SIFT Matching with CNN Evidences for Particular Object Retrieval

Guixuan Zhang, Zhi Zeng, Shuwu Zhang, Yuan Zhang, Wanchun Wu

1. Introduction

This paper considers the task of particular object retrieval. Given a query image in which a particular object has been selected, the retrieval system should return from its corpus a set of relevant images in which that object appears. This is a harder problem than whole-image retrieval, since the query object may be cluttered or occluded with diverse backgrounds in the returned images. Some examples are shown in Fig. 1.

Most object retrieval systems are based on matching of local features. There are two steps for extracting local features. The first step detects keypoints of interest. This step also delineates a local patch around every keypoint and normalizes the local patch into fixed-size. The second step describes the normalized patches based on algorithms such as SIFT [1]. However, matching the 128-dimensional SIFTs directly costs a lot of time. To speed up the matching process, bag-of-words (BoW) model is widely used [2,3]. BoW model defines a visual dictionary and quantizes the local features to visual words. An image can be represented by a frequency histogram of visual words. Local features are matched if they are quantized to the same visual word, so the similarity between two images can be expressed by the inner product of their BoW representations. Inverted index which exploits the sparsity of BoW representation also makes the search efficient. Since matching the SIFTs based on visual words may be too coarse, Jegou et al. [4] propose Hamming embedding (HE) to improve the matching accuracy. HE adds a compact binary signature to each SIFT when quantizing it. Then two coarsely matched SIFTs will be filtered out if their Hamming distance between binary signatures is larger than a threshold.

However, SIFT only describes the local gradient distribution and serves as a low-level representation, which is often not sufficiently distinctive to prevent false matches. As shown in Fig. 2, some local patches are very similar in the SIFT feature space, but they depict different contents. This challenging problem is mainly due to the semantic gap. So seeing the big picture and adopting semantic clues may be a good solution to bridge the semantic gap.

Recently, deep convolutional neural networks (CNN) have been proven to achieve state-of-the-art performance in many computer vision tasks such as image classification [6,7], object detection [8] or semantic segmentation [9]. With the deep architectures, semantic abstractions that are close to human cognition can be learned. A number of recent works show that CNN features trained
on large and diverse datasets such as ImageNet [10] can be used to solve tasks for which they have not been trained. Particularly for image retrieval, many works have adopted solutions based on off-the-shelf features extracted from a pre-trained CNN, achieving promising performance on popular benchmarks [11–14].

In this paper, we propose to adopt the semantic-aware CNN features to improve the SIFT matching accuracy, which helps to improve the particular object retrieval performance. Considering that global CNN representation is sensitive to background clutter, which is common in object retrieval (see Fig. 1), we extract CNN feature at the object-level. We detect a number of potential objects in the images and extract CNN feature on each object. For each pre-matched SIFT pair, we choose appropriate semantic evidence from the candidate CNN features in a query-adaptive manner, and use them to verify the SIFT match quality. By fusing low-level and high-level clues, high visual matching accuracy can be achieved.

The major contributions of this paper are summarized in the following aspects. First, we adopt the object-level CNN features to improve the SIFT matching accuracy. A query-adaptive method is proposed to choose appropriate CNN evidences to verify the SIFT match quality. Second, two different visual matching verification functions are introduced. We evaluate these two functions and show that they have different effectiveness for different types of datasets. Third, we explore the suitability of fine-tuning the CNN to obtain better semantic evidences for our proposed method. Extensive experiments on benchmark datasets demonstrate that our results compare favorably to the state-of-the-art methods with acceptable memory usage and query time.

The remainder of the paper is organized as follows. Related works are reviewed in Section 2. We show our approach in details in Section 3. After that, we provide the experimental results and comparisons in Section 4. Final conclusions are in Section 5.

2. Related work

The combination of BoW model and local features is widely used in many object retrieval systems. The BoW model is an approximation to the direct matching of local features, so that it can achieve scalability for large scale object retrieval. Popular local features include SIFT [1] and SURF [15]. SIFT descriptor and its extension RootSIFT [16] have shown good performance for most applications. In BoW model, a visual dictionary is trained on an independent set of local features. Then two local features match if they are quantized to the same visual word. However, the coarse quantization reduces the discriminative power of local features. Dissimilar local features may be quantized to the same visual word and cannot be distinguished from each other. Some methods [4,17,18] improve the discriminative power of local features in the quantizing process. Hamming embedding (HE) [4] is a widely used solution, which refines a binary code for each SIFT when quantizing it. Two coarsely matched SIFTs will be filtered out if their Hamming distance between binary codes is larger than a threshold.

Since SIFT only describes the local gradient distribution, it is not sufficiently distinctive to prevent false matches. As illustrated in Fig. 2, even though some patches are similar in the SIFT feature space, they depict different contents and belong to the false matches. To increase the discriminative power, Zheng et al. [19] adopt color names to further describe the local patches. Some works adopt spatial constraints to verify the SIFT matches [20–23]. Though these methods can alleviate the impact of false matches to some extent, they introduce more complexity.
Another line for object retrieval is adopting the global representation. Local features are aggregated into a global representation with encoding algorithms such as Fisher vector [24] and VLAD [25,26]. Since every image is represented with a single vector, the retrieval efficiency can be improved by adopting approximate nearest neighbor search techniques [27–29]. However, these encoding methods often use a small visual vocabulary, which limits their performance for object retrieval.

Recently, with the success of convolutional neural networks [30] in many computer vision areas [6,8,9], the research focus has transferred to deep learning methods. Many works adopt the semantic-aware features extracted from a pre-trained CNN model for image retrieval tasks. [12,31] use a global CNN representation extracted from the whole image. [13,32] extract CNN features from multi-scale image regions and encode them to a single vector, resulting in a holistic representation. [11,14] perform multiple search for each query since they extract a number of CNN features from an image. All these methods show promising results for image retrieval.

Due to the semantic awareness of CNN, it is straightforward to adopt CNN features to bridge the semantic gap, which is encountered when matching the low-level SIFTs. The closest work to ours is [33]. This work defines “true match” as a pair of keypoints which are similar on local, regional and global levels. The authors adopt the global and specified regional CNN features to verify the SIFT match. However, they overlook the interference of background noise. Since the object of interest often occupies only a small portion of the image, many relevant images are not similar on the global level in the CNN feature space. This problem may even lower the retrieval performance.

3. Our approach

The framework of our method is shown in Fig. 3. The feature extraction process is similar in both off-line and on-line phases. The difference is that we detect multiple object proposals only on the database side, which will be shown in Section 3.2. Both SIFT and CNN feature are stored in the indexing structure. In the on-line phase, we first match the SIFTs using bag-of-words model. Then for each pre-matched SIFT pair, the appropriate CNN evidences are chosen in a query-adaptive manner. With the visual matching verification function, we improve the SIFT matching accuracy based on these CNN evidences. Consequently, the object retrieval performance can be improved.

3.1. SIFT matching for object retrieval

In the BoW model, a visual vocabulary \( V = \{v_1, v_2, \ldots, v_k\} \) is trained with k-means algorithm, where \( k \) indicates the vocabulary size. Each SIFT is quantized to a nearest visual word \( v \) by quantizer \( q(\cdot) \). Two SIFTs are matched if they are quantized to the same word. Let \( X = \{x_1, x_2, \ldots, x_m\} \) and \( Y = \{y_1, y_2, \ldots, y_n\} \) be the SIFT set extracted from a query image \( Q \) and a database image \( I \), respectively. The similarity between two images is defined as

\[
\text{sim}(Q, I) = \frac{1}{\|x\| \cdot \|y\|} \sum_{x \in X, y \in Y} f_{\text{bow}}(x, y) \cdot \text{idf}^2,
\]

where \( \|\cdot\| \) is normalization factor and \( \text{idf} \) means inverse document frequency, which reduces the impact of the visual words that are more frequent [34]. The matching function \( f_{\text{bow}}(x, y) \) measures the similarity between two SIFTs. In conventional BoW model, the SIFT matching function is

\[
f_{\text{bow}}(x, y) = \begin{cases} 1 & \text{if } q(x) = q(y) \\ 0 & \text{otherwise} \end{cases}.
\]

Since the matching scheme based on visual words may be too coarse, Jégou et al. [4] introduce Hamming embedding to improve the matching accuracy. HE technique refines each SIFT \( x \) by a binary \( h(x) \). HE compares two SIFTs \( x \) and \( y \) assigned to the same word by computing the Hamming distance \( h_y = h(b(x), b(y)) \) between their binary signatures. Only if the Hamming distance is below a threshold \( \kappa \), they are considered as a match and non-zero similarity score is attached. The matching function of HE is:

\[
f_{\text{HE}}(x, y) = \begin{cases} \exp(-\frac{h_y^2}{\sigma^2}) & \text{if } q(x) = q(y) \text{ and } h(b(x), b(y)) \leq \kappa \\ 0 & \text{otherwise} \end{cases}.
\]

The weight \( w(h_y) = \exp(-h_y^2/\sigma^2) \) is a decreasing function of the Hamming distance, which provides a benefit in matching accuracy.
3.2. Object regions detection

An image usually contains several objects. The object appearance is more likely to keep consistent no matter how the background and image layout change. In particular object retrieval, the object of interest may occupy only a small portion of a positive database image. If we concentrate on the object-level content, the interference of background noise can be weakened or even eliminated. Moreover, the semantic information extracted from an object region could give a further description for the characteristics of keypoints located in this object region. In this paper, the object-level semantic information will be used as candidate evidence to verify the SIFT (or the corresponding keypoint) match.

To detect potential objects, we might refer to object proposal methods including selective search [35], BING [36] and EdgeBoxes [37], which are used to guide the search for object detection tasks. Considering the efficiency and effectiveness, we adopt EdgeBoxes. EdgeBoxes starts from a coarse sliding window pattern, and builds on object boundary estimation. It adds a subsequent refinement step to improve location. EdgeBoxes is an unsupervised method, so no parameters are learned. It provides a number of image windows which are the possible locations of objects. We sort the confidence scores of windows and preserve the top-ranked object regions. In order to take account of global clues, we also treat the whole image as an object region. For each image I, a set of object regions is constructed,

\[ R_I = \{ r_{i,1}, r_{i,2}, \ldots, r_{i,M-1} \}. \]

where \( r_{i,j} \) is the whole image, \( \{ r_{i,1}, r_{i,2}, \ldots, r_{i,M-1} \} \) means the regions detected by EdgeBoxes. The final number of object regions for each image is \( M \).

Assume that the corresponding keypoint of a SIFT \( y \) is located in \( T_y \) object regions. Then a set of relevant object regions for \( y \) is created,

\[ R_{Iy} = \{ r_{y,1}, r_{y,2}, \ldots, r_{y,r} \}. \quad R_{Iy} \subseteq R_I. \]

The whole image is treated as one object region, so every SIFT has at least one relevant region, i.e. \( T_y \geq 1 \).

Since the object which we want to search has been specified in the query image, we only detect objects on the database side. For a SIFT \( x \) in the query image \( Q \), its relevant object set is \( R_{Qx} = \{ r_Q \} \), where \( r_Q \) represents the query object.

We do not require EdgeBoxes to provide accurate object locations, since our objective is to weaken the interference of background noise when verifying the SIFT matches, rather than conducting the object detection task. An object region with coarse location could also provide effective information.

3.3. Compact CNN feature extraction

Deep convolutional neural networks have demonstrated good performance in image classification [6,7]. Recent works show that CNN features trained on ILSVRC [38] have semantic awareness and can be transferred to other image applications [11]. In this work, the CNN architecture proposed by Krizhevsky [6] and implemented by Caffe [39] is used. We refer to this architecture as AlexNet. In AlexNet, there are eight layers (5 convolutional layers and 3 fully connected layers). Given an object region, we sample it to 227 x 227 pixels, subtract the mean of pixel values, and feed the region through the network. The activations of the last convolutional layer and the first two connected layers are often employed as the off-the-shelf features. We name these three layers as Conv5, FC6 and FC7, and evaluate them in our experiments. The dimensionality of FC6 and FC7 feature is 4096. Since the activations of Conv5 are less invariant to translation, we adopt the max-pooling step [32] to address this problem. The dimension of Conv5 feature after max-pooling is 256, which is equal to the number of convolutional filters. All these CNN features are L2 normalized.

Considering the efficiency and memory usage, we transform the CNN features to compact binary codes. The locality-sensitive hashing (LSH) algorithm [40] is used and every CNN feature is transformed to a 128-bit binary signature. For the object region set \( R_I = \{ r_{i,1}, r_{i,2}, \ldots, r_{i,M-1} \} \) in image \( I \), its corresponding CNN feature set is \( C_I = \{ c_{i,1}, c_{i,2}, \ldots, c_{i,M-1} \} \), where \( c \) is 128-bit binary signature.

This paper also explores the suitability of fine-tuning CNN to obtain better feature representation for the object retrieval task. Each kind of the query object is treated as an independent class. We modify the CNN architecture so that the last output layer is adapted to the number of class labels of the new dataset. More implementation details and the evaluation of the fine-tuned CNN models will be shown in the experiment section.

3.4. Appropriate CNN evidences selection

\( x \) and \( y \) are two SIFT descriptors from query image \( Q \) and database image \( I \). Assume that \( (x, y) \) is a matched pair based on the matching function \( f_{bow} \) or \( f_{be} \). We have detected \( M \) object regions for the database image. According to the position of \( y \), we can obtain a relevant object region set \( R_{Iy} = \{ r_{y,1}, r_{y,2}, \ldots, r_{y,r} \} \). \( R_{Iy} \subseteq R_I \) as shown in Section 3.2. The corresponding compact CNN feature set for \( R_{Iy} \) is \( c_{Iy} = \{ c_{y,1}, c_{y,2}, \ldots, c_{y,r} \} \).

Note that there is only one object region in the query image. Denote its CNN feature as \( c_Q \). We choose the appropriate CNN evidence on the database side with the following function

\[ c_{(y \rightarrow x)} = \arg \min_{c_{y,r} \in c_{Iy}} h(c_{Q}, c_{y,r}), \]

where \( h(\cdot, \cdot) \) denotes the Hamming distance function. Such procedure is illustrated in Fig. 3.

In fact, the final clue that we use to verify the SIFT match is the Hamming distance \( h_t = \min_{r} h(c_{Q}, c_{y,r}) \). To demonstrate the effectiveness of this procedure, we show the distribution of the Hamming distance \( h_t \) in Fig. 4. We collect a set of true-matching and false-matching SIFT pairs from three datasets, which will be used to evaluate the proposed method in the experiment section. The true-matching SIFT pairs are selected according to the ground truth and have been geometrically verified. The false-matching SIFT pairs are denoted as the pairs 1) which have non-zero matching score and 2) which are not true matches according to the ground truth. It can be seen that the Hamming distance \( h_t \) separates the true-matching from the false-matching SIFT pairs quite well, which can serve as effective evidence to verify the SIFT match.

3.5. SIFT match verification function

Since we have obtained appropriate CNN evidence, it remains to design a function to verify the SIFT match. We introduce two different verification functions. The first one adopts the product rule:

\[ \phi(x, y) = \begin{cases} f(x, y) \times \exp\left(\frac{-\|x - y\|}{\theta}\right) & \text{if } f(x, y) \neq 0 \\ 0 & \text{otherwise} \end{cases}, \]

where \( f(x, y) \) is the SIFT matching function, either \( f_{bow}(x, y) \) or \( f_{be}(x, y) \) can be used. \( \theta \) is a parameter of the exponential function.
The second verification function adopts the sum rule:

\[ \phi_+(x, y) = \begin{cases} f(x, y) + \exp\left(\frac{-h_c^2}{\beta^2}\right) & \text{if } f(x, y) \neq 0 \\ 0 & \text{otherwise} \end{cases}, \]

where \( \beta \) is a parameter of the exponential function.

We conduct the verification process only for the SIFT pair \((x, y)\) which has been pre-matched according to the matching function \(f(x, y)\), i.e., the pair has non-zero matching score. Based on the verification function, the similarity between two images updates as follows:

\[ \text{sim}'(Q, I) = \frac{1}{\|x\| \|y\|} \sum_{x \in X, y \in Y} \phi(x, y) \cdot \text{idf}^2. \]

where \( \phi(x, y) \) can be either \( \phi_+(x, y) \) or \( \phi_-(x, y) \).

3.6. Indexing structure and searching scheme

We use inverted file to index the database images. As illustrated in Fig. 5(a), the SIFTs quantized to a particular visual word are stored in the corresponding inverted list. Each entry stores the image ID and the SIFT binary signature. In order to mark which object regions the keypoint is located in, we also store an indicator in the posting entry. We have detected \( M \) object regions for each database image. \( M \) bits are allocated for each keypoint (SIFT). If the keypoint is located in one region, the corresponding bit is set to 1, otherwise the bit is set to 0.

The compact CNN features (128-bit binary signature) are stored in separate memory space. Each image needs \( 16 \times M \) bytes to store the CNN features. The strategy is illustrated in Fig. 5(b).

The search scheme of our method is summarized in Algorithm 1. For each pre-matched SIFT pair, we should choose the appropriate CNN evidence to verify the match. Computing the Hamming distance of CNN signatures every time may make the search process complex. To speed up the process, we allocate extra memory to serve as look up table. All the Hamming distances are pre-computed and stored in the memory when a database image is visited for the first time. Then the Hamming distance can be read directly when the entry of this database image is visited next time.

4. Experiments

4.1. Datasets

We evaluate our method on three datasets for particular object retrieval: Oxford5k [20], Paris6k [41] and INSTRE [5]. All these three datasets specify a rectangular region delimiting the object in the image as a query. The correct results for a query are the other images which contain this object.

The Oxford5k dataset consists of 5063 images and the Paris6k dataset contains 6392 images. Both Oxford5k and Paris6k have 55 query images, comprising 11 different buildings. The Flickr100k [41] is often added to Oxford5k to test the scalability of retrieval algorithm, resulting in Oxford105k. INSTRE is a new
Algorithm 1 Search scheme with CNN verification.

1: $S_n$, the score of the nth database image; //Initialize with 0
2: $F_n$, the visit flag of the nth database image; //Initialize with false
3: $D_n = \{d_1, d_2, \ldots, d_M\}$, the Hamming distance look up table of the n-th database image;
4: for each SIFT $x$ in query $Q$ do
5: Quantize $x$ to the nearest visual word $v$ with quantizer $q(\cdot)$;
6: for all indexed features $y$ in the inverted list of $v$ do
7: if $f(x, y) > 0$ then
8: $n = \text{imageID}(y)$;
9: if $F_n == \text{false}$ then
10: Calculate all the Hamming distances between $C_l$ and $C_q$, store them as $D_n$;
11: $F_n = \text{true}$;
12: end if
13: Read the corresponding Hamming distances from $D_n$ according to the object region indicator, and find the minimum Hamming distance, denoted as $h_c$;
14: $S_n += \phi(x, y) \cdot \text{idf}^2$;
15: end if
16: end for
17: end for
18: Results $R = \text{sort}(S)$;

benchmark for instance-level object retrieval and recognition. This dataset covers 200 objects and 23070 labelled images with over 100 images per class. The natural backgrounds and occlusions in this dataset capture the problems encountered in real world. Some examples of INSTRE are shown in Fig. 1. Following the standard evaluation protocols, we report mean Average Precision (mAP) for these datasets.

4.2. Implementation details

4.2.1. Local descriptors

We use the Hessian–Affine detector [42] to detect keypoints for INSTRE. Since Oxford and Paris datasets are correctly oriented (sky-is-up), we use the modified Hessian–Affine detector [43], which includes the gravity vector assumption to fix rotation uncertainty. Our experiments use the default detector threshold values. We adopt the extension of SIFT descriptor. The descriptor is first L1-normalized, then followed by component-wise square rooting. The RootSIFT has proven to yield superior performance at no cost [16].

4.2.2. Visual vocabulary

We use approximate k-means to train our visual vocabularies [44]. Vocabulary used for Oxford is trained on Paris, and vice versa. For INSTRE, the vocabulary is trained on an independent dataset. We use a vocabulary of 65k visual words.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Oxford5k</th>
<th>Paris6k</th>
<th>INSTRE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Function</td>
<td>$\phi_1$</td>
<td>$\phi_2$</td>
<td>$\phi_3$</td>
</tr>
<tr>
<td>Conv5</td>
<td>77.8</td>
<td>78.2</td>
<td>80.3</td>
</tr>
<tr>
<td>FC6</td>
<td>79.5</td>
<td>80.2</td>
<td>81.8</td>
</tr>
<tr>
<td>FC7</td>
<td>78.6</td>
<td>79.2</td>
<td>81.0</td>
</tr>
</tbody>
</table>

4.2.3. SIFT Hamming embedding

The 64-bit SIFT Hamming embedding is used. The Hamming distance threshold $\kappa$ in Eq. 3 is set 22, and the weighting parameter $\sigma$ is set to 16.

4.2.4. Burstiness and multiple assignment

Some complementary techniques, such as burstiness (Burst) [45] and multiple assignment (MA) [4,41], are often integrated into BoW based systems. The burstiness phenomenon is that a given visual word occurs many times in an image, which may dominate the voting score. An intra-image burstiness normalization is proposed to deal with this phenomenon. Multiple assignment aims to improve the recall, which is often applied on the query side only. A query SIFT is quantized to several nearest visual words.

4.3. Parameter analysis

In our method, there are two major parameters: $\theta$ in the verification function $\phi_4$, and $\beta$ in the function $\phi_3$. We tune $\theta$ and $\beta$ under different conditions and present the impact on performance in Figs. 6 and 7. It can be seen that our verification scheme with CNN evidences gives a significant improvement compared to the SIFT matching baseline. In Fig. 6, all curves have similar shapes and rise to the peak at $\theta = 38$. In Fig. 7, the best performance is achieved at $\beta = 36$ for both Paris6k and INSTRE. For Oxford5k, the mAP scores are almost equivalent for values from $\beta = 34$ to $\beta = 36$. Based on these results, we set $\theta = 38$ and $\beta = 36$ for the following experiments.

4.4. Evaluation

The comparison for pre-trained CNN features extracted from different layers is shown in Table 1. In our method, the CNN feature from FC6 has the best performance. FC7 works better than Conv5 and falls a little behind FC6. Unless stated otherwise, CNN features of FC6 are used.

We evaluate the two different verification functions and present the results in Table 1. The function $\phi_4$ with the product rule is more effective than the function $\phi_3$ for Paris6k and INSTRE.
Fig. 6. Impact of $\theta$ in the verification function $\phi_\times$. In the first row, SIFT matching function $f_{bow}$ is used. In the second row, SIFT matching function $f_{he}$ is used. The performance of MA is also included. We tune $\theta$ on the three datasets with CNN features extracted from Conv5, FC6 and FC7, respectively. The object region number $M$ is set to 32.

Fig. 7. Impact of $\beta$ in the verification function $\phi_\times$. In the first row, SIFT matching function $f_{bow}$ is used. In the second row, SIFT matching function $f_{he}$ is used. The performance of MA is also included. We tune $\beta$ on the three datasets with CNN features extracted from Conv5, FC6 and FC7, respectively. The object region number $M$ is set to 32.

However, $\phi_\times$ works better than $\phi_\times$ for Oxford5k. This is due to the large intraclass variance of Oxford5k in the CNN feature space (see Fig. 4(a)). The right tail of the true match curve in Oxford5k seems to have larger $h_c$ than the right tails in Paris6k and INSTRE. If the Hamming distance $h_c$ of a true match SIFT pair is large, it perhaps weakens the high matching score obtained by function $f(\cdot)$ under the product rule. So $\phi_\times$ is a good choice for the Oxford dataset. In the rest of our experiments, we adopt $\phi_\times$ for Paris6k and INSTRE, and employ $\phi_\times$ for Oxford5k.

Table 2 summarizes the respective contributions of the different elements of our search system. The SIFT verification function $\phi$ with CNN evidences significantly improves the retrieval performance. The complementary techniques of Burst and MA also help to boost the accuracy. That is why these techniques are widely used. Note that no matter whether these complementary techniques are used or not, our approach brings large improvement for all the three datasets. The results in Table 2 demonstrate the effectiveness of our method.
The impact of $M$, i.e., the number of object regions detected by EdgeBoxes is shown in Fig. 8. The mAP score rises with the increase of the object regions, since more regions could provide more possible locations of the object of interest, which helps to obtain accurate evidence to verify the SIFT match. $M = 1$ corresponds to the case that the whole image is treated as the object region (see Section 3.2). In this case, the verification with CNN evidence even lowers the performance on the INSTRE dataset. This is because that the object of interest often occupies a small portion of an database image in this dataset. Adopting the global CNN feature may introduce background noise, which may give a wrong verification.

In order to demonstrate the compatibility of our approach, we test the retrieval performance with different kinds of object regions. We make a comparison among three object proposal methods: EdgeBoxes, Selective Search [35] and BING [36]. Both EdgeBoxes and Selective Search are unsupervised methods. We use the Selective Search’s “fast mode” to extract object proposals. BING employs two stages cascaded SVM to measure the objectness. We train BING on the PASCAL VOC dataset [46]. The average detection time of EdgeBoxes, Selective Search and BING is 0.2s, 8s and 0.006s, respectively. The comparison is shown in Fig. 9. No matter which object detection method is used, our approach brings consistent improvement over the baseline. Though BING is very efficient, its performance is a little worse than EdgeBoxes and Selective Search. Based on the results, EdgeBoxes is a good choice for our method.

### 4.5. Fine-tuning CNN

We wonder how the performance of our method can be improved if we fine-tune the CNN models on the target dataset. Each kind of query object is considered as an independent class. In the case of Oxford and Paris, there are 11 different class labels. For INSTRE, 200 class labels are used. We modify the last layer in the network so that the number of output class probabilities is adapted to the target dataset. For each object class, we adopt EdgeBoxes to detect multiple object proposals from the relevant images according to the ground truth. Then we manually pick out the proposals which enclose or partially overlap the object of interest and use them as the training data. All the layers except the last layer are initialized with the parameters pre-trained on ImageNet, while the output layer is initialized from a zero-mean Gaussian distribution with standard deviation 0.01.

The performance of our method with fine-tuned CNN features are shown in Table 3. By fine-tuning the CNN for the target dataset, the performance can be further improved. The fine-tuned networks could provide more distinctive evidence to prevent false SIFT matches, which helps to boost the object retrieval accuracy. It is interesting that FC7 works best after fine-tuning, which is different from the pre-trained CNN features. So we will adopt FC7 features when the CNN model is fine-tuned.
Table 6
Comparison with the state-of-the-art methods. The methods are divided into three types: SIFT based methods (only SIFT is used), CNN based methods (only CNN feature is used) and SIFT-CNN methods (the fusion of SIFT and CNN). * denotes the result obtained by adopting 128-bit SIFT binary signature. ** denotes the result obtained by fine-tuning the CNN models. * indicates the results by our own implementation.

<table>
<thead>
<tr>
<th>Type</th>
<th>Methods</th>
<th>Oxford5k</th>
<th>Oxford105k</th>
<th>Parisi6k</th>
<th>INSTRE</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIFT-based</td>
<td>Burst[45]</td>
<td>64.7</td>
<td>49.0</td>
<td>64.0*</td>
<td>60.5*</td>
</tr>
<tr>
<td></td>
<td>ThreeThings[16]</td>
<td>80.9</td>
<td>72.2</td>
<td>76.5</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>SCSM[22]</td>
<td>75.2</td>
<td>72.9</td>
<td>74.1</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Q.adal[77]</td>
<td>78.0</td>
<td>72.8</td>
<td>73.6</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>ASMC[18]*</td>
<td>80.4</td>
<td>75.0</td>
<td>77.0</td>
<td>68.5*</td>
</tr>
<tr>
<td></td>
<td>EarlyBurst[47]*</td>
<td>81.3</td>
<td>-</td>
<td>77.5</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Off-the-Shelf[11]</td>
<td>68.0</td>
<td>-</td>
<td>79.5</td>
<td>58.6*</td>
</tr>
<tr>
<td></td>
<td>DeepIndex[48]</td>
<td>66.3</td>
<td>-</td>
<td>81.2</td>
<td>52.5*</td>
</tr>
<tr>
<td>CNN-based</td>
<td>MAC[32]</td>
<td>66.8</td>
<td>-</td>
<td>83.0</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>FasterRCNN[49]**</td>
<td>71.0</td>
<td>-</td>
<td>79.8</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>BLCT[50]</td>
<td>73.9</td>
<td>59.3</td>
<td>81.9</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>NetVLAD[51]**</td>
<td>70.8</td>
<td>-</td>
<td>78.3</td>
<td>-</td>
</tr>
<tr>
<td>SIFT-CNN</td>
<td>DeepEmbedding[33]*</td>
<td>79.9</td>
<td>-</td>
<td>78.6*</td>
<td>59.2*</td>
</tr>
<tr>
<td></td>
<td>Ours</td>
<td>80.2</td>
<td>74.5</td>
<td>81.8</td>
<td>74.8</td>
</tr>
<tr>
<td></td>
<td>Ours*</td>
<td>81.6</td>
<td>75.9</td>
<td>83.6</td>
<td>77.9</td>
</tr>
<tr>
<td></td>
<td>Ours**</td>
<td>81.4</td>
<td>76.1</td>
<td>83.6</td>
<td>78.8</td>
</tr>
</tbody>
</table>

4.6. Complexity

We compare the complexity of the proposed method with BoW and HE, as shown in Tables 4 and 5. For each local feature, 4 bytes are allocated to store the image ID. In HE, 8 bytes are needed to store the 64-bit binary SIFT signature. Since we detect 32 object regions for each database image, another 4 bytes are required to store the region indicator. Each object region has a 128-bit CNN signature, so 512 bytes are consumed to store the binary features. Assume that every image has 500 SIFT descriptors. On the 1M dataset, our method consumes a memory of 7.93 GB, which can be easily stored in the main memory of a commodity server.

The average query time for the 1M dataset is presented in Table 5. The SIFT extraction takes an average of 0.7s. Detecting object regions with EdgeBoxes costs 0.2 s. Extracting CNN features is efficient, consuming only 0.05 s. Comparing to HE, our method needs extra time to find appropriate CNN evidences to verify the SIFT matches. Instead of computing Hamming distance for every pre-matched SIFT pair, we adopt the look up table to speed up search process (see Algorithm 1). All the Hamming distances of CNN signatures between the query object and a database image are computed and stored in memory when this database image is visited at the first time. About 40% database images may be visited for each query, so at most 32 × 0.4 × N Hamming distance computations are required, where N is the database image number. Each keypoint is located in 4 object regions on average, so 4 comparison operations are also needed to find the minimum Hamming distance for each pre-matched SIFT pair. The final search time on 1M dataset is 2.12 s.

4.7. Comparison with the state-of-the-art

We compare our results with the state-of-the-art methods. These image retrieval methods are divided into three types: SIFT based methods (only SIFT is used), CNN based methods (only CNN feature is used) and SIFT-CNN methods (the fusion of SIFT and CNN). Post-processing algorithms, such as query expansion and re-ranking are not included.

The comparison is shown in Table 6. We achieve the best performance on all the four datasets. Our approach yields a retrieval accuracy of 81.6%, 76.1%, 83.6% and 78.8% on Oxford5k, Oxford105k, Parisi6k and INSTRE, respectively. Even without fine-tuning CNN, our results are still very competitive.

It should be noticed that, CNN-based retrieval methods have quickly demonstrated their strengths in object retrieval. As shown in Table 6, CNN-based methods outperform all the SIFT-based methods on Parisi6k, which demonstrates the effectiveness of semantic-aware CNN features. However, SIFT may be superior to CNN if the images are easily confused in the semantic space. That is why SIFT still performs better than CNN on the Oxford dataset. SIFT describes low-level details, while CNN describes general semantics. It seems that none of them is systematically better than the other. So it is a good choice to adopt both features to achieve a better result.

5. Conclusions

This paper proposes to employ CNN evidences to improve SIFT matching accuracy, which plays a critical role in improving the object retrieval performance. We decompose the image into several object regions and extract CNN features from them. A query-adaptive method is proposed to select appropriate evidence from the regional CNN features, which weakens the interference of background noise. Two different verification functions are introduced to verify the SIFT matches. Extensive experiments demonstrate the effectiveness of our method with acceptable memory usage and time cost. We also explore the suitability of fine-tuning CNN models for our method. The experiments show that fine-tuning CNN models on the target dataset could provide more effective CNN features.

Acknowledgment

This work has been supported by the National Science and Technology Supporting Program of China under Grant No. 2015BAH49F01 and the Key Technology R&D Program of Beijing under Grant No.D161100005216001.

References


Guixuan Zhang received the B.S. degree in Measurement and Control Technology from University of Science and Technology, Beijing, China, in 2012. He is currently pursuing the Ph.D. degree at the Digital Content Technology and Media Service Research Center, Institute of Automation, Chinese Academy of Sciences, Beijing, China. His research interests include image retrieval, machine learning and pattern recognition.

Zhi Zeng received the B.S. and M.S. degree in Computer Science in 2003 and 2006, respectively, both from Chongqing University, China, and the Ph.D. degree in Pattern Recognition from the Institute of Automation, Chinese Academy of Sciences, in 2009. He is currently a Senior Engineer in the Institute of Automation, Chinese Academy of Sciences. His research interests include information retrieval, machine learning, multimedia, and digital rights management.

Shuwu Zhang got his Ph. D from Chinese Academy of Sciences in 1997. Currently, he is a professor of Institute of Automation, Chinese Academy of Sciences. His research interests are focused on digital content analysis, digital right management, and web-based cultural content service technologies.
Yuan Zhang received the B.S. degree in Automation from Beijing University of Posts and Telecommunications in 2009 and the Ph.D. degree in Pattern Recognition from the Institute of Automation, Chinese Academy of Sciences in 2014. He is currently an Algorithm Engineer in Alibaba Group. His research interests include image retrieval, object detection, machine learning, and pattern recognition.

Wanchun Wu received the B.S. and M.S. degree in Computer Science in 2003 and 2006, respectively, both from Chongqing University, China. He is currently an engineer of Children's Hospital of Chongqing Medical University, Chongqing, China. His research interests include relational database, medical information and computer vision.