PAPER

Learning Local Similarity with Spatial Interrelations on Content-Based Image Retrieval

Longjiao ZHAO[†], **Yu WANG**[††], ***Nonmembers***, **Jien KATO**[†††a)], ***and*** **Yoshiharu ISHIKAWA**[†], ***Members***

SUMMARY Convolutional Neural Networks (CNNs) have recently demonstrated outstanding performance in image retrieval tasks. Local convolutional features extracted by CNNs, in particular, show exceptional capability in discrimination. Recent research in this field has concentrated on pooling methods that incorporate local features into global features and assess the global similarity of two images. However, the pooling methods sacrifice the image's local region information and spatial relationships, which are precisely known as the keys to the robustness against occlusion and viewpoint changes. In this paper, instead of pooling methods, we propose an alternative method based on local similarity, determined by directly using local convolutional features. Specifically, we first define three forms of local similarity tensors (LSTs), which take into account information about local regions as well as spatial relationships between them. We then construct a similarity CNN model (SCNN) based on LSTs to assess the similarity between the query and gallery images. The ideal configuration of our method is sought through thorough experiments from three perspectives: local region size, local region content, and spatial relationships between local regions. The experimental results on a modified open dataset (where query images are limited to occluded ones) confirm that the proposed method outperforms the pooling methods because of robustness enhancement. Furthermore, testing on three public retrieval datasets shows that combining LSTs with conventional pooling methods achieves the best results.

key words: *image retrieval, convolutional neural network, deep local features, local similarity, 4D convolution, spatial correlation*

1. Introduction

Content-based image retrieval (CBIR) aims at ranking images in a huge dataset based on the information given from a query image. It is essential for many artificial intelligence technologies such as search engines. Generally, the pipeline of image retrieval includes two main steps: image representation and similarity evaluation. The first step aims to produce a global feature that well represents the input image, while the second step calculates the similarity scores between the query image and gallery images based on the representation generated in the first step. Euclidean distance and cosine similarity are commonly used for similarity evaluation. According to the rank list of similarity scores, the top-k images are considered as similar results with the query image.

Since 2012, because of the success of AlexNet [1] in object recognition, most researchers have transferred to working on CNN (convolutional neural network)-based approaches. As CNNs show remarkable performance in feature extraction, the deep features produced by CNNs attract much attention. Initially, works [2] and [3] utilize the activations of fully-connected layers as a global feature. Since the CNN architecture limits the size of input images, the global feature from fully-connected layers lacks multi-scale information. In contrast, local features from deep convolutional layers support any input image's size. So, researchers start to turn their attention to extracting such local features and then pooling them into a global feature [3], [4].

However, most pooling methods, which incorporate local features into global features via pooling first and then assess the global similarity between images, sacrifice the information of local regions in exchange for gaining global features. On the other hand, for image retrieval tasks, since such information plays a crucial role in the robustness against occlusion and viewpoint changes [5], losing or weakening this information always causes a negative impact on retrieval performance. Moreover, because pooling methods also weaken the information of spatial relationships among local regions, they lose most of the structure information of input images. In order to keep both abundant local region information [6], [7] and structure information, we think that evaluating the similarity by directly using deep local features could be an effective breakthrough. Below we refer to the similarity of local features as local similarity and the similarity of pooled global features as global similarity. To the best of our knowledge, there are few studies on image retrieval based on local similarity. Although the paper [8] proposed a local similarity-based approach, it couldn't provide a qualitative and comprehensive analysis of the impact of local similarity on image retrieval.

From the perspective described above, in this paper we propose a local similarity-based method for image retrieval tasks. Figure 1 shows the overview of our method, where the lower branch consisting of two modules: local similarity tensor (LST) generation and local similarity evaluation, is the key component of the proposed method. In particular, deep local features are extracted from a pre-trained CNN first, and then the similarity score between a pair of deep local features from the query and gallery images is calculated to produce an LST. The LST is finally fed into a similarity-CNN (SCNN) to obtain the local similarity score.

Manuscript received September 8, 2022.
Manuscript revised December 26, 2022.
Manuscript publicized February 14, 2023.

[†]The authors are with Graduate School of Informatics, Nagoya University, Nagoya-shi, 464–8601 Japan.
[††]The author is with Center for Information and Communication Technology, Hitotsubashi University, Tokyo, 186–8601 Japan.
[†††]The author is with College of Information Science and Engineering, Ritsumeikan University, Kusatsu-shi, 525–8577 Japan.
a) E-mail: jien@fc.ritsumei.ac.jp

DOI: 10.1587/transinf.2022EDP7163

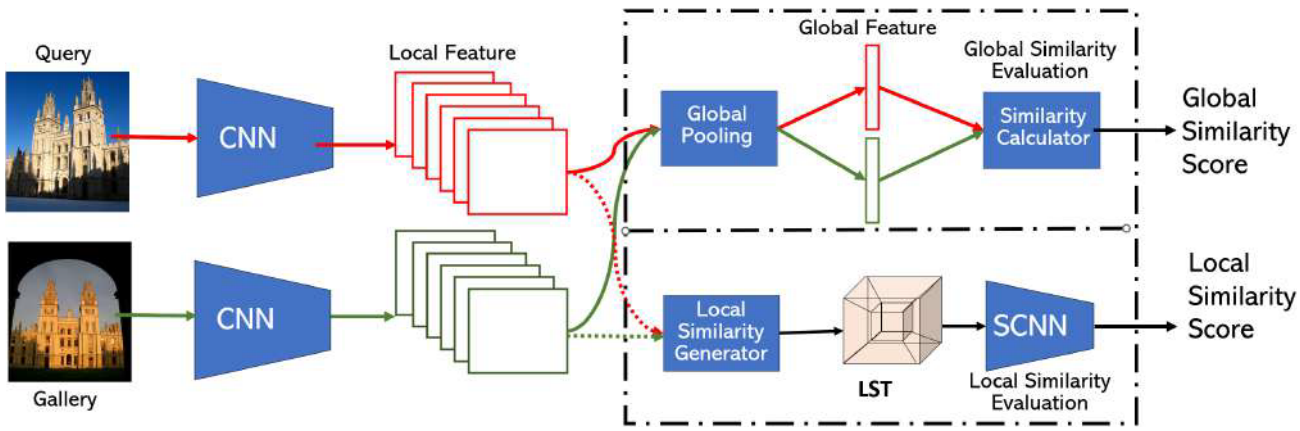

Fig. 1 Overview and workflow of the proposed image retrieval approach based on local similarity and global similarity. From the last convolutional layer, the global similarity score is calculated by global features that are generated by a pooling method. For the local similarity, an LST generated from local convolutional features is fed into the similarity CNN (SCNN) to produce the local similarity score.

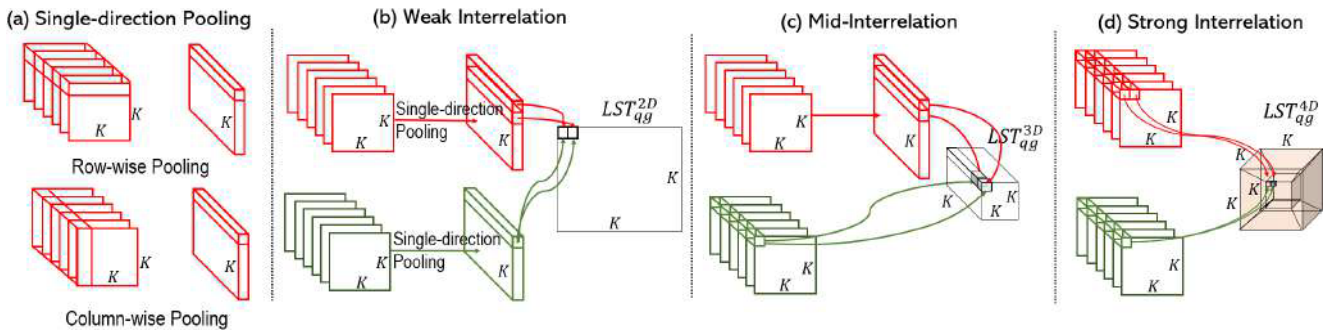

Fig. 2 Generating local similarity tensors. A single-direction pooling in (a) is defined to decrease the structure information of local features, which acts as the base for generating three different types of local similarity tensors (LSTs). Three types of tensors, named 2D, 3D, and 4D LSTs, are generated to represent weak-to-strong spatial interrelationships between local features of query and gallery images, that is, (b) weak interrelatio, (c) mid-interrelation, and (d) strong interrelation, respectively.

To validate our approach, we evaluate the proposed method by practising it on three forms of LSTs (i.e. 2D, 3D, and 4D LSTs) that correspond to three modes of spatial relationships among local regions: (1) the structure information of both query and gallery images is retained; (2) the structure information of either query or gallery image is retained; and (3) the relative structure information between query and gallery images is retained, as illustrated in Fig. 2. To adapt to three forms of LSTs, we train two SCNN models, a 2D SCNN based on a 2D convolution/pooling module, and a 4D SCNN based on a 4D convolution/pooling module, to learn the local similarity.

We conduct extensive experiments to investigate the impact of local region information and spatial relationships between local regions, as well as to identify the best practices for the proposed local similarity-based method. The experimental results on a modified open dataset, where query images are limited to occluded ones, confirm that our proposed method outperforms existing pooling methods based on global similarity, due to a significant enhancement of robustness against occlusion and viewpoint change. In addition, testing on three public retrieval datasets shows that the combination of local similarity and global similarity, as shown in the whole framework in Fig. 1, achieves the best results.

Our contributions are summarized as follows: We thoroughly investigate the influence of local region information and spatial relationships among local regions on image retrieval. We introduce a local similarity-based method by proposing 4D LSTs (to represent local information) as well as a 4D SCNN (to accomplish local similarity evaluation based on 4D LSTs), which outperform all embedded global features in a public retrieval dataset. To the best of our knowledge, this is the first time using 4D convolution in image retrieval tasks. We demonstrate that the local similarity-based method is advantageous in the situation where target objects are occluded in query images. We show that combining the local similarity with the global similarity achieves the best performance in the datasets Oxford5k, Paris6k, and CUB-200-2011.

2. Related Work

Content-based image retrieval (CBIR) Since the 1990s, CBIR has attracted a lot of attention from the computer vision community [9]. Many important works had been actively conducted from 1990s to 2000s, as summarized in [10]–[12]. In the late 1990s, with the introduction of many powerful handcrafted local features, for instance, the scale-invariant local visual feature (SIFT) [13] and histogram of oriented gradients (HOG) [14], the accuracy of CBIR significantly advanced. Another turning point appeared in 2003, the proposal of the Bag-of-Visual-Words (BoVW) [15] enabled to easily generate compact global features from local features. Owing to the BoVW, researchers greatly improved the performance of CBIR by using large visual codebooks and spatial verification. Afterwards, a lot of compact global features were introduced at the image representation level, such as Fisher vector [16] and VLAD [17].

Deep features With the development of deep learning in the past decade, convolutional neural networks have gradually replaced the status of traditional handcrafted feature-based methods in image retrieval. Due to the great success of the AlexNet [1] in classification tasks, researchers try to explore CNN-based methods also for image retrieval, which can go beyond handcrafted feature-based methods. At first, CNN was just used simply as a local feature extractor because the activations from the last convolutional layer showed extraordinary performance on image representation. So, researchers paid much attention to how to generate a compact global feature. A standard way is to use a pre-trained network as the feature extractor and encode the local features into a global feature by means of, for example, Fisher vector [16] or VLAD [17]. It has been practiced in many studies. Then, with the introduction of the contrastive loss and triplet loss, it became clear that fine-tuned networks [18]–[24] with a retrieval dataset achieved much better performance than simply using pre-trained networks as feature extractors. So, the global pooling methods started to be a popular research topic [3], [4], [25]–[30], since they could be easily included in the fine-tuning procedure. SPoC [3] and MAC [4] are usually used to generate global features by sum or maximum procedure, and GeM [29] is generally used to pool local features with a power mean, which can transform to SPoC and MAC depending on different hyperparameters. However, global features lose inevitably most of local region information and spatial relationships among local regions because of pooling process.

Similarity evaluation There are many studies on similarity evaluation using global features. Among them, it is no exaggeration to say Euclidean distance and cosine similarity are the most widely adopted evaluation methods. Even though similarity evaluation based on global features is a mainstream way, recently, researchers have become conscious that local features can positively affect the similarity evaluation in image retrieval tasks. For example, the works [31], [32] tried to use the similarity of local features in the test procedure to increase precision, and Yang et al. [33] attempt to advance the retrieval performance by fusing the local and global features. In addition, Chen et al. [8] presented a method that increased the retrieval accuracy by employing local similarity to align the similar region pair between the query and gallery images, but it led to additional overhead costs required for extra localization labels. A quantitative and thorough investigation and analysis of how local features affect similarity evaluation is still a remaining problem.

3. Local Similarity Tensor

In this section, we present three methods to generate local similarity tensors (LSTs) according to three patterns of spatial relationships among local regions.

3.1 Global Similarity

The upper branch in Fig. 1 shows the general workflow of image retrieval based on global features. Let $X \in R^{M\times N\times C}$ denote a set of feature maps from the last convolutional layer of an input image. M and N are the spatial sizes of feature maps, and C is the number of channels. With a pooling method F, local convolutional features X are aggregated to global features G as:

$$G = F(X); \quad G \in R^{1\times C}. \tag{1}$$

We use G_q and G_g to represent the global features of the query and gallery images, respectively. The global similarity between the query and gallery images can be defined as:

$$S^{\mathscr{G}}_{qg} = D(G_q, G_g), \tag{2}$$

where D is a distance function, and we adopt cosine similarity as D in this work.

3.2 Local Feature Extraction

CNNs such as VGG [34], ResNet [35], and DenseNet [36] generally include five convolutional blocks. Local features from each block deliver the information from different levels [37]. Earlier works made a lot of efforts to prove that feature maps from the final layer achieve the best results for global similarity evaluation. To the best of our knowledge, how local features should be chosen and used for local similarity has not yet been thoroughly investigated. So, it is necessary to start with seeking the layer that achieves the best performance for local similarity evaluation. As shown in Fig. 3, we extract local features from each convolutional block. For a query image q, we extract local convolutional feature maps $X^L_q \in R^{M_q\times N_q\times C}$ with spatial shape $M_q \times N_q$ from block L. Similarly, local convolutional feature maps of a gallery image g from block L are extracted and indicated by $X^L_g \in R^{M_g\times N_g\times C}$ with spatial shape $M_g \times N_g$.

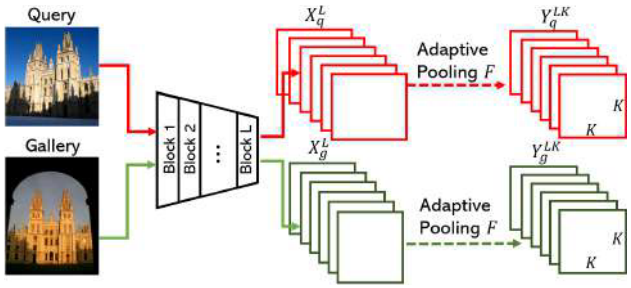

Fig. 3 Procedure for local feature extraction. The local features are extracted from each convolutional block (from Block 1 to Block L).

Considering that it is infeasible to analyze local similarity with all possible spatial forms of local features caused by the input image, we implement an adaptive pooling method F that produces feature maps with a certain spatial shape $K \times K$, where K can be controlled by the pooling window size. So, for the local feature maps X^L, the output feature maps with F are described as:

$$Y^{LK} = F(X^L, K), \quad Y^{LK} \in R^{K \times K \times C}. \tag{3}$$

We employ the generalized-mean (GeM [29]) as the pooling kernel because of its superior performance in image retrieval tasks. Thus, a feature vector y_{ij}^{LK} at location (i, j) of feature maps Y^{LK} is calculated as:

$$y_{ij}^{LK} = \left(\frac{1}{|X_{r_{ij}}^L|} \sum_{x \in X_{r_{ij}}^L} x^p \right)^{\frac{1}{p}}, \tag{4}$$

where $X_{r_{ij}}^L$ represents the feature maps of the pooling region r_{ij} $(i, j = 1, \ldots, K)$, and p is a parameter that controls the types of pooling methods. For example, when $p = 1$, Eq. (4) means an average pooling; when p verges to infinity, it becomes to max pooling.

Note that the feature maps Y^{LK} divide the input image $I \in R^{M_I \times N_I \times 3}$ into a $K \times K$ grid, indicating that K is inversely proportional to the local region size.

3.3 Local Similarity Tensor Generation

We first introduce a single-direction pooling, row-wise pooling or column-wise pooling, to decrease the structure information of local features Y^{LK}, as shown in Fig. 2 (a). Namely, each row or column of the local features are averagely pooled into a vector that acts as the base of generating three different types of similarity tensors. Thus, by such a single-direction pooling processing, the local features Y^{LK} are pooled into $\hat{Y}^{LK} \in R^{K \times C}$.

We define three types of local similarity tensors (LSTs), named 2D, 3D, and 4D LSTs, to represent weak-to-strong spatial interrelationships between the local features of the query and gallery images. As shown in Fig. 2 (b) ~ (d), these tensors correspond to three patterns of spatial relationships, i.e., (1) weak interrelation that only keeps the relative structure information between query and gallery images, (2) mid-interrelation that keeps the structure information of either query or gallery images, and (3) strong interrelation that keeps the structure information of both query and gallery images, respectively.

For the weak interrelation presented by 2D LSTs, both query and gallery images lose their structure information. The single-direction pooling is first applied to local features of the query and gallery images. The 2D local similarity tensor t between query image q and gallery image g is then generated by:

$$t^{LK}(i, j) = D(\hat{Y}_q^{LK}(i), \hat{Y}_g^{LK}(j)), \tag{5}$$

where $i, j = 1, \ldots, K$, and D means a distance function.

For the mid-interrelation presented by 3D LSTs, the single-direction pooling is only applied to either of the images. You can choose to only keep the spatial information of the query image or gallery image. We found that keeping the gallery's information led to better results from preliminary studies, so in this paper, we focus our discussion on the case of keeping the gallery's information. The local feature of the query image is pooled into $\hat{Y}_q^{LK}$, and 3D LST t is generated by:

$$t^{LK}(u, v, i) = D(Y_g^{LK}(u, v), \hat{Y}_q^{LK}(i)), \tag{6}$$

where $u, v, i = 1, \ldots, K$.

For the strong interrelation presented by 4D LSTs, both the query and gallery images keep the structure information. A 4D LST t is generated by:

$$t^{LK}(u, v, i, j) = D(Y_g^{LK}(u, v), Y_q^{LK}(i, j)), \tag{7}$$

where $i, j, u, v = 1, \ldots, K$.

4. Similarity-CNN

In this section, we propose a lightweight CNN model named similarity-CNN (SCNN) to learn the local similarity between the query and gallery images based on LSTs. As a result, the local similarity score $S_{qg}^{\mathscr{L}}$ between the query image q and gallery image g can be obtained by the SCNN as noted below:

$$S_{qg}^{\mathscr{L}} = \mathrm{SCNN}(t_{qg}^{LK}). \tag{8}$$

We define a set of LSTs $\left[t_{qp}, t_{qn}\right]$ as the input of the model in the training procedure as shown in Fig. 4 (a). t_{qn} is a negative pair between a query image q and a negative image n, while t_{qp} means a positive pair between a query image q and a positive image p. $S_{qn}^{\mathscr{L}}$ and $S_{qp}^{\mathscr{L}}$ are the local similarity scores for t_{qn} and t_{qp}, respectively. The parameters of the network are learned by optimizing the triplet loss function, i.e.

$$L_d = \frac{1}{2}\max(0, m + S_{qp}^{\mathscr{L}} - S_{qn}^{\mathscr{L}}), \tag{9}$$

where m is a constant that represents a margin between the

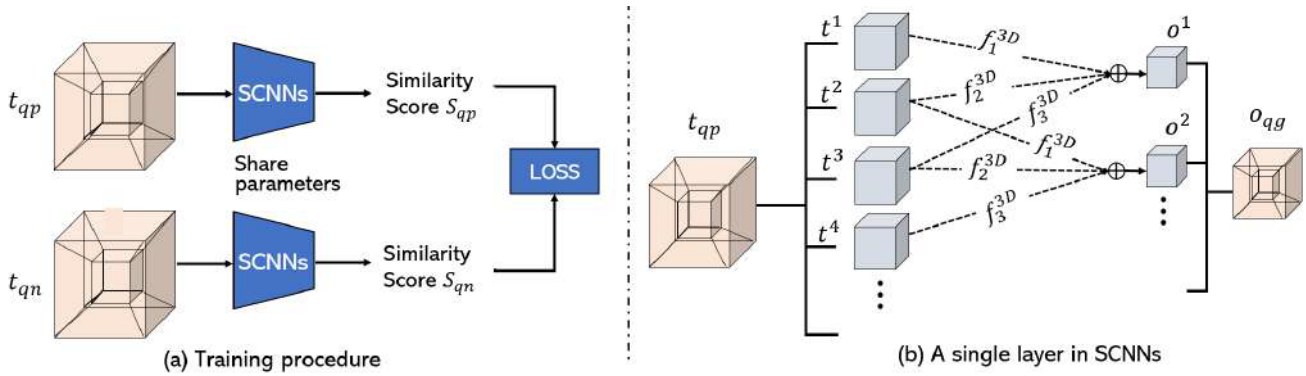

Fig. 4 Structure of SCNN. The lightweight CNN model SCNN is shown in (a), which learns the local similarity between query and gallery images. For 4D SCNN, we employ multiple 3D convolution/pooling operators f_w^{3D} to realize a 4D convolution/pooling, as illustrated in (b). A 4D tensor t_{qp} is composed of a set of 3D tensors $\{t^1, t^2, t^i, \ldots\}$. Similarly, the 4D tensor o_{qg} is composed of a set of 3D tensors $\{o^1, o^2, o^j, \ldots\}$. o^j is generated by the sum of the set $\{f_1^{3D}(t^i), f_2^{3D}(t^{i+1}), \ldots f_w^{3D}(t^{(i+w-1)}), \ldots\}$, where $f_1^{3D}(t^i)$ is a simple 3D convolution/pooling operation for t^i.

Table 1 Structure of SCNNs. The 2D SCNN is used for 2D or 3D LST input, and the 4D SCNN is used for 4D LST input.

2D SCNN				4D SCNN		
Type	Filter Shape	Input Size of 2D LST	Input Size of 3D LST	Type	Filter Shape	Input Size of 4D LST
2D Conv	[3×3,32]×3	K×K×1	K×K×K	4D Conv	[3×3×3×3,32]×3	K×K×K×K
2D Max Pool	[2×2]	K×K×32	K×K×32	4D Max Pool	[2×2×2×2]	K×K×K×K×32
2D Conv	[3×3,64]×2	$\frac{K}{2}\times\frac{K}{2}\times 32$	$\frac{K}{2}\times\frac{K}{2}\times 32$	4D Conv	[3×3×3×3,64]×2	$\frac{K}{2}\times\frac{K}{2}\times\frac{K}{2}\times K\times 32$
2D Max Pool	$[\frac{K}{2}\times\frac{K}{2}]$	1×1×64	1×1×64	4D Max Pool	$[\frac{K}{2}\times\frac{K}{2}\times\frac{K}{2}\times\frac{K}{2}]$	1×1×1×64
Norm	-	1×1×64	1×1×64	Norm	-	1×1×1×64
FC	64 ×1	1×1×64	1×1×64	FC	64 ×1	1×1×1×64

positive and negative pair.

The back propagation can be calculated as

$$\frac{\partial L_d}{\partial S_{qp}^{\mathscr{L}}} = \begin{cases} \frac{1}{2} & if\ m + S_{qp}^{\mathscr{L}} - S_{qn}^{\mathscr{L}} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{10}$$

$$\frac{\partial L_d}{\partial S_{qn}^{\mathscr{L}}} = \begin{cases} -\frac{1}{2} & if\ m + S_{qp}^{\mathscr{L}} - S_{qn}^{\mathscr{L}} > 0 \\ 0 & \text{otherwise.} \end{cases} \tag{11}$$

We adopt a two-branch siamese network to implement the triplet loss, where the two branches share the same parameters. We use a 2D SCNN for 2D or 3D LST input, and a 4D SCNN for 4D LST input, which are all composed of two convolution blocks and one fully connected layer. The network input varies according to the different forms of LSTs. In the case of 2D LSTs, the network takes an input of spatial size $K \times K$ and one channel, and in the case of 3D LSTs, it takes the $K \times K$ as the spatial shape and K as the channel number. The details of network configurations are summarized in Table 1.

Inspired by [38], we implement a 4D module with 4D convolution or pooling in each block. As shown in Fig. 4 (b), the 4D module can be implemented by multiple 3D modules. Namely, the input 4D tensor $t_{qg} \in R^{K\times K\times K\times K}$ can be calculated by a set of 3D tensor $t^i \in R^{K\times K\times K}$. Similarly, the output tensor $o_{qg} \in R^{K'\times K'\times K'\times K'}$ can be calculated by a set of 3D tensor $o^j \in R^{K'\times K'\times K'}$, where K' means the spatial resolution of output tensor o_{qg}. We define f_w^{3D} as a 3D module with 3D convolution or pooling. With the kernel size (W, W, W) and stride 1, the o^j can be calculated as

$$o^j = f_1^{3D}(t^j) + f_2^{3D}(t^{j+1}) + \ldots + f_W^{3D}(t^{j+W-1}) \tag{12}$$

5. Implementation Detail

This section introduces setups used in experiments, such as training/test datasets and feature extraction networks. In addition, we present the pipeline to demonstrate the entire experimentation process.

5.1 Datasets Used in Experiments

- **Retrieval-SfM-120k** [29] totally includes 117,365 images from Flicker filmed in 713 popular cities and landmarks worldwide. There are 91,642 images with 181,697 queries for training and 6,403 images with 1,691 queries for validation.
- **Retrieval-SfM-30k** [39] is a compact version of Retrieval-SfM-120k. There are 22,156 images with 5,974 queries for training and 6,403 images with 1,691 queries for validation.

- **Oxford5k** [40] includes 5,063 images filmed in 11 landmarks and provides 5 query images for each places. This dataset is especially used for test.
- **Paris6k** [41] includes 6,932 images for 12 landmarks. Same as Oxford5k, it is particularly used for test.
- **CUB** [42] includes 200 classes of birds with 11,788 images in total. There are 5,994 images for training and 5,794 images for test.

5.2 Experiment Pipeline

As shown in Fig. 1, the experiment pipeline is composed of three procedures. First, the local features for the query, positive and negative images are extracted by a fine-tuned retrieval CNN with a triplet loss [29]. A trainable whitening layer [29] is added in some experiments to accelerate training convergence. Then, an LST pair, i.e. (t_{qp}, t_{qn}), is generated from the local features in the manner described in Sect. 3.3. Finally, the LST pair is fed into the SCNN to produce the local similarity score.

In the test phase, we retain the original size (i.e., without cropping) of the query image from Oxford5k and Paris6k in order to extract more information. For a fair comparison, we employ the same setting for all four global similarity-based methods. For evaluation, we employ mean average precision (mAP) on Oxford5k, Parsi6k, and recall@K on the CUB dataset.

5.3 Training Details

We implement all experiments in this work using the PyTorch framework [43] and train all the models on three NVIDIA GeForce GTX 1080 Ti GPUs and one NVIDIA GeForce RTX 3090Ti. For all the experiments, we train the models using Stochastic Gradient Descent (SGD) with the epoch number of 100, momentum of 0.9, weight decay of 5×10^{-4}, and a batch size of 16.

6. Result and Discussion

We conduct through experiments to seek the best practice for the proposed local similarity-based method, and compare its performance with that of conventional global similarity-based methods.

6.1 Best Practice of LST

We define different forms of LSTs by varying the block L for local feature extraction, the spatial shape K of local regions and the pattern of spatial relationships between the query and gallery images. Among them, the first two are related with local region information. We quantitatively and thoroughly analyze the impact of each factor on the retrieval performance and identify the best practice for them.

Ablation studies of single-direction pooling We first confirm the configuration of the single-direction pooling. As introduced in Sect. 3.3, single-direction pooling has two choices: row-wise pooling or column-wise pooling. To assess the impact of the pooling direction, we run retrieval experiments on 2D and 3D LSTs with $L = 5$ and $K = 7$ settings and use ResNet101 as the backbone. The results on Oxford5k are shown in Table 2. Row-wise and column-wise poolings produce highly similar results, which means that the pooling direction insignificantly affects the final retrieval accuracy. Thus, in the following experiments, we narrow our discussion only to the case of using row-wise pooling.

Table 2 Results of single-direction pooling on Oxford5k: Pooling direction insignificantly affects the final retrieval accuracy.

Pooling direction	2D	3D
Row-wise	79.41%	78.22%
Column-wise	79.42%	78.21%

Table 3 Statistics of 4D LSTs on Oxford5k in each block.

Block	Standard Deviation σ	Mean μ
Block 1	0.00039	0.9926
Block 2	0.00059	0.9849
Block 3	0.00055	0.9866
Block 4	0.00200	0.9248
Block 5	0.01420	0.6366

Influence of extracted block We evaluate the performance of the LSTs using local features from different blocks. Using ResNet 101 as the backbone, we extract local features from Block 1 to Block 5 and set the spatial shape to $K = 7$. In Fig. 5, we plot accuracy for five blocks on Oxford5k and Paris6K, using 2D, 3D, 4D, and GeM (a global similarity-based method). The same trend can be observed across the different datasets, and moreover, as the block gets deeper, the accuracy shows a noticeable improvement. The worse performance is related to Block 1 (mAP $\approx$ 0.096), Block 2 (mAP $\approx$ 0.136), and Block 3 (mAP $\approx$ 0.290), which confirms that low-level image information like edge or blobs is not sufficient to the local similarity evaluation. In contrast, Block 4 and Block 5 present a reasonable accuracy, since these blocks include high-level image information about the entire object.

We conduct a statistical analysis of 4D LSTs on the Oxford5k to further explain this phenomenon. As each element of an LST is a similarity measure for the corresponding region of the query and gallery images, a similar region produces a high score, while a dissimilar region produces a low score. As discussed above, the shallower blocks (e.g., Block 1, 2, or 3) extract local features, which leads to poor discrimination because those local features are commonly included in all images. So, high similarity scores should be obtained for these shallower blocks regardless of whether the LSTs are t_{qp} or t_{qn}. We calculate the mean of standard deviation σ and the average μ of the LST for each image pair in different blocks as shown in Table 3. It is easy to see that the first three blocks have extremely low σ (around 10^{-4}) and high μ (the upper limit is 1). That means most image pairs look the same in terms of the similarity score, and a positive pair and a negative pair cannot be distinguished. On

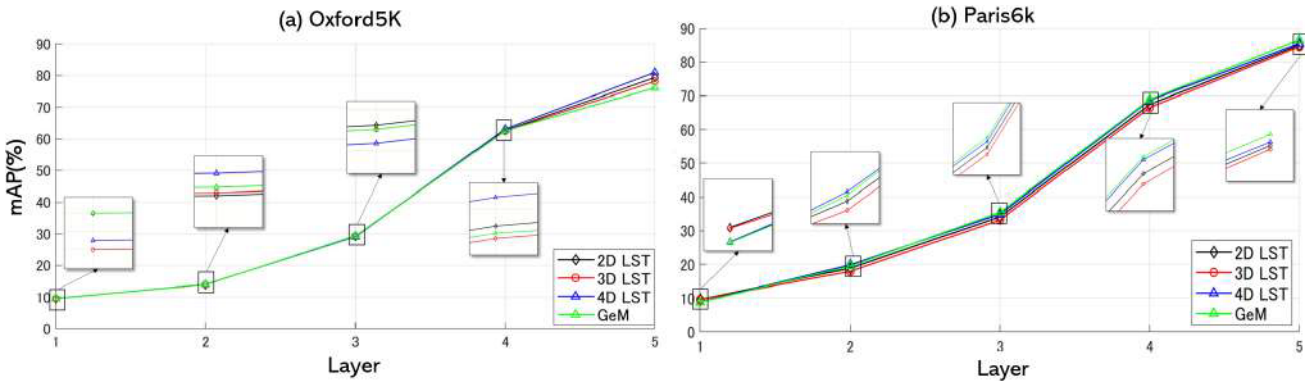

Fig. 5 Evaluation of the influence of extracted blocks. The accuracy for five blocks on Oxford5k and Paris6K using 2D, 3D, 4D, and GeM (a global similarity-based method) shows that as the block gets deeper, the accuracy has a noticeable improvement. The same trend can be observed across the different datasets.

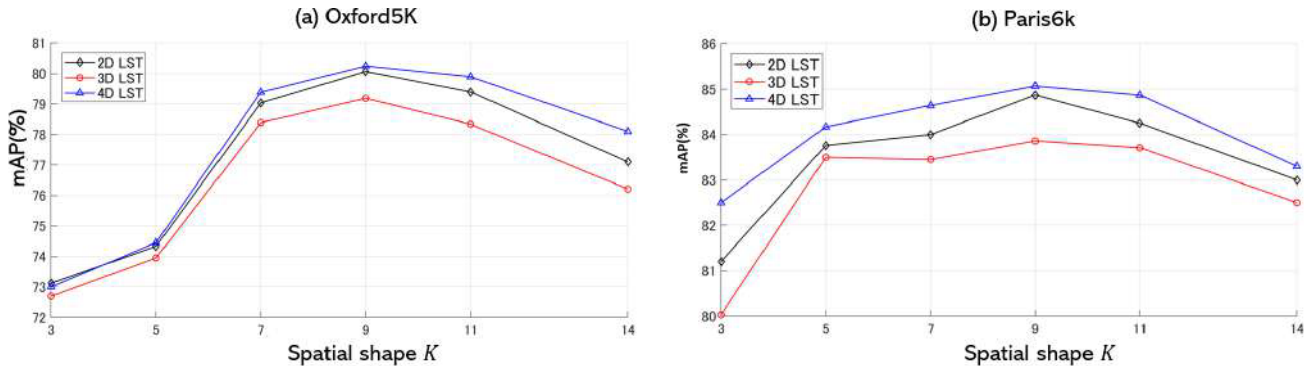

Fig. 6 Influence of the local region size. The mean average precision varies as K gets bigger using three patterns of spatial relationships related to Block 5. The best performance appears at $K = 9$ in all the experiments.

the other hand, Block 4 and Block 5 have higher σ (almost 10 or 100 times larger) and lower μ than shallower blocks, which support that the deeper blocks certainly bring higher distinguishability.

Influence of local region size Since the information included by a local region is different depending on the region size, we want to investigate through experiments what size of local regions most positively affects image retrieval. As mentioned in Sect. 3.2, the spatial shape K of local features can be used to control the local region size. We conduct experiments by setting values 3, 5, 7, 9, 11, and 14, on Oxford5k and Paris6k.

Figure 6 shows how the mean average precision varies as K gets bigger, using three types of spatial relationships related to Block 5. The best performance appears at $K = 9$ in all the experiments. To further confirm the validity of this value, we investigate the effect of different spatial shapes on each individual query image. Specifically, we select the optimal spatial shape K^* for each query, by comparing its average precision on different values of K, and plot the results in Fig. 7 as a bar graph, where each bin indicates the number of query images that present the highest average precision. As shown in Fig. 7, the optimal spatial shape K^* may differ depending on images, since each image has different texture, object size, etc. that directly influence the value of K^*. We choose the statistically optimal shape $K = 9$. This value precisely consist with that we described above.

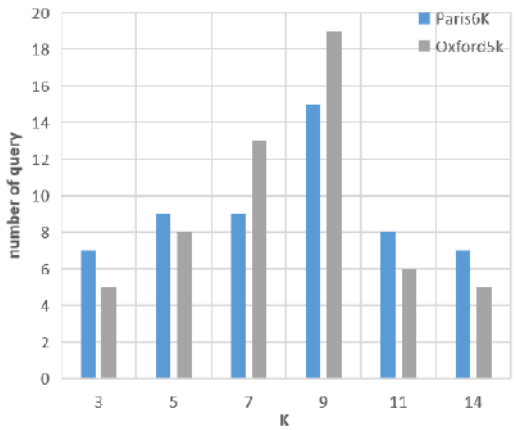

Fig. 7 Statistical information of LSTs in each local region size. Each bin represents the number of query images that present the highest average precision in the spatial shape K.

Influence of spatial relationships In Sect. 3, we have defined three types of LSTs that correspond to three patterns of spatial relationships: weak, mid-, and strong interrelations.

Fig. 8 Some occluded query images from Oxford5k dataset, where the targets are occluded by something such as vegetation, road sign, people or other building more or less.

Table 4 Experimental results on three datasets with two backbones (VGG16 and ResNet101) as well as the setting of Block 5 for local feature extraction and the spatial shape $K = 9$.

Dataset	Backbone	Whitening	2D	3D	4D
mAP					
Oxford5k	ResNet101	no	79.40%	78.20%	81.00%
Oxford5k	VGG16	no	78.38%	77.62%	78.50%
Paris6k	ResNet101	no	85.00%	84.50%	85.47%
Paris6k	VGG16	no	81.64%	80.27%	82.27%
Oxford5k	ResNet101	yes	81.30%	79.81%	84.32%
Oxford5k	VGG16	yes	80.15%	78.94%	82.52%
Paris6k	ResNet101	yes	86.95%	86.20%	88.38%
Paris6k	VGG16	yes	85.29%	84.34%	87.42%
Recall@10					
CUB	ResNet101	no	86.46%	85.72%	87.25%
CUB	VGG16	no	85.45%	84.55%	86.50%
CUB	ResNet101	yes	90.21%	88.50%	94.53%
CUB	VGG16	yes	89.50%	86.48%	92.60%

Since the observation obtained from previous experiments shows that LSTs from Block 5 achieve the best performance, we focus our attention on the spatial relationships of LSTs related only to Block 5. In addition, the spatial shape is set to $K = 9$. We implement the experiments on three datasets with two backbones: VGG16 and ResNet101. As shown in Table 4, among the three types of LSTs, the 3D LST presents the worst performance. The accuracy of 2D LST and 4D LST is very close, and 4D LST gives the highest accuracy in all experiments. So, the performance of different spatial layouts represented by LSTs from high to low is strong, weak and mid-interrelations.

Both the weak and strong interrelations represented by 2D and 4D LSTs satisfy the symmetry between the query and gallery images. Namely, the spatial relationship keeps even if the query and gallery images are exchanged. In contrast, since the mid-interrelation is produced by the query image with the weak interrelation and the gallery image with the strong interrelation, it does not satisfy this property. This kind of imbalance in information seems to negatively affect accuracy. So, the results suggest that keeping spatial layouts between the query and gallery images consistent has a positive impact on image retrieval based on local similarity. Moreover, the best performance of strong interrelationships makes it clear that richer structure information achieves better accuracy. In conclusion, from three evaluations, we find that the 4D LSTs produced on Block 5 in combination with the feature map size $K = 9$ give the best performance among all configurations.

6.2 Comparison to Global Similarity

This section compares the performance of our proposed 4D LSTs-based method with well-known global similarity-based methods. We first compare their performance in cases where the target objects in query images are occluded, and then evaluate their accuracy on some public datasets individually. Finally, we investigate the effectiveness of the combination of local similarity and global similarity.

Performance under occluded objects In this experiment, we compare the robustness against occlusion between the local and global similarities. As shown in Fig. 8, we rebuild Oxford5k dataset by only selecting occluded query images. Same with the original dataset, we chose 55 query images where the targets are obscured by something such as vegetation, a road sign, people, or another building, more or less. We select the latest four global similarity-based methods, SPoC [3], GeM [29], MAC [4], and R-MAC [30] to compare. We conduct the experiments on VGG16 and ResNet101. The results summarized in Table 5 show that LSTs-based method outperforms all global similarity-based methods by 8% higher accuracy on average. These results prove that local similarity performs much more robustly against occlusion than global similarity.

Comparison with the state-of-the-art We also compare the 4D LSTs-based method with state-of-the-art results on three public datasets: Oxford5k, Paris6k and CUB. Moreover, we investigate the combination method by simply averaging the global similarity score $S^{\mathscr{G}}$ and the local similarity score $S^{\mathscr{L}}$.

All the methods are evaluated on two widely used backbones, namely VGG16 and ResNet101. Note that we use the paper-provided pretrained weights [29] of GeM and MAC with whitening. We obtain the results for the remaining methods by implementing the code provided in Paper [29]. The highlighted results in Table 6 indicate the extent to which our methods outperform comparable global methods. Without whitening, all 4D LSTs-based methods achieve greater accuracy on Oxford5K than global similarity-based methods. Especially on the CUB dataset, the combination methods improve the accuracy by 10% at most. So, in conclusion, we say that the local similarity advances the accuracy and the combination of local and global similarities achieves the best result.

Table 5 Comparing the robustness against occlusion between the local and global similarities. The proposed 4D LSTs-based method outperforms all global similarity-based methods.

	VGG16	ResNet101
GeM [29]	35.48%	33.97%
SPoC [3]	19.51%	23.95%
MAC [4]	38.21%	42.18%
R-MAC [30]	31.32%	31.88%
4D LST (ours)	43.39%	48.01%

6.3 Limitation and Discussion

We discuss the computation cost incurred by the proposed method during LST and local similarity score generation.

Table 6 Comparing the proposed 4D LST-based method with state-of-the-art results on three public datasets (Oxford5k, Paris6k and CUB). In addition, the performance of a strategy that combines global and local similarity scores by averaging them is demonstrated. retrieval-SfM-30k is used to fine-tune the outcomes of global similarity-based methods without whitening, whereas retrieval-SfM-120k is used to fine-tune the whitening versions. All fine-tuned trainings involve paper-based code [29]. Only GeM and MAC with whitening used the weights provided by paper [29]. The highlighted findings indicate where our methods outperform their respective global methods.

Method	Backbone	Oxford5k(mAP)	Paris6k(mAP)	CUB(recall)			
				1	2	4	8
Without whitening							
GeM [29]	VGG16	77.57%	82.40%	59.54%	70.44%	78.57%	82.77%
GeM [29]	ResNet101	76.17%	86.53%	65.10%	76.40%	80.24%	85.45%
SPoC [3]	VGG16	68.40%	71.86%	54.14%	64.12%	72.51%	79.01%
SPoC [3]	ResNet101	69.69%	79.91%	63.79%	73.59%	78.14%	80.21%
MAC [4]	VGG16	76.84%	78.81%	60.15%	71.20%	78.18%	82.24%
MAC [4]	ResNet101	78.51%	83.76%	64.78%	75.49%	81.07%	85.90%
R-MAC [30]	VGG16	75.10%	81.64%	59.24%	69.89%	78.00%	82.59%
R-MAC [30]	ResNet101	79.3%	84.51%	64.53%	75.93%	81.21%	86.10%
OURS							
4D LST	VGG16	78.5%	82.27%	60.12%	69.54%	75.41%	84.34%
4D LST	ResNet101	80.21%	85.47%	62.31%	72.01%	79.95%	85.77%
4D LST+GeM	VGG16	79.79%	87.92%	67.01%	75.12%	82.11%	88.12%
4D LST+GeM	ResNet101	83.91%	88.37%	70.52%	81.34%	85.65%	90.38%
4D LST+SPoC	VGG16	81.28%	85.70%	62.54%	72.98%	79.52%	86.95%
4D LST+SPoC	ResNet101	80.65%	85.10%	68.21%	75.64%	83.54%	89.38%
4D LST+MAC	VGG16	81.02%	85.05%	67.86%	76.54%	83.34%	89.44%
4D LST+MAC	ResNet101	80.46%	84.86%	71.75%	81.95%	85.73%	91.14%
4D LST+R-MAC	VGG16	80.85%	85.22%	67.46%	75.47%	82.95%	88.37%
4D LST+R-MAC	ResNet101	80.62%	85.01%	71.07%	81.55%	85.24%	90.72%
With whitenning							
GeM [29]	ResNet101	87.78%	93.22%	71.35%	81.25%	87.95%	92.34%
SPoC [3]	ResNet101	76.73%	86.18%	63.79%	74.49%	83.14%	89.63%
MAC [4]	ResNet101	83.95%	92.85%	71.79%	81.19%	87.99%	92.42%
R-MAC [30]	ResNet101	83.23%	92.02%	71.55%	80.45%	87.50%	92.05%
OURS							
4D LST	ResNet101	84.32%	88.38%	64.67%	76.91%	85.30%	91.35%
4D LST+GeM	ResNet101	87.25%	93.57%	81.50%	88.97%	94.18%	97.10%
4D LST+SPoC	ResNet101	84.04%	89.41%	79.03%	87.52%	93.51%	97.05%
4D LST+MAC	ResNet101	88.64%	93.35%	81.55%	88.78%	93.87%	97.05%
4D LST+R-MAC	ResNet101	87.84%	92.80%	80.95%	88.34%	93.91%	97.01%
4D LST[1]	ResNet101	86.53%	92.25%	66.76%	77.46%	89.36%	92.75%
4D LST[1]+GeM	ResNet101	88.25%	94.65%	82.33%	89.56%	95.28%	97.70%

[1] This result is generated by the network that trained with retrieval-sfm-120k.

Table 7 Memory footprint and cost time. The cost time is the amount of time it takes for a pair of images to traverse a network and receive a similarity score. The cost time is measured on NVIDIA RTX 3090ti.

	2D LST	3D LST	4D LST	GeM
cost time	39.21ms	39.99ms	40.44 ms	38.59 ms
memory footprint (Paris6k)	3.50 GB	3.02 GB	6.78 GB	0.05GB
memory footprint (Oxford5k)	3.39 GB	3.00 GB	5.98 GB	0.04GB

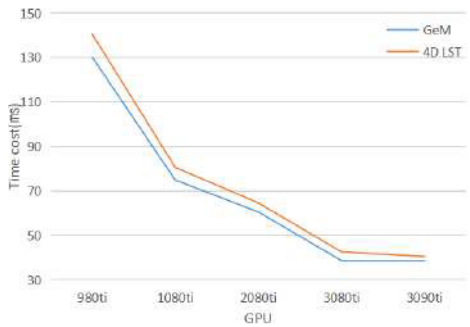

Fig. 9 The time cost of local and global similarity-based methods on the different GPU devices.

The memory footprint and time cost of local and global similarity are shown in Table 7. 4D LST takes the most memory space among local similarity-based approaches, whereas the time costs of the other two LSTs are comparable. Despite the 4D LST's evident advantage over other LSTs in terms of retrieval outcomes and the closer time cost, we maintain that it produced the best performance. Although local similarity requires more storage space than global similarity, the difference in computing speed is only 3 ms per pair. In addition, as demonstrated in Fig. 9, the speed gap between local and global similarity decreases as GPU performance increases. However, the exceptional performance of the proposed method on the occlusion dataset is clear according to the above discussion. This robustness against occlusion is of tremendous assistance for location recognition tasks, particularly for large search engines with powerful GPUs and ample memory. In the future, we will concentrate on lowering memory prices due to the memory's high utilisation.

7. Conclusion

In this paper, we propose a local similarity-based method (LST+SCNN) for image retrieval tasks. We conduct comprehensive experiments to evaluate the impact of local region information and spatial relationships among local regions. According to the experimental results, we get some meaningful observations: (1) As with global similarity, high-level image information advances the accuracy of the local similarity; (2) the optimal local region size is different for each image, but a statistically optimal region size can be chosen to positively affect the local similarity evaluation; (3) keeping the spatial layout between the query and gallery images consistent shows the positive impact on local similarity; (4) the more abundant structure information produces higher accuracy. Based on the above observations, we propose the best pipeline of our method, i.e. 4D LSTs from the last convolutional block together with the spatial shape 9. In addition, we propose a novel deep learning model called SCNN (similarity CNN) to enable local similarity evaluation using LSTs. Experiments confirm that compared with global similarity, the local similarity provides higher robustness when the query image is occluded, and the 4D LST+SCNN performs better than the three well-known global similarity-based methods on the Oxford5k dataset. Furthermore, the methods combining local and global similarity achieve the best performance on three public retrieval datasets.

Acknowledgements

This work is supported by the JSPS Grant-in-Aid for Scientific Research (C) (No.22K12103).

References

[1] A. Krizhevsky, I. Sutskever, and G.E. Hinton, "Imagenet classification with deep convolutional neural networks," Advances in Neural Information Processing Systems, pp.1097–1105, 2012.
[2] Y. Gong, L. Wang, R. Guo, and S. Lazebnik, "Multi-scale orderless pooling of deep convolutional activation features," ECCV, vol.8695, pp.392–407, 2014.
[3] A.B. Yandex and V. Lempitsky, "Aggregating local deep features for image retrieval," The IEEE International Conference on Computer Vision (ICCV), pp.1269–1277, Dec. 2015.
[4] A.S. Razavian, J. Sullivan, S. Carlsson, and A. Maki, "Visual instance retrieval with deep convolutional networks," ITE Transactions on Media Technology and Applications, vol.4, no.3, pp.251–258, 2016.
[5] B. Babenko, P. Dollar, and S. Belongie, "Task specific local region matching," 2007 IEEE 11th International Conference on Computer Vision, pp.1–8, 2007.
[6] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp.2921–2929, 2016.
[7] R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," Proc. IEEE International Conference on Computer Vision, pp.618–626, 2017.
[8] Z. Chen, Z. Kuang, W. Zhang, and K.-Y.K. Wong, "Learning local similarity with spatial relations for object retrieval," Proc. 27th ACM International Conference on Multimedia, MM '19, pp.1703–1711, 2019.
[9] Y. Rui, T.S. Huang, M. Ortega, and S. Mehrotra, "Relevance feedback: a power tool for interactive content-based image retrieval," IEEE Trans. Circuits Syst. Video Technol., vol.8, no.5, pp.644–655, 1998.
[10] A.W.M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-based image retrieval at the end of the early years," IEEE Trans. Pattern Anal. Mach. Intell., vol.22, no.12, pp.1349–1380, 2000.
[11] M.S. Lew, N. Sebe, C. Djeraba, and R. Jain, "Content-based multimedia information retrieval: State of the art and challenges," ACM Trans. Multimedia Comput. Commun. Appl., vol.2, no.1, pp.1–19, Feb. 2006.
[12] Y. Liu, D. Zhang, G. Lu, and W.-Y. Ma, "A survey of content-based

image retrieval with high-level semantics," Pattern Recognition, vol.40, no.1, pp.262–282, 2007.
[13] D.G. Lowe, "Object recognition from local scale-invariant features," Proc. 7th IEEE Conference on International Conference on Computer Vision, vol.2, pp.1150–1157, 1999.
[14] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp.886–893, 2005.
[15] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," IEEE International Conference on Computer Vision, vol.2, pp.1470–1470, 2003.
[16] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the fisher kernel for large-scale image classification," Computer Vision – ECCV 2010, vol.6314, pp.143–156, 2010.
[17] H. Jégou, F. Perronnin, M. Douze, J. Sánchez, P. Pérez, and C. Schmid, "Aggregating local image descriptors into compact codes," IEEE Trans. Pattern Anal. Mach. Intell., vol.34, no.9, pp.1704–1716, 2012.
[18] P. Wu, S.C.H. Hoi, H. Xia, P. Zhao, D. Wang, and C. Miao, "Online multimodal deep similarity learning with application to image retrieval," Proc. 21st ACM International Conference on Multimedia, pp.153–162, 2013.
[19] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky, "Neural codes for image retrieval," Proc. 12th European conference on computer vision, vol.8689, pp.584–599, Springer, 2014.
[20] C.-Q. Huang, S.-M. Yang, Y. Pan, and H.-J. Lai, "Object-location-aware hashing for multi-label image retrieval via automatic mask learning," IEEE Trans. Image Process., vol.27, no.9, pp.4490–4502, 2018.
[21] N. Garcia and G. Vogiatzis, "Learning non-metric visual similarity for image retrieval," Image and Vision Computing, vol.82, pp.18–25, 2019.
[22] E.J. Ong, S. Husain, and M. Bober, "Siamese network of deep fisher-vector descriptors for image retrieval," arXiv preprint arXiv:1702.00338, 2017.
[23] A. Gordo, J. Almazán, J. Revaud, and D. Larlus, "Deep image retrieval: Learning global representations for image search," Proc. 14th European conference on computer vision, vol.9910, pp.241–257, Springer, 2016.
[24] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "Netvlad: Cnn architecture for weakly supervised place recognition," Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp.5297–5307, 2016.
[25] Y. Kalantidis, C. Mellina, and S. Osindero, "Cross-dimensional weighting for aggregated deep convolutional features," European conference on computer vision, vol.9913, pp.685–701, Springer, 2016.
[26] A. Jimenez, J.M. Alvarez, and X. Giro-i-Nieto, "Class-weighted convolutional features for visual instance search," arXiv preprint arXiv:1707.02581, 2017.
[27] T.-T. Do, T. Hoang, D.-K.L. Tan, H. Le, T.V. Nguyen, and N.-M. Cheung, "From selective deep convolutional features to compact binary representations for image retrieval," ACM Trans. Multimedia Comput. Commun. Appl., vol.15, no.2, pp.1–22, June 2019.
[28] J. Xu, C. Shi, C. Qi, C. Wang, and B. Xiao, "Unsupervised part-based weighting aggregation of deep convolutional features for image retrieval," Proc. AAAI Conference on Artificial Intelligence, vol.32, no.1, 2018.
[29] F. Radenović, G. Tolias, and O. Chum, "Fine-tuning cnn image retrieval with no human annotation," IEEE Trans. Pattern Anal. Mach. Intell., vol.41, no.7, pp.1655–1668, 2019.
[30] G. Tolias, R. Sicre, and H. Jégou, "Particular object retrieval with integral max-pooling of cnn activations," arXiv preprint arXiv:1511.05879, 2015.
[31] B. Cao, A. Araujo, and J. Sim, "Unifying deep local and global features for image search," Computer Vision – ECCV 2020, ed. A. Vedaldi, H. Bischof, T. Brox, and J.M. Frahm, Cham, vol.12365, pp.726–743, Springer International Publishing, 2020.
[32] S. Hausler, S. Garg, M. Xu, M. Milford, and T. Fischer, "Patch-netvlad: Multi-scale fusion of locally-global descriptors for place recognition," Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.14141–14152, 2021.
[33] M. Yang, D. He, M. Fan, B. Shi, X. Xue, F. Li, E. Ding, and J. Huang, "Dolg: Single-stage image retrieval with deep orthogonal fusion of local and global features," 2021 IEEE/CVF International Conference on Computer Vision (ICCV), pp.11752–11761, 2021.
[34] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014.
[35] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," Proc. IEEE conference on computer vision and pattern recognition, pp.770–778, 2016.
[36] G. Huang, Z. Liu, L. Van Der Maaten, and K.Q. Weinberger, "Densely connected convolutional networks," Proc. IEEE conference on computer vision and pattern recognition, pp.2261–2269, 2017.
[37] M.D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," Proc. 12th European conference on computer vision, vol.8689, pp.818–833, Springer, 2014.
[38] S. Zhang, S. Guo, W. Huang, M.R. Scott, and L. Wang, "V4d: 4d convolutional neural networks for video-level representation learning," Proc. International Conference on Learning Representations, 2020.
[39] F. Radenović, G. Tolias, and O. Chum, "Cnn image retrieval learns from bow: Unsupervised fine-tuning with hard examples," Proc. 14th European Conference on Computer Vision, vol.9905, pp.3–20, 2016.
[40] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp.1–8, 2007.
[41] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Lost in quantization: Improving particular object retrieval in large scale image databases," Proc. IEEE Conference on Computer Vision and Pattern Recognition, pp.1–8, 2008.
[42] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD Birds-200-2011 Dataset," Tech. Rep. CNS-TR-2011-001, California Institute of Technology, 2011.
[43] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in Advances in Neural Information Processing Systems 32, ed. H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, pp.8024–8035, Curran Associates, Inc., 2019.

Longjiao Zhao received the M.S. degree in Information Science, from Nagoya University in 2018. She is currently a Ph.D. Candidate with the Graduate School of Informatics, Nagoya University.

Yu Wang received the M.S. degree in Information Science and Ph.D. degree in Engineering, from Nagoya University, in 2010 and 2013, respectively. Then he became an assistant professor with the College of Information Science and Engineering, Ritsumeikan University. He is currently an associate professor with Center for Information and Communication Technology, Hitotsubashi University. He is a member of IEEE.

Jien Kato received the M.E. and Ph.D. degrees in Information Engineering from Nagoya University in 1990 and 1993, respectively. Then she became an assistant professor at Toyama University. She was a visiting researcher at the University of Oxford from 1999 for one year. She became an associate professor at the Graduate School of Engineering of Nagoya University in 2000. She has been a professor at the College of Information Science and Engineering of Ritsumeikan University since 2018. Her research interests include object recognition, visual event recognition and machine learning. She is a member of IPSJ and JSAI, and also a senior member of IEICE and IEEE.

Yoshiharu Ishikawa is a Professor in the Graduate School of Informatics, Nagoya University. He received B.E., M.E., and Dr. Eng. degrees from University of Tsukuba in 1989, 1991, and 1995, respectively. His research interests include database system technologies, spatio-temporal databases, scientific databases, data mining, and Web information systems. He is a member of the Database Society of Japan, IPSJ, IEICE, JSAI, ACM, and the IEEE Computer Society.