Efficient and Interactive Spatial-Semantic Image Retrieval

Ryosuke Furuta(✉), Naoto Inoue, and Toshihiko Yamasaki

Department of Information and Communication Engineering,
The University of Tokyo, Tokyo, Japan
{furuta,inoue,yamasaki}@ay-lab.org

Abstract. This paper proposes an efficient image retrieval system. When users wish to retrieve images with semantic and spatial constraints (*e.g.*, a horse is located at the center of the image, and a person is riding on the horse), it is difficult for conventional text-based retrieval systems to retrieve such images exactly. In contrast, the proposed system can consider both semantic and spatial information, because it is based on semantic segmentation using fully convolutional networks (FCN). The proposed system can accept three types of images as queries: a segmentation map sketched by the user, a natural image, or a combination of the two. The distance between the query and each image in the database is calculated based on the output probability maps from the FCN. In order to make the system efficient in terms of both the computation time and memory usage, we employ the product quantization technique (PQ). The experimental results show that the PQ is compatible with the FCN-based image retrieval system, and that the quantization process results in little information loss. It is also shown that our method outperforms a conventional text-based search system.

1 Introduction

With the increase in number of images that have been captured and uploaded to the Internet, the importance of image retrieval system has been increasing. The most widely employed image retrieval systems are based on text. Namely, the query consists of text and relevant images are retrieved. Conventional text-based retrieval systems require tags or captions to be attached to each image in the database. To tackle this problem, many retrieval methods based on machine learning techniques have been proposed, such as caption generation [13,16,26] or the mapping of features into a common latent space between text and images [8,17]. However, such methods still cannot deal with spatial constraints such as object positions.

To this end, Xu et al. [32] proposed an image retrieval system based on concept maps. A query consists of a canvas, where the textual information is distributed to represent spatial constraints. Although that method can deal with object names and positions simultaneously, it cannot consider object shapes and scales. Recently, a novel image retrieval method has been proposed in which a

K. Schoeffmann et al. (Eds.): MMM 2018, Part I, LNCS 10704, pp. 190–202, 2018.
https://doi.org/10.1007/978-3-319-73603-7_16

query is a canvas with a set of bounding boxes representing semantic and spatial constraints [22]. Although this method can deal with object scales and locations, it still cannot treat object shapes. Moreover, it also cannot take background information into consideration.

In this paper, we propose an efficient image retrieval system based on semantic segmentation. The proposed system can accept three types of queries: a natural image, a segmentation map drawn by the user, and a combination of the two. We employ a fully convolutional network (FCN) [21], which is composed only of convolution and pooling layers. By retrieving images based on the probability maps from the FCN, our system can deal with object scales, shapes, positions, and background information. As shown in Fig. 3, our system also enables users to search for images interactively, by selecting one of the retrieved images as the new query, and adding a partial segmentation map to the new query. To the best of our knowledge, this is the first work to propose an interactive image retrieval system based on semantic segmentation.

In order to make the proposed system efficient in terms of both the search speed and memory usage, we employ the product quantization (PQ) technique [11], which is compatible with our system. Using subjective evaluations in Sect. 4.2, we show that the proposed system provides a superior performance compared with a conventional text-based image retrieval system. Furthermore, in Sect. 4.5 we demonstrate that the PQ makes our system orders of magnitude faster while maintaining the retrieval quality, by quantizing each probability map independently.

2 Related Work

2.1 Semantic Image Retrieval

As pointed out in [22], the majority of early methods for spatial-semantic retrieval extracted low-level features from exemplars [3,19,32]. Inspired by the recent success of convolutional neural networks (CNNs) for image classification, some studies have employed CNNs to learn and extract effective features for image retrieval, where queries consist of images [7] or sketches [20,28–30,33]. The objective of such a method is to retrieve images that have similar appearances, even in cross-modal domains, which differs from our approach.

To capture the context or object topology, some methods have incorporated graphs into the image retrieval, which represent attributes and the relationships between them [14,27]. However, these methods do not enable users to search for images interactively, because users cannot create the graph as a query directly. In contrast, in our method users can draw a segmentation map as a query on the canvas, or even on the natural image.

The most relevant work to ours is in [9], where a query consists of a single source object and a target object sketched by the user. Similar to our method, that one is based on semantic segmentation. However, their objective is to retrieve images considering the interaction between two objects by extracting RAID (relation-augmented image descriptor) features. In contrast, our method uses the probability maps directly for image retrieval.

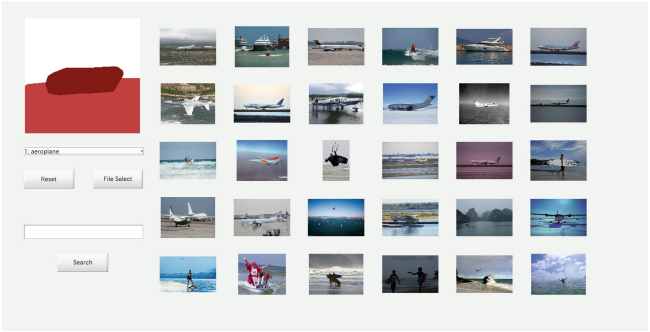


Fig. 1. Interface of the proposed system.

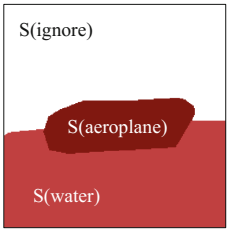


Fig. 2. Example of segmentation map.

2.2 Quantization for Efficient Retrieval

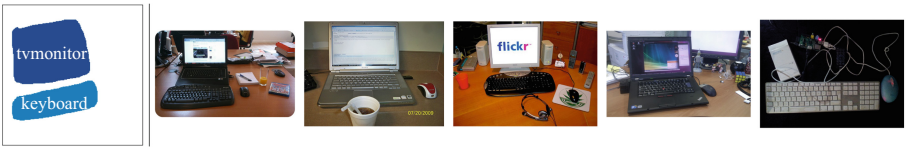
Many techniques have been proposed for efficient retrieval, such as binary coding and hashing (summarized in [31]). Product quantization (PQ) [11] is one of the most popular techniques among these, because it is efficient in terms of both computational cost and memory usage. By partitioning the vectors in the database into subvectors and quantizing them using k-means clustering, the approximate distances between the query vector and those in the database can be calculated efficiently via lookup tables. Although variants of PQ have also been proposed and shown superior performances [1,6,15,23,25], in this paper we use the original PQ [11], because of its simplicity and compatibility with our system. Recently, Hinami and Satoh [10] proposed an efficient image retrieval system using an adaptive quantization technique. However, their method is tailored for R-CNN-based object detection, and cannot be applied to semantic segmentation.

3 Proposed Method

3.1 System Overview

Figure 1 presents the interface of the proposed system. The top-left shows the canvas that is treated as the query. The retrieved images are shown in the right area. The proposed system can accept three types of queries: (i) a segmentation map drawn by a user, (ii) a natural image, and (iii) a combination of the two.

(i) Users can easily create a segmentation map with predefined C class labels by drawing it using a mouse input. For example, when the user wants to retrieve images in which a horse is located at the center, he/she chooses the *horse* label and roughly draws its shape at the corresponding location as he/she likes.



(a) Users can draw a segmentation map as a query.

(b) Users can choose an image from the retrieved images as the next query.

(c) Users can add a partial segmentation map to the query image.

Fig. 3. Interactive image retrieval. Query and top five retrieved images are shown.

(ii) Users can also use a natural image as a query. In this case, the proposed system retrieves images that contain objects and backgrounds whose shapes and locations are similar to those of the query image. In addition, the user can choose a query image from the retrieved images shown in the right area.
(iii) In addition, users can draw a partial segmentation map on a natural image. In this case, the proposed system retrieves images by considering both the objects and backgrounds in the query image and those drawn by the user.

3.2 Semantic-Spatial Image Retrieval

We use a fully convolutional network (FCN) trained for C class semantic segmentation to extract spatial-semantic information. The FCN takes an image whose size is $n' \times n'$ as an input, and outputs C probability maps of size $n \times n$. In general, n is smaller than n', because the resolutions of the intermediate feature maps are decreased by pooling layers (the scale difference n'/n depends on which FCN we employ). In this paper, we use DeepLab-v2 [4], which has demonstrated a state-of-the-art performance, and $n' = 8n$ in this case.

Offline Pre-process. Let I^i denote the i-th reference image ($i = 1, \cdots, N$) in the database. We input I^i into the FCN and obtain the C probability maps. The j-th location ($j = 1, \cdots, n^2$) of the c-th probability map ($c = 1, \cdots, C$) has the probability $p_c^i(j)$ that the c-th class label is assigned to that location. The probability is normalized by the final softmax layer in the FCN, and satisfies the following:

$$\forall i, j, \quad p_c^i(j) \in [0, 1], \quad \sum_{c=1}^{C} p_c^i(j) = 1. \tag{1}$$

We reshape this c-th probability map as a vertical vector $\boldsymbol{p}_c^i = [p_c^i(1), \cdots, p_c^i(n^2)]^\top$. Furthermore, we vectorize the C probability maps as $\boldsymbol{p}^i = [\boldsymbol{p}_1^{i\top}, \cdots, \boldsymbol{p}_C^{i\top}]^\top$. As an offline pre-process, we obtain the probability vectors $\boldsymbol{p}^i$ for all reference images I^i $(i = 1, \cdots, N)$ in the database.

When the Query is a Natural Image. We first consider the case that the query consists of a natural image. Given a query image, we input the query image into the FCN and obtain the probability vector $\boldsymbol{p}^{query} \in [0, 1]^{n^2 C}$ online. We define the distance between the query image I^{query} and the reference image I^i as the L2 distance between $\boldsymbol{p}^{query}$ and $\boldsymbol{p}^i$:

$$dist(I^{query}, I^i) = ||\boldsymbol{p}^{query} - \boldsymbol{p}^i||^2 = \sum_{c=1}^{C} ||\boldsymbol{p}_c^{query} - \boldsymbol{p}_c^i||^2. \tag{2}$$

By calculating the rankings of the reference images based on the above distance, we can retrieve the images that contain objects whose shapes and locations are similar to those of the query image. In addition, we can consider background information if scene labels such as *sky*, *building*, and *grass* are included in the C class labels.

When the Query is a Segmentation Map Drawn by a User. Next, we consider the case that the query consists of a segmentation map drawn by a user. Let $\boldsymbol{y}$ denote the query segmentation map whose size is $n \times n$, and let $y(j) \in \{0, \cdots, C\}$ be the label assigned to the j-th location. Here, $y(j) = 0$ denotes the *ignore* label, which is assigned when the user does not specify any label at the j-th location. We define the region where the c-th class label is assigned as $S(c)$:

$$S(c) = \{j \mid y(j) = c\}. \tag{3}$$

Figure 2 presents an example of $S(c)$.

Given a query image, we construct a vector $\boldsymbol{q} \in \{0, 1\}^{n^2 C}$ as follows:

$$\boldsymbol{q} = [\boldsymbol{q}_1^\top, \cdots, \boldsymbol{q}_C^\top]^\top, \quad \boldsymbol{q}_c = [q_c(1), \cdots, q_c(n^2)]^\top, \quad q_c(j) = \begin{cases} 1 & if \quad y(j) = c \\ 0 & otherwise. \end{cases} \tag{4}$$

$\boldsymbol{q}_c$ can be interpreted as the binary probability map for the c-th class, which is calculated from the segmentation map drawn by the user. In $\boldsymbol{q}_c$, a location at which the user has assigned the c-th label has the value 1, and all other locations have the value 0. Using this vector $\boldsymbol{q}$, we define the distance between the query and a reference image as following:

$$dist(\boldsymbol{y}, I^i) = \sum_{c=1}^{C} \mathbf{1}[S(c) \neq \emptyset] ||\boldsymbol{q}_c - \boldsymbol{p}_c^i||^2, \tag{5}$$

where $\mathbf{1}[\cdot]$ is the indicator function, which is 1 if the statement in the blanket is true and 0 otherwise. This indicator function is introduced in order to only consider the labels that the user has specified. The rankings of the reference images are obtained by using the above distance.

When the Query is the Combination of a Natural Image and a Partial Segmentation Map Drawn by a User. In the proposed system, the user can search for images interactively by adding a partial segmentation map $\boldsymbol{y}$ to a natural image I^{query}. In this case, we define a query vector $\boldsymbol{q} = [\boldsymbol{q}_1^\top, \cdots, \boldsymbol{q}_C^\top]^\top$ as follows:

$$\boldsymbol{q}_c = [q_c(1), \cdots, q_c(n^2)]^\top, \quad q_c(j) = \begin{cases} 1 & if \quad y(j) = c \\ 0 & if \quad y(j) \neq c \wedge y(j) \neq 0 \\ p_c^{query}(j) & if \quad y(j) = 0. \end{cases} \tag{6}$$

In $\boldsymbol{q}_c$, locations at which the user has assigned the c-th label have the value 1. The locations where other labels are assigned have the value 0, and all other locations have the value $p_c^{query}(j)$. Similarly to the above cases, we define the distances between the query and the reference images as follows:

$$dist(\boldsymbol{y}, I^i) = ||\boldsymbol{q} - \boldsymbol{p}^i||^2 = \sum_{c=1}^{C} ||\boldsymbol{q}_c - \boldsymbol{p}_c^i||^2. \tag{7}$$

By calculating the rankings using Eq. (7), we can consider both the objects in the query image and those drawn by the user.

3.3 Product Quantization for Efficient Retrieval

In Sect. 3.2, we introduced image retrieval based on semantic segmentation, which considers spatial-semantic information. However, storing the long vectors $\boldsymbol{p}^i \in [0,1]^{n^2C}$ is memory consuming, because of the dimensions of $n^2C = 64^2 \times 60 = 245,760$ in our setting in Sect. 4. In addition, the naive computation of Eqs. (2), (5) or (7) is slow. Therefore, we employ PQ [11] in order to make the system efficient in terms of both the computational time and memory usage. We show that the approximate values of Eqs. (2), (5) and (7) can be efficiently computed by applying PQ.

Offline Pre-process. We partition the vector $\boldsymbol{p}^i$ into M distinct subvectors $\boldsymbol{u}_1^i, \cdots, \boldsymbol{u}_M^i$, and quantize them independently, *i.e.*, $f_1(\boldsymbol{u}_1^i), \cdots, f_M(\boldsymbol{u}_M^i)$. Following the original PQ technique [11], we learn the quantizer f_m ($m = 1, \cdots, M$) using k-means. We perform k-means on the set of vectors $\{\boldsymbol{u}_m^i \mid i = 1, \cdots, N\}$ and obtain K centroids $\mathcal{A}_m = \{\boldsymbol{a}_{m,k} \mid k = 1, \cdots, K\}$. The quantizer f_m is the mapping function to the nearest centroids:

$$f_m(\boldsymbol{u}_m^i) = \underset{\boldsymbol{a}_{m,k} \in \mathcal{A}_m}{\arg\min} ||\boldsymbol{u}_m^i - \boldsymbol{a}_{m,k}||^2. \tag{8}$$

When we set $M = C$, this quantization process corresponds to partitioning $\boldsymbol{p}^i$ into each probability map $\boldsymbol{p}_c^i (= \boldsymbol{u}_m^i)$ and quantizing these. If the probability maps are independent of each other, this is the optimal setting of M, because this partitioning results in no information loss. In our experiments, we assume that they are almost independent, and set $M = C$. We show that PQ with this setting does not decrease the retrieval quality in Sect. 4.5.

Online Search. By using PQ, we can efficiently compute the approximate value of Eq. (2) as follows:

$$dist(I^{query}, I^i) = ||\boldsymbol{p}^{query} - \boldsymbol{p}^i||^2 = \sum_{m=1}^{M} ||\boldsymbol{u}_m^{query} - \boldsymbol{u}_m^i||^2 \tag{9}$$

$$\approx \sum_{m=1}^{M} ||\boldsymbol{u}_m^{query} - f_m(\boldsymbol{u}_m^i)||^2. \tag{10}$$

Because $f_m(\boldsymbol{u}_m^i)$ is the mapping function to the nearest centroid, as shown in Eq. (8), we can efficiently compute Eq. (10) for all reference images by constructing a lookup table of the distances between the query subvector $\boldsymbol{u}_m^{query}$ and each of the centroids. Similarly, Eq. (7) can be also computed efficiently using the lookup table.

There is a further benefit of setting $M = C$. Namely, we can efficiently compute Eq. (5) by approximating as follows:

$$dist(\boldsymbol{y}, I^i) = \sum_{c=1}^{C} \mathbf{1}[S(c) \neq \emptyset] ||\boldsymbol{q}_c - \boldsymbol{p}_c^i||^2 \tag{11}$$

$$\approx \sum_{c=1}^{C} \mathbf{1}[S(c) \neq \emptyset] ||\boldsymbol{q}_c - f_c(\boldsymbol{p}_c^i)||^2. \tag{12}$$

Similarly, we can construct a lookup table of the distances between the query subvector $\boldsymbol{q}_c$ and each of the centroids. To exploit this approximation, we set $M = C$ in Sect. 4.

4 Experimental Results

4.1 Implementation Details

As the FCN, we used DeepLab-v2 [4] implemented on the Caffe library [12], which is publicly available. We trained this network on the trainval set of the PASCAL-Context Dataset [24], which contains 5,105 images with groundtruth pixel-level labels. This is a dataset for $60 (= C)$ class semantic segmentation, where a variety of classes are included, such as *car*, *building*, *sky*, and *road*. We denote the set of 60 class names as $\mathcal{C}$. Similarly to [4], we employed poly-learning, where the learning rate started at 2.5×10^{-4} and was multiplied by

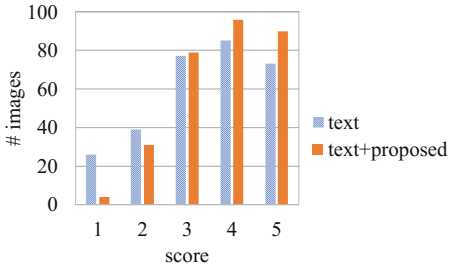


Fig. 4. Histogram of scores.

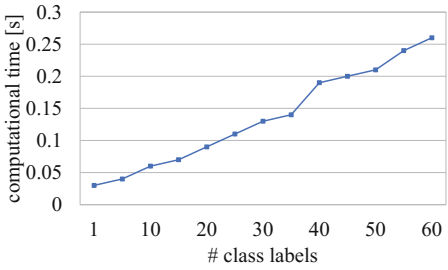


Fig. 5. Computational time versus number of class labels.

$(1 - (\frac{iter}{max\_iter})^{power})$ at each iteration. We set the max_iter to 20,000, $power$ to 0.9, momentum to 0.9, and weight decay to 5.0×10^{-4}. We used the pixel-wise softmax cross-entropy between the groundtruth label and the predicted score. The input size is $n' \times n' = 512 \times 512$, and the output size is $n \times n = 64 \times 64$. We implemented the user interface and PQ on MATLAB, and set $K = 256$.

4.2 Subjective Evaluation

We conducted subjective evaluation tests to verify the efficacy of the proposed system. As the database, we used the MSCOCO2014 training set [18], which contains 82,783 images. Each image has five caption annotations written by Amazon Mechanical Turk workers.

We compared the following two methods.

Text-based retrieval. The user can input a set of words as a query. The rankings of images are simply calculated based on the number of words in the captions that are the same as the query words.

Text+proposed system. The user can input both a set of words and the three types of images described in Sect. 3.1 as a query. Given the input, we first obtain the rankings in a similar manner to the text-based retrieval method above. Subsequently, we sort the images that have the same rank based on the distance in Eq. (10) or (12). When the user does not input any images on the canvas, this system is equivalent to the text-based retrieval method above. In contrast, when the user does not input query words, the ranking is calculated simply based on the distance in Eq. (10) or (12).

The procedure of the test is as follows.

1. We used the MSCOCO2014 validation set, which has 202,654 captions, to construct a caption set $\mathcal{T}$. We picked up the captions that contain three or more class names in $\mathcal{C}$, and there consequently remained 2,896 captions.
2. We randomly chose a caption from $\mathcal{T}$, and asked subjects to imagine an image that the caption describes.

3. We asked subjects to choose ten images that they think are similar to the imagined image using the text-based and text+proposed systems, respectively. Subjects could make queries and search for images any number of times.
4. We showed the twenty chosen images in random order, and asked subjects to assign relevance scores from 1 to 5 to the each image. A score of 5 indicates the highest relevance, and 1 the lowest.
5. Steps 2–4 were repeated three times.

The number of subjects was ten, and their ages were 21–26. Figure 4 presents the histograms of the scores. We observe that the number of images scored as 1 using the text+proposed system is significantly lower than that for the text-based system. Accordingly, the number of images scored as 4 and 5 increased when using the text+proposed system. This is reasonable, because the text-based system cannot deal with semantic-spatial information, such as the shapes and locations of objects that subjects imagine. The average score of the all 300 images for the text-based system is 3.5, and that of the text+proposed system is 3.8. There is a significant difference ($\rho < 0.01$) between these according to the Student's t-test.

4.3 Computational Time Analysis

Using the 82,783 images in the MSCOCO train set, we analyze the computational time required to calculate the ranking and sorting based on Eq. (10) or (12) with PQ. Because we set $M = C$, the size of lookup table for computing Eq. (10) is $C \times K$, where M is the quantization level, C is the number of classes, and K is the number of centroids. The average computational time for Eq. (10) and sorting for all 82,783 images was about 0.3 s on a machine with an Intel Core i7-6600U and 12GB RAM, which is sufficiently fast for real application.

When the query consists of a segmentation map drawn by a user, the size of the lookup table required for Eq. (12) depends on the number of classes the user specifies (*i.e.*, $|\mathcal{C}'|$ where $\mathcal{C}' = \{c \mid S(c) \neq \emptyset\}$). Figure 5 shows the computational time versus $|\mathcal{C}'|$ for all 82,783 images. The computational time increases linearly as $|\mathcal{C}'|$ becomes large. However, even when the user specifies all classes (*i.e.*, $|\mathcal{C}'| = 60$), the computational time is only 0.26 sec, which is sufficiently fast.

We could not measure the computational time without PQ, because we could not store 82,783 vectors of length $n^2C = 245,760$ on the RAM.

4.4 Qualitative Evaluation

Figure 6 shows some examples of the retrieved images with the proposed system on the MSCOCO2014 train set. We observe that the proposed system successfully performs retrieval considering the spatial-semantic contexts of the query images.

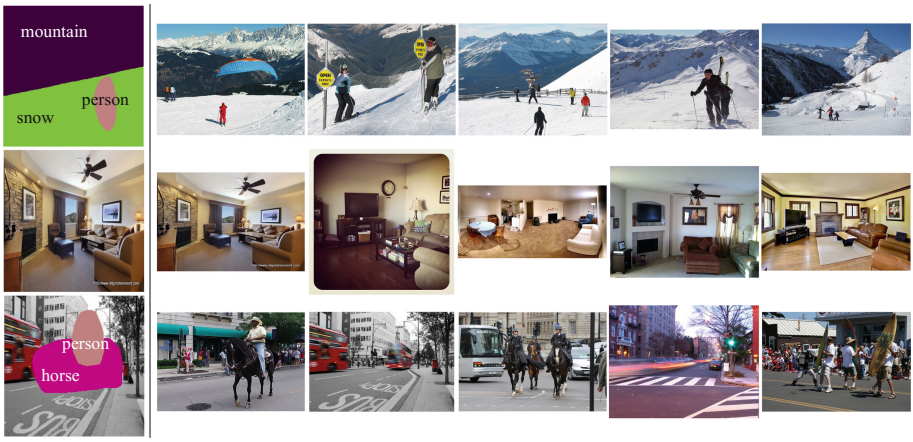
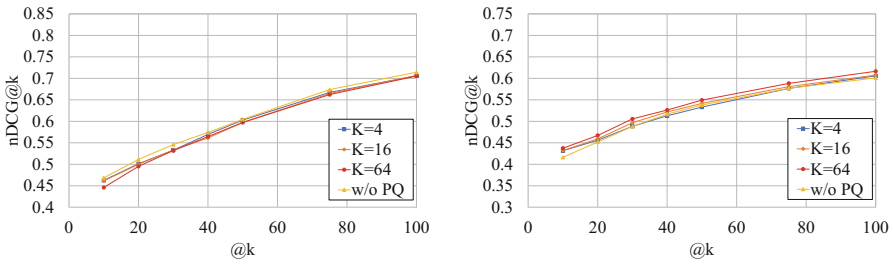


Fig. 6. Examples of retrieved images using the proposed method on the MSCOCO2014 train set. Query and top five images are shown.

4.5 Performance for Structured Retrieval

We used the rPascal and rImageNet datasets [27], which contain 1,895 and 3,354 images, respectively. The rPascal dataset contains 50 query images, and each query image has 180 reference images on average. The rImageNet dataset contains 50 query images, with 305 reference images per query on average. Both datasets contain relevance score annotations for each pair of query and reference images. Using these datasets, we evaluated the performance of the proposed method for structured retrieval. Similarly to [27], we used the normalized discounted cumulative gain (nDCG).

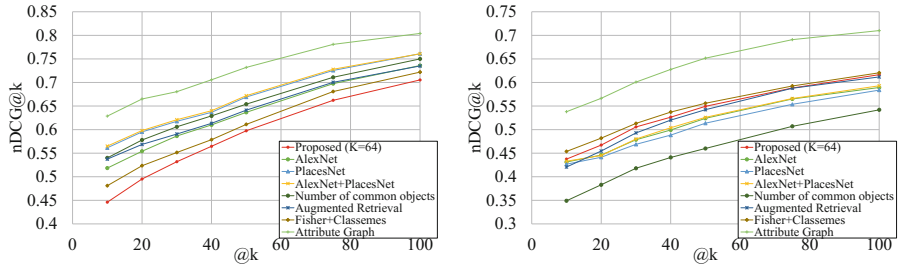
Figure 7a and b show the results of the proposed method with various values of K (the number of centroids) and without PQ. We observe that the search speed is enhanced without degrading the search accuracy.



(a) rPascal (b) rImageNet

Fig. 7. nDCG of the proposed method with various numbers of centroids (K).

Figure 8a and b present comparisons with other methods on rPascal and rImageNet, respectively. The proposed method is significantly inferior to Attribute-Graph [27], which is reasonable, because that method is tailored for structured retrieval and ours is not. Although the proposed method performs worse than other methods on rPascal, it shows a competitive performance with other methods except for Attribute-Graph [27] on rImageNet.



(a) rPascal (b) rImageNet

Fig. 8. Comparison of nDCG. Except for the proposed method, the plots are from [27]. *Augmented Retrieval*, *Fisher+Classemes*, and *Attribute Graph* indicate [2,5,27], respectively.

Table 1 presents a comparison of the computational time with and without PQ on the rPascal and rImageNet datasets. PQ makes the computation orders of magnitude faster, especially when K is small. When K is large, PQ is not as effective. However, this is because the numbers of reference images are extremely small (180 and 305, respectively). We believe that PQ will be more effective when the dataset size is large.

Table 1. Comparison of the computational times [s] with and without PQ.

	rPascal	rImageNet
$K = 4$	0.006	0.006
$K = 16$	0.016	0.016
$K = 64$	0.054	0.055
w/o PQ	0.079	0.135

5 Conclusion

In this paper, we have proposed an efficient and interactive image retrieval system using FCN and PQ. The FCN is used to treat spatial-semantic information, and PQ is applied for efficient computation and memory usage. The experimental

results showed that the proposed system is effective in reflecting the intentions of users. It was also shown that PQ is compatible with the proposed system, and makes it considerably faster while maintaining the retrieval quality. The limitation of the proposed method is that it cannot treat new classes that are not included in the training dataset of semantic segmentation.

Acknowledgement. This work was partially supported by the Grants-in-Aid for Scientific Research (no. 26700008 and 16J07267) from JSPS, JST-CREST (JPMJCR1686), and Microsoft IJARC core13.

We would like to thank Nikita Prabhu and R. Venkatesh Babu for providing their data.

References

1. Babenko, A., Lempitsky, V.: Additive quantization for extreme vector compression. In: CVPR (2014)
2. Cao, X., Wei, X., Guo, X., Han, Y., Tang, J.: Augmented image retrieval using multi-order object layout with attributes. In: ACMMM (2014)
3. Cao, Y., Wang, H., Wang, C., Li, Z., Zhang, L., Zhang, L.: Mindfinder: interactive sketch-based image search on millions of images. In: ACMMM (2010)
4. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: DeepLab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs. IEEE TPAMI (2017). http://ieeexplore.ieee.org/document/7913730
5. Douze, M., Ramisa, A., Schmid, C.: Combining attributes and fisher vectors for efficient image retrieval. In: CVPR (2011)
6. Ge, T., He, K., Ke, Q., Sun, J.: Optimized product quantization for approximate nearest neighbor search. In: CVPR (2013)
7. Gordo, A., Almazán, J., Revaud, J., Larlus, D.: Deep image retrieval: learning global representations for image search. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9910, pp. 241–257. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46466-4_15
8. Gordo, A., Larlus, D.: Beyond instance-level image retrieval: leveraging captions to learn a global visual representation for semantic retrieval. In: CVPR (2017)
9. Guerrero, P., Mitra, N.J., Wonka, P.: RAID: a relation-augmented image descriptor. ACM TOG **35**(4), 46:1–46:12 (2016)
10. Hinami, R., Satoh, S.: Large-scale R-CNN with classifier adaptive quantization. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9907, pp. 403–419. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46487-9_25
11. Jegou, H., Douze, M., Schmid, C.: Product quantization for nearest neighbor search. IEEE TPAMI **33**(1), 117–128 (2011)
12. Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S., Darrell, T.: Caffe: convolutional architecture for fast feature embedding. In: ACMMM (2014)
13. Johnson, J., Karpathy, A., Fei-Fei, L.: DenseCap: fully convolutional localization networks for dense captioning. In: CVPR (2016)
14. Johnson, J., Krishna, R., Stark, M., Li, L.J., Shamma, D., Bernstein, M., Fei-Fei, L.: Image retrieval using scene graphs. In: CVPR (2015)

15. Kalantidis, Y., Avrithis, Y.: Locally optimized product quantization for approximate nearest neighbor search. In: CVPR (2014)
16. Karpathy, A., Fei-Fei, L.: Deep visual-semantic alignments for generating image descriptions. In: CVPR (2015)
17. Kim, G., Moon, S., Sigal, L.: Ranking and retrieval of image sequences from multiple paragraph queries. In: CVPR (2015)
18. Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48
19. Liu, C., Wang, D., Liu, X., Wang, C., Zhang, L., Zhang, B.: Robust semantic sketch based specific image retrieval. In: ICME (2010)
20. Liu, L., Shen, F., Shen, Y., Liu, X., Shao, L.: Deep sketch hashing: fast free-hand sketch-based image retrieval. In: CVPR (2017)
21. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: CVPR (2015)
22. Long Mai, H.J., Lin, Z., Fang, C., Brandt, J., Liu, F.: Spatial-semantic image search by visual feature synthesis. In: CVPR (2017)
23. Matsui, Y., Yamasaki, T., Aizawa, K.: Pqtable: fast exact asymmetric distance neighbor search for product quantization using hash tables. In: ICCV (2015)
24. Mottaghi, R., Chen, X., Liu, X., Cho, N.G., Lee, S.W., Fidler, S., Urtasun, R., Yuille, A.: The role of context for object detection and semantic segmentation in the wild. In: CVPR (2014)
25. Norouzi, M., Fleet, D.J.: Cartesian k-means. In: CVPR (2013)
26. Ordonez, V., Han, X., Kuznetsova, P., Kulkarni, G., Mitchell, M., Yamaguchi, K., Stratos, K., Goyal, A., Dodge, J., Mensch, A., et al.: Large scale retrieval and generation of image descriptions. IJCV **119**(1), 46–59 (2016)
27. Prabhu, N., Venkatesh Babu, R.: Attribute-graph: a graph based approach to image ranking. In: ICCV (2015)
28. Qi, Y., Song, Y.Z., Zhang, H., Liu, J.: Sketch-based image retrieval via siamese convolutional neural network. In: ICIP (2016)
29. Sangkloy, P., Burnell, N., Ham, C., Hays, J.: The sketchy database: learning to retrieve badly drawn bunnies. ACM TOG **35**(4), 119 (2016)
30. Wang, F., Kang, L., Li, Y.: Sketch-based 3D shape retrieval using convolutional neural networks. In: CVPR (2015)
31. Wang, J., Zhang, T., Sebe, N., Shen, H.T., et al.: A survey on learning to hash. IEEE TPAMI (2017). http://ieeexplore.ieee.org/document/7915742/
32. Xu, H., Wang, J., Hua, X.S., Li, S.: Image search by concept map. In: SIGIR (2010)
33. Yu, Q., Liu, F., Song, Y.Z., Xiang, T., Hospedales, T.M., Loy, C.C.: Sketch me that shoe. In: CVPR (2016)