Neurocomputing 237 (2017) 235–241

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Exploiting the complementary strengths of multi-layer CNN features for image retrieval

Wei Yu[a], Kuiyuan Yang[b], Hongxun Yao[a,*], Xiaoshuai Sun[a], Pengfei Xu[b]

[a] School of Computer Science and Technology, Harbin Institute of Technology, China
[b] Microsoft Research, Beijing, China

ARTICLE INFO

Communicated by Dr Xinmei Tian

Keywords:
Image retrieval
CNN
Multi-level features

ABSTRACT

Deep convolutional neural networks have demonstrated breakthrough accuracies for image classification. A series of feature extractors learned from CNN have been used in other computer vision tasks. However, CNN features of different layers aim to encode different-level information. High-layer features care more about semantic information but less detail information, while low-layer features contain more detail information but suffer from the problem of background clutter and semantic ambiguity. We propose to exploit complementary strengths of different layers in a simple but effective way. A mapping function is designed to highlight the effectiveness of low-layer similarity, when measuring fine-grained similarity between query image and its nearest neighbors with similar semantic. Extensive experiments show that our method can achieve competitive performance on popular retrieval benchmarks. Extensive experiments show that the proposed method outperforms the features extracted from single layers and their direct concatenations. Meanwhile, our method achieves competitive performance on popular retrieval benchmarks.

1. Introduction

Recently, deep Convolutional Neural Network (CNN) has achieved the state-of-the-art performance in image classification task [1–4]. With the rebirth of CNN, a series of feature extractors stacked from low-level to high-level can be automatically learned from large-scale training data in an end-to-end manner. A number of works have shown that these learned feature extractors can be successfully transferred to other computer vision tasks [5–7]. As an active research topic, content-based image retrieval (CBIR) also can utilize CNN activations as universal representation for image. In this paper, we attempt to achieve better retrieval performance based on the complementarity of CNN activation maps of different layers.

CBIR always relies on the descriptor's ability of representing image. The powerful hand-crafted descriptors can capture local characteristics of object, such as SIFT [8]. Most existing approaches encode these gradient-based features to overcome the semantic gap, such as Bog-of-Words (BoW) [9], Fisher Vectors (FV) [10] or Vector Locally Aggregated Descriptors (VLAD) [11] and their variants [12–14]. However, these encoding methods are still far from capturing high-level semantic information.

When the pre-trained CNN is applied to CBIR, the feature maps of higher layers are selected to present whole image [15]. In recent findings, feature maps of lower layers can achieve better results in instance-level image retrieval [16]. Actually, feature maps of different layers extract information of different levels from input image. Fig. 1 illustrates some patches corresponding to top activations on some filters learned by CNN.[1] The dimensions of low-layer features tends to response the patches with similar simple patterns and with more ambiguity. The feature maps of higher layers care more about semantic information but less detail information about image, since higher layers are closer to the last layer with category labels. The feature maps of lower layers contain more structural information about image, but suffer from the problem of background clutter and semantic ambiguity. Although the feature maps of lower layers share similar semantic gap as hand-crafted features, the feature extractors of lower layer can capture local patterns for describing instance-level detail.

High-layer feature is used to measure semantic similarity and low-layer feature is used to measure fine-grained similarity. Giving an easy-to-understand example, when query image is a building, high-layer similarity captures the images contains a building and low-layer similarity captures the building with same subordinate-class even same instance. Obviously, the complementarity of low-layer and high-layer features can improve the similarity measuring between query image

* Corresponding author.
E-mail addresses: w.yu@hit.edu.cn (W. Yu), kuyang@microsoft.com (K. Yang), h.yao@hit.edu.cn (H. Yao), xiaoshuaisun@hit.edu.cn (X. Sun), penxu@microsoft.com (P. Xu).

[1] We retrain a AlexNet model on ImageNet ILSVRC 2012

http://dx.doi.org/10.1016/j.neucom.2016.12.002
Received 19 July 2016; Received in revised form 29 September 2016; Accepted 2 December 2016
Available online 23 December 2016
0925-2312/

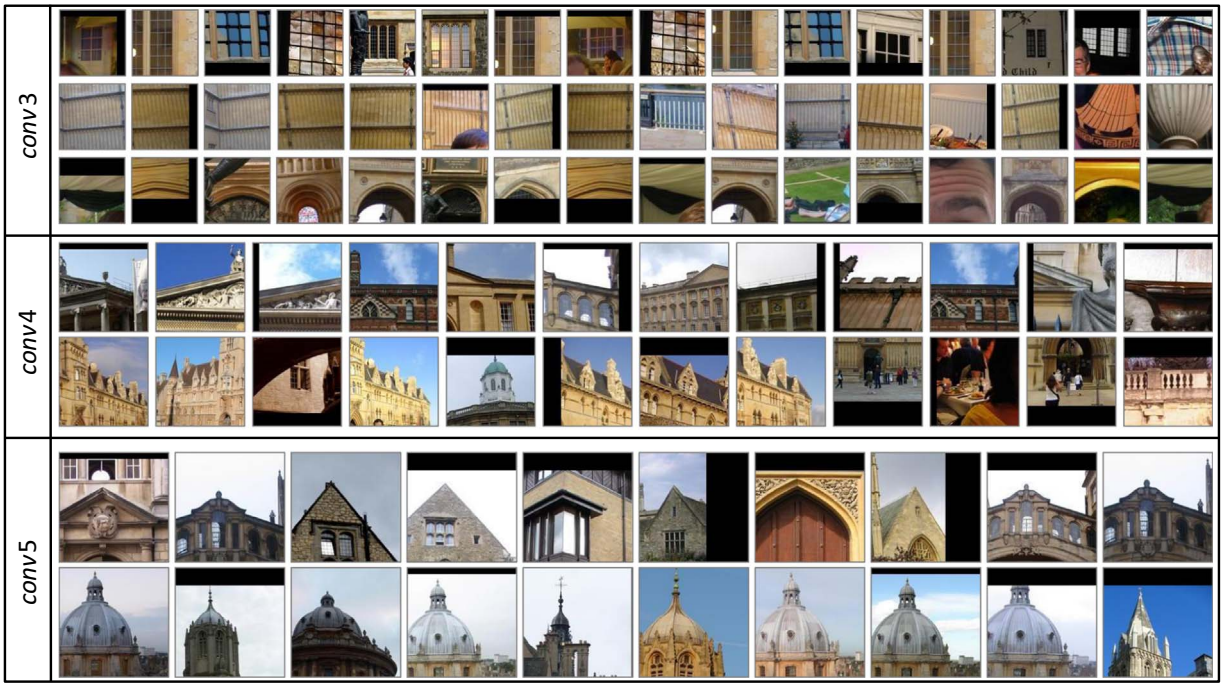

Fig. 1. The patches with top activations on sampled dimensions of CNN features from different convolutional layers. Each row shows the patches with largest activations in the corresponding dimension. All patches are sampled from Oxford 5K dataset.

and other candidate images. Some existing methods attempt to utilize multi-scale orderless pooling for CNN activations. For instance, CNN features are extracted and encoded from different layers respectively, then these aggregated features of different layers are concatenated to measure images [17]. However, direct concatenation can not use the complementary of high-layer and low-layer features adequately. High-layer feature can search a collection of candidate images with similar semantic for query image, yet it is not powerful enough to describe the fine-grained detail. Thus, high-layer similarity will weaken the effectiveness of low-layer similarity, when fine-grained distinctions are distinguished between the nearest neighbors with similar semantic.

In this paper, we propose to exploit more complementary strengths of CNN features of different layers in a simple but effective way. Our method attempt to highlight the effectiveness of low-layer similarity, when measuring fine-grained similarity between query image and its nearest neighbors with similar semantic. In other words, low-layer feature is used to refine the ranking result of high-layer feature, rather than concatenating multiple layers directly. As shown in Fig. 2, high-layer feature is not powerful enough to describe the detail information, while low-layer feature suffers from background clutter and semantic ambiguity. In the manner of direct concatenation, low-layer similarity can not play a vital role in distinguishing fine-grained distinction due to the influence of high-layer similarity. Using a mapping function, our method takes more advantage of the low-layer feature measuring fine-grained similarity between query image and its nearest neighbors with same semantic. In experiments, we demonstrate that our method performs better than single-layer feature, concatenation of multiple layers and other hand-crafted features based methods.

2. Related work

Traditional CBIR approaches rely on powerful hand-crafted features and effective encoding methods. For much of the past decades, Bag-of-Words (BoW) based methods [9,18] were considered to be the state of the art. Built on the powerful locally invariant features like SIFT [8], BoW can be robust to represent images with the variations on scaling, translation, rotation, and so on. As a replacement for BoW, vector locally aggregated descriptor(VLAD) [11] was proposed to capture a compact representation for image and achieved great result. Later, some additional technologies can be applied to further boost VLAD's performance, such as intra-normalization [13], power-law normalization [14], singed square root [19] and so on. Multi-VLAD [13] constructed multi-level VLAD descriptors and matched them to improve localization accuracy, where RootSIFT was utilized to boost the performance of retrieval task. Covariant-VLAD (CVLAD) [14] also used RootSIFT and applied the power-law normalization to update the individual components of VLAD descriptor.

Other holistic features encode more global spatial information, where Fisher Vector(FV) [10] is the best known descriptor of this kind. There are also many variants of FV. For example, [20] applied some stand binary encoding techniques due to the density of FV, and introduced a simple normalization procedure to improve retrieval accuracy. Considering the dimensionality reduction of FV, [12] utilized ImageNet to discover discriminative low-dimensional subspace, where one hidden unit layer and one classifier output layer were added on top of FV. Although the low-dimensional activations of hidden layer were used as descriptors for image retrieval, the representations were still based on the hand-crafted features (SIFT and local color histograms), rather than the feature map learned from the input image in CNN.

Inspired by the success of image classification, feature extractors of CNN also were applied to other recognition tasks. DeCAF [21] firstly released such feature extractors along with all associated network parameters. Since low layers are unlikely to contain rich semantic information, only feature extractors from the last convolutional layer and the first two fully-connected layers were evaluated. Some pixel labeling tasks tried to enrich pixel representation with semantic information by concatenating feature extractors from different layers, such as object segmentation [22,23] and boundary detection [24]. These tasks achieved promising results with the concatenation of features from high layer and low layer.

OverFeat [25] firstly investigated the use of CNN features for image retrieval task, where the features of last layer did not outperform the SIFT based methods with BoW and VLAD encoding. Neural Codes [15] examined neuron activations of high layers for image retrieval, where the first two fully-connected layers and the last pooling layer were used to evaluate respectively. Although the feature exactors of retrained

Fig. 2. The ranking results searched by different CNN features. From top to bottom, the rows show the ranking results of high-layer feature, low-layer feature, direct concatenation and our method. High-layer feature is extracted from *fc*1 layer, low-layer feature is extracted form *conv*4 layer, direct concatenation and our method use these two layers. The green border denotes positive candidate image, while red border denotes false candidate image. Both query image and candidate images are sampled from Oxford 5K dataset. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

CNN can achieve great performance, collecting training samples and retraining stage require significant amounts of human and computing resources. MOP-CNN [17] aggregated features extracted from different layers with VLAD, where one fully-connected layer and two pooling layers were preselected. The feature exactors of low layers can perform better on instance-level image retrieval. This observation was further confirmed and extended by many encouraging experimental results [16], where all convolutional layers extracted from OxordNet [2] and GoogLeNet [3] are evaluated respectively. Obviously, different layers focus on extracting different information from image, the complementarity of low layer and high layer can improve the image retrieval task further.

3. The method

To describe image, the activation map of convolutional layer need to be aggregated into a feature vector. In this paper, the feature of fully-connected layer is the output of corresponding layer, while the feature of convolutional layer is the aggregated feature. In particular, the similarity is measured using cosine distance, thus the range of single-layer similarity is [0,1].

3.1. The similarity using multi-layer features

In the existing methods, the similarity of feature concatenation S is obtained as a sum over different layers:

$$S = \sum_{i=1}^{C} S_i \tag{1}$$

where C layers are selected to measure similarity, and S_i is the similarity of i^{th} selected layer between two images.

In order to highlight low-layer feature, we design a mapping function f_i, and the similarity of our method can be summarized as:

$$S = \sum_{i=1}^{C} f_i(S_i) \tag{2}$$

The mapping function f_i is designed as:

$$f_i(S_i) = \begin{cases} t_i + (1 - t_i)\left(\dfrac{S_i - t_i}{1 - t_i}\right)^p & S_i > t_i \\ S_i & S_i <= t_i \end{cases} \tag{3}$$

where t_i is a threshold of similarity of the ith selected layer, and the range of p is (0,1).

With this designed function, we aim to weaken the difference between the similarities in range $[t_i,1]$. Considering the combination of two layers, the effectiveness of high-layer similarity in range $[t_i,1]$ will be weakened, and low-layer similarity can distinguish fine-grained distinction better. In practical experiments, higher layer uses a lower threshold, that is $t_1 \le \ldots \le t_i \le \ldots \le t_C$, and the threshold t_C of lowest layer is set to 1.

3.2. Analysis of mapping function

The graph of f_i can explain this seemingly complex function, as shown in Fig. 3(a). The mapping function is designed to decrease the difference between the similarities greater than threshold t_i. The exponent p controls the degree of weakening. In particular, when p is set to 0, our method will be turned into re-ranking high-layer ranking results using low-layer similarity. When p is set to 1 or t_i is set to 1, the similarity of ith layer will be used directly and our method will be turned into concatenating different layers directly.

Obviously, the mapping function of Eq. (3) is the key to tap complementary strengths of different layers. To avoid a ad-hoc problem, we attempt to set the fixed thresholds t and p. For ease of explanation, we use two layers (*fc*2 and *conv*4) to analyze the impact of threshold t and exponent p of *fc*2 layer. As previously mentioned, the threshold t of the lowest layer is set to 1, thus the similarity of *conv*4 is used directly. In this condition, the mapping function only works on the similarity of *fc*2 layer. In particular, the curve shows the performance of direct concatenation when t of *fc*2 layer is set to 1.

The results with different settings are shown in Fig. 3(b)–(d), which demonstrate the mapping function can exploiting more complementary strengths. Though the different settings lead to improvement differences, most settings make improvements on three datasets. Meanwhile, the parameters t and p achieving best improvement are similar across datasets, where t is set to 0.2 and p is set to 0.45. The proposed method will not lead to a ad-hoc problem. The details of parameters setting will be introduce in the following experiments.

4. Preliminary

4.1. Single-layer feature

We train a CNN from ImageNet ILSVRC 2012 [26] as an example. The structure of this network is introduced by Krizhevsky et al. [1],

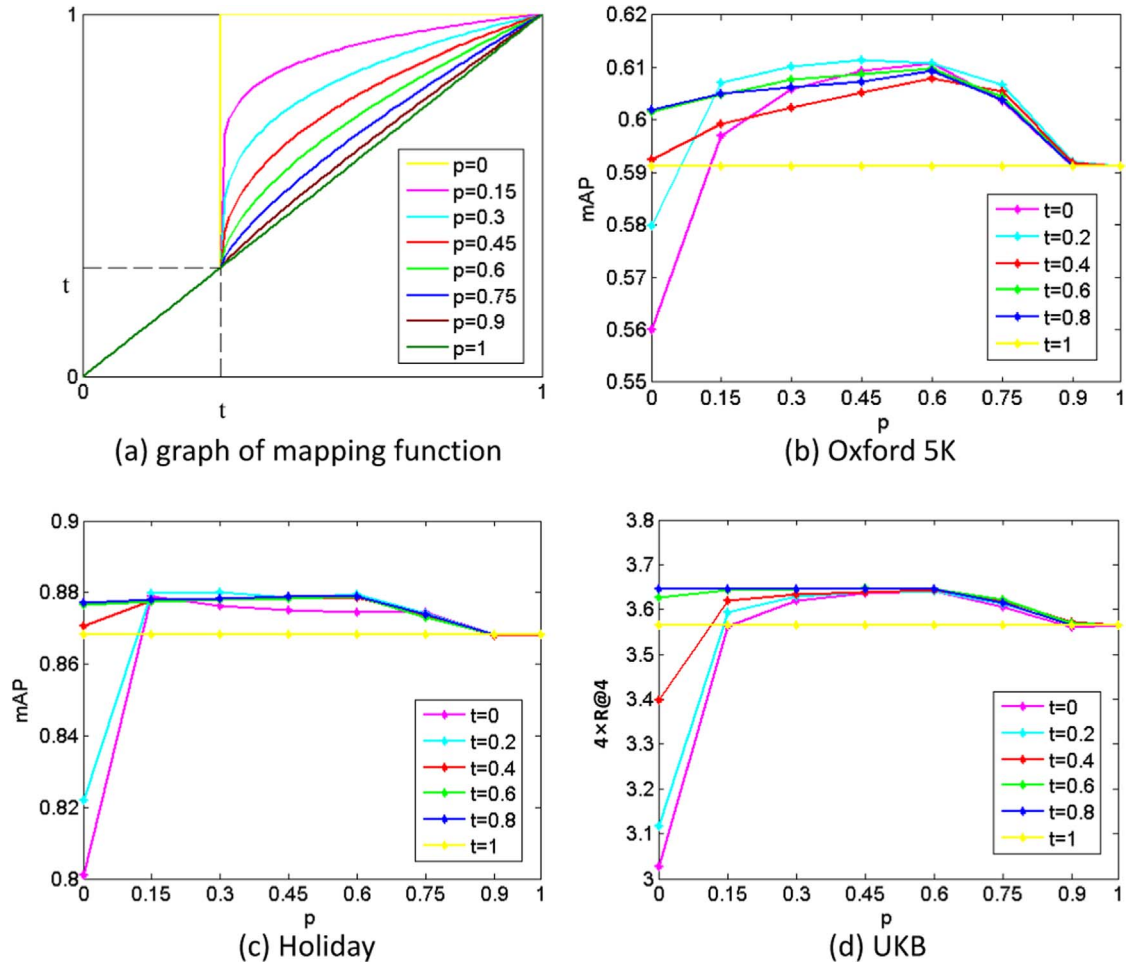

Fig. 3. Part (a) shows the graph of mapping function with fixed t and different p. Part (b)–(d) show the performances with different settings on Oxford 5K, Holiday and UKB respectively. (Better viewed in color). (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

which contains five convolutional layers and three fully-connected layers. The settings of all layers are the same as [1] except without using local contrast normalization layers, since they have been proven to be not helpful for ImageNet classification. In the following sections, *fci* denotes the *i*th fully-connected layer for short, and *convi* denotes the *i*th convolutional layer. In experiments, we select three convolutional layers (*conv*3, *conv*4 and *conv*5) and two fully-connected layers (*fc*1 and *fc*2) to measure the similarity between images.

As previously mentioned, the output of fully-connected layer is extracted as feature directly. We extract the output of convolutional layer as a set of local features, and train the visual words using approximate k-means (AKM) [27] on over 4 M features. Then we adopt BoW to encode the output of convolutional layer as a universal presentation for image.

4.2. The detail of benchmarks

We demonstrate our methods on three well-known benchmarks of image retrieval: Oxford 5 K [27], Holiday [28] and UKB [29].

Oxford 5K is the Oxford Buildings dataset, which consists of 5062 photographs corresponding to major Oxford landmarks. Images corresponding to 11 landmarks are manually annotated. The 55 hold-out queries evenly distributed over those 11 landmarks are provided.

Holiday is the INRIA Holidays datset, which consists of 1491 vacation photographs corresponding to 500 groups based on same scene or object. One image from each group serves as a query.

UKB is the University of Kentucky Benchmark dataset, which includes 10,200 indoor photographs of 2550 objects (4 photos per object). Each image is used to query the rest of the dataset. Following conventional setting, the performance of Oxford 5 K and Holiday is reported as a mean average precision (mAP) over the provided queries. The performance of UKB is reported as the average number of same-object images within the top-4 results, and is a number between 0 and 4. In particular, the shorter edges of all images are resized as 256.

5. Experiment

5.1. The result using single-layer feature

For ease of comparison, we report the performances of single-layer features extracted from our CNN. Table 1 shows the result comparison on three benchmarks. The numbers of visual words are set to 50 K, 100 K and 100 K for Oxford 5 K, Holiday and UKB respectively.

The *conv* layers can outperform the *fc* layers on Oxford 5 K, since the query images are category-level similar and the *fc* features are not powerful enough to describe fine-grained information. Although the Holiday and UKB datasets also aim to solve the task of instance-level

Table 1
The performance comparison between different layers.

	*conv*3	*conv*4	*conv*5	*fc*1	*fc*2
Oxford 5K	0.489	0.560	0.487	0.458	0.488
Holiday	0.809	0.801	0.806	0.847	0.855
UKB(4×R@4)	3.01	3.03	2.23	3.28	3.45

image retrieval, the query images belong to various categories. Thus, the *fc* features with semantic information can achieve better performance on these two datasets.

5.2. Comparison of different combinations of two layers

We begin from testing the combinations of two layer to validate the effectiveness of mapping function. Our method aims to utilize the complementarity between different layers, that is semantic similarity of high-layer and fine-grained similarity of low-layer, hence we extract high-layer feature from *fc* layer and low-layer feature from *conv* layer. For a fair comparison, we set $t_1 = 0.2$, $p_1 = 0.45$ and $t_2 = 1$ for all combinations of two layers.

Table 2 reports the performance of the combinations between *fc* layers and *conv* layers on three benchmarks, where our method outperforms the direct concatenation method. To some extend, the performance of combination is determined by single layer. For example, *fc*2 layer outperforms *fc*1 layer generally, and the combinations with *fc*2 layer can achieve better performance. On Oxford 5 K, *conv*4 achieve competitive performance and the combinations with it can outperform other combinations.

5.3. Comparison of different combinations of three layers

In this experiments, the image similarity is measured by three layers, thus we need to set two group parameters for top two layers. For the first layer, we set $t_1 = 0.2$, $p_1 = 0.45$ following the setting of two-layer combination. For the second layer, we set $t_2 = 0.4$ and $p_2 = 0.45$. Similarly, we set $t_3 = 1$ for the bottom layer.

Based on the foregoing analysis and result, the performance of layer combination is determined by single layer. We select *fc*2 layer as the high-level layer, since the two-layer combinations with *fc*2 outperform the combinations with *fc*1 significantly. Thus, we test the three-layer combinations with *fc*2 layer. Table 3 reports the performance, where our method also outperforms the direct concatenation method.

Obviously, the combination of three layers can achieve better performance. However, the improvement from combination of threes layers to combination of two layers is less than the improvement from single-layer to combination of two layers. To analyze this issue, we test the combinations of two *conv* layers. The performance is reported in Table 4, where the parameters setting follows Section 5.2. Comparing

Table 2
The performance comparison between different combinations of two layers.

		Oxford 5K		Holiday		UBK	
		*fc*1	*fc*2	*fc*1	*fc*2	*fc*1	*fc*2
Direct concatenation	conv3	0.511	0.532	0.859	0.864	3.42	3.54
	conv4	0.568	0.591	0.857	0.865	3.42	3.57
	conv5	0.507	0.527	0.857	0.863	3.46	3.56
Our Method	conv3	0.544	0.562	0.875	0.878	**3.52**	**3.65**
	conv4	**0.594**	**0.611**	0.874	**0.880**	3.51	**3.65**
	conv5	0.548	0.564	**0.877**	0.879	3.48	3.62

Table 3
The performance comparison between some selected combinations of three layers.

		Oxford 5 K	Holiday	UKB
Direct concatenation	*fc*2+conv5+conv4	0.602	0.872	3.62
	*fc*2+conv4+conv3	0.601	0.872	3.59
	*fc*2+conv5+conv3	0.566	0.873	3.62
Our Method	*fc*2+conv5+conv4	0.612	**0.889**	**3.69**
	*fc*2+conv4+conv3	**0.613**	0.883	3.66
	*fc*2+conv5+conv3	0.578	0.884	**3.69**

Table 4
The performance comparison between the combinations of two *conv* layers.

	Oxford 5K	Holiday	UKB
conv5+conv4	**0.567**	0.809	**3.243**
conv5+conv3	0.518	**0.822**	3.240
conv4+conv3	0.563	0.820	3.131

Table 5
The single-layer performance comparison of VGG-16.

	c3_1	c4_1	c5_1	*fc*1	*fc*2
Oxford 5K	0.381	0.504	0.570	0.439	0.390
Holiday	0.837	0.872	0.847	0.845	0.835
UKB(4×R@4)	2.77	2.91	3.31	3.54	3.45

Table 6
The two-layer performance comparison of VGG-16.

		Oxford 5 K		Holiday		UBK	
		*fc*1	*fc*2	*fc*1	*fc*2	*fc*1	*fc*2
Direct concatenation	c3_1	0.480	0.428	0.867	0.879	3.54	3.47
	c4_1	0.561	0.522	0.892	0.885	3.54	3.52
	c5_1	0.587	0.557	0.887	0.881	3.58	3.58
Our Method	c3_1	0.508	0.467	0.876	0.881	3.58	3.54
	c4_1	0.573	0.548	**0.910**	**0.910**	3.61	3.57
	c5_1	**0.608**	**0.598**	0.904	0.907	**3.65**	**3.63**

Tables 2 and 4, we can find that the improvement from single *conv* layer to the combination of two *conv* layers is less than the improvement from single layer to the combination of high- and low-level layers. For instance, the layer combination of *fc*2 and *conv*3 achieves the improvement from 0.488(*fc*2) to 0.562 on Oxford 5 K dataset, while the layer combination of *conv*4 and *conv*3 achieves the improvement from 0.560(*conv*4) to 0.563. The improvement is small from single *conv* layer to two *conv* layers, which also leads to small improvement from two-layer combination to three-layer combination. As previously mentioned, the *conv* layers extract the local information, while the *fc* layers represent global information. Though different layers tend to encode the information of different levels, the information difference between *conv* layers is smaller than the difference between *conv* layer and *fc* layer. The complementarity between *fc* layer and *conv* layer is stronger, thus the two-layer combination of *fc* layer and *conv* layer can achieve remarkable improvement.

5.4. The fusion strategy on state-of-the-art pre-trained model

In this part, we attempt introduce the state-of-the-art CNN model to show the generalization of our method. To this end , we employ the pre-trained VGG-16 model [2] in the multi-layer fusion strategy to make results more convincing.[2] It is very time-consuming to test all layer combinations of VGG-16, since there are 13 convolutional layers and 3 fully-connected layers. Thus, we also select two fully-connected layers (*fc*1 and *fc*2) and three convolutional layers (c3_1, c4_1 and c5_1) to measure the similarity between images.

Table 5 reports the single-layer results of VGG-16 on three benchmarks. Table 6 reports the performance of the combinations between *fc* layers and *conv* layers. For the three-layer combination of VGG-16, we select *fc*1 layer as the high-level layer, since the two-layer combinations with *fc*1 outperform the combinations with emphfc2. Thus, we test the

[2] The released VGG-16 model achieved 29.5% top-1 classification error rate on ILSVRC-2012. (http://www.robots.ox.ac.uk/vgg/research/very_deep/).

Table 7
The three-layer performance comparison of VGG-16.

		Oxford 5 K	Holiday	UKB
Direct concatenation	*fc*1+c5_1+c4_1	0.604	0.900	3.57
	*fc*1+c4_1+c3_1	0.563	0.896	3.55
	*fc*1+c5_1+c3_1	0.592	0.895	3.57
Our Method	*fc*1+c5_1+c4_1	**0.615**	**0.914**	**3.68**
	*fc*1+c4_1+c3_1	0.565	0.911	3.63
	*fc*1+c5_1+c3_1	0.610	0.912	**3.68**

Table 8
The performance comparison between our method and other existing methods.

Method	Oxford 5K	Holiday	UKB
Sparse-coded features [30]	–	0.727	3.67
Triangulation embedding [33]	0.433	0.617	3.40
BoW-200 K [19]	0.364	0.540	2.81
VLAD [19]	0.304	0.556	3.28
Multi-VLAD [13]	0.558	0.653	–
CVLAD [14]	0.514	0.827	3.62
Improved FV [20]	0.414	0.626	3.44
Fisher+color [12]	–	0.723	3.08
Neural Codes [15]	0.545	0.793	3.29
MOP-CNN [17]	–	0.802	–
CNNaug-ss [31]	0.680	0.843	–
Multi-resolution Spatial Search [32]	0.844	0.897	–
Features from OxfordNet [16]	0.649	0.838	–
Features from GooLeNet [16]	0.581	0.840	–
Our method (two layers, AlexNet)	0.611	0.880	3.65
Our method (three layers, AlexNet)	0.612	0.889	**3.69**
Our method (two layers, VGG-16)	0.608	0.904	3.65
Our method (three layers, VGG-16)	**0.615**	**0.914**	3.68

three-layer combinations with *fc*1 layer, Table 7 reports the performance of the combinations between *fc*1 layer and two *conv* layers. Considering Tables 5–7, the proposed method also reaches similar conclusions when employing the pre-trained VGG-16 model.

5.5. Comparison of state-of-the-art methods

As the last part of our experiment, we compare our method with the state-of-the-art methods performed on image retrieval task. To verify our method, we employ two CNN models: our trained AlexNet and pre-trained VGG-16. For our trained AlexNet, we utilize the two-layer combination of *fc*2 and *conv*4 and the three-layer combination of *fc*2, *conv*5 and *conv*4, where the parameters setting follows the Sections 5.2 and 5.3. For pre-trained VGG-16, we utilize the two-layer combination of *fc*1 and c5_1 and the three-layer combination of *fc*1, c5_1 and c4_1. Table 8 reports the performances for image retrieval on three datasets, where top part shows the results of hand-crafted features based methods (from Sparse-coded features [30] to Fisher+color [12]) and bottom part shows the results of CNN features based methods (from Neural Codes [15] to Our method).

With no doubt, our method outperforms all previous methods using hand-crafted features, which confirms CNN features can work better on CBIR task. Our method also significantly better than the method using pre-selected single-layer CNN feature (denoted as Neural Codes) and the method using direct concatenation of three layers (denoted as MOP-CNN). In particular, [16] extracted features from different scales, and selected the layer with best performance. Our method extract feature from single scale and achieve competitive performance on Oxford 5K dataset. It should be noted that some CNN feature based methods introduce the spatial information and achieve great improvement, such as spatial search [31,32]. Our method demonstrates the complementarity between high-layer and low-layer, and proposes the mapping function to boost the performance. Similarly, we believe that the performance of our method will be improved further using the spatial information.

6. Conclusion

Inspired by the success of CNN, we propose to further improve the performance of image retrieval using complementary information in different layers. Preliminary results demonstrate the effectiveness of our method in measuring similarity for CBIR. Our future work contains two parts. On one hand, we will improve our method further. As foregoing analysis, the performance of combination is determined by single-layer performance. Other methods can be used to encode the features of convolutional layer, such as VLAD. Meanwhile, our method can be applied on more powerful networks, such as VGGNet and GooLeNet. On another hand, we will explore the trade-off between retrieval performance and storage space, and attempt to improve our method's generality. For example, the aggregated feature of convolutional layers need to be compressed to a low-dimensional representation.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (No. 61472103) and Key Program (No. 61133003).

References

[1] A. Krizhevsky, I. Sutskever, G.E. Hinton, Imagenet classification with deep convolutional neural networks, Adv. Neural Inf. Process. Syst. 25 (2) (2012) 2012.
[2] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, Eprint Arxiv.
[3] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, Going deeper with convolutions, in: Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1–9.
[4] H. Kaiming, Z. Xiangyu, R. Shaoqing, S. Jian, Deep residual learning for image recognition, in: arXiv preprint arXiv:1512.03385v1, 2015.
[5] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Computer Vision and Pattern Recognition (CVPR), 2014, pp. 580–587.
[6] Y. Taigman, M. Yang, M. Ranzato, L. Wolf, Deepface: closing the gap to human-level performance in face verification, in: Computer Vision and Pattern Recognition (CVPR), 2014, pp. 1701–1708.
[7] Y. Wei, Y. Kuiyuan, B. Yalong, Y. Hongxun, R. Yong, Dnn flow: Dnn feature pyramid based, in: British Machine Vision Conference (BMVC), 2014.
[8] D.G. Lowe, Object recognition from local scale-invariant features, in: Proceedings of the International Conference on Computer Vision (ICCV), 1999, pp. 1150–1157.
[9] J. Sivic, A. Zisserman, Video google: A text retrieval approach to object matching in videos, in: Proceedings of the International Conference on Computer Vision (ICCV), 2003, pp. 1470–1477.
[10] F. Perronnin, J. Sánchez, T. Mensink, Improving the fisher kernel for large-scale image classification, in: Proceedings of the European Conference on Computer Vision (ECCV), 2010.
[11] H. Jegou, M. Douze, C. Schmid, P. Perez, Aggregating local descriptors into a compact image representation, in: Computer Vision and Pattern Recognition (CVPR), 2010, pp. 3304–3311.
[12] A. Gordoa, J.A. Rodriguez-Serrano, F. Perronnin, E. Valveny, Leveraging category-level labels for instance-level image retrieval, in: Computer Vision and Pattern Recognition (CVPR), 2012, pp. 3045–3052.
[13] R. Arandjelovic, A. Zisserman, All about vlad, in: Computer Vision and Pattern Recognition (CVPR), 2013, pp. 1578–1585.
[14] W.L. Zhao, G. Gravier, H. Jegou, Oriented pooling for dense and non-dense rotation-invariant features, in: British Machine Vision Conference (BMVC), 2013, pp. 99.1–99.11.
[15] A. Babenko, A. Slesarev, A. Chigorin, V. Lempitsky, Neural codes for image retrieval, in: Proceedings of the European Conference on Computer Vision (ECCV), 2013, pp. 584–599.
[16] Y.H. Ng, F. Yang, L.S. Davis, Exploiting local features from deep networks for image retrieval, in: Computer Vision and Pattern Recognition Workshops (CVPRW), 2015 IEEE Conference on, 2015.
[17] Y. Gong, L. Wang, R. Guo, S. Lazebnik, Multi-scale orderless pooling of deep convolutional activation features, in: Proceedings of the European Conference on Computer Vision (ECCV).
[18] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, Y. Gong, Locality-constrained linear coding for image classification, in: Computer Vision and Pattern Recognition (CVPR), 2010, pp. 3360–3367.
[19] J. Herv, P. Florent, D. Matthijs, S. Jorge, P. Patrick, S. Cordelia, Aggregating local

image descriptors into compact codes, IEEE Trans. Pattern Anal. Mach. Intell. 34 (9) (2012) 1704–1716.
[20] F. Perronnin, Y. Liu, J. Sanchez, H. Poirier, Large-scale image retrieval with compressed fisher vectors, in: Computer Vision and Pattern Recognition (CVPR), 2010, pp. 3384–3391.
[21] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, T. Darrell, Decaf: A Deep Convolutional Activation Feature for Generic Visual Recognition, University of California Berkeley Brigham Young University, 2013, pp. 647–655.
[22] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: Computer Vision and Pattern Recognition (CVPR), 2015, pp. 1337–1342.
[23] B. Hariharan, P. Arbelaez, R. Girshick, J. Malik, Hypercolumns for object segmentation and fine-grained localization, in: Computer Vision and Pattern Recognition (CVPR), 2015, pp. 447–456.
[24] G. Bertasius, J. Shi, L. Torresani, High-for-low and low-for-high: efficient boundary detection from deep object features and its applications to high-level vision, in: International Conference on Computer Vision (ICCV), 2015, pp. 504–512.
[25] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, Y. Lecun, Overfeat: Integrated Recognition, Localization and Detection Using Convolutional Networks, Eprint Arxiv.
[26] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A.C. Berg, L. Fei-Fei, ImageNet Large Scale Visual Recognition Challenge, Int. J. Comput. Vis. (IJCV) 115 (3) (2015) 211–252. http://dx.doi.org/10.1007/s11263-015-0816-y.
[27] J. Philbin, O. Chum, M. Isard, J. Sivic, A. Zisserman, Object retrieval with large vocabularies and fast spatial matching, in: Computer Vision and Pattern Recognition (CVPR), 2007, pp. 1–8.
[28] H. Jegou, M. Douze, C. Schmid, Hamming embedding and weak geometric consistency for large scale image search, in: European Conference on Computer Vision (ECCV), 2008, pp. 1.1–1.1.
[29] D. Nister, H. Stewenius, Scalable recognition with a vocabulary tree, in: Computer Vision and Pattern Recognition (CVPR), 2006, pp. 2161–2168.
[30] T. Ge, Q. Ke, J. Sum, Sparse-coded features for image retrieval, in: Proceedings of the British Machine Vision Conference (BMVC), 2013.
[31] A.S. Razavian, H. Azizpour, J. Sullivan, S. Carlsson, Cnn features off-the-shelf: An astounding baseline for recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2014, pp. 512–519.
[32] A.S. Razavian, J. Sullivan, A. Maki, S. Carlsson, A baseline for visual instance retrieval with deep convolutional networks, Computer Science.
[33] H. Jegou, A. Zisserman, Triangulation embedding and democratic aggregation for image search, in: Computer Vision and Pattern Recognition (CVPR), 2014, pp. 3310–3317.

Wei Yu received the B.S and M.S. degrees from the Harbin Institute of Technology, Harbin, China, in 2009 and 2012, respectively, and is currently working toward the Ph.D. degree in computer science and technology at the Harbin Institute of Technology, Harbin, China. He was a Research Intern with Web Search and Mining Group, Microsoft Research, Beijing, China, from 2013 to 2015. His research interests include computer vision, multimedia, and machine learning.

Kuiyuan Yang received the B.E. and Ph.D. degrees in automation from the university of Science and Technology of China, Hefei, China, in 2007 and 2012, respectively. He is currently a Research Staff Member with the Web Search and Mining Group, Microsoft Research, Beijing, China. His current research interests include computer vision, multimedia, and machine learning.

Hongxun Yao (M'11) received the B.S. and M.S. degrees from the Harbin Shipbuilding Engineering Institute, Harbin, China, in 1987 and 1990, respectively, and the Ph.D. degree from the Harbin Institute of Technology, Harbin, China, in 2003, all in computer science. She is currently a Professor with the School of Computer Science and Technology, Harbin Institute of Technology. She has authored or coauthored five books and over 200 scientific papers. Her research interests include pattern recognition, multimedia processing, and digital watermarking.

Xiaoshuai Sun received the B.S. degree in computer science from Harbin Engineering University, Harbin, China, in 2007. He is currently pursuing the Ph.D. degree with the School of Computer Science and Technology, Harbin Institute of Technology, Harbin. He was a Research Intern with Microsoft Research Asia, Beijing, China, from 2012 to 2013, and also a recipient of the Microsoft Research Asia Fellowship in 2011. He holds two authorized patents and has authored over 40 referred journal and conference papers in the field of multimedia and computer vision.

Pengfei Xu received the B.S., M.S. and Ph.D. degrees from the Harbin Institute of Technology, Harbin, China, in 2007, 2009 and 2013 respectively. His research interests include image annotation, CBIR, video analysis and machine learning.