Learning Spatial-context-aware Global Visual Feature Representation for Instance Image Retrieval

Zhongyan Zhang[1], Lei Wang[1], Luping Zhou[2], Piotr Koniusz[3,4]
University of Wollongong[1], University of Sydney[2], Data61♥CSIRO[3], Australian National University[4]
zhongyan@uow.edu.au, leiw@uow.edu.au, luping.zhou@sydney.edu.au, peter.koniusz@data61.csiro.au

Abstract

In instance image retrieval, considering local spatial information within an image has proven effective to boost retrieval performance, as demonstrated by local visual descriptor based geometric verification. Nevertheless, it will be highly valuable to make ordinary global image representations spatial-context-aware because global representation based image retrieval is appealing thanks to its algorithmic simplicity, low memory cost, and being friendly to sophisticated data structures. To this end, we propose a novel feature learning framework for instance image retrieval, which embeds local spatial context information into the learned global feature representations. Specifically, in parallel to the visual feature branch in a CNN backbone, we design a spatial context branch that consists of two modules called online token learning and distance encoding. For each local descriptor learned in CNN, the former module is used to indicate the types of its surrounding descriptors, while their spatial distribution information is captured by the latter module. After that, the visual feature branch and the spatial context branch are fused to produce a single global feature representation per image. As experimentally demonstrated, with the spatial-context-aware characteristic, we can well improve the performance of global representation based image retrieval while maintaining all of its appealing properties. Our code is available at https://github.com/Zy-Zhang/SpCa.

1. Introduction

Given a query image, the purpose of instance-level image retrieval is to search and retrieve the images containing the identical object described by the query from a large-scale image dataset. In this task, visual feature representations of images play a crucial role in measuring the similarities between a query and candidate images. A variety of handcrafted feature-based methods [14, 38, 2] have been proposed to significantly improve the performance in the past two decades. Recently, due to the development of deep learning technologies, deep feature representations have been overtaking the position of conventional handcrafted ones and bringing great progress in the task of instance image retrieval [1, 3, 17, 28, 15].

Generally, deep feature representations used in instance image retrieval can be categorized into two types. One type is global feature representation, which describes the visual content of an image as a whole. As a multi-dimensional vector, it can be efficiently used to measure the similarity of two images, say, via Euclidean distance or cosine similarity. For a retrieval task, the total number of global feature representations is just the size of image database, and they can be pre-extracted and economically stored for use. In addition, one global feature representation per image works well with the classic data structures designed for searching. The other type is local descriptors, which describe the local information within an image and collectively reflect the spatial information of the visual cues in an image. In image retrieval, they are important for conducting geometric verification to confirm if two images truly match or not. However, the total number of local descriptors per image could be large (*e.g.*, $1,000$) and the verification involves non-trivial computation, making this process expensive in computational cost and memory footprint.

This situation leads to a wide use of "two-stage" paradigm in instance image retrieval [23, 3, 17]. An initial retrieval result is firstly obtained via the global feature representation. After that, a re-ranking step utilizes the local descriptors to refine a small number of top-retrieved images. Nevertheless, this two-stage paradigm not only results in two separate procedures [26, 10] but also considerably increases retrieval time and memory expense for practical retrieval tasks [18], due to the presence of the local descriptor-based spatial verification step. Therefore, it will be highly valuable to make ordinary global image representations spatial-context-aware by considering their appealing properties of algorithmic simplicity, low memory cost, and being friendly to data structures.

To achieve this, we propose a novel feature learning framework to effectively embed spatial context information into global feature representation of images. Doing so will help to boost the retrieval performance of global feature representation, improving its efficacy when the local descriptor based re-ranking becomes costly or infeasible.

Specifically, two types of information are extracted in our framework. One is conventional visual information obtained by a visual feature branch in a CNN backbone. The other one is spatial context information that describes *for each local descriptor learned in CNN, what kind of surrounding local descriptors are* and *how they spatially distribute on a feature map*. To obtain this information, we develop a spatial context branch that operates in parallel to the visual feature branch in CNN. This branch consists of two modules called online token learning and distance encoding. The module of online token learning addresses the "what kind" issue. A set of semantic tokens that could be regarded as anchors in the space of local descriptors is learned in an online manner. Comparing the tokens to the visual words in a visual dictionary, each local descriptor can be uniquely labelled via soft coding as a token-based identification. The module of distance encoding cares about the "spatial distribution" issue. A probability transition based encoding is devised to reflect the relative spatial distance between each pair of local descriptors, so as to capture their spatial distribution information in a feature map. After that, the token-based identification and the spatial distribution information are integrated to produce a spatial context clue for each local descriptor. With both visual and spatial information, a feature fusion operation is conducted to fuse them together to generate spatial-context-aware local descriptors. Finally, a global pooling followed by a whitening layer is added to embed the context-aware local descriptors into a global feature representation for each image.

Our contributions are summarised as follows:

i. We propose an end-to-end feature learning framework to characterise and embed spatial context information into the process of information processing and extraction in CNN. It makes global feature representations become more capable for instance image retrieval.

ii. To implement the framework, we design a spatial context branch. With its online token learning module, the types of local descriptors are identified and easily compared. With the distance encoding module, the information about the spatial distribution of different types of local descriptors around a given descriptor is obtained.

iii. To verify the efficacy of the proposed framework, we conduct extensive experiments on instance image retrieval benchmark datasets, $\mathcal{R}$Oxford and $\mathcal{R}$Paris, with one million distractors. As demonstrated, our global feature representations effectively improve the performance of instance image retrieval, making global representation a more competitive option for practical tasks.

2. Related Work

In this section, we briefly review deep instance image retrieval, including local and global feature learning methods.

2.1. Global features

Global feature has been widely used to represent the content of images for a wide range of computer vision tasks [21, 13, 8, 36, 37, 12, 11]. Due to its compactness and efficiency, a variety of global methods [24, 32, 35, 15] have been proposed to learn discriminative feature representations via deep models in the instance image retrieval community. Some of the studies focus on applying more effective loss functions, including triplet losses [29], list-wise losses [22], and classification losse s [5] to train the model. Others explore how to pool the entire feature maps into a compact feature vector while maintaining their discrepancies. For instance, weighted-sum-pooling (CroW) [9], regional-max-pooling (R-MAC) [29], and GeM pooling [20]. Recently, several methods try to modify the training model architecture to capture more useful information during the training process. SOLAR [15] combines normal training with second-order information by adding self-attention modules in the conventional CNN backbones. Token [32] jointly learns a set of tokens and the corresponding visual representations to mimic the conventional dictionary learning plus feature encoding scheme to represent images. Some other studies [32, 35] propose multiple branches for CNN models to mine more complementary information and fuse them into the final feature vectors. We follow the global feature type to learn compact features to conduct retrieval.

2.2. Local features

Deep local descriptors [28, 30, 33, 17] have been extensively researched and made significant progress in instance image retrieval [4]. The key to their success lies in the spatial information these local descriptors hold, and it is usually utilized by kernel alignment aggregation (*e.g.*, ASMK [27]) or geometric verification (*e.g.*, RANSAC [6]) for image similarity measure. In recent studies, several methods [26, 10] individually train a matching model to directly compare the similarity of two images (or their pre-extracted local descriptors) in a data-driven manner. However, they need more inference time and memory footprint. Therefore, they are usually applied to only top k retrieved images obtained by a global retrieval method as a reranking.

2.3. Learning local and global features jointly

Driven by the complementary properties of global and local features, jointly learning the two types of features

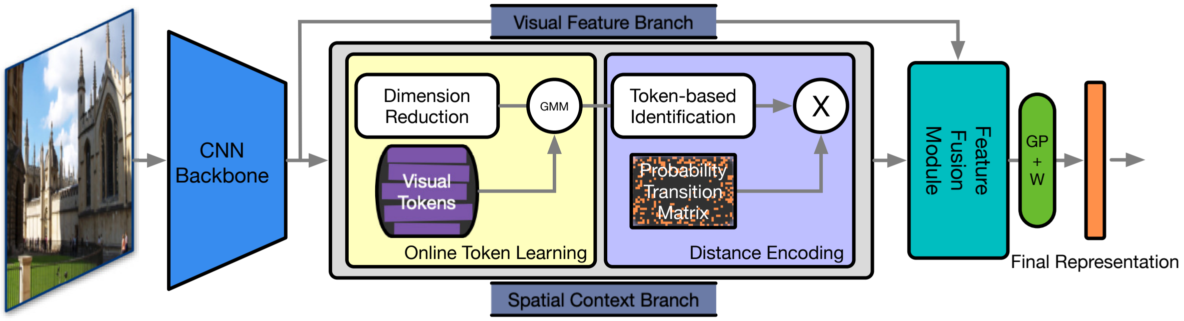

Figure 1. Overview of our framework. After an initial feature extraction by a CNN backbone, two branches of "Visual Feature Branch" and "Spatial Context Branch" are used to extract visual and spatial context information. After that, a cross-attention module is used to fuse the visual feature and the corresponding spatial context feature to generate spatial-context-aware visual representations for images. In particular, "GMM" denotes the Gaussian Mixture Model process, and "GP+W" refers to global pooling followed by whitening.

becomes a straightforward idea. Recently, several studies demonstrate the advantages of learning features in this manner. DELG [3] creatively unifies the two separate learning processes into one training framework. However, even though the training procedure is unified, the learning processes of the two feature types are still somehow irrelevant. To further explore this approach, DOLG [35] and DALG [25] firstly separate the feature extractor into two branches to learn the local and global features and then fuse them via a manual or learnable way to obtain compact feature representations for images. Both of them try to extract individual visual information from the two types of features to make them complementary to each other. However, the powerful spatial context information held by local features is neither exploited nor embedded into the final feature representations. In this work, we argue that spatial context information is more beneficial than visual information from local descriptors to the final global feature representations for instance image retrieval. To this end, we propose a novel feature learning framework to effectively embed the spatial context information into final global features.

3. The Proposed Method

An overview of our framework is in Figure 1. Given an image I, a conventional CNN backbone is used to generate a feature map $\mathbf{V} \in \mathbb{R}^{D\times H\times W}$, in which each spatial position corresponds to a D-dimensional local visual descriptor. Besides, a spatial context branch is split from the feature map to extract the spatial context information of all the descriptors, which is represented in the form of $\mathbf{S} \in \mathbb{R}^{C\times H\times W}$ with C being the dimensions. After that, for each local descriptor, its two kinds of information are fused together, say, with a cross-attention module or a simple concatenation operation. After the fusion, a global pooling and a fully-connected layer are used to generate the final global spatial-context-aware feature $\mathbf{f}_{spca}$ to represent the image. In the following, we focus on describing the spatial context branch since the visual branch follows its original design in CNN backbones.

3.1. Spatial Context Branch

In the spatial context branch, there are two main issues to be addressed for each spatial position on the feature map: 1) what kinds of local descriptors are there in its surrounding region (*i.e.*, identification task); 2) how they spatially distribute on the feature map (*i.e.*, spatial distribution task). We now introduce how we address the two issues orderly.

Token-based Identification. For existing local feature-based retrieval methods [28, 30, 33], learning tokens to capture discriminative visual patterns in a dataset has proven reliable to identify local descriptors. With the tokens, the encoded features could become more robust against illumination change, the information loss due to occlusion, and the adverse impact of the background in an image. In our framework, we generally follow this "tokens plus encoding" pipeline to identify local descriptors. However, in the literature, the tokens are usually learned by clustering (*e.g.*, by k-means) a sampled subset of local descriptors obtained from an image dataset in an offline manner. This way separates the process of token learning from encoding, potentially causing a discrepancy between the training and the inference stages. In addition, using a sampled subset of local descriptors could lead to learning biased tokens since the information available in an image dataset is not fully utilized. To improve this situation, we perform token learning via an online mini-batch based Gaussian Mixture Model (GMM) process. This helps to adequately utilize all the local descriptors to learn tokens and makes our framework fully end-to-end trainable.

Specifically, we model the distribution of local descriptors by a K-component Gaussian mixture model $p(\mathbf{v}|\Theta) = \sum_k \alpha_k N(\mathbf{v}; \boldsymbol{\mu}_k, \Sigma_k)$, where $\mathbf{v}$ denotes the a local descriptor, while α_k, $\boldsymbol{\mu}_k$, and $\Sigma_k = \sigma_k^2 I_D$ denote the prior probability, the mean, and the (spherical) covariance matrix for component k, respectively. In this case, token learning boils down to the parameter estimation for this Gaussian mixture model. At the beginning, $\boldsymbol{\mu}$ and σ^2 are randomly generated and α is set as $1/K$. After that, we conduct the Expectation and Maximization (EM) process for all the local descriptors within each mini-batch. To be concise, we show the key results below and the details can be found in the literature [16].

Within each mini-batch, for the E-Step, the probability that the ith local descriptor $\mathbf{v}_i$ belongs to the kth component can be expressed as

$$\omega_{ik} = \frac{\alpha_k N(\mathbf{v}_i; \boldsymbol{\mu}_k, \Sigma_k)}{\sum_{j=1}^{K} \alpha_j N(\mathbf{v}_i; \boldsymbol{\mu}_j, \Sigma_j)} \tag{1}$$

For M-Step, the prior probability, the mean and the covariance are updated as

$$\begin{aligned} \alpha_k &\leftarrow \alpha_k + \beta\left(\frac{\sum_i \omega_{ik}}{HW} - \alpha_k\right) \\ \boldsymbol{\mu}_k &\leftarrow \boldsymbol{\mu}_k + \beta\left(\frac{\sum_i \omega_{ik}\mathbf{v}_i}{\sum_i \omega_{ik}} - \boldsymbol{\mu}_k\right) \\ \sigma_k^2 &\leftarrow \sigma_k^2 + \beta\left(\frac{\sum_i \omega_{ik}\|\mathbf{v}_i - \boldsymbol{\mu}_k\|^2}{\sum_i \omega_{ik}} - \sigma_k^2\right), \end{aligned} \tag{2}$$

where HW is the number of local descriptors obtained from an image, and β is a momentum to make the estimation process more stable in the mini-batch clustering [16]. One E-Step and M-Step are conducted in each iteration during the training.

After the above process, $(\boldsymbol{\mu}_1, \boldsymbol{\mu}_2, \cdots, \boldsymbol{\mu}_K)$ represents the learned tokens. $\boldsymbol{\omega}_i = (\omega_{i1}, \omega_{i2}, \cdots, \omega_{iK}) \in \mathbb{R}^K$ obtained by Eq. (1) can be used as the token-based identification for local descriptor $\mathbf{v}_i$ since it represents the posterior probability that this descriptor belongs to each of the K components.

Spatial Distribution. To extract spatial distribution information for a local descriptor $\mathbf{v}_i$, the distances of this descriptor to all the descriptors $V = \{\mathbf{v}_j, j = 1, 2, \ldots, HW\}$ presented on the same feature map should be effectively measured and collected. To achieve this, simply using the coordinate distances among them is inadequate. We utilize a probability transition score obtained by applying random walk on the graph constituted by all the local descriptors as the nodes to measure the distances. By doing so, more possible paths among the local descriptors will be considered and the number of random walk steps could also be used to control the spatial scale in the distance measurement. Inspired by the literature [11], we perform the spatial distribution extraction as an aggregation of the encoded distances between $\mathbf{v}_i$ and all $\mathbf{v}_j \in V$. By "encoded," we mean that the distance is measured by a probability transition process as explained below.

Given an undirected graph with an affinity matrix A, the distance of its nodes v and u can be encoded by a probability transition score ζ obtained via l steps of random walk

$$\zeta(v, u)^{(l)} = (M^l)_{vu}, \; M^1 = AD^{-1}, \; M^l = M^{l-1} \cdot M, \tag{3}$$

where M^1 denotes the random walk matrix, D^{-1} refers to the normalization matrix of the graph, and l is the number of random walk steps. Based on this, the distance of a node v from all nodes u in a target set S can be represented by an aggregation operation as

$$\zeta(v, S)^{(l)} = \text{agg}(\{\zeta(v, u)^{(l)} | u \in S\}) \tag{4}$$

Linking the above result to our case, the undirected graph consists of all the local descriptors V in an image, and they collectively form the target set S in this distance encoding process. In addition, we compute the affinity matrix A based on the coordinate distance on the feature map as

$$A_{ij} = \exp(-\text{co\_dist}(\mathbf{v}_i, \mathbf{v}_j)), \; (1 \le i, j \le HW), \tag{5}$$

where co_dist$(\cdot, \cdot)$ denotes the Euclidean distance between the coordinates of two local descriptors. After that, the encoded distance can be obtained by applying Eq. (3). Furthermore, we enrich Eq. (4) by considering the learned token-based identification (*i.e.*, the posteriori probability $\boldsymbol{\omega}$ obtained in Eq. (1)), and this leads to an instantiated "agg" operation in Eq. (4) to produce the spatial context feature for local descriptor $\mathbf{v}_i$.

$$\mathbf{s}_i^{(l)} \triangleq \zeta(v_i, S)^{(l)} = \sum_{j=1}^{HW} \zeta(\mathbf{v}_i, \mathbf{v}_j)^{(l)} \boldsymbol{\omega}_j. \tag{6}$$

Recall that the token-based identification $\boldsymbol{\omega}_i$ can be treated as a vector of probability transition scores between local descriptor $\mathbf{v}_i$ and each of the learned tokens $\boldsymbol{\mu}$. By combining the encoded distance and the token-based identification, the spatial context feature $\mathbf{s}_i$ in Eq. (6) essentially reflects the probability transition scores between node $\mathbf{v}_i$ and the tokens $\boldsymbol{\mu}$ when regarding the spatial structure of the given $H \times W$ feature map as a (static) transition matrix. This is illustrated in Figure 2. That means the spatial distribution information is used to control the extraction of the spatial context features. In this sense, the spatial distribution information is effectively embedded into the spatial context features. At last, we define the final spatial context feature for the local descriptor $\mathbf{v}_i$ obtained through an l-step random walk as

$$\mathbf{s}_i = \mathbf{s}_i^{(1)} \oplus \mathbf{s}_i^{(2)} \oplus \cdots \oplus \mathbf{s}_i^{(l)}, \tag{7}$$

where $\oplus$ denotes concatenation operation. The concatenation of the features obtained after each of the l steps considers the spatial context information at various scales.

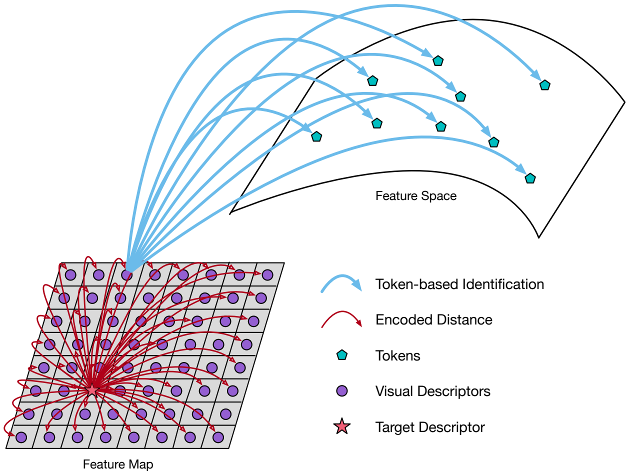

Figure 2. Spatial context feature generation. See the detailed description after Eq. (6).

3.2. Fusion of Visual and Spatial Information

With both visual and spatial context information available, a proper fusion scheme is needed to make them complement to benefit the final feature representation. To accomplish it, various feature fusion schemes could be used, including concatenation, orthogonal fusion [35], Hadamard product, or cross-attention fusion. We utilize a cross-attention module for information fusion in our method since it is commonly used to align features from different modalities in the recent literature. Meanwhile, we provide an ablation study on all the four fusion schemes in the experimental part. For the cross-attention scheme, the spatial context feature $\mathbf{S} = (\mathbf{s}_1, \mathbf{s}_2, \cdots, \mathbf{s}_{HW})$ is projected into the same space as the visual feature $\mathbf{V}$ with a linear projection layer. After that, both of them are flattened into sequences and mapped into Queries ($\mathbb{Q}$), Keys ($\mathbb{K}$), and Values ($\mathbb{V}$), respectively, to put through into the cross-attention layer:

$$\begin{aligned} &\mathbb{Q} = proj(\mathbf{V}), \mathbb{K} = proj(\mathbf{S}), \mathbb{V} = proj(\mathbf{S}) \\ &\mathbf{F}^*_{spca} = softmax\left(\frac{\mathbb{Q} \cdot \mathbb{K}^\top}{\sqrt{D}}\right) \cdot \mathbb{V} \\ &\mathbf{F}_{spca} = \mathbf{F}^*_{spca} + \mathrm{MLP}(\mathbf{F}^*_{spca}) \end{aligned} \tag{8}$$

where MLP denotes a feed-forward layer and $\mathbf{F}_{spca}$ denotes the spatial-context-aware feature sequence. After that, the feature sequence is rearranged as a feature map, followed by a global pooling and a fully-connected layer to generate a global feature representation vector $\mathbf{f}_{spca}$.

3.3. Objective Function for Training

Following previous state-of-the-art work [3, 35, 32], we train our model on a dataset with object class annotations and utilize ArcFace margin loss [5] with learnable parameters $\hat{\mathbf{W}} \in \mathbb{R}^{D \times N}$ to train the whole model:

$$L = -\log\left(\frac{\exp(\gamma \times AF(\hat{\mathbf{w}}_t^\top \hat{\mathbf{f}}_{spca}, 1))}{\sum_n \exp(\gamma \times AF(\hat{\mathbf{w}}_n^\top \hat{\mathbf{f}}_{spca}, y_n))}\right) \tag{9}$$

where $\hat{\mathbf{w}}_t$ denotes the tth column of $\hat{\mathbf{W}}$ and $\hat{\mathbf{f}}_{spca}$ is the L_2-normalized $\mathbf{f}_{spca}$. y is the one-hot label vector and t is the index to indicate the true class (*i.e.*, $y_t = 1$). n is the total number of classes in the dataset. γ is a scaling factor. $AF(\cdot)$ refers to the ArcFace cosine similarity and it is calculated as

$$AF(s, c) = \begin{cases} \cos(acos(s) + m), & \text{if } c = 1 \\ s, & \text{if } c = 0 \end{cases} \tag{10}$$

where s is the cosine similarity between $\hat{\mathbf{w}}_t$ and $\hat{\mathbf{f}}_{spca}$, m is the ArcFace margin and c indicates if this is the true class.

4. Experimental Result

4.1. Dataset and Evaluation Metric

Our model is trained on "Google Landmark v2 clean" dataset, which is a clean subset of Google Landmark v2 [31]. It contains 1,580,470 images of 81,313 landmark classes, and it has been widely used in existing methods [35, 32, 3] as the training dataset. To make a fair comparison, we follow the previous work [32, 35] to randomly divide the dataset into two subsets 'train'/'val' with 80%/20% split. For evaluation, we use two widely used benchmark datasets, $\mathcal{R}$Oxford and $\mathcal{R}$Paris [19] to test our model. Both datasets have 70 query images but contain 4,993 and 6,322 gallery images, respectively. In addition, a 1M distractor set [19] is added to test our model for the case of large-scale retrieval. We use mean average precision (mAP), as the criterion of retrieval performance for both "Medium" and "Hard" splits of $\mathcal{R}$Oxford and $\mathcal{R}$Paris.

4.2. Implementation Details

We use ResNet101/ResNet50 [8] as the CNN backbone to learn visual features. For image augmentation, random crop and color jittering are used first, and all the augmented images are resized into 512 × 512 pixels as the model input. The model is trained on 4 Nvidia V100-SXM2-32GB GPU cards with a batch size of 128 for 50 epochs. We use SGD optimizer with a momentum of 0.9. A warming-up of 5-epoch's training is used to initialize the learning rate from 0.0001 to the base learning rate of 0.005. As for the ArcFace loss, the margin m and the scale γ are empirically set as 0.2 and 45. For the global pooling, we utilize GeM [20] pooling with the fixed coefficient $p = 3$ during the whole training. β in the M-Step of GMM is empirically set as 0.999. In addition, we set the number of random walk steps l as 3 and the number of tokens as $K = 16$. The dimensions of the

Method	Medium				Hard			
	$\mathcal{R}$Oxf	$\mathcal{R}$Oxf+1M	$\mathcal{R}$Par	$\mathcal{R}$Par+1M	$\mathcal{R}$Oxf	$\mathcal{R}$Oxf+1M	$\mathcal{R}$Par	$\mathcal{R}$Par+1M
Global features + local features re-ranking								
R101-DELG [3]+SP	81.20	69.10	87.20	71.50	64.00	47.50	72.80	48.70
R101-DELG [3]+SP†	81.82	70.15	88.64	76.14	64.97	49.54	76.81	53.62
Global features								
R101-R-MAC [7]	60.90	39.30	78.90	54.80	32.40	12.50	59.40	28.00
R101-GeM [20]	64.70	45.20	77.20	52.30	38.50	19.90	56.30	24.70
R101-SOLAR [15]	69.90	53.50	81.60	59.20	47.90	29.90	64.50	33.40
R101-DELG [3]	76.30	63.70	86.60	70.60	55.60	37.50	72.40	46.90
R101-GLAM [24]	78.60	68.00	88.50	73.50	60.20	43.50	76.80	53.10
Swin-S-DALG [25]	79.94	-	90.04	-	57.55	-	79.06	-
R101-DOLG [35]★	81.50	77.43	91.02	83.29	61.10	54.81	80.30	66.69
R101-Token [32]	82.28	70.52	89.34	76.66	66.57	47.27	78.56	55.90
R101-Token [32]†	77.82	66.74	87.93	75.22	60.10	43.07	74.66	53.19
R101-DOLG [35]†	78.35	(75.51)	(88.48)	(78.25)	58.59	(52.36)	(75.39)	(61.85)
R101-DELG [3]†	79.82	67.85	87.21	72.45	61.16	42.58	73.84	50.79
R101-SOLAR [15]†	(81.63)	71.84	88.21	72.86	(63.29)	45.28	75.21	51.26
R50-SpCa$_{cro}$ (Ours)	79.91	72.78	87.43	78.01	59.27	49.26	73.14	58.3
R50-SpCa$_{cat}$ (Ours)	81.55	73.20	88.60	78.23	61.19	48.76	76.21	60.91
R101-SpCa$_{cro}$ (Ours)	82.73	**77.84**	90.20	79.12	65.60	**53.35**	79.29	**65.84**
R101-SpCa$_{cat}$ (Ours)	**83.24**	77.82	**90.56**	**79.48**	**65.85**	53.27	**79.97**	64.98

Table 1. Performance (mAP) comparison with recent state-of-the-art instance image retrieval methods on datasets $\mathcal{R}$Oxford5k and $\mathcal{R}$Paris6k [19]. R101 and R50 denote ResNet101 and ResNet50. Swin-S denotes Swin-Transformer-small [13]. The subscripts $_{cro}$ and $_{cat}$ refer to cross-attention and concatenation feature fusion strategies. SP denotes spatial verification via RANSAC. † denotes our re-implementation. ★ means no query cropping. Results in gray are those reported in the published papers. The best and second best performances are highlighted in **bold** and with underlines, respectively. The best previous performance is shown in brackets.

final global feature are set as 2048. For inference, the multiple scales $[0.3535, 0.5, 0.7071, 1.0, 1.4142]$ are applied to each original image to extract the feature representations. To fuse the features obtained from different scales, they are L_2-normalized, summed, and normalized again to generate the final global features for retrieval.

4.3. Comparison with State-of-the-art Methods

Setting for fair comparison Note that there are setting differences among the previous state-of-the-art methods. For example, they are different at 1) pretrained model sources for initialization (Caffe[1], PyTorch[2], and Facebook[3]); 2) training datasets (SfM-120k [20], GLDv1 [17], GLDv2 [31], and different 80%/20% split); 3) the number of multiple scales used at the inference stage (*e.g.*, 3, 5, or 7); 4) with or without query cropping scheme. These discrepancies could affect the fairness of the comparison if not well handled. To address this situation, we carefully re-implement multiple SOTA methods by using their officially published codes and the training parameters reported in their papers, while keeping the same settings for the above factors between these methods and ours. Specifically, to ensure fair comparison, our experimental study utilizes the same 80% of GLDv2-clean [31] as the training dataset, the same remaining 20% of GLDv2-clean as the validation dataset, the same pretrained model from PyTorch for initialization, and the same 5 scales with query cropping scheme for inference to conduct instance image retrieval.

Retrieval results Two groups of state-of-the-art methods are compared in this experimental study: 1) methods based on global feature representations. They include two milestone works R-MAC [7] and GeM [20] and another six more recent works including DELG [3], SOLAR [15], GLAM [24], DOLG [35], DALG [25], and Token [32]; 2) a method using global feature based initial retrieval plus local feature based re-ranking scheme, DELG+SP [3], is included as well. Also, note that we focus on global feature based image retrieval in this paper, and we do not consider training a verification model [10] or adding any post-processing steps like diffusion or query expansion to boost retrieval performance [34].

The retrieval result is reported in Table 1. The methods under "Global features" are partitioned into three sections. The results in gray color are those reported in the original

[1] http://cmp.felk.cvut.cz/cnnimageretrieval/data/networks/imagenet
[2] https://pytorch.org/vision/stable/models/generated/torchvision.models
[3] https://github.com/facebookresearch/pycls

published papers. Since they involve different settings as discussed above, the results are included here mainly as a reference. Following the gray-colored results are those obtained by our re-implementation for the sake of fair comparison, and they are used to compare with our methods shown at the bottom of this table.

As seen, our method R101-SpCa (where "SpCa" is short for "Spatial-context-aware"), with either cross-attention (*_cro*) or concatenation (*_cat*) fusion strategy, clearly outperforms the others on all the test cases. Compared with the closest results obtained by DOLG and SOLAR, our method achieves 1.6% and 2.0% performance gain on the Medium split of $\mathcal{R}$Oxford and $\mathcal{R}$Paris, respectively. And higher improvements of 2.5% and 4.5% are obtained on the Hard split of the two datasets. After the 1M distractors are added, our method still demonstrates its excellent retrieval performance. All these results validate the efficacy of our method.

Compared with the global initial retrieval plus local feature re-ranking scheme, our method still maintains its advantage. As listed in Table 1, even without using the pairwised re-ranking step as done in R101-DELG+SP [3], our method has outperformed it by around 2% on all dataset splits. More importantly, our method costs no extra inference time as taken by the re-ranking step. In addition, the memory footprint of our global feature representation (*i.e.*, a 2048-dimensional vector per image) is also much less than that used by the local descriptor (there are usually $1,000$ vectors per image, each of which is of 128 dimensions) based re-ranking. This again shows the computational advantage brought by our method.

Computational Overhead We report the number of parameters, feature extraction latency, and memory footprint for different SOTA models in Table 2. As seen, our method is comparable to that of DOLG in terms of computational overhead and comparable to that of Token in terms of the number of parameters. Also, our method takes the same memory footprint as the others that generate 2048-dimensional vectors as feature representations.

Method	#Para (M)	Latency (ms)	Mem. (MB) On $\mathcal{R}$Oxford
SOLAR [15]	52.5	110	39.0
DELG [3]	**44.5**	**104**	39.0
Token [32]	76.6	112	19.5
DOLG [35]	61.5	150	**9.7**
SpCa (Ours)	76.8	153	39.0

Table 2. Latency is measured on a 1080Ti GPU for a 1024 × 1024 image with 3 scaling factors [1, $\sqrt{2}$, 1/ $\sqrt{2}$].

4.4. Ablation Studies

In this study, we uniformly utilize ResNet50 as the CNN backbone and set the token number and the number of random walk steps as $K = 8$ and $l = 1$, respectively, for all the following experiments if not mentioned otherwise. In addition, all the models are trained by 25 epochs. More ablation studies utilizing ResNet101 are provided in the Appendix.

Global	OTL	DE	Medium		Hard	
			$\mathcal{R}$Oxf	$\mathcal{R}$Par	$\mathcal{R}$Oxf	$\mathcal{R}$Par
✓			77.51	87.82	54.76	73.82
✓	✓		78.82	88.21	58.73	74.03
✓	✓	✓	**79.64**	**89.06**	**60.90**	**76.97**

Table 3. Ablation studies on the impact of different components in the spatial context branch. The baseline containing only the visual feature branch is shown in the first row. OTL and DE denote online token learning and distance encoding, respectively.

Impact of each component in spatial context branch The proposed spatial context branch consists of two components which are online token learning and distance encoding. We validate the contribution of each of the components by adding them one by one to a bare visual feature branch. In particular, for the first comparison, we directly set the whole probability transition matrix M as a unit matrix (*i.e.*, all of its entries are "1") to erase the distance encoding information. By doing so, only the token-based identification generated by the online token learning module is involved in the final global features. As seen in Table 3, with this single module (2nd row), the retrieval performance could be improved from 77.51% to 78.82% and 87.82% to 88.21% on $\mathcal{R}$Oxf-Medium and $\mathcal{R}$Par-Medium, respectively. It shows the benefit of utilizing the tokens to identify the local descriptors. After adding the distance encoding module, the spatial distribution information is involved. It can be seen, with the complete branch (3rd row), the retrieval performance is further improved, especially on Hard splits. The performance (mAP) is increased from 58.73% to 60.90% and 74.03% to 76.97% on $\mathcal{R}$Oxf-Hard and $\mathcal{R}$Par-hard, respectively. This verifies the importance of capturing the spatial distribution information with the distance encoding module. Altogether, this result demonstrates the effectiveness of the spatial context branch as a whole in our method.

Fusion Strategy	Medium		Hard	
	$\mathcal{R}$Oxf	$\mathcal{R}$Par	$\mathcal{R}$Oxf	$\mathcal{R}$Par
Orthogonal	79.04	86.69	59.49	72.87
Hadamard	79.83	87.90	59.44	74.60
Concatenation	**79.94**	88.45	60.72	76.00
Cross-Attention	79.64	**89.06**	**60.90**	**76.97**

Table 4. Ablation studies on the impact of fusion strategy.

Fusion Strategy We validate the impact of different feature fusion strategies, including concatenation, orthogonal [35], Hadamard product, and cross-attention to fuse the information learned by our two branches in Table 4. As seen, the cross-attention scheme outperforms all the others in most cases except on the Medium split of $\mathcal{R}$Oxford. It demonstrates the effectiveness of utilizing cross-attention as the feature fusion strategy. Compared with cross-attention, orthogonal and Hadamard product perform worse. Surprisingly, we find simply concatenating the information from two branches has been able to obtain competitive performance. The result obtained by the concatenation strategy has been included in Table 1 too.

Token Number (K)	Medium		Hard	
	$\mathcal{R}$Oxf	$\mathcal{R}$Par	$\mathcal{R}$Oxf	$\mathcal{R}$Par
4	80.76	88.35	62.02	75.33
8	79.64	89.06	60.90	76.97
16	**81.22**	88.61	**62.41**	75.78
24	80.46	**89.42**	61.87	**77.01**
32	80.11	88.28	60.64	75.06

Table 5. Ablation studies on the impact of token number.

Number of Tokens We investigate the impact of the token number in Table 5. As seen, when the number is 16, the performance on $\mathcal{R}$Oxford is much better than other options. Meanwhile, when we utilize 24 tokens, the performance on $\mathcal{R}$Paris outperforms the others instead. We believe the optimal token number is related to the characteristic of the test dataset. When the number is too small, only a few visual prototypes could be explored to describe the whole dataset, leading to incomplete spatial context information extraction. Conversely, if the number is too large, more noisy tokens will be included. This could adversely impact the token-based identification process and consequently decrease the final retrieval performance. In our experiment, we just empirically use $K = 16$ since we do not access any test dataset during the training stage.

Random Walk (l)	Medium		Hard	
	$\mathcal{R}$Oxf	$\mathcal{R}$Par	$\mathcal{R}$Oxf	$\mathcal{R}$Par
1	79.64	89.06	60.90	76.97
2	80.76	88.88	62.15	76.24
3	**81.31**	**89.64**	**62.57**	**77.48**
4	80.25	88.54	61.12	75.61

Table 6. Ablation studies on the impact of random walk steps.

Number of Random Walk Steps To inspect how many random walk steps are needed to encode the distance for our spatial context information extraction, we apply different number of random walk steps $l = \{1, 2, 3, 4\}$. As seen in Table 6, $l = 3$ gives the best performance. When the step number is small (such as 1 or 2), only a small area around the nodes in the matrix M of Eq. (3) is considered for the final distance encoding process. Meanwhile, with the number of steps increased, the random walk process tends to converge and its results become similar (*i.e.*, $\mathbf{s}^{(l)} \approx \mathbf{s}^{(l+1)}$ in Eq. (6)). Keeping adding or concatenating this information leads to redundant or even noisy information to the distance encoding, and thus adversely impacts the spatial context information extracted and further undermines retrieval.

Qualitative Results We demonstrate two images retrieved from $\mathcal{R}$Oxford. In addition, we provide the classification activation map obtained by using query feature representation as classification projection weights. As seen in Figure 3, in addition to the most discriminative part of the queried object (as shown in the second column obtained by DELG), the proposed SpCa focuses more on the spatial-context area. This makes the ranking of the true positive image dramatically increased compared with DELG.

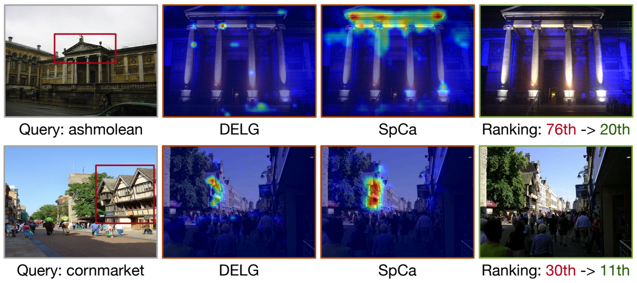

Figure 3. Qualitative results and the activation maps.

5. Conclusion

Motivated by the benefit of using local descriptor based geometric verification in a re-ranking step, we propose a novel feature learning framework that effectively embeds the spatial context information into the global visual feature representation for instance image retrieval. To mine the spatial context information, a spatial context branch consisting of an online token learning module and a distance encoding module is proposed. The former module is utilized to identify the type of each local descriptor on a feature map. The latter is used to capture the information on how the surrounding descriptors are spatially distributed. By fusing visual and spatial context information together, our framework learns spatial-context-aware global feature representations for images to conduct retrieval. Extensive experiments demonstrate the effectiveness of the proposed framework for retrieval. In future work, we will explore the incorporation of other types of information into global feature representations to further improve instance image retrieval.

Acknowledgement

This work is supported by the Australian Research Council (grant number DP200101289) and undertaken with the assistance of resources and services from the National Computational Infrastructure (NCI), which is supported by the Australian Government.

References

[1] Relja Arandjelovic, Petr Gronát, Akihiko Torii, Tomás Pajdla, and Josef Sivic. Netvlad: CNN architecture for weakly supervised place recognition. In *CVPR*, pages 5297–5307. IEEE Computer Society, 2016. 1

[2] Herbert Bay, Andreas Ess, Tinne Tuytelaars, and Luc Van Gool. Speeded-up robust features (SURF). *Comput. Vis. Image Underst.*, 110(3):346–359, 2008. 1

[3] Bingyi Cao, André Araujo, and Jack Sim. Unifying deep local and global features for image search. In *ECCV (20)*, volume 12365 of *Lecture Notes in Computer Science*, pages 726–743. Springer, 2020. 1, 3, 5, 6, 7

[4] Wei Chen, Yu Liu, Weiping Wang, Erwin M Bakker, Theodoros Georgiou, Paul Fieguth, Li Liu, and Michael S Lew. Deep learning for instance retrieval: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 2

[5] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *CVPR*, pages 4690–4699. Computer Vision Foundation / IEEE, 2019. 2, 5

[6] Martin A. Fischler and Robert C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981. 2

[7] Albert Gordo, Jon Almazán, Jérôme Revaud, and Diane Larlus. End-to-end learning of deep visual representations for image retrieval. *Int. J. Comput. Vis.*, 124(2):237–254, 2017. 6

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778. IEEE Computer Society, 2016. 2, 5

[9] Yannis Kalantidis, Clayton Mellina, and Simon Osindero. Cross-dimensional weighting for aggregated deep convolutional features. In *ECCV Workshops (1)*, volume 9913 of *Lecture Notes in Computer Science*, pages 685–701. Springer, 2016. 2

[10] Seongwon Lee, Hongje Seong, Suhyeon Lee, and Euntai Kim. Correlation verification for image retrieval. In *CVPR*, pages 5364–5374. IEEE, 2022. 1, 2, 6

[11] Pan Li, Yanbang Wang, Hongwei Wang, and Jure Leskovec. Distance encoding: Design provably more powerful neural networks for graph representation learning. In *NeurIPS*, 2020. 2, 4

[12] Tsung-Yu Lin, Subhransu Maji, and Piotr Koniusz. Second-order democratic aggregation. In *ECCV (3)*, volume 11207 of *Lecture Notes in Computer Science*, pages 639–656. Springer, 2018. 2

[13] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, pages 9992–10002. IEEE, 2021. 2, 6

[14] David G. Lowe. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.*, 60(2):91–110, 2004. 1

[15] Tony Ng, Vassileios Balntas, Yurun Tian, and Krystian Mikolajczyk. SOLAR: second-order loss and attention for image retrieval. In *ECCV (25)*, volume 12370 of *Lecture Notes in Computer Science*, pages 253–270. Springer, 2020. 1, 2, 6, 7

[16] Hien Duy Nguyen, Florence Forbes, and Geoffrey J. McLachlan. Mini-batch learning of exponential family finite mixture models. *Stat. Comput.*, 30(4):731–748, 2020. 4

[17] Hyeonwoo Noh, Andre Araujo, Jack Sim, Tobias Weyand, and Bohyung Han. Large-scale image retrieval with attentive deep local features. In *ICCV*, pages 3476–3485. IEEE Computer Society, 2017. 1, 2, 6

[18] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*. IEEE Computer Society, 2007. 1

[19] Filip Radenovic, Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondrej Chum. Revisiting oxford and paris: Large-scale image retrieval benchmarking. In *CVPR*, pages 5706–5715. Computer Vision Foundation / IEEE Computer Society, 2018. 5, 6

[20] Filip Radenovic, Giorgos Tolias, and Ondrej Chum. Fine-tuning CNN image retrieval with no human annotation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(7):1655–1668, 2019. 2, 5, 6

[21] Saimunur Rahman, Piotr Koniusz, Lei Wang, Luping Zhou, Peyman Moghadam, and Changming Sun. Learning partial correlation based deep visual representation for image classification. In *CVPR*, 2023. 2

[22] Jérôme Revaud, Jon Almazán, Rafael S. Rezende, and César Roberto de Souza. Learning with average precision: Training image retrieval with a listwise loss. In *ICCV*, pages 5106–5115. IEEE, 2019. 2

[23] Oriane Siméoni, Yannis Avrithis, and Ondrej Chum. Local features and visual words emerge in activations. In *CVPR*, pages 11651–11660. Computer Vision Foundation / IEEE, 2019. 1

[24] Chull Hwan Song, Hye Joo Han, and Yannis Avrithis. All the attention you need: Global-local, spatial-channel attention for image retrieval. In *WACV*, pages 439–448. IEEE, 2022. 2, 6

[25] Yuxin Song, Ruolin Zhu, Min Yang, and Dongliang He. Dalg: Deep attentive local and global modeling for image retrieval. *arXiv preprint arXiv:2207.00287*, 2022. 3, 6

[26] Fuwen Tan, Jiangbo Yuan, and Vicente Ordonez. Instance-level image retrieval using reranking transformers. In *ICCV*, pages 12085–12095. IEEE, 2021. 1, 2

[27] Giorgos Tolias, Yannis Avrithis, and Hervé Jégou. Image search with selective match kernels: Aggregation across sin-

gle and multiple images. *Int. J. Comput. Vis.*, 116(3):247–261, 2016. 2

[28] Giorgos Tolias, Tomás Jenícek, and Ondrej Chum. Learning and aggregating deep local descriptors for instance-level recognition. In *ECCV (1)*, volume 12346 of *Lecture Notes in Computer Science*, pages 460–477. Springer, 2020. 1, 2, 3

[29] Giorgos Tolias, Ronan Sicre, and Hervé Jégou. Particular object retrieval with integral max-pooling of CNN activations. In *ICLR (Poster)*, 2016. 2

[30] Philippe Weinzaepfel, Thomas Lucas, Diane Larlus, and Yannis Kalantidis. Learning super-features for image retrieval. In *ICLR*. OpenReview.net, 2022. 2, 3

[31] Tobias Weyand, André Araujo, Bingyi Cao, and Jack Sim. Google landmarks dataset v2 - A large-scale benchmark for instance-level recognition and retrieval. In *CVPR*, pages 2572–2581. Computer Vision Foundation / IEEE, 2020. 5, 6

[32] Hui Wu, Min Wang, Wengang Zhou, Yang Hu, and Houqiang Li. Learning token-based representation for image retrieval. In *AAAI*, pages 2703–2711. AAAI Press, 2022. 2, 5, 6, 7

[33] Hui Wu, Min Wang, Wengang Zhou, and Houqiang Li. Learning deep local features with multiple dynamic attentions for large-scale image retrieval. In *ICCV*, pages 11396–11405. IEEE, 2021. 2, 3

[34] Fan Yang, Ryota Hinami, Yusuke Matsui, Steven Ly, and Shin'ichi Satoh. Efficient image retrieval via decoupling diffusion into online and offline processing. In *AAAI*, pages 9087–9094. AAAI Press, 2019. 6

[35] Min Yang, Dongliang He, Miao Fan, Baorong Shi, Xuetong Xue, Fu Li, Errui Ding, and Jizhou Huang. DOLG: single-stage image retrieval with deep orthogonal fusion of local and global features. In *ICCV*, pages 11752–11761. IEEE, 2021. 2, 3, 5, 6, 7, 8

[36] Shan Zhang, Dawei Luo, Lei Wang, and Piotr Koniusz. Few-shot object detection by second-order pooling. In *ACCV (4)*, volume 12625 of *Lecture Notes in Computer Science*, pages 369–387. Springer, 2020. 2

[37] Shan Zhang, Naila Murray, Lei Wang, and Piotr Koniusz. Time-reversed diffusion tensor transformer: A new TENET of few-shot object detection. In *ECCV (20)*, volume 13680 of *Lecture Notes in Computer Science*, pages 310–328. Springer, 2022. 2

[38] Yantao Zheng, Ming Zhao, Yang Song, Hartwig Adam, Ulrich Buddemeier, Alessandro Bissacco, Fernando Brucher, Tat-Seng Chua, and Hartmut Neven. Tour the world: Building a web-scale landmark recognition engine. In *CVPR*, pages 1085–1092. IEEE Computer Society, 2009. 1