, where each sample $x_i \in R^d (1 \leq i \leq n)$ is a data point of d dimension. Assuming the m-th layer consists of $u^{(m)}$ units, the output of each layer is computed as:

$$h^{(1)}(x_i) = s(x_i W^{(1)} + c^{(1)}), i = 1, \cdots, n \tag{2}$$
$$h^{(m)}(x_i) = s(h^{(m-1)}(x_i) W^{(m)} + c^{(m)}), i = 1, \cdots, n \tag{3}$$

where $s(\cdot)$ is a nonlinear activation function such as the *tanh* function, and the projection matrix $W^{(m)}$ and the bias vector $c^{(m)}$ are the parameters to be learned for the m-th layer of the network.

And for a I-layer network, we could have the output in the form of:

$$F(x) = h^{(I)}(x) \in R^{u^{(I)}} \tag{4}$$

where the mapping $F : R^d \rightarrow R^{u(I)}$ is a parametric nonlinear function determined by $\{W^{(m)}, c^{(m)}\}_{m=1}^{I}$. We treat the sign of the output of the network as

the binary code of these n samples and put the binary code of all the samples together as:

$$B = sgn(F(X)) \in \{-1, +1\}^{(n \times r)} \tag{5}$$

Specifically, we treat the 0 as +1. The formula is as follows:

$$sgn(b) = \begin{cases} 1, & b \geq 0 \\ -1, & b < 0 \end{cases} \tag{6}$$

The goal is to find a binary matrix that minimizes the value of loss function. The formula is as follows:

$$arg \min_{B} Q = Q(L, B) + Q(B, X) \tag{7}$$

where $Q(L, B)$ means the difference between the predicted labels through the hash code matrix B and the ground truth labels of all samples, and the $Q(B, X)$ means the information loss caused by transforming to binary code. Denoting the classifier weight matrix as C, the first term $Q(L, B)$ can be considered as:

$$Q_C(L, B) = ||L - CB^T||_F^2 \tag{8}$$

where the L is the ground truth label of the samples and $|| \cdot ||_F$ means the Frobenius norm.

$Q(B, X)$ measures the discrepancy between the binary codes and the data samples including the quantization loss term and the similarity preserving term

$$\begin{aligned} Q_F^{(I)}(B, X) &= \|B - F^{(I)}\|_F^2 \\ &+ \alpha \sum_{i=1}^{n} \sum_{j=1}^{n} S_{ij} \|F^{(I)}(i,:) - F^{(I)}(j,:)\|_F^2 \\ &s.t. B \in \{-1, 1\}^{n \times r}, B^T B = nI_r \end{aligned} \tag{9}$$

where S is the similarity matrix. To reduce the redundancy of information, $B^T B = nI_r$ is added. But the problem of constraint makes optimization difficult, so a real-valued matrix Y is introduced in $\Omega = \{Y \in \mathbb{R}^{n \times r} \| Y^T Y = nI_r\}$ approaching to B. So the Eq. 10 is introduced to substitute the independent constraint.

$$Q_I(B) = \|B - Y\|_F^2 \tag{10}$$

We notice that the loss function Eq. 7 only considers the outputs of the top layer of the network, but the hidden layers are not included. So the companion loss function is introduced as follows:

$$Q_F(B, X) = Q_F^{(I)}(B, X) + \sum_{m=1}^{I-1} \alpha^{(n)} h(Q_F^{(m)} - \tau^m) \tag{11}$$

where $h(x) = \max(x, 0)$ and $Q_F^{(m)} = \sum_{i=1}^{n} \sum_{j=1}^{n} S_{ij} \|F^{(m)}(i,:) - F^{(m)}(j,:)\|_F^2$, $m = 1, 2, \ldots, I-1$.

In consideration of all the mentioned above, the overall function is defined as follow:

$$\arg \min_{\mathbf{B},\mathbf{P},\{\mathbf{F}^{(m)}\}_{m}^{\mathbf{I}}=1,\mathbf{Y}} = Q_C + \lambda_1 Q_I + \lambda_2 Q_F + \lambda_3 Q_R$$
$$s.t. B \in \{-1, 1\}^{n \times r} \tag{12}$$

where $Q_R = \|C\|_F^2 + \sum_{m=1}^{I} \|W^{(m)}\|_F^2 + \sum_{m=1}^{I} \|c^m\|_F^2$ contains the regularizer to control the scales of the parameters.

Since the above joint optimization problem is non-convex and difficult to solve. Sub-optimal problems with respect to one variable while keeping other variable fixed is used. So we could iterate each variable of optimal solution in sub-optimal problem one by one. And this problem could be solved by Singular Value decomposition (SVD) and Gram-Schmidt process. More details could be found in [3].

3 Experiments

In this section, we conduct experiments on three datasets: MNIST [4], CIFAR10 [11], and SUN379 [26], to evaluate the performance of the proposal and try to answer the question we posed in the introduction.

Table 1. Retrieval performance on MNIST with 16, 32 and 64 bit length of binary codes

Method	mAP (%)			Precision@500 (%)		
	16	32	64	16	32	64
LSH [1]	15.81	25.41	32.78	28.08	38.56	48.39
SMLSH [25]	31.68	38.28	43.42	41.93	49.16	55.14
ITQ [5]	38.11	42.13	43.63	54.35	60.15	62.03
SPLH [24]	48.67	49.38	48.71	59.69	60.57	63.06
CCA-ITQ [5]	58.61	60.34	62.51	67.95	69.37	71.42
FastH [15]	95.04	96.19	96.71	93.60	94.67	95.27
SDH [20]	92.28	93.74	94.81	91.45	92.07	92.88
DeepH [16]	70.91	74.10	76.34	76.75	79.13	81.55
NDH [3]	94.64	95.88	96.29	93.82	94.81	94.99
SSDH [28]	-	**98.20**	-	-	98.50	-
ResHNN-50	**98.01**	98.03	**98.07**	**98.60**	**98.62**	**98.63**

3.1 Databases

MNIST: It is a handwritten digit dataset consisting of 70000 images with the size of 28×28. Each image is associated with a digit from 0 to 9 and represented as a 784-dimensional gray-scale feature vector by concatenating all pixels [3]. It's a simple dataset, so we extract a 256 dimensional feature vector by ResNet for each image. Following the same setting in [24], 1000 images with 100 images per class are randomly selected from original test set to form the query set, and use the remaining 69000 images as gallery database.

CIFAR10: It is a set of 60000 manually labeled color images. They are from 10 classes, and each class has 6000 images. Each image is with the size of 32×32. ResNet is used to extract a 1024 dimensional feature vector for each image. Similar to the MNIST, we use 1000 images consist of 100 images per class from original test set as query set and construct the gallery database with the remaining images.

SUN397: This dataset contains 108754 images which are classified into 397 categories. It is bigger and more complex than the two mentioned above databases, it could be a challenge to retrieve semantic neighbors. Each image is represented by a 2048 dimensional feature vector extracted by ResNet. Following the same protocol of the referred methods, 8000 images are randomly sampled as query images and the remaining images are left to form the gallery database.

3.2 Evaluation Metric

All experiments are repeated 10 times and the averaged values are took as the final result. Two metrics are used to measure the performance of different methods: precision at N samples and mean Average Precision (mAP). Given top N returned samples, precision at N samples is calculated as the percentage of relevant retrieved images:

$$Precision@N = \frac{\sum_{k=1}^{N} rel(k)}{N} \tag{13}$$

where $rel(k) = 1$ if k-th image is a relevant retrieved image, otherwise, $rel(k) = 0$. The mean Average Precision (mAP) presents an overall measurement of the retrieval performance by computing the area under the precision-recall curve, which delivers good discrimination and stability. It is calculated as follows:

$$AveP = \frac{\sum_{k=1}^{N}(P(k) \times rel(k))}{number\ of\ relevant\ images},$$
$$MAP = \frac{\sum_{q=1}^{Q} AveP(q)}{Q} \tag{14}$$

where k is the rank in the sequence of retrieved documents, N is the number of retrieved images, $P(k)$ is the precision at cut-off k in the list, and $rel(k)$ is equal to 1 if the item at rank k is a relevant image, otherwise, it is equal to 0 [23]. Q is the number of the queries.

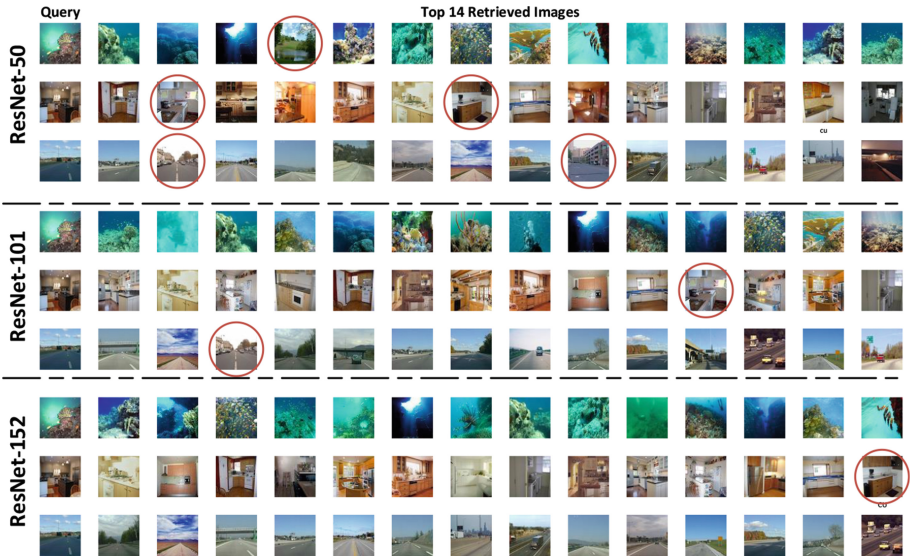

Fig. 2. Top 14 retrieved images from SUN397 dataset by different number of layers of ResNet with 128 bits binary codes. The results of ResHNN-50 are shown in the first three rows, the results of ResHNN-101 are shown in the middle three rows, and the results of ResHNN-152 are shown in the last three rows. The irrelevant images are marked by red circle. (Color figure online)

3.3 Results and Analysis

Result on MNIST: The training set used for hashing net is with the size of 5000 images by selecting 500 images from each class. The ResNet and HNN are trained separately and the depth of the ResNet we used in this database is 50. For the HNN, we take the tanh function as the nonlinear activation function and initialize the biases $c^{(m)}$ to be 0. Each element of $W^{(m)}$ is uniformly sampled from the range $\left[-\sqrt{\frac{6}{row(m)+col(m)}}, \sqrt{\frac{6}{row(m)+col(m)}}\right]$, where $row(m)$ is the number of rows of $W^{(m)}$ and $col(m)$ is the number of columns of $W^{(m)}$. The numbers R and L are set as 5 and 3. And we set $\alpha^{(1)}$ and $\alpha^{(2)}$ as 20, $\alpha^{(3)}$ as 100, $\tau^{(1)}$ and $\tau^{(2)}$ as 1000, λ_1 as 1e−3, λ_2 and λ_3 as 1e−5, learning rate η as 1e−3. And the same setting is adopted for all the other datasets. The experimental results are shown in Table 1. The results are compared on hash code with lengths of 16, 32, and 64 bits. As MNIST is a simple handwrite characters dataset, a lot of methods

could achieve good performance, so does ResHNN. As the performance achieved in this database is quiet high, the impact of depth of the DCNN is difficult to estimate, so we did not conduct further experiments on it.

Result on CIFAR10: Similar as the setting of experiments on the MNIST dataset, the training set is constructed with 5000 images with 500 images per category. We compare the results in different depth of the ResNet with 50, 101 and 152 to evaluated their impacts on feature extracting and hashing. Results are shown in Table 2. It is obviously that our method ResHNN outperforms referred methods obviously in different kinds of bit length, both in the aspect of mAP and Precision@500 and ResHNN-152 is the best. And with the ResNet going deeper, the retrieval performance improves slightly. We believe that the reason is that deeper networks could extract features from images more efficiently, which is preserved in our hash layers.

Table 2. Retrieval performance on CIFAR-10 with 16, 32 and 64 bit length of binary codes

Method	mAP (%)			Precision@500 (%)		
	16	32	64	16	32	64
LSH [1]	12.63	13.70	14.62	15.32	17.23	19.36
SMLSH [25]	14.96	16.41	16.98	17.82	19.75	20.36
ITQ [5]	15.57	15.80	16.57	19.91	21.04	22.53
SPLH [24]	17.08	19.38	21.21	21.22	26.39	29.34
CCA-ITQ [5]	16.21	16.02	16.49	24.63	24.44	26.77
FastH [15]	27.94	33.09	36.55	37.74	43.13	46.84
SDH [20]	29.21	29.22	32.67	39.08	39.62	42.15
DeepH [16]	24.04	25.96	27.53	32.45	34.09	36.85
NDH [3]	33.75	35.93	37.90	43.58	46.67	48.24
LPMH [14]	67.54	72.17	73.59	-	-	-
SSDH [28]	-	81.20	-	-	82.80	-
ResHNN-50	93.04	93.31	93.68	92.81	92.98	93.40
ResHNN-101	93.69	94.32	94.46	93.45	94.10	94.28
ResHNN-152	**94.16**	**94.65**	**94.92**	**93.87**	**94.35**	**94.74**

Result on SUN397: In order to verify whether the proposed ResHNN works well under large and complex conditions, more experiments were conducted in SUN397 database. As this database is a larger collection, we evaluate the impacts of the different number of the layers of ResNet also, with respects of 50, 101 and 152. Results are shown in Table 3. We notice that the proposed ResHNN could

Table 3. Retrieval performance on SUN397 with 48, 64 and 128 bit length of binary codes respectively

Method	mAP (%)			Precision@2000 (%)		
	48	64	128	48	64	128
ITQ [5]	5.16	5.58	6.73	6.14	6.43	6.98
SPLH [24]	1.27	1.89	0.99	2.90	3.33	2.65
CCA-ITQ [5]	7.22	6.38	6.08	6.21	5.90	5.56
FastH [15]	2.71	4.98	8.28	2.90	3.90	5.22
SDH [20]	9.87	9.65	11.85	7.57	7.81	8.52
DeepH [16]	9.31	9.73	8.32	7.54	7.52	6.76
NDH [3]	**11.39**	**12.96**	13.86	**7.81**	**8.32**	9.05
ResHNN-50	9.96	10.41	16.61	6.67	7.01	9.36
ResHNN-101	10.12	11.32	18.95	6.74	7.44	10.09
ResHNN-152	10.23	11.67	**19.58**	6.96	7.70	**10.26**

achieve the comparable performance with referred methods, and outperforms in long length of bits. Furthermore, the increase of the depth of the ResNet could trigger the obvious improvements on the retrieval with the longer length of binary codes. This could be explained by the fact that the deeper of the DCNN layers, the more information of the visual content of the image could be extracted, and with the longer of the length of the binary codes, such information could be preserved better. The examples of retrieval results with different layers of ResNet are shown in Fig. 2. The wrong returned images are marked by red circles.

4 Conclusion and Perspective

In this paper, we propose a ResNet based compact feature representation for image retrieval, namely, ResHNN, which integrates Residual net and hashing neural networks to generate the binary code for CBIR. Extensive experimental results on three widely used public databases demonstrate the superiority of the proposed ResHNN. Furthermore, we explore the impact of ResNet's depth on the performance. The impacts of the different deep features and different hashing method will be discovered further.

Acknowledgement. This work is supported by the National Natural Science Foundation of China under Grants No. 61502424 and U1509207, Zhejiang Provincial Natural Science Foundation of China under Grant No. LY18F020032 and LY16F020033.

References

1. Andoni, A., Indyk, P.: Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. In: Proceedings - Annual IEEE Symposium on Foundations of Computer Science, FOCS, pp. 459–468 (2006)
2. Bai, C., Huang, L., Pan, X., Zheng, J., Chen, S.: Optimization of deep convolutional neural network for large scale image retrieval. Neurocomputing **303**, 60–67 (2018)
3. Chen, Z., Lu, J., Feng, J., Zhou, J.: Nonlinear discrete hashing. IEEE Trans. Multimedia **19**(1), 123–135 (2017)
4. Deng, L.: The MNIST database of handwritten digit images for machine learning research [best of the web]. IEEE Sig. Process. Mag. **29**(6), 141–142 (2012)
5. Gong, Y., Lazebnik, S., Gordo, A., Perronnin, F.: Iterative quantization: a procrustean approach to learning binary codes for large-scale image retrieval. IEEE Trans. Pattern Anal. Mach. Intell. **35**(12), 2916–2929 (2013)
6. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778 (2016)
7. He, K., Zhang, X., Ren, S., Sun, J.: Identity mappings in deep residual networks. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9908, pp. 630–645. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46493-0_38
8. Hong, R., Hu, Z., Wang, R., Wang, M., Tao, D.: Multi-view object retrieval via multi-scale topic models. IEEE Trans. Image Process. **25**(12), 5814–5827 (2016). https://doi.org/10.1109/TIP.2016.2614132
9. Hong, R., Zhang, L., Tao, D.: Unified photo enhancement by discovering aesthetic communities from flickr. IEEE Trans. Image Process. **25**(3), 1124–1135 (2016). https://doi.org/10.1109/TIP.2016.2514499
10. Hong, R., Zhang, L., Zhang, C., Zimmermann, R.: Flickr circles: aesthetic tendency discovery by multi-view regularized topic modeling. IEEE Trans. Multimedia **18**(8), 1555–1567 (2016). https://doi.org/10.1109/TMM.2016.2567071
11. Krizhevsky, A.: Learning multiple layers of features from tiny images. Technical report (2009)
12. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: International Conference on Neural Information Processing Systems, pp. 1097–1105 (2012)
13. Lew, M., Sebe, N., Djeraba, C., Jain, R.: Content-based multimedia information retrieval: state of the art and challenges. ACM Trans. Multimedia Comput. Commun. Appl. **2**(1), 1–19 (2006)
14. Li, K., Qi, G.J., Hua, K.A.: Learning label preserving binary codes for multimedia retrieval: a general approach. ACM Trans. Multimedia Comput. Commun. Appl. **14**(1), 2:1–2:23 (2017)
15. Lin, G., Shen, C., Shi, Q., Van Den Hengel, A., Suter, D.: Fast supervised hashing with decision trees for high-dimensional data. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 1971–1978 (2014)
16. Liong, V.E., Lu, J., Wang, G., Moulin, P., Zhou, J.: Deep hashing for compact binary codes learning. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 07–12 June 2015, pp. 2475–2483 (2015)
17. Liu, Y., Zhang, D., Lu, G., Ma, W.: A survey of content-based image retrieval with high-level semantics. Pattern Recogn. **40**(1), 262–282 (2007)

18. Mei, T., Rui, Y., Li, S., Tian, Q.: Multimedia search reranking. ACM Comput. Surv. **46**(3), 1–38 (2014)
19. Russakovsky, O., et al.: ImageNet large scale visual recognition challenge. Int. J. Comput. Vis. **115**(3), 211–252 (2015)
20. Shen, F., Shen, C., Liu, W., Shen, H.T.: Supervised discrete hashing. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 07–12 June 2015, pp. 37–45 (2015)
21. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations (ICRL), pp. 1–14 (2015)
22. Szegedy, C., et al.: Going deeper with convolutions. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 07–12 June 2015, pp. 1–9 (2015)
23. Turpin, A., Scholer, F.: User performance versus precision measures for simple search tasks. In: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2006, pp. 11–18. ACM, New York (2006)
24. Wang, J., Kumar, S., Chang, S.F.: Semi-supervised hashing for large-scale search. IEEE Trans. Pattern Anal. Mach. Intell. **34**(12), 2393–2406 (2012)
25. Weng, L., Jhuo, I.H., Shi, M., Sun, M., Cheng, W.H., Amsaleg, L.: Supervised multi-scale locality sensitive hashing. In: Proceedings of the 5th ACM on International Conference on Multimedia Retrieval, ICMR 2015, pp. 259–266. ACM, New York (2015)
26. Xiao, J., Hays, J., Ehinger, K.A., Oliva, A., Torralba, A.: Sun database: large-scale scene recognition from abbey to zoo. In: 2010 IEEE conference on Computer vision and pattern recognition (CVPR), pp. 3485–3492. IEEE (2010)
27. Ye, L., Liu, Z., Li, L., Shen, L., Bai, C., Wang, Y.: Salient object segmentation via effective integration of saliency and objectness. IEEE Trans. Multimedia **19**(8), 1742–1756 (2017)
28. Zhang, J., Peng, Y.: SSDH: semi-supervised deep hashing for large scale image retrieval. IEEE Trans. Circ. Syst. Video Technol. **PP**(99), 1 (2017)
29. Zheng, J., Yang, P., Chen, S., Shen, G., Wang, W.: Iterative re-constrained group sparse face recognition with adaptive weights learning. IEEE Trans. Image Process. **26**(5), 2408–2423 (2017)