Multimedia Tools and Applications (2019) 78:18713–18733
https://doi.org/10.1007/s11042-018-7148-1

# Efficient and interactive spatial-semantic image retrieval

**Ryosuke Furuta**[1] · **Naoto Inoue**[1] · **Toshihiko Yamasaki**[1]

Received: 3 April 2018 / Revised: 25 October 2018 / Accepted: 28 December 2018 /
Published online: 1 February 2019
© The Author(s) 2019

**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 computational time and memory usage, we employ the product quantization (PQ) technique. 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.

**Keywords** Image retrieval · Fully convolutional networks · Semantic segmentation · Product quantization

## 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

✉ Ryosuke Furuta
furuta@hal.t.u-tokyo.ac.jp

Naoto Inoue
inoue@hal.t.u-tokyo.ac.jp

Toshihiko Yamasaki
yamasaki@hal.t.u-tokyo.ac.jp

[1] Department of Information and Communication Engineering, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku 113-8656, Tokyo, Japan

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 [21, 24, 38] or the mapping of features into a common latent space between text and images [9, 25]. However, such methods still cannot deal with spatial constraints such as object positions.

To this end, Xu et al. [45] 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 query is a canvas with a set of bounding boxes representing semantic and spatial constraints [32]. Hinami et al. [12] also proposed an image retrieval system based on bounding boxes and relative attributes between them. Although these methods can deal with object scales and locations, they still cannot treat object shapes. Moreover, it also cannot take background information into consideration.

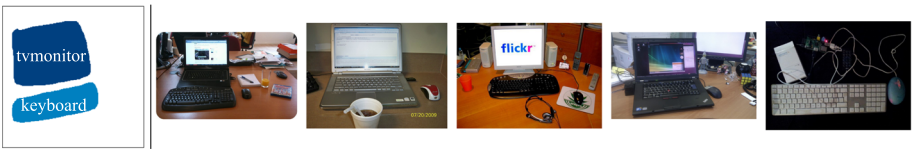
In this paper, we propose an efficient image retrieval system based on semantic segmentation. The proposed system can accept three types of queries: a natural image, a segmentation map drawn by the user, and a combination of the two. We employ a fully convolutional network (FCN) [31], which is composed 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. 1, 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 [15], which is compatible with our system. Using subjective evaluations in Section 4.2, we show that the proposed system provides a superior performance compared with a conventional text-based image retrieval system. Furthermore, in Section 5 we demonstrate that the PQ makes our system orders of magnitude faster while maintaining the retrieval quality, by quantizing each probability map independently. The average computational time and the memory usage of the proposed system are about 0.3 second and 850MB on MATLAB for 82,783 images, respectively[1].

The fundamental algorithm of our method and some experimental results have already been presented in our preliminary study [6]. In this paper, we present the results of additional experiments for further analysis.

The rest of this paper is organized as follows. Section 2 reviews related works. Section 3 describes the details of the proposed image retrieval system. Sections 4 and 5 presents the experimental results on MSCOCO and rPascal & rImageNet datasets, respectively. Finally, Section 6 concludes the paper.

[1]Note that MATLAB uses 400MB even without running a program.




(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. 1** Interactive image retrieval. Query and top five retrieved images are shown

## 2 Related work

### 2.1 Semantic image retrieval

As pointed out in [32], the majority of early methods for spatial-semantic retrieval extracted low-level features from exemplars [3, 29, 45]. For example, Jian et al. [19] used wavelet coefficients to detect directional and salient patches and extracted color moments and Gabor texture features from them. 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 [8] or sketches [30, 40–43, 47]. Song et al. [42] introduced attention modeling and proposed a new loss based on higher-order energy function in order to deal with the semantic gap and the misalignment between sketches and natural images. The objective of those CNN-based methods 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, 22, 39]. 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 [10], 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, which is the different focus from ours.

Ji et al. [16] proposed a feature embedding into low-dimensional space by learning a projection matrix. This method is relevant to our work because it not only reduces the dimensionality but also reranks the retrieved images based on visual features. The projection matrix is optimized based on the similarities between the image that the user clicked and the other retrieved images. In [17], they also proposed another reranking method, where the projection matrix is optimized to satisfy the constraints from the relevance graph and irrelevance graph. Different from ours, their methods [16, 17] need the initial retrieved results using text-based queries because their learning algorithms are tailored for reranking. Their methods can be incorporated into our system to rerank the initial results.

### 2.2 Product quantization for efficient retrieval

Many techniques have been proposed for efficient retrieval, such as binary coding and hashing (summarized in [44]). Product quantization (PQ) [15] is one of the most popular techniques, 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, 7, 23, 27, 33, 34, 36, 37, 46], in this paper we use the original PQ [15], because of its simplicity and compatibility with our system. Recently, Hinami and Satoh [11] 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 2 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. As shown in Fig. 1, 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.
(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.

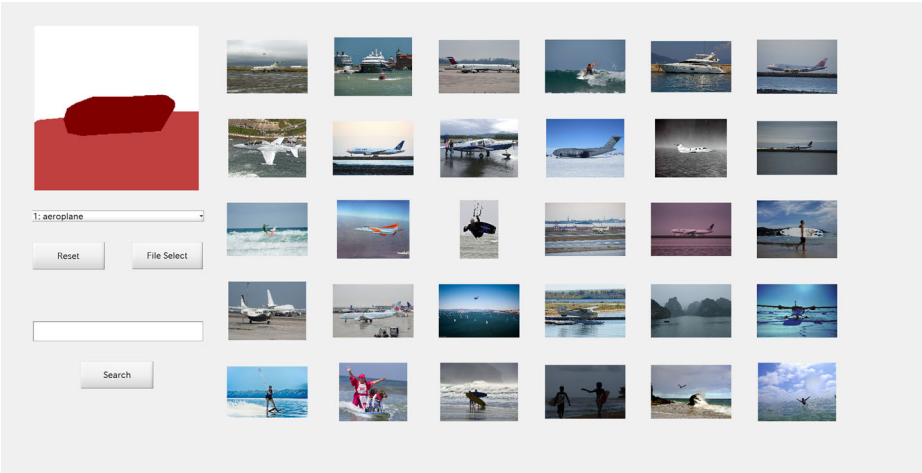



**Fig. 2** Interface of the proposed system

### 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.

#### 3.2.1 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 $p_c^i = [p_c^i(1), \cdots, p_c^i(n^2)]^\top$. Furthermore, we vectorize the $C$ probability maps as $p^i = [{p_1^i}^\top, \cdots, {p_C^i}^\top]^\top$. As an offline pre-process, we obtain the probability vectors $p^i$ for all reference images $I^i$ ($i = 1, \cdots, N$) in the database.

#### 3.2.2 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 $p^{query} \in [0, 1]^{n^2 C}$

online. We define the distance between the query image $I^{query}$ and the reference image $I^i$ as $p$-th power of the L$p$ distance between $p^{query}$ and $p^i$:

$$dist(I^{query}, I^i) = ||p^{query} - p^i||_p^p \tag{2}$$

$$= \sum_{c=1}^{C} ||p_c^{query} - p_c^i||_p^p. \tag{3}$$

In Section 5.4, we use L1 and squared L2 distances ($p = \{1, 2\}$), which are commonly used in image retrieval, and compare the performance. Hereafter, we denote $||\cdot||_p^p$ as $||\cdot||$ for notational clarity. 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.

### 3.2.3 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 $y$ 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 $S(c)$:

$$S(c) = \{j \mid y(j) = c\}. \tag{4}$$

$S(c)$ has the following properties:

$$\forall c, c' \in \{0, \cdots, C\},\ S(c) \cap S(c') = \emptyset. \tag{5}$$

$$S(0) \cup S(1) \cup \cdots \cup S(C) = \mathcal{Y}, \tag{6}$$

where $\mathcal{Y} = \{j \mid j = 1, \cdots, n^2\}$ is the set of all locations in $y$. Equation (5) means that the region $S(c)$ does not overlap each other, and (6) means the union of all the regions constitutes an entire segmentation map. Figure 3 presents an example of $S(c)$.

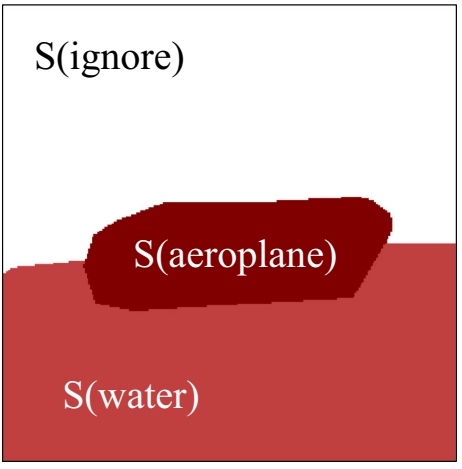



**Fig. 3** Example of segmentation map

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

$$q = [q_1^\top, \cdots, q_C^\top]^\top, \tag{7}$$

$$q_c = [q_c(1), \cdots, q_c(n^2)]^\top, \tag{8}$$

$$q_c(j) = \begin{cases} 1 \ \ if \ \ y(j) = c \\ 0 \ \ otherwise. \end{cases} \tag{9}$$

$q_c$ 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 $q_c$, 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 $q$, we define the distance between the query and a reference image as following:

$$dist(y, I^i) = \sum_{c=1}^{C} 1[S(c) \neq \emptyset] ||q_c - p_c^i||, \tag{10}$$

where $1[\cdot]$ 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.

#### 3.2.4 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 $y$ to a natural image $I^{query}$. In this case, we define a query vector $q = [q_1^\top, \cdots, q_C^\top]^\top$ as follows:

$$q_c = [q_c(1), \cdots, q_c(n^2)]^\top, \tag{11}$$

$$q_c(j) = \begin{cases} 1 & if \ \ y(j) = c \\ 0 & if \ \ y(j) \neq c \wedge y(j) \neq 0 \\ p_c^{query}(j) & if \ \ y(j) = 0. \end{cases} \tag{12}$$

In $q_c$, 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:

$$dist(y, I^i) = ||q - p^i|| \tag{13}$$

$$= \sum_{c=1}^{C} ||q_c - p_c^i||. \tag{14}$$

By calculating the rankings using (14), we can consider both the objects in the query image and those drawn by the user.

### 3.3 Product Quantization for Efficient Retrieval

In Section 3.2, we introduced image retrieval based on semantic segmentation, which considers spatial-semantic information. However, storing the long vectors $p^i \in [0, 1]^{n^2C}$ is memory consuming, because of the dimensions of $n^2C = 64^2 \times 60 = 245,760$ in our setting in Section 4 and 5. In addition, the naive computation of (3), (10) or (14) is slow. Therefore, we employ PQ [15] in order to make the system efficient in terms of both the

computational time and memory usage. We show that the approximate values of (3), (10) and (14) can be efficiently computed by applying PQ.

#### 3.3.1 Offline pre-process

We partition the vector $p^i$ into $M$ distinct subvectors $u^i_1, \cdots, u^i_M$, and quantize them independently, *i.e.*, $f_1(u^i_1), \cdots, f_M(u^i_M)$. Following the original PQ technique [15], we learn the quantizer $f_m$ ($m = 1, \cdots, M$) using k-means. We perform k-means on the set of vectors $\{u^i_m \mid i = 1, \cdots, N\}$ and obtain $K$ centroids $\mathcal{A}_m = \{a_{m,k} \mid k = 1, \cdots, K\}$. The quantizer $f_m$ is the mapping function to the nearest centroids:

$$f_m(u^i_m) = \underset{a_{m,k} \in \mathcal{A}_m}{\arg\min} ||u^i_m - a_{m,k}||. \tag{15}$$

When we set $M = C$, this quantization process corresponds to partitioning $p^i$ into each probability map $p^i_c (= u^i_m)$ 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 Section 5.

#### 3.3.2 Online search

By using PQ, we can efficiently compute the approximate value of (3) as follows:

$$\begin{aligned} dist(I^{query}, I^i) &= ||p^{query} - p^i|| && (16)\\ &= \sum_{m=1}^{M} ||u^{query}_m - u^i_m|| && (17)\\ &\approx \sum_{m=1}^{M} ||u^{query}_m - f_m(u^i_m)||. && (18) \end{aligned}$$

Because $f_m(u^i_m)$ is the mapping function to the nearest centroid, as shown in (15), we can efficiently compute (18) for all reference images by constructing a lookup table of the distances between the query subvector $u^{query}_m$ and each of the centroids. Similarly, (14) can be also computed efficiently using the lookup table.

There is a further benefit of setting $M = C$. Namely, we can efficiently compute (10) by approximating as follows:

$$\begin{aligned} dist(y, I^i) &= \sum_{c=1}^{C} 1[S(c) \neq \emptyset] ||q_c - p^i_c|| && (19)\\ &\approx \sum_{c=1}^{C} 1[S(c) \neq \emptyset] ||q_c - f_c(p^i_c)||. && (20) \end{aligned}$$

Similarly, we can construct a lookup table of the distances between the query subvector $q_c$ and each of the centroids. To exploit this approximation, we set $M = C$ in Sections 4 and 5.

Figure 4 summarizes the framework of the proposed method.

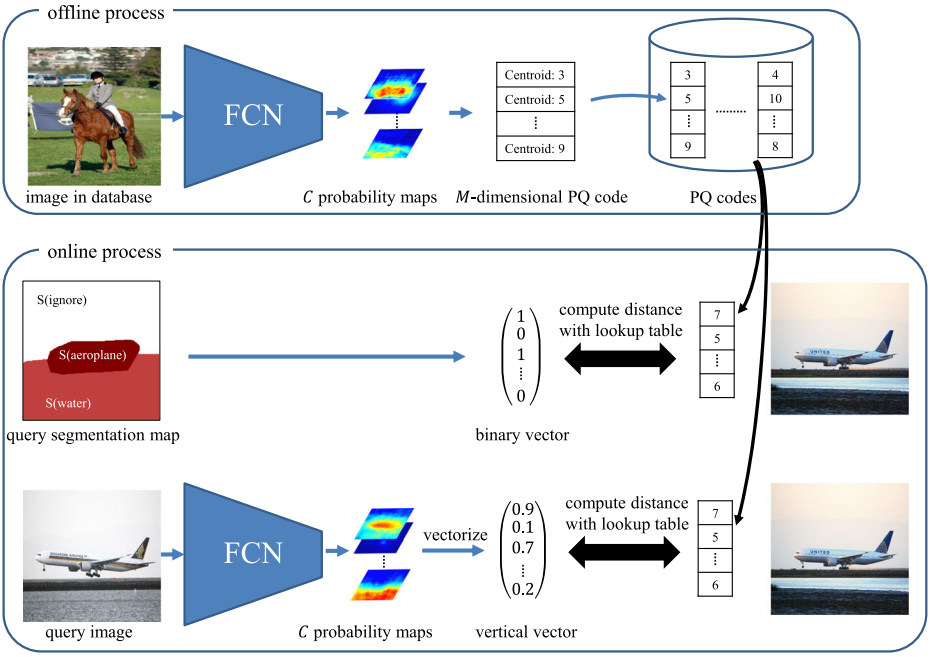



**Fig. 4** Framework of the proposed method

### 3.3.3 Asymmetric vs. symmetric distance computation

PQ has two options to efficiently calculate approximate distances: asymmetric distance computation (ADC) and symmetric distance computation (SDC). The distance computation in (18) is ADC because only the subvector $u_m^i$ is quantized with $f_m$ and the query subvector $u_m^{query}$ is not. In SDC, not only $u_m^i$ but also $u_m^{query}$ is quantized with $f_m$. While ADC is more accurate than SDC, we have to construct a lookup table every time a query is given in ADC. In contrast, only one lookup table of the distances between the centroids is enough for every query in SDC. However, the computational time to construct a lookup table is negligibly small in our image retrieval application. Although we can use the both options for our application, we use ADC from the above reason.

## 3.4 Comparison to other image retrieval systems

It is important to note that our objective is not showing high performance, compared with the state-of-the-art methods on image retrieval benchmarks. Rather than that, one of our contributions is that we propose a novel application of semantic segmentation. Although semantic segmentation has been attracting much attention in recent years, its application has been limited so far (e.g., recognition for autonomous driving).

Here, we compare the properties of the proposed system with state-of-the-art image retrieval systems with the different types of queries. Table 1 summarizes the comparison. As discussed in Section 1, [45] cannot deal with the scales and shapes of the objects. The retrieval systems based on bounding boxes [12, 32] cannot deal with the object shapes. In contrast to them, one of the advantages of the proposed method is that the object shapes

**Table 1** Comparison of the properties with the different types of state-of-the-art image retrieval systems

| | Query type | Can be specified? | | | Object | Can the retrieved |
|---|---|---|---|---|---|---|
| | | location | scale | shape | classes | images reused? |
| [45] | concept map | ✓ | - | - | any | - |
| [32] | bonding boxes (BBs) | ✓ | ✓ | - | any | (✓) |
| [12] | BBs with relative attributes | ✓ | ✓ | - | any | ✓ |
| [42] | sketch | ✓ | ✓ | ✓ | - | (✓) |
| Ours | natural image /segmentation map | ✓ | ✓ | ✓ | fixed | ✓ |

that users specify can be considered (e.g., diagonal road and round dish) as shown later in Fig. 12. Although the sketch-based retrieval system [42] also can treat thee object shapes, the object classes cannot be explicitly specified. Another advantage of the proposed method is that users can reuse one of the retrieved images as the next query. Although [32] and [42] also can possibly do it by performing object/edge detection on the retrieved images, that has not been discussed in their papers. The disadvantage of the proposed method is that it can deal with only pre-defined classes. In contrast, [32, 45] and [12] can deal with any object class although the relative attributes in [12] must be pre-defined (e.g., *stand next to*).

## 4 Experiments on MSCOCO

### 4.1 Implementation details

As the FCN, we used DeepLab-v2 [4] implemented on the Caffe library [18], which is publicly available. We trained this network on the trainval set of the PASCAL-Context Dataset [35], 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 $2.5 \times 10^{-4}$ and was multiplied by $(1 - (\frac{iter}{max\_iter})^{power})$ at each iteration. We set the $max\_iter$ to 20,000, $power$ to 0.9, momentum to 0.9, and weight decay to $5.0 \times 10^{-4}$. 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 $K = 256$. Our implementation of the PQ is based on the publicly available code[2], and the core part for distance computation is implemented on C-based mex function. Unless we state, we use squared $L2$ distance.

### 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 [28], which contains 82,783 images. Each image has five caption annotations written by Amazon Mechanical Turk workers.

We compared the following two methods.

[2] http://people.rennes.inria.fr/Herve.Jegou/projects/ann.html

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 Section 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 (18) or (20). 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 (18) or (20).

The procedure of the test for each subject 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 the subject to imagine an image that the caption describes.
3. We asked the subject to choose ten images that he/she think are similar to the imagined image using the text-based system. As shown in Fig. 2, top 30 retrieved images were shown on the display. The subject could make queries and search for images any number of times until the total number of the chosen images reached to ten.
4. We asked the subject to do the same task as step 3 using the text+proposed system instead of text-based system.
5. We showed the twenty chosen images in random order, and asked the subject to assign a relevance score from 1 to 5 to each image, which means how much each image is relevant to the image that he/she imagined at step 2. A score of 5 indicates the highest relevance, and 1 the lowest.
6. Steps 2-5 were repeated three times.

The number of subjects was ten, and their ages were 21-26. Nine subjects were male, and one was female. The chosen captions were, for example, “A person skiing alone on snow covered mountain,” “A school bus on the road behind a truck” and so on. Fig. 5 presents

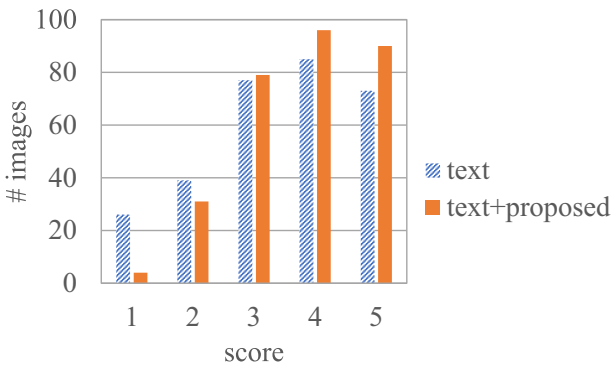



**Fig. 5** Histogram of scores

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 (18) or (20) with PQ. Because we set $M = C$, the size of lookup table for computing (18) is $C \times K$, where $M$, $C$ and $K$ are the number of partitions, classes, and centroids, respectively. The average computational time for (18) and sorting for all 82,783 images was about 0.3 sec 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 (20) depends on the number of classes the user specifies (i.e., $|\mathcal{C}'|$ where $\mathcal{C}' = \{c \mid S(c) \neq \emptyset\}$). Figure 6 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 12 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.

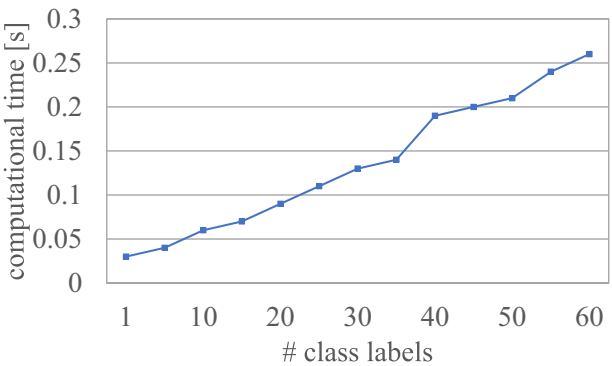



**Fig. 6** Computational time versus number of class labels

**Table 2** 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 Experiments on rPascal & rImageNet

### 5.1 Dataset

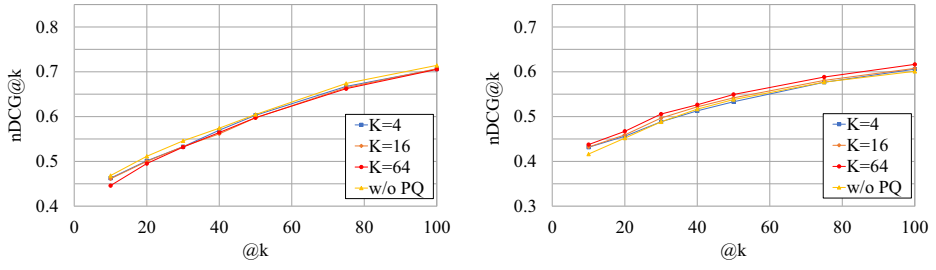
We used the rPascal and rImageNet datasets [39], which contain 1,895 and 3,354 images, respectively. The rPascal dataset contains 50 query images, and each query image has 180 reference images on average. The rImageNet dataset contains 50 query images, with 305 reference images per query on average. Both datasets contain relevance score annotations for each pair of query and reference images. Using these datasets, in this section, we evaluated the performance of the proposed method for structured retrieval. Similarly to [39], we used the normalized discounted cumulative gain (nDCG), which is defined as:

$$DCG@k = \sum_{i=1}^{k} \frac{2^{rel_i} - 1}{\log_2(i+1)} nDCG@k = \frac{DCG@k}{IDCG@k}, \tag{21}$$

where $rel^i$ is the relevance score of the $i$-th ranked image, and IDCG is the ideal discounted cumulative gain.

### 5.2 Computational time

Table 2 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




(a) rPascal (b) rImageNet

**Fig. 7** nDCG of the proposed method with squared $L2$ distance and various numbers of centroids ($K$)

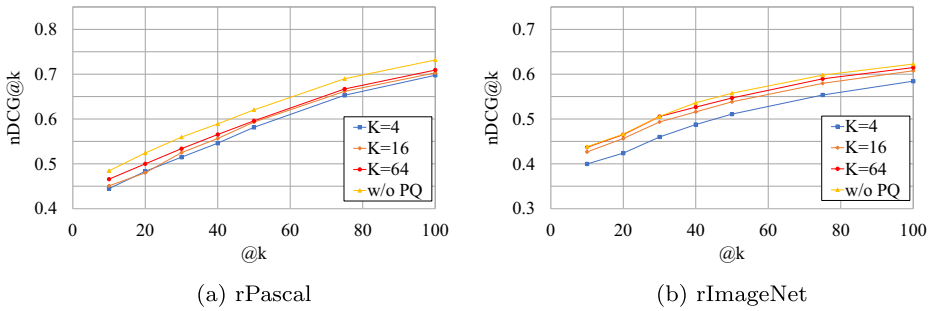



**Fig. 8** nDCG of the proposed method with $L1$ distance and various numbers of centroids ($K$)

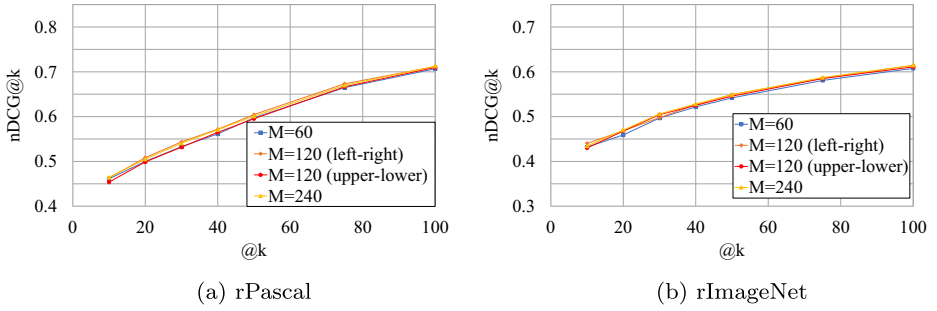



**Fig. 9** nDCG of the proposed method with various number of partitions ($M$)

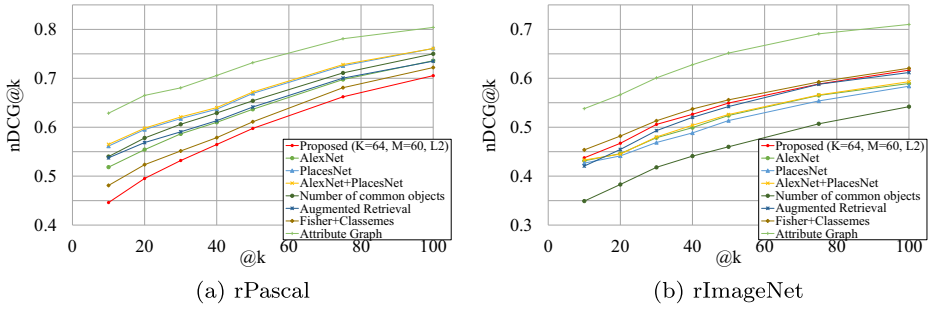



**Fig. 10** Comparison of nDCG with other methods. Except for the proposed method, the plots are from [39]. *Augmented Retrieval*, *Fisher+Classemes*, and *Attribute Graph* indicate [2, 5], and [39], respectively

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.

### 5.3 Number of centroids *K*

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.

### 5.4 *L*1 vs. squared *L*2 distance

Figure 8a and b show the results of the proposed method with $L1$ distance. When we compare Figs. 7 and 8, we observe that the performance of $L1$ distance without PQ is a little better than that of squared $L2$. However, when we use PQ ($K = 16$ and 64), the performance with $L1$ distance slightly drops and is almost same as squared $L2$. Especially, when the quantization is rough ($K = 4$), it gets much worse, which means $L1$ distance is less robust for PQ in this application. That is why we used squared $L2$ distance, not $L1$.

### 5.5 Number of partitions *M*

One may think "Is the number of partitions $M = C$ really optimal setting?". To answer this question, we tried other number of partitions $M = 2C$ and $4C$. When we set $M = 2C$, each probability map is partitioned into two parts ("left and right parts" or "upper and lower parts") and quantized separately. Similarly, when $M = 4C$, each probability map is partitioned into four parts.

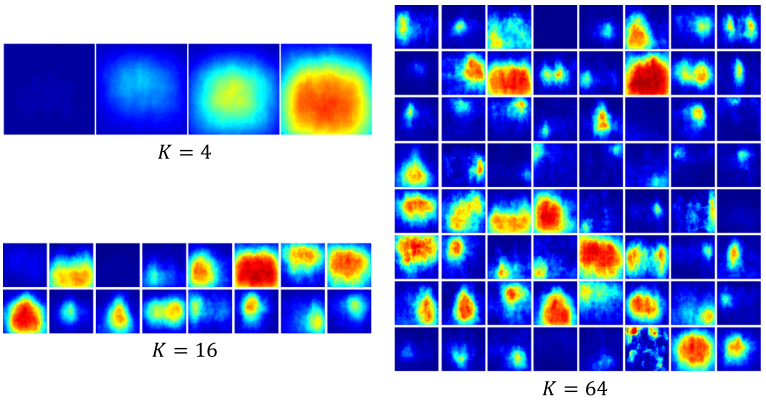
Figure 9a and b show their results. We observe that the number of partitions has little effect on the retrieval performance.

### 5.6 Comparison with other methods

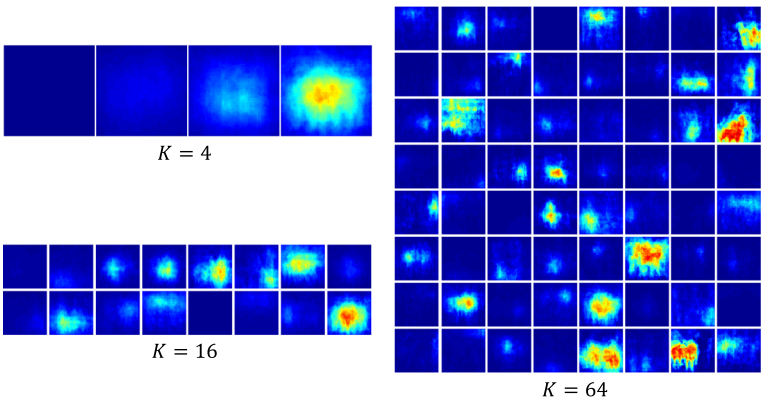
Figure 10a and b present comparisons with other methods on rPascal and rImageNet, respectively. The proposed method is significantly inferior to Attribute-Graph [39], which is reasonable, because that method is tailored for structured retrieval and ours is not. Although the proposed method performs worse than other methods on rPascal, it shows a competitive performance with other methods except for Attribute-Graph [39] on rImageNet.

### 5.7 Visualization of Centroids

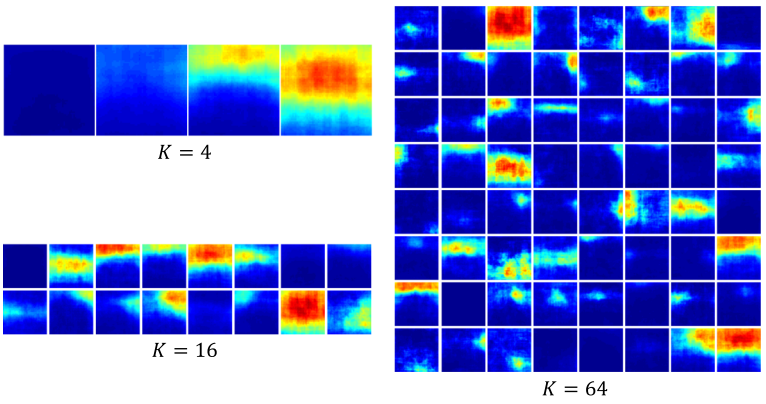
Figure 11a, b and c show the centroids of the probability maps for *person*, *horse* and *building* classes, respectively. When the number of centroids is small ($K = 4$), each centroid represents only the possibility that persons exist (i.e., "do not exist", "probably do not exist", "probably exist", and "exist"), and cannot represent the locations of persons because the quantization is too rough. When $K = 16$, the centroids represent not only the possibility but also the locations of persons in the image. When we set $K = 64$, much more variety of patterns are represented (e.g., multiple persons in different locations), which leads to accurate retrieval (Fig. 12).




(a) *person*




(b) *horse*




(c) *building*

**Fig. 11** Centroids of the probability maps for *person*, *horse* and *building* classes

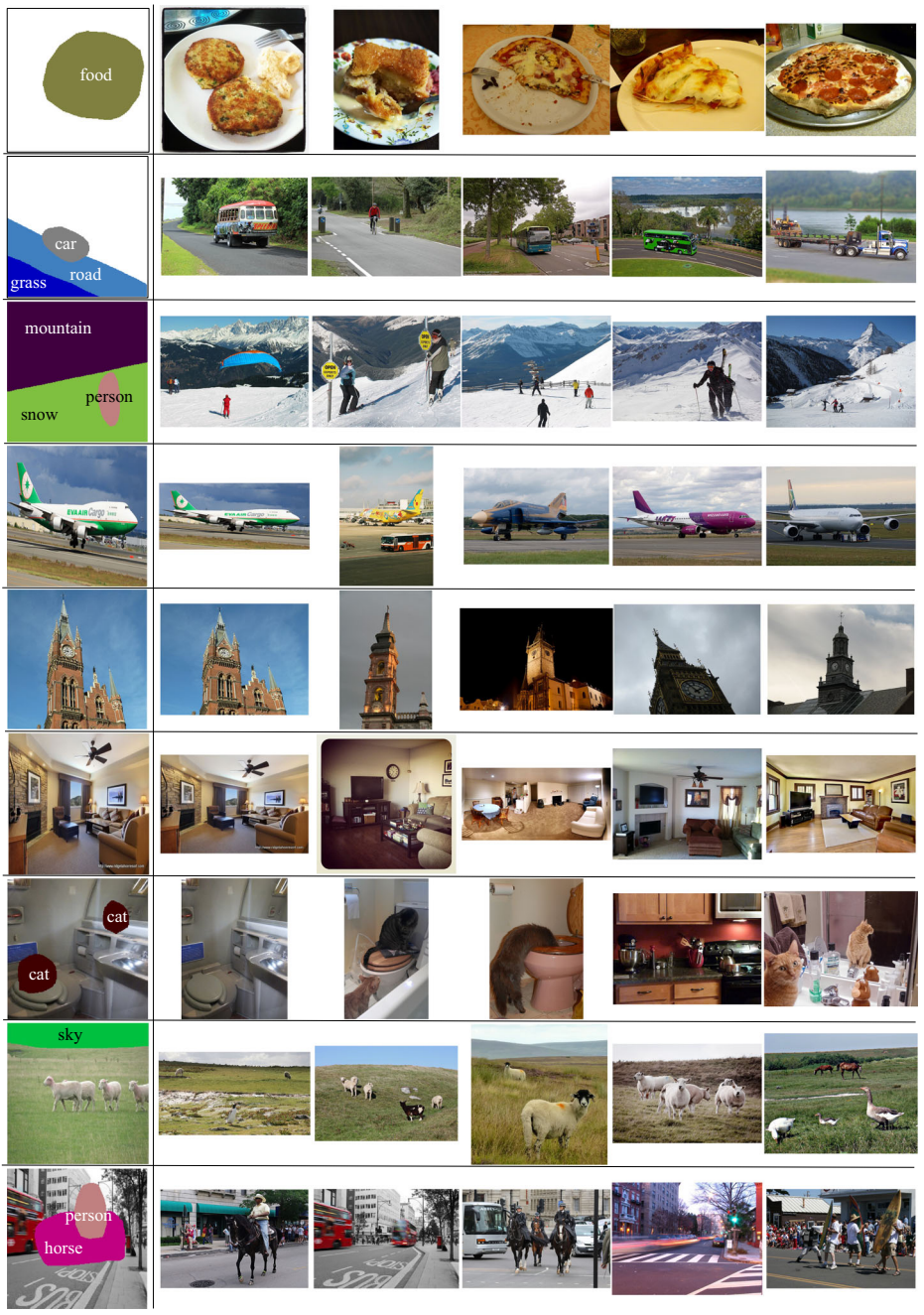



**Fig. 12** Examples of retrieved images using the proposed method on the MSCOCO2014 train set. Query and top five images are shown

## 6 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. Although we employed DeepLab-v2 [4] and original PQ [15] in this paper, we believe that the performance of the proposed system, in terms of retrieval quality, computational time, and memory usage, can be improved by using more powerful network (e.g., pyramid scene parsing network [48]) and advanced PQ techniques such as Optimized PQ [7] or PQTable [34].

The limitation of the proposed method is that it cannot treat new classes that are not included in the training dataset of semantic segmentation. To train the FCN from image-level labels or videos by employing weakly-supervised semantic segmentation techniques [13, 20, 26] is one possible solution for the limitation.

**Acknowledgements** 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 in Fig. 10.

**Open Access** This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (http://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

## References

1. Babenko A, Lempitsky V (2014) Additive quantization for extreme vector compression. In: CVPR
2. Cao X, Wei X, Guo X, Han Y, Tang J (2014) Augmented image retrieval using multi-order object layout with attributes. In: ACMMM
3. Cao Y, Wang H, Wang C, Li Z, Zhang L, Zhang L (2010) Mindfinder: interactive sketch-based image search on millions of images. In: ACMMM
4. Chen LC, Papandreou G, Kokkinos I, Murphy K, Yuille A (2017) Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE TPAMI
5. Douze M, Ramisa A, Schmid C (2011) Combining attributes and fisher vectors for efficient image retrieval. In: CVPR
6. Furuta R, Inoue N, Yamasaki T (2018) Efficient and interactive spatial-semantic image retrieval. In: MMM
7. Ge T, He K, Ke Q, Sun J (2013) Optimized product quantization for approximate nearest neighbor search. In: CVPR
8. Gordo A, Almazán J, Revaud J, Larlus D (2016) Deep image retrieval: Learning global representations for image search. In: ECCV
9. Gordo A, Larlus D (2017) Beyond instance-level image retrieval: Leveraging captions to learn a global visual representation for semantic retrieval. In: CVPR
10. Guerrero P, Mitra NJ, Wonka P (2016) RAID: A relation-augmented image descriptor. ACM TOG 35(4):46:1–46:12
11. Hinami R, Satoh S (2016) Large-scale r-cnn with classifier adaptive quantization. In: ECCV
12. Hinami R, Matsui Y, Satoh S (2017) Region-based image retrieval revisited. In: ACMMM, pp 528–536

13. Hong S, Yeo D, Kwak S, Lee H, Han B (2017) Weakly supervised semantic segmentation using web-crawled videos. In: CVPR
14. Inoue N, Furuta R, Yamasaki T, Aizawa K (2017) Object detection refinement using markov random field based pruning and learning based rescoring. In: ICASSP
15. Jegou H, Douze M, Schmid C (2011) Product quantization for nearest neighbor search. IEEE TPAMI 33(1):117–128
16. Ji Z, Pang Y, Li X (2015) Relevance preserving projection and ranking for web image search reranking. IEEE TIP 24(11):4137–4147
17. Ji Z, Pang Y, Yuan Y, Pan J (2016) Relevance and irrelevance graph based marginal fisher analysis for image search reranking. Signal Process 121:139–152
18. Jia Y, Shelhamer E, Donahue J, Karayev S, Long J, Girshick R, Guadarrama S, Darrell T (2014) Caffe: Convolutional architecture for fast feature embedding. In: ACMMM
19. Jian M, Yin Y, Dong J, Lam KM (2018) Content-based image retrieval via a hierarchical-local-feature extraction scheme. Multimed Tools Appl 77(21):29099–29117
20. Jin B, Ortiz-Segovia MV, Süsstrunk S (2017) Webly supervised semantic segmentation. In: CVPR
21. Johnson J, Karpathy A, Fei-Fei L (2016) Densecap: Fully convolutional localization networks for dense captioning. In: CVPR
22. Johnson J, Krishna R, Stark M, Li LJ, Shamma D, Bernstein M, Fei-Fei L (2015) Image retrieval using scene graphs. In: CVPR
23. Kalantidis Y, Avrithis Y (2014) Locally optimized product quantization for approximate nearest neighbor search. In: CVPR
24. Karpathy A, Fei-Fei L (2015) Deep visual-semantic alignments for generating image descriptions. In: CVPR
25. Kim G, Moon S, Sigal L (2015) Ranking and retrieval of image sequences from multiple paragraph queries. In: CVPR
26. Kolesnikov A, Lampert CH (2016) Seed, expand and constrain: Three principles for weakly-supervised image segmentation. In: ECCV, pp 695–711
27. Li J, Lan X, Li X, Wang J, Zheng N, Wu Y (2017) Online variable coding length product quantization for fast nearest neighbor search in mobile retrieval. IEEE TMM 19(3):559–570
28. Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL (2014) Microsoft coco: Common objects in context. In: ECCV
29. Liu C, Wang D, Liu X, Wang C, Zhang L, Zhang B (2010) Robust semantic sketch based specific image retrieval. In: ICME
30. Liu L, Shen F, Shen Y, Liu X, Shao L (2017) Deep sketch hashing: Fast free-hand sketch-based image retrieval. In: CVPR
31. Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: CVPR
32. Long Mai HJ, Lin Z, Fang C, Brandt J, Liu F (2017) Spatial-semantic image search by visual feature synthesis. In: CVPR
33. Matsui Y, Yamasaki T, Aizawa K (2015) Pqtable: Fast exact asymmetric distance neighbor search for product quantization using hash tables. In: ICCV
34. Matsui Y, Yamasaki T, Aizawa K (2017) Pqtable: Non-exhaustive fast search for product-quantized codes using hash tables IEEE TMM
35. Mottaghi R, Chen X, Liu X, Cho NG, Lee SW, Fidler S, Urtasun R, Yuille A (2014) The role of context for object detection and semantic segmentation in the wild. In: CVPR
36. Ning Q, Zhu J, Zhong Z, Hoi SC, Chen C (2017) Scalable image retrieval by sparse product quantization. IEEE TMM 19(3):586–597
37. Norouzi M, Fleet DJ (2013) Cartesian k-means. In: CVPR
38. Ordonez V, Han X, Kuznetsova P, Kulkarni G, Mitchell M, Yamaguchi K, Stratos K, Goyal A, Dodge J, Mensch A et al (2016) Large scale retrieval and generation of image descriptions. IJCV 119(1):46–59
39. Prabhu N, Venkatesh Babu R (2015) Attribute-graph: a graph based approach to image ranking. In: ICCV
40. Qi Y, Song YZ, Zhang H, Liu J (2016) Sketch-based image retrieval via siamese convolutional neural network. In: ICIP
41. Sangkloy P, Burnell N, Ham C, Hays J (2016) The sketchy database: learning to retrieve badly drawn bunnies. ACM TOG 35(4):119
42. Song J, Yu Q, Song YZ, Xiang T, Hospedales TM (2017) Deep spatial-semantic attention for fine-grained sketch-based image retrieval. In: ICCV
43. Wang F, Kang L, Li Y (2015) Sketch-based 3d shape retrieval using convolutional neural networks. In: CVPR
44. Wang J, Zhang T, Sebe N, Shen HT et al (2017) A survey on learning to hash IEEE TPAMI

45. Xu H, Wang J, Hua XS, Li S (2010) Image search by concept map. In: SIGIR
46. Yu L, Huang Z, Shen F, Song J, Shen HT, Zhou X (2017) Bilinear optimized product quantization for scalable visual content analysis. IEEE TIP 26(10):5057–5069
47. Yu Q, Liu F, Song YZ, Xiang T, Hospedales TM, Loy CC (2016) Sketch me that shoe. In: CVPR
48. Zhao H, Shi J, Qi X, Wang X, Jia J (2017) Pyramid scene parsing network. In: CVPR

**Ryosuke Furuta** received the B.S. and M.S. degrees in information and communication engineering from The University of Tokyo, in 2014 and 2016, respectively. He is currently working towards the Ph. D. degree at The University of Tokyo. His research interests include computer vision, machine learning, and image processing, especially MRF optimization. He is a member of IEEE, ACM, ITE.

**Naoto Inoue** received the B.E. and M.S. in information and communication engineering from the University of Tokyo in 2016 and 2018, respectively. He is currently a Ph.D. student in The University of Tokyo, Japan. His research interests lie in computer vision, with particular interest in object detection and domain adaptation. He is a member of IEEE.

**Toshihiko Yamasaki** received the B.S. degree in electronic engineering, the M.S. degree in information and communication engineering, and the Ph.D. degree from The University of Tokyo in 1999, 2001, and 2004, respectively.

From April 2004 to Oct. 2006, he was an Assistant Professor at Department of Frontier Informatics, Graduate School of Frontier Sciences, The University of Tokyo. He is currently an Associate Professor at Department of Information and Communication Engineering, Graduate School of Information Science and Technology, The University of Tokyo. He was a JSPS Fellow for Research Abroad and a visiting scientist at Cornell University from Feb. 2011 to Feb. 2013.

His current research interests include attractiveness computing based on multimedia big data analysis, pattern recognition, machine learning, and so on. His publication includes three book chapters, more than 60 journal papers, more than 160 international conference papers, more than 500 domestic conference papers. He has received around 50 awards.

Dr. Yamasaki is a member of IEEE, ACM, IEICE, ITE, IPSJ.