Learning Local Similarity with Spatial Relations for Object Retrieval

Zhenfang Chen
The University of Hong Kong, Hong Kong
zfchen@cs.hku.hk

Zhanghui Kuang
SenseTime Research, Hong Kong
kuangzhanghui@sensetime.com

Wayne Zhang
SenseTime Research, Hong Kong
wayne.zhang@sensetime.com

Kwan-Yee K. Wong
The University of Hong Kong, Hong Kong
kykwong@cs.hku.hk

ABSTRACT

Many state-of-the-art object retrieval algorithms aggregate activations of convolutional neural networks into a holistic compact feature, and utilize global similarity for an efficient nearest neighbor search. However, holistic features are often insufficient for representing small objects of interest in gallery images, and global similarity drops most of the spatial relations in the images. In this paper, we propose an end-to-end local similarity learning framework to tackle these problems. By applying a correlation layer to the locally aggregated features, we compute a local similarity that can not only handle small objects, but also capture spatial relations between the query and gallery images. We further reduce the memory and storage footprints of our framework by quantizing local features. Our model can be trained using only synthetic data, and achieve competitive performance. Extensive experiments on challenging benchmarks demonstrate that our local similarity learning framework outperforms previous global similarity based methods.

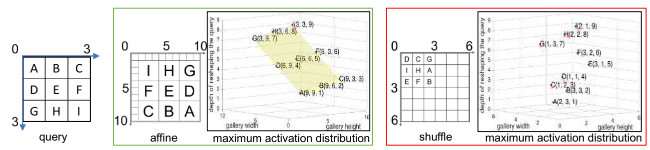
CCS CONCEPTS

• **Information systems → Image search**.

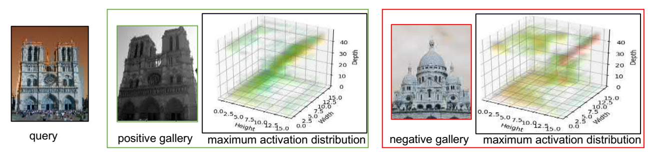
KEYWORDS

Object Retrieval, Local Spatial Relations, Computer Vision

ACM Reference Format:
Zhenfang Chen, Zhanghui Kuang, Wayne Zhang, and Kwan-Yee K. Wong. 2019. Learning Local Similarity with Spatial Relations for Object Retrieval. In *Proceedings of the 27th ACM International Conference on Multimedia (MM '19), October 21–25, 2019, Nice, France.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3343031.3351005



(a) A toy example



(b) A real example

Figure 1: Illustration of spatial relations encoded in the correlation volume. In (a), we show how the spatial relations are encoded in the correlation volume using a toy example. The query contains 9 local regions (*i.e.,* $A, B, \ldots I$). For each local region, it has a best-matching region in the gallery (indicated by the same character) which gives the maximum activation along the depth channel. In the green rectangle, we show a positive gallery which is an affine transformation of the query. Since affine transformation is linear, we can see the coordinates of all the maximum activations fall on a plane. In the red rectangle, we show a gallery image which contains the same local regions as the query but with different spatial relations, and we see the coordinates of the maximum activations distribute randomly. In (b), we show a real example with feature maps of the query and the gallery pooled to 7×7 and 15×15, respectively. ResNet101 is used as the feature extractor and only the strongest 20% activations are shown. We observe that the coordinates of most activations fall on a plane in the positive correlation volume while activations randomly distribute in the negative case (best viewed on screen). See section 3.2 for further explanation.

1 INTRODUCTION

Object retrieval has been a fundamental and popular research topic in computer vision. It aims at retrieving images containing objects same as the query image from a large database. Classical methods often represent each image by one or more descriptors, and formulate object retrieval as a nearest neighbor search in the descriptor space [2, 6, 7, 16, 22, 25–28, 41]. Recently, deep convolutional features have been extensively explored for image representation, and they have shown excellent performance over conventional features in many vision tasks. Many efforts have therefore been devoted to designing and learning good holistic image representations based on deep convolutional features, and object retrieval

MM '19, October 21–25, 2019, Nice, France

ACM ISBN 978-1-4503-6889-6/19/10…$15.00
https://doi.org/10.1145/3343031.3351005

can be accomplished by computing global similarities from these representations [4, 5, 10, 11, 18, 32–35, 42, 45].

Despite its success on standard benchmarks [28, 30], object retrieval based on global similarities computed from deep convolutional features has a number of limitations. First, it often has difficulty in retrieving gallery images with small objects of interest and cluttered background. Second, it may produce false positives which do not preserve the structures of the query objects. These problems originate from the averaging of the aggregated features of local regions in computing the global descriptor of an image. For the first problem, the global descriptor of the image is likely to be dominated by features of the background. For the second problem, the averaging process discards the spatial relations of local regions in both the query and the gallery images. To solve the first problem, Razavian *et al.* proposed R-match [33] which first performs a winner-take-all region cross-matching and then accumulates the maximum similarity of each query region. However, their method does not take the spatial relations of local regions into account, resulting in performance degradation. Later, Gordo *et al.* [10, 11] attempted to overcome some of the limitations by replacing a fixed grid of local regions with object proposals [36], but the performance of their method depends heavily on the trained region proposal network which may overfit to the object classes in the training set. Recently, Noh *et al.* [23] proposed to select a fixed number of deep local features using an attention mechanism, perform an approximate nearest neighbor search for each local descriptor in the gallery feature set, and aggregate all the matches per gallery image. However, their image-level similarity is not defined explicitly and therefore cannot be learned end-to-end to achieve an optimal retrieval performance.

In this paper, we propose a simple and effective end-to-end local similarity learning network for object retrieval. Similar to many state-of-the-art methods, we generate aggregated features of local regions for both the query and the gallery images. Instead of combining these aggregated features into a global descriptor through averaging, we introduce a correlation layer (first used in stereo matching and optical flow estimation [8, 20]) to compute a correlation volume describing the similarities between local regions of the query and gallery images. In particular, we show that by preserving the spatial relations of local regions in both the query and gallery images, maximum responses in this correlation volume should lie in a specific subspace which can be used to infer the existence and location of the object of interest. We propose using a small convolutional network to learn the existence and location of the object of interest from such a correlation volume. Our local similarity learning network is composed of two subnetworks, namely a feature extraction subnetwork for generating local regional features and an object localization subnetwork for predicting the existence and location of the object of interest. Since all operations are differentiable, our local similarity network can be trained end-to-end. To reduce data annotation effort, we deploy a simple approach to synthesize training data by cutting and pasting object instances on random backgrounds as in [9]. To scale up our retrieval algorithm, we further use product quantization [17] to reduce our memory and storage footprints, and prefiltering with global descriptors to accelerate the search process on large-scale databases. Our main contributions are summarized as follows:

(1) We introduce a correlation layer to compute a correlation volume describing the similarities between local regions of the query and gallery images, and show that consistency between the spatial relations of local regions in the query and gallery images is reflected by maximum responses in this volume lying in a specific subspace (see Fig. 1).
(2) We propose an end-to-end trainable local similarity network that can capture the spatial relations of local regions for object retrieval.
(3) We deploy cutting-and-pasting synthesis method [9] to train the proposed network and achieve competitive performance on standard benchmarks. To our best knowledge, this is the first study of training an object retrieval model with synthetic data only.
(4) Our algorithm outperforms other methods on Oxford [29], Paris [30] and Instre [44] datasets, especially for cases of positive gallery images with cluttered background and negative gallery images with similar local regions but different spatial relations.

2 RELATED WORKS

Object retrieval has been studied for more than a decade. A comprehensive survey, which categorizes existing methods into local feature based methods and global feature based methods, can be found in [48]. In [41], Sivic and Zisserman proposed the Bag-of-the-Word (BoW) model which encodes a set of local invariant features (*e.g.*, SIFT [21]) in a sparse feature vector for image retrieval. Following this seminal work, researchers introduced various components, such as large visual codebooks [3, 22], spatial verification [25, 28] and query expansion [6, 7], into the pipeline to improve the retrieval performance. Other classical research works, such as Fisher Vector [26, 27] and VLAD [2, 16], focused on designing schemes for aggregating local features into compact global features.

In the past few years, convolutional neural networks (CNNs) were widely adopted for the object retrieval task, and they demonstrated extraordinary performance. Many of such works [4, 5, 10, 11, 18, 31–35, 42, 45] focused on designing and learning holistic compact features. Early works [4, 33] extracted features directly from the fully connected layers of a pre-trained CNN, while recent works commonly aggregated regional features from the convolutional layers [1, 42, 45]. Researchers also found that fine-tuning CNNs on data similar to the target task can significantly boost the retrieval performance [4, 10, 11, 32]. These methods typically adopted a nearest neighbor search based on global similarities of the holistic compact features. Although searching based on global similarity is efficient, it does not perform well under more challenging conditions, such as complicated clutter, large occlusion and variations in viewpoints. This is because global similarity is ineffective in representing similarities between local regions as it drops the spatial relations of local regions in the images.

In [23], Noh *et al.* proposed a deep local feature image retrieval method which utilizes an attention mechanism to extract local features from a query image and performs nearest neighbor search for each local feature. Their method then aggregates all the matches per gallery image. Iscen *et al.* [15] proposed a diffusion mechanism to capture the manifold in the local feature space. The diffusion is carried out on descriptors of overlapping image regions rather than on a holistic image descriptor. The use of local features has boosted the retrieval performance significantly in these approaches. However, their local feature based similarities are not defined explicitly, and their pipelines are rather complicated. For instance, the regional diffusion method [15] requires storing and computing on a huge graph. In contrast, we propose a simple and effective end-to-end local similarity learning network for object retrieval.

CNNs have also demonstrated great success in image matching. Many research works [12, 43, 46, 47] focused on part of or the whole pipeline for detecting local feature keypoints and comparing local features. Recently, Rocco *et al.* [37] proposed a CNN architecture trained on synthetically generated images for predicting an affine or thin-plate-spline transformation for image matching. However, all these methods were not originally designed for the object retrieval task. In this paper, we cast the problem of object retrieval as object localization (*i.e.*, checking whether the query object exists in the gallery image and finding its position), and supervise our proposed end-to-end trainable network with an object localization loss. We also demonstrate how our object retrieval algorithm can be scaled up to handle large-scale datasets.

3 LOCAL SIMILARITY WITH SPATIAL RELATIONS

In this section, we first briefly describe the working principle of existing methods that are based on global similarity, and the problems associated with them. This leads us to explore taking the spatial relations of local regions into account in computing local similarity for object retrieval. Specifically, we propose to compute similarities between local regions of the query and gallery images using correlation, and rearrange the results into a correlation volume that allows easy indexing the similarities by spatial indices of the local regions. We prove that by preserving the spatial relations of local regions in both the query and gallery images, maximum responses in this correlation volume should lie in a specific subspace. Based on this proof, we propose using a small CNN to learn the existence and location of the object of interest from the correlation volume.

3.1 Deficiency of Global Similarity

Many recent CNN-based object retrieval algorithms follow the global R-MAC pipeline [42] which subdivides an image into a grid of rectangular regions, extracts local features from the regions, and aggregates these regional features to a holistic image representation [5, 10, 11, 14, 18, 32]. The R-MAC extraction process can be summarized as follows. First, activation features are extracted from the convolutional layers of a pre-trained network. These activation features are then max-pooled in each region. The pooled regional features are independently ł$_2$-normalized, whitened with PCA and ł$_2$-normalized again. Finally, these normalized regional features are sum-aggregated and ł$_2$-normalized to produce a holistic descriptor. A global similarity between two images can then be computed as the dot-product of their holistic descriptors.

Let $\mathbf{f}_q(\mathbf{x}_q) \in R^D$ denote the normalized D-dimensional regional feature with spatial index $\mathbf{x}_q \in R^2$ for the query image. Similarly, let $\mathbf{f}_g(\mathbf{x}_g) \in R^D$ denote the normalized regional feature with spatial index $\mathbf{x}_g \in R^2$ for the gallery image. The global R-MAC descriptors for the query and gallery images can be written as $\mathbf{g}_q = \sum_{\mathbf{x}_q} \mathbf{f}_q(\mathbf{x}_q)$ and $\mathbf{g}_g = \sum_{\mathbf{x}_g} \mathbf{f}_g(\mathbf{x}_g)$[1] respectively. The global similarity between the query and gallery images is then given by

$$S_{\text{R-MAC}} = \mathbf{g}_g^{\mathrm{T}} \mathbf{g}_q = \sum_{\mathbf{x}_g, \mathbf{x}_q} \mathbf{f}_g(\mathbf{x}_g)^{\mathrm{T}} \mathbf{f}_q(\mathbf{x}_q). \tag{1}$$

It follows from (1) that the global R-MAC similarity can be interpreted as a cross matching between all regions of the query and gallery images. This formulation has two potential problems. First, each pair of regions contributes equally to the final similarity, and this makes the contributions of the true corresponding regions less significant (*i.e.*, a low signal-to-noise ratio). Second, the sum-aggregation is an *orderless* operation over the set of regional features, and is incapable of preserving the spatial relations of local regions in the image. The seminal work R-Match [34] solves the first problem by adopting a winner-take-all approach. It greedily finds the best matched local region in the gallery image for each local region in the query image, and then sum-aggregates the similarities of the best matches into the final similarity. Formally, the R-Match similarity is given by

$$S_{\text{R-Match}} = \sum_{\mathbf{x}_q} \max_{\mathbf{x}_g} \left(\mathbf{f}_g(\mathbf{x}_g)^{\mathrm{T}} \mathbf{f}_q(\mathbf{x}_q) \right). \tag{2}$$

Note that R-Match does not take the spatial relations of local regions into account in computing the local similarity. It would therefore output an incorrect high similarity when the gallery image contains local regions which are similar to those of the query image but have a different spatial composition (see the shuffle case in Fig. 1a).

3.2 Preserving Spatial Relations of Local Regions

In order to allow us to take the spatial relations of local regions into account in computing local similarity, we propose to compute similarities between local regions of the query and gallery images using correlation, and rearrange the results into a correlation volume that indices the similarities by spatial locations. Let $\mathbf{f}_q \in R^{W_q \times H_q \times D}$ denote a 2D map of D-dimensional regional features for the query image, where $W_q \times H_q$ is the spatial resolution of the subdivision grid for the query image. Similarly, let $\mathbf{f}_g \in R^{W_g \times H_g \times D}$ denote a 2D map of D-dimensional regional features for the gallery image, where $W_g \times H_g$ is the spatial resolution of the subdivision grid for the gallery image. We define the similarity between the local region of the query image with spatial index (w_q, h_q) and the local region of the gallery image with spatial index (w_g, h_g) as

$$s(w_g, h_g, w_q, h_q) = \mathbf{f}_g(w_g, h_g)^{\mathrm{T}} \mathbf{f}_q(w_q, h_q). \tag{3}$$

By considering the similarities between all regions of the query and gallery images, we produce a tensor of similarities with a dimension of $W_g \times H_g \times W_q \times H_q$. We rearrange the last two dimensions related to the query image into a single dimension referred to as the depth channel, and obtain a volume of similarities with a dimension of $W_g \times H_g \times D'$ where $D' = W_q \times H_q$. We can interpret this volume as a 2D map (*i.e.*, $W_g \times H_g$) of D'-dimensional correlation vectors $\mathbf{f}_c(w_g, h_g)$, where each correlation vector $\mathbf{f}_c(w_g, h_g)$ encodes how

[1] The normalization scales are omitted in the expressions of $\mathbf{g}_q$ and $\mathbf{g}_g$ for the sake of simplicity.

well the local region of the gallery image with spatial index (w_g, h_g) matches with each local region of the query image.

The above computation can be conveniently implemented with a correlation layer which is first used in CNN-based stereo matching and optical flow estimation [8, 20]. Note that the correlation volume generated by the correlation layer stores the similarities in such a way that allows easy indexing the similarities by spatial indices of the local regions. Next, we are going to show that by preserving the spatial relations of local regions in both the query and gallery images, maximum responses in this correlation volume should lie in a specific subspace. Consider the simple case where the query and the gallery images are identical. Let $d' \in [1, W_q \times H_q]$ denote the index of the maximum response for the correlation vector $\mathbf{f_c}(w_g, h_g)$. Since the query and gallery images are identical, maximum response should happen when the local regions of the query and gallery images have the same spatial index (*i.e.*, $(w_q, h_q) = (w_g, h_g)$). Hence, we have

$$d' = \mathbf{K}[w_q \;\; h_q]^{\mathrm{T}} - W_q, \tag{4}$$

where $\mathbf{K} = [1 \;\; W_q]$. Now consider the case where there exists a transformation between the query and gallery images. Let Φ be a function that maps each local region of the gallery image to a corresponding local region of the query image, *i.e.*,

$$[w_q \;\; h_q]^{\mathrm{T}} = \Phi[w_g \;\; h_g]^{\mathrm{T}}. \tag{5}$$

Substituting (5) into 4 gives

$$d' = \mathbf{K}\Phi([w_g \;\; h_g]^{\mathrm{T}}) - W_q, \tag{6}$$

which defines the subspace in which the maximum responses corresponding to true matching of local regions between the query and gallery images should lie. It is easy to see that if the mapping function Φ is linear (*e.g.*, translation, rotation, scaling, affine transformation), this subspace will become a plane. We illustrate the existence of such a subspace for maximum responses in Fig. 1 using both toy and real examples. It can be observed that maximum responses lie on a plane for positive gallery images, whereas there is a more 'random' distribution of maximum responses for negative gallery images.

3.3 Learning Local Similarity with Object Localization

From the analysis above, we find that the correlation volume $\mathbf{f_c}$ not only encodes the pairwise similarity between all the local region pairs, but also captures the spatial relations. Since our goal is to estimate the similarity between the query and the gallery image, we design a lightweight CNN $\mathcal{F}$ to classify the pattern encoded in the correlation volume. The predicted confidence for a positive pair is used as the estimated similarity between the query and the gallery. Formally,

$$S_{local} = \mathcal{F}(\mathbf{f}_c). \tag{7}$$

As the correlation volume encodes information of whether the query exists in the gallery as well as the location(s) of the query, we model the retrieval problem as object localization on the correlation volume. Following the spirit of Faster R-CNN [36], we apply classification and localization in parallel. Formally, during training, we define a multi-task loss as

$$\mathcal{L}(y, \overline{y}, l, \overline{l}) = \mathcal{L}_{cls}(y, \overline{y}) + \lambda \mathcal{L}_{loc}(l, \overline{l}), \tag{8}$$

Table 1: Comparing the performance of different loss functions. Models are trained with ResNet-101 as backbone on synthetic data. $\mathcal{L}_{trp}$, $\mathcal{L}_{cls}$ and $\mathcal{L}_{loc}$ denote triplet loss, the classification loss and localization loss, respectively.

	Oxf5k	Par6k
$\mathcal{L}_{trp}$	78.2	87.6
$\mathcal{L}_{cls}$	80.3	88.9
$\mathcal{L}_{cls} + \mathcal{L}_{loc}$	**81.1**	**89.3**

where y is the probability of object existence predicted by the convolutional subnetwork, $\overline{y}$ is the ground truth label of object existence ($\overline{y} = 1$ if the query object exists in the gallery image, and $\overline{y} = 0$ otherwise), l and $\overline{l}$ denote the predicted and ground truth normalized bounding boxes of the query object in the gallery image, respectively. λ is simply a balancing weight which is set to 0.05 in our implementation.

The classification loss $\mathcal{L}_{cls}(y, \overline{y})$ is computed based on cross entropy loss. We use the classification loss rather than the triplet loss that is used in [10, 11], because the objective of the proposed CNN $\mathcal{F}$ is to classify the correlation volume, not to differentiate between the features from two images.

The localization loss $\mathcal{L}_{loc}(l, \overline{l})$ is computed as smooth l_1 loss as defined in [36], with two major differences. First, the localization loss is normalized by the number of positive query gallery pairs in a mini-batch, *i.e.*, $N_{loc} = \sum_{i=1}^{N} y_i$ where N is the size of a mini-batch, but not the number of positive region proposals as in [36]. Second, the bounding box is encoded and normalized with the height and width of the input image as the reference rectangle but not region proposals or default boxes as in [36]. We emphasize that our task is object retrieval but not object detection. We assume there is at most one query object in one gallery image during training. This is actually true for our training data. For testing, we have no such limitation.

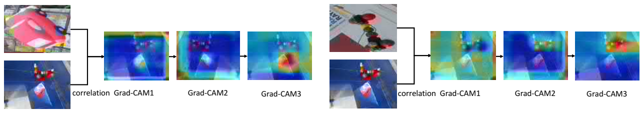
After training, the probability of object existence predicted by the CNN $\mathcal{F}$ can be used directly as the similarity between the query and gallery images. Since the correlation layer encodes the matching between local regions, we name our similarity as local similarity, in contrast to global similarity which compares holistic image representations.

To demonstrate the effectiveness of the proposed loss, we compare the performance of models trained with different loss functions in Table 1. We can see that the classification loss outperforms the triplet loss, and the localization loss further improves the performance since it utilizes additional information for supervision.

By making use of the above multi-task loss, we find that the CNN $\mathcal{F}$ can learn to mine the spatial relation consistency. To apprehend its capability, we apply Grad-CAM [39] to visualize the convolutional feature maps. Fig. 2 illustrates that the CNN $\mathcal{F}$ gradually focuses on the region with spatial relation consistency and discards those without it.

3.4 Local Similarity Learning CNN

Our local similarity learning CNN consists of two major components, namely a feature extraction subnetwork and an object localization subnetwork. Fig. 3 shows the overall architecture of the proposed framework.



(a) (b)

Figure 2: Visualizing the capability of mining spatial relation consistency. Top and bottom are the query and gallery images. (a) and (b) use the same gallery image but different query objects. Grad-CAM1, Grad-CAM2 and GradCAM3 show activations of the 1st, 2nd and 4th convolutional layer feature maps, respectively (best viewed on screen).

Feature extraction subnetwork. Theoretically, any kind of architectures can be used as the backbone for our feature extraction subnetwork. For simplicity, we take ResNet-101 as the backbone for analysis here. To extract better representations for small objects, we do not resize the input image[2] before feeding it into the feature extraction subnetwork. Hence, our feature extraction subnetwork produces feature maps of different sizes depending on the input image sizes. To handle the differences in size of the feature maps, on the last convolution of ResNet-101, we use one ROI-pooling layer (with ROI being the whole image) to pool the feature map of the gallery image to a fixed size of $15 \times 15 \times 2048$, and that of the query image to $7 \times 7 \times 2048$. Note that the feature map of the query image has a smaller size than that of the gallery image. Such a design is based on the fact that the (cropped) query image typically contains only the object of interest, whereas the object of interest often occupies only a small region of the gallery image. A small query feature map results in a smaller number of depth channels in the correlation volume and hence a faster similarity computation. We further $l2$-normalize the pooled feature maps at each location to avoid magnitude unbalance between the query and gallery feature maps before feeding them into the object localization subnetwork.

Object localization subnetwork. With a correlation layer taking the query and gallery feature maps as input, our object localization subnetwork utilizes a lightweight CNN to predict whether the query exists in the gallery image with a localization auxiliary task. It has only 5 convolutional layers with a small number of channels, and two small fully connected layers. Each convolutional layer is followed by an instance normalization layer and a ReLU layer. The detailed configuration of the object localization subnetwork can be see in Table 2.

Our object localization subnetwork is lightweight. The total runtime for retrieving a query is the sum of the runtime for extracting query features and the runtime for predicting similarity. The runtime of our feature extraction subnetwork is similar to that of the global feature extractor [11], both taking about 0.09s on average for an image of size 1024×768 (typical size for images in Oxford building and Paris datasets) using GPU (*e.g.*, Titan X). Our object location subnetwork can make a prediction in less than 0.07ms, achieving about 16,000 frames per second, which is less than 0.1% of the runtime for feature extraction.

[2]Query images are first cropped with the given bounding boxes. During training, for ease of implementation, all gallery images in a mini-batch are resized to their average size, with a maximum size of 800, and the same for query images in a mini-batch. During testing, we keep the original sizes of the gallery images.

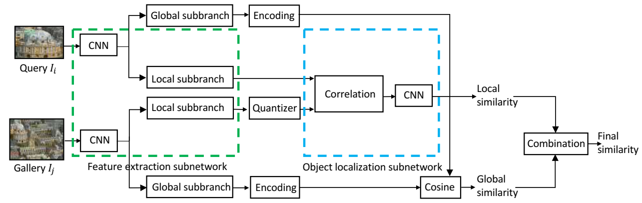


Figure 3: Architecture of the proposed local similarity learning framework. The proposed feature extraction subnetwork and object localization subnetwork are shown in the dashed boxes.

Table 2: Detailed configuration of the proposed object localization subnetwork.

Type	Filter Shape	Input Size
Conv Block	$[3 \times 3, 128] \times 3$	15×15×49
Max Pool	Pool 2×2	15×15×128
Conv Block	[3×3, 64]×2	7×7×128
Max Pool	Pool 2×2	7×7×128
FC	576×2	3×3×64
	576×4	

3.5 Scaling Up for Efficient Object Retrieval

Scalable storage. We pre-compute and store the outputs of the feature extraction subnetwork for the gallery images for saving time and computational resource. Given a gallery image, its feature map has a dimension of $W_d \times H_d \times D$ (where D denotes the number of channels). For a large gallery set, it would mean a high memory and storage cost. In order to reduce the memory and storage footprints, each D-dimensional feature vector is first subdivided into M sub-vectors and then k-means clustering is run on the uncompressed sub-vectors to generate C centers for each subpart. Finally, each subvector is approximated by one of the C centers. In this way, each feature vector can be approximated using only $M * log_2(C)$ bits. Typically, C is set to 256, and M is set to 128 in our case. The total storage can therefore be reduced by 64 times. Experimental results show that, after quantization, the retrieval performance decreases only by 1.2% and 0.5%, respectively on Oxf5k [28] and Par6k [30] benchmarks. It suggests that the proposed object localization subnetwork is robust against small perturbation of the input, and has a good generalization ability. Fine-tuning the object localization subnetwork after quantization can further improve its performance consistently. A fine-tuned subnetwork with $M = 128$ (*i.e.*, 64 times compression) has negligible drops (0.9% on Oxf5k and 0.2% on Par6k) on performance as its uncompressed counterpart on both benchmarks. More detailed results can be found in Section 4.4.

Scalable search. For large-scale databases, we extract a global feature on top of the convolutional layer to accelerate the whole retrieval process. Specifically, we first use the global feature to filter out most irrelevant images, and get a short list of K gallery images. The proposed local similarity is then computed only between the query and the shortlisted images. Theoretically, any global feature can be used to accelerate the search process. To harvest the power

Figure 4: Examples of the synthetic data. Black/red/yellow borders on images denote original images, masks provided and synthetic samples generated, respectively. Regions of interest are bordered with green color for better visualization.

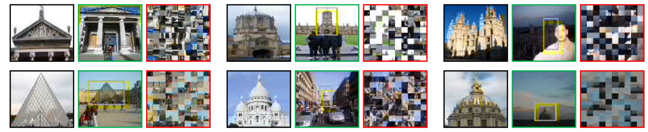


Figure 5: Examples of the synthetic distractors. Black, green and red borders on images denote example query objects, positive gallery images and synthetic negative gallery distractors, respectively. Regions of interest in the positive galleries are bordered with yellow color for better visualization.

of deep features and reduce computational cost, we extract the popular and effective R-MAC [42] features. Directly encoding R-MAC on activations of the last convolutional layer of ResNet-101 resulted in poor performance of the local similarity learning task. This is reasonable since global similarity and local similarity need different semantic representation. We gradually reduce the shared layers between R-MAC and the proposed feature extraction subnetwork, and experimentally find that sharing the first 91 layers can achieve a good trade-off between effectiveness and efficiency. Therefore, in our final design, R-MAC and the proposed feature extraction subnetwork each have one separated small branch of 10 convolutional layers ($92 \sim 101$ layers in ResNet-101). When testing on a GPU (*e.g.*, Titan X), the separated 10 convolutional layers add about $0.004s$ for the total feature extraction and take up 5% of the total feature extraction time. Although the global feature R-MAC is introduced here for efficient search initially, we find that the performance of the proposed method can be boosted further by combining local similarity with the global similarity based on R-MAC.

3.6 Training with Synthetic Data.

Previous research [10, 11, 32] proposed to collect a large real dataset from the web and label training images with sophisticated methods for image retrieval. Inspired by recent cutting-and-pasting approach for instance detection [9], we generated a large set of training images with bounding boxes annotations in a simpler and more scalable way. The procedure for generating synthetic data can be summarized as follows.

Cutting. To cut objects from images, we first need the instance masks for the object instances. We use the masks of object instances provided by the annotations of COCO dataset [19]. We randomly sample 5,000 images from COCO validation set [19]. For each image, we automatically remove those instances whose areas are smaller than 80×80 and keep only one instance mask per image, resulting in about 4,000 objects.

Pasting. We use images from ImageNet dataset [38] as the background dataset because it provides diverse backgrounds. We paste the query objects into any random position of the background images. Data augmentation, including rotation, scale transformation, photo-metric distortion and random cropping, is carried out before blending to further increase the diversity of the training dataset.

Samples of the original images, masks and synthetic data can be seen in Fig 4. During training, we use the whole synthetic images as gallery images and the cropped regions of interest as query images.

4 EXPERIMENTS

In this section, we demonstrate the performance of our proposed object retrieval algorithm with extensive experiments. First, we show that our pipeline can be trained effectively with only synthetic data and achieve competitive performance. We then show the effectiveness of local similarity. We also evaluate the techniques of scalable storage and scalable search. Finally, we compare our object retrieval algorithm with state-of-the-art methods using global similarity (*e.g.*, DIR [10, 11]) and regional features (*e.g.*, R-Match [34] and DELF [23]).

4.1 Experimental Settings

Evaluation datasets and criterion. We evaluate our methods on popular object retrieval datasets, namely Oxford Building [28], Paris [30] and INSTRE [44]. Oxford building dataset [28] and Paris [30] dataset, composed of $5,063$ and $6,412$ images, are referred to as Oxf5k and Par6k, respectively. Besides, 100k Flickr images [28] are added to these two datasets to form Oxf105k and Par106k datasets for evaluation at a larger scale. To better show the effectiveness of the proposed method to capture the spatial relations, we also create new datasets by adding more challenging synthetic negative distractors into Oxf5k and Par6k. Specifically, every positive gallery image is first subdivided into a grid of 10×10 and then we shuffle these 100 patches randomly. We finally manually check that the region of interest is fully destroyed. Fig 5 shows some examples of synthetic distractors, which contain local patches similar to the query images but with a total different spatial arrangement. For each positive gallery image, we generate 3 synthetic distractors and combine them with the original Oxf5k and Par6k to form two new testing sets which contain $6,764$ and $11,782$ images, respectively. We refer to these two new datasets as OxfShf and ParShf, respectively. As for INSTRE [44], it contains 250 different objects and include more variations such as scales, rotations and occlusions than Oxford and Paris, which make it more challenging. Retrieval performance is measured in terms of mean average precision (mAP) which is widely used in the image retrieval community.

Training data. We construct two training datasets, namely CocoSyn and Landmark-clean-half. CocoSyn is a synthetic dataset of $20,000$ images. As described in Section 3.6, we synthesize 4 images for each segmented object instance. Together with the original $4,000$ Coco [19] images, we obtain $20,000$ training images. Landmark-clean-half is a subset of Landmark-clean dataset created by Gordo *et al.* [10, 11]. The original Landmark-clean dataset used for training DIR [10, 11] consists of $49,000$ images with approximate

Table 3: Effectiveness of training with synthetic data. The proposed local similarity is marked with ⋆. syn and lch indicate CocoSyn dataset and Landmark-clean-half dataset, respectively.

(a) With VGG-16 as backbone

	Oxf5k	Par6k
SiaMac [32]	77.0	84.1
⋆(syn)	75.6	81.2
⋆(syn+lch)	**86.4**	**88.1**

(b) With ResNet-101 as backbone

	Oxf5k	Par6k
DIR-R-MAC [11]	83.9	93.8
⋆(syn)	81.1	89.3
⋆(syn+lch)	**90.3**	**94.4**

bounding boxes. However, due to link failure, we are only able to download a subset of 27, 699 images (23, 843 for training and 3, 856 for validation), which is about 56.5% of the whole Landmark-clean dataset.

Implementation details. We use VGG-16 [40] and ResNet-101[13] as our backbones. For models trained with synthetic data only, we use the model pre-trained on ImageNet as a starting point. For models that share feature extraction subnetwork with DIR-R-MAC [11], we use their released R-MAC model [11] to initialize them. We train the networks with stochastic gradient descent, with a learning rate of 10^{-3}, momentum 0.9, weight decay 10^{-2} and batch size of 16. In each batch, we use 8 pairs of images (*i.e.*, 8 query images and their corresponding positive gallery images) to generate 8 positive pairs and 56 negative pairs. We select all 8 positive pairs and 8 most difficult negative pairs for loss calculation. Our implementation is based on PyTorch [24] library and trained on a PC with 4 NVIDIA GTX 1080Ti cards.

4.2 Effectiveness of Training with Synthetic Data

We evaluate the effectiveness of training with synthetic data using both VGG-16 and ResNet-101 pre-trained on ImageNet as backbones. As shown in Table 3, the proposed local similarity trained on CocoSyn can achieve competitive performance compared with SiaMac [32] and DIR-R-MAC [11]. Note that SiaMac [32] and DIR-R-MAC [11][3] are trained on large-scale real datasets (160, 000 and 192, 000 images, respectively, compared to 20, 000 images in CocoSyn) which have contents similar to the test images in Oxf5k and Par6k, and are constructed with sophisticated methods. With 23, 843 additional real images from Landmark-clean-half, the proposed local similarity outperforms SiaMac and DIR-R-MAC. As the combination of CocoSyn and Landmark-clean-half has a size comparable to the original Landmark-clean dataset [11], we conduct all the rest of our experiments using the combined training dataset except as otherwise noted.

4.3 Effectiveness of Local Similarity

We use the released R-MAC model [11] to initialize the proposed feature extraction network. We further fine-tune the last 10 layers of the feature extraction subnetwork and the whole object localization subnetwork. As shown in Table 4, the proposed local similarity consistently outperforms global similarity [11] on all datasets. Moreover, combining local and global similarities with a linear weight

[3]For ease of comparison with DIR on all the datasets, we test the performance of DIR with the source code and models released by the authors. It is slightly different from the performance reported in the paper (84.1 on Oxf5k and 93.6 on Par6k).

Table 4: Comparison of object retrieval performance of global similarity, local similarity and their combination on seven datasets. We use ResNet-101 as backbone. Our methods are marked with ⋆.

	Oxf5k	Par6k	Oxf105k	Par106k	OxfShf	ParShf	INSTRE
local⋆	91.6	95.3	89.5	91.7	**91.4**	**93.3**	76.9
global [11]	83.9	93.8	80.8	89.9	74.7	81.5	62.6
local+global⋆	**91.8**	**95.6**	**89.8**	**92.5**	91.0	93.1	**77.3**

of 0.9 for the local similarity has the best performance. This shows that our local similarity is complementary to the global similarity.

Robustness to small objects with cluttered background. To illustrate the advantage of local similarity, we show some examples on which our local similarity outperforms global similarity in Fig. 6. We find that global similarity is good at retrieving gallery images with relatively large objects and fails to retrieve gallery images with a small query object and complicated background (*e.g.*, the images with green borders in Fig. 6). In contrast, our local similarity is able to find region-level matches between images, which enables it to successfully retrieve such difficult images. Particularly, for the INSTRE [44] dataset which contains a lot of small objects, our local similarity outperforms global similarity by 14.3% (see Table 4).

Capturing Spatial Relations. We also find that global similarity may retrieve false positive gallery images with similar local patterns but different spatial relations (*e.g.*, the images with red borders in Fig. 6). In contrast, our local similarity is good at capturing spatial relations of local patterns, which helps to suppress such false alarms. Particularly, for OxfShf and ParShf which contain challenging synthetic negative distractors, our local similarity outperforms global similarity by 16.7% and 11.8%, respectively. Local similarity achieves slightly better performance than local+global on oxfShf and ParShf. We suspect that the reason is that global similarity fails to capture the spatial relations and achieves poor performance on OxfShf and ParShf, which makes the combination of local+global achieves slightly worse performance. However, local+global achieves the best performance on most datasets.

4.4 Impact of Scaling Up

Impact of storage compression. In product quantization, we fix the number of the centers to be 256 and the compression ratio is determined by the number of subvectors M, which we divide each D-dimensional local feature into. Table 5 shows the retrieval performance when M is 32, 64 and 128, respectively. We can see that fine-tuning the object localization subnetwork can consistently boost the retrieval performance. The overall performance is good and stable with varying parameters from 32 to 128. Since the model with $M = 128$ to encode the local features has the most stable performance, we use this value for the rest of our experiments.

Impact of prefiltering with global similarity. We do cross validation for linear combination weight w from 0 to 1 and the size K of the short list from 0 to 5,000. The results are shown in Fig. 7 and 8. We find that the performance is stable for a wide range of both parameters. The combined similarity is defined as $S = (1 - w) \times S_{global} + w \times S_{local}$, where S_{global} and S_{local} are the

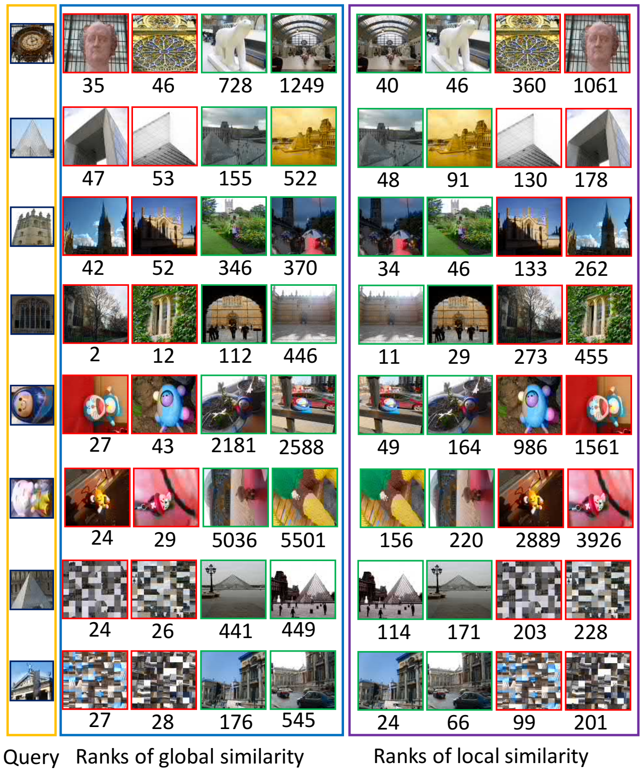
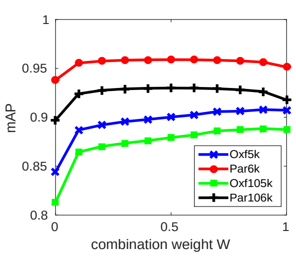


Figure 6: Typical examples on which our local similarity outperforms global similarity. Black/red/green borders on images denote query/negative gallery/positive gallery images, respectively. Yellow/blue/purple rectangles denote query images, results of the global similarity [11] and that of the proposed local similarity, respectively. Ranks are shown below the gallery images (best viewed on screen).



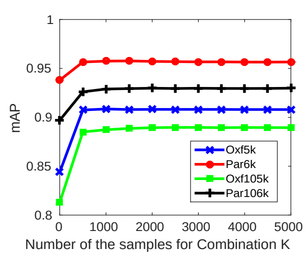


Figure 7: Cross-validation for linear combination weight w. **Figure 8: Cross-validation for the size of the short list K.**

global similarity and local similarity, respectively. We set $w = 0.9$ and $K = 5,000$.

4.5 Comparison with State-of-the-Art Methods

We compare our algorithm with state-of-the-art methods including global similarity based method such as DIR [10, 11] and local similarity based methods such as R-Match [34] and DELF [23] on Oxford, Paris and INSTRE datasets. All of the implementations of the compared methods, except DELF [23] and siaMAC [32], use the ResNet-101 model released by Gordo *et al.* [11] and a single scale as input. All implementations are carefully reproduced using the public source code released by the original authors. As shown in the first part of Table 6, our method trained with only the Landmark-clean-half outperforms all the other methods (compared without any post-processing). It can achieve a further gain when training with both synthetic data and real data. Again, with query expansion, our method performs best on all the datasets.

Table 5: Comparing the performance of using different numbers of subvectors for compressing local features. $*$ denotes the results after fine-tuning the object localization network with quantization. *Full* denotes the results without any memory and storage compression. *CR* denotes the compression ratio.

M	32	32*	64	64*	128	128*	Full
CR	256	256	128	128	64	64	-
Oxf5k	84.9	87.3	88.1	89.0	90.4	90.7	**91.6**
Par6k	93.2	94.2	94.3	94.8	94.8	95.1	**95.3**

Table 6: Comparison with state-of-the-art methods. Our methods are marked with ⋆. lch denotes the model trained with Landmark-clean-half only. lch+syn denotes the model trained with both Landmark-clean-half and CocoSyn. QE denotes query expansion.

	Oxf5k	Par5k	Oxf105k	Par106k	OxfShf	ParShf	Ins
Without post-processing							
siaMAC [32]	77.7	84.1	70.1	76.8	-	-	-
DIR-RMAC [11]	83.9	93.8	80.8	89.9	74.7	81.5	62.6
R-Match [11, 34]	88.1	94.9	85.7	91.3	83.5	86.9	71.0
DELF [23]	83.8	85.0	82.6	81.7	83.9	84.2	-
⋆(lch)	90.5	**95.7**	88.6	92.5	90.4	**93.3**	71.1
⋆(syn-lch)	**90.8**	**95.7**	**88.9**	**93.0**	**90.5**	92.7	**76.5**
With query expansion							
siaMAC+QE [32]	82.9	85.6	77.9	78.3	-	-	-
DIR-RMAC+QE [11]	89.6	95.3	88.3	92.7	75.2	82.1	70.5
R-Match+QE [11, 34]	91.0	95.5	89.6	92.5	84.9	86.8	77.1
DELF+DIR+QE [11, 23]	90.0	95.7	88.5	92.8	84.4	84.6	-
⋆+QE(lch)	91.5	**95.8**	90.0	92.8	90.9	**92.4**	75.2
⋆+QE(syn-lch)	**91.9**	**95.8**	**90.4**	**93.3**	**91.3**	91.7	**78.2**

5 CONCLUSIONS

We proposed an end-to-end trainable CNN for local similarity learning by modeling the problem of object retrieval as object localization. Our CNN consists of a feature extraction subnetwork and an object localization subnetwork. Correlation layer is used to capture the spatial relations in images. We found that the correlation volume encodes whether the spatial relations of the gallery and those of the query are consistent or not. Thanks to the spatial relation harvesting, the proposed local similarity has excellent retrieval performance, and is complementary to global similarity. Besides, we proposed a scalable retrieval algorithm, by utilizing product quantization to compress gallery features, and global similarity to prefilter the gallery images and enhance the search results. Extensive experiments on challenging benchmarks demonstrate the effectiveness of the proposed algorithm.

REFERENCES

[1] R. Arandjelović, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. 2016. NetVLAD: CNN architecture for weakly supervised place recognition. In *CVPR*.

[2] Relja Arandjelovic and Andrew Zisserman. 2013. All About VLAD. In *CVPR*. 1578–1585.

[3] Yannis Avrithis and Yannis Kalantidis. 2012. Approximate gaussian mixtures for large scale vocabularies. In *ECCV*. 15–28.

[4] Artem Babenko, Anton Slesarev, Alexandr Chigorin, and Victor Lempitsky. 2014. Neural codes for image retrieval. In *ECCV*. 584–599.

[5] Zhenfang Chen, Zhanghui Kuang, Kwan-Yee K. Wong, and Wei Zhang. 2017. Aggregated Deep Feature from Activation Clusters for Particular Object Retrieval. In *Thematic Workshops, MM*. 44–51.

[6] Ondrej Chum, Andrej Mikulik, Michal Perdoch, and Jiri Matas. 2011. Total recall II: Query expansion revisited. In *CVPR*. 889–896.

[7] Ondrej Chum, James Philbin, and Josef Sivic. 2007. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *ICCV*.

[8] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick van der Smagt, Daniel Cremers, and Thomas Brox. 2015. Flownet: Learning optical flow with convolutional networks. In *ICCV*.

[9] Debidatta Dwibedi, Ishan Misra, and Martial Hebert. 2017. Cut, Paste and Learn: Surprisingly Easy Synthesis for Instance Detection. In *ICCV*.

[10] Albert Gordo, Jon Almazán, Jerome Revaud, and Diane Larlus. 2016. Deep image retrieval: Learning global representations for image search. In *ECCV*. 241–257.

[11] Albert Gordo, Jon Almazan, Jerome Revaud, and Diane Larlus. 2017. End-to-end learning of deep visual representations for image retrieval. *IJCV* 124, 2 (2017), 237–254.

[12] Xufeng Han, Thomas Leung, Yangqing Jia, Rahul Sukthankar, and Alexander C Berg. 2015. Matchnet: Unifying feature and metric learning for patch-based matching. In *CVPR*. 3279–3286.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*. 770–778.

[14] Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondrej Chum. 2018. Mining on manifolds: Metric learning without labels. In *CVPR*. 7642–7651.

[15] Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, Teddy Furon, and Ondrej Chum. 2017. Efficient diffusion on region manifolds: Recovering small objects with compact cnn representations. In *CVPR*.

[16] Hervé Jégou and Matthijs Douze. 2010. Aggregating local descriptors into a compact image representation. In *CVPR*.

[17] Hervé Jégou, Matthijs Douze, and Cordelia Schmid. 2011. Product quantization for nearest neighbor search. *TPAMI* 33, 1 (jan 2011), 117–128.

[18] Yannis Kalantidis, Clayton Mellina, and Simon Osindero. 2016. Cross-dimensional weighting for aggregated deep convolutional features. In *ECCVW*. 42.

[19] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *ECCV*. Springer, 740–755.

[20] Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji. 2015. Bilinear cnn models for fine-grained visual recognition. In *ICCV*. 1449–1457.

[21] David G Lowe. 2004. Distinctive image features from scale-invariant keypoints. *IJCV* 60, 2 (2004), 91–110.

[22] David Nister and Henrik Stewenius. 2006. Scalable recognition with a vocabulary tree. In *CVPR*, Vol. 2. 2161–2168.

[23] Hyeonwoo Noh, Andre Araujo, Jack Sim, Tobias Weyand, and Bohyung Han. 2017. Large-Scale Image Retrieval with Attentive Deep Local Features. In *ICCV*.

[24] Adam Paszke, Sam Gross, and Soumith Chintala. 2017. PyTorch. (2017).

[25] Michal Perdoch, Ondrej Chum, and Jiri Matas. 2009. Efficient representation of local geometry for large scale object retrieval. In *CVPR*. 9–16.

[26] Florent Perronnin and Christopher Dance. 2007. Fisher kernels on visual vocabularies for image categorization. In *CVPR*. 1–8.

[27] Florent Perronnin, Yan Liu, Jorge Sánchez, and Herve Poirier. 2010. Large-scale image retrieval with compressed fisher vectors. In *CVPR*. 3384–3391.

[28] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. 2007. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*. 1–8.

[29] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. 2007. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*. 1–8.

[30] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and A Zisserman. 2008. Lost in Quantization: Improving Particular Object Retrieval in Large Scale Image Databases. In *CVPR*.

[31] Filip Radenović, Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, and Ondrej Chum. 2018. Revisiting oxford and paris: Large-scale image retrieval benchmarking. In *CVPR*. 5706–5715.

[32] Filip Radenović, Giorgos Tolias, and Ondrej Chum. 2016. CNN image retrieval learns from BoW: Unsupervised fine-tuning with hard examples. In *ECCV*. 3–20.

[33] Ali Sharif Razavian, Hossein Azizpour, Josephine Sullivan, Stefan Carlsson, Ali Sharif, Razavian Hossein, Azizpour Josephine, Sullivan Stefan, and K T H Royal. 2014. CNN Features off-the-shelf : an Astounding Baseline for Recognition. In *CVPRW*. 512–519.

[34] Ali Sharif Razavian, Josephine Sullivan, Stefan Carlsson, and Atsuto Maki. 2014. Visual instance retrieval with deep convolutional networks. *arXiv preprint arXiv:1412.6574* (2014).

[35] Ali Sharif Razavian, Josephine Sullivan, Atsuto Maki, and Stefan Carlsson. 2014. A Baseline for Visual Instance Retrieval with Deep Convolutional Networks. *CoRR* abs/1412.6574 (2014).

[36] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. 2015. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*. 91–99.

[37] Ignacio Rocco, Relja Arandjelović, and Josef Sivic. 2017. Convolutional neural network architecture for geometric matching. *CVPR* (2017).

[38] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, and Others. 2015. Imagenet large scale visual recognition challenge. *IJCV* 115, 3 (2015), 211–252.

[39] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2016. Grad-cam: Visual explanations from deep networks via gradient-based localization. *See https://arxiv. org/abs/1610.02391 v3* 7, 8 (2016).

[40] Karen Simonyan and Andrew Zisserman. 2015. Very deep convolutional networks for large-scale image recognition. In *ICLR*.

[41] Josef Sivic and Andrew Zisserman. 2003. Video Google: a text retrieval approach to object matching in videos. In *ICCV*, Vol. 2. 1470–1477.

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

[43] Yannick Verdie, Kwang Yi, Pascal Fua, and Vincent Lepetit. 2015. TILDE: A temporally invariant learned detector. In *CVPR*. 5279–5288.

[44] Shuang Wang and Shuqiang Jiang. 2015. INSTRE: a new benchmark for instance-level object retrieval and recognition. *TOMM* 11, 3 (2015), 37.

[45] Artem Babenko Yandex and Victor Lempitsky. 2015. Aggregating local deep features for image retrieval. In *ICCV*.

[46] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. 2016. LIFT: Learned invariant feature transform. In *ECCV*. 467–483.

[47] Sergey Zagoruyko and Nikos Komodakis. 2015. Learning to compare image patches via convolutional neural networks. In *CVPR*. 4353–4361.

[48] Liang Zheng, Yi Yang, and Qi Tian. 2017. SIFT meets CNN: A decade survey of instance retrieval. *TPAMI* (2017).