Deep Image Retrieval: Learning Global Representations for Image Search

Albert Gordo[(✉)], Jon Almazán, Jerome Revaud, and Diane Larlus

Computer Vision Group, Xerox Research Center Europe, Meylan, France
{Albert.Gordo,Jon.Almazan,Jerome.Revaud,Diane.Larlus}@xrce.xerox.com

Abstract. We propose a novel approach for instance-level image retrieval. It produces a global and compact fixed-length representation for each image by aggregating many region-wise descriptors. In contrast to previous works employing pre-trained deep networks as a black box to produce features, our method leverages a deep architecture trained for the specific task of image retrieval. Our contribution is twofold: (i) we leverage a ranking framework to learn convolution and projection weights that are used to build the region features; and (ii) we employ a region proposal network to learn which regions should be pooled to form the final global descriptor. We show that using clean training data is key to the success of our approach. To that aim, we use a large scale but noisy landmark dataset and develop an automatic cleaning approach. The proposed architecture produces a global image representation in a single forward pass. Our approach significantly outperforms previous approaches based on global descriptors on standard datasets. It even surpasses most prior works based on costly local descriptor indexing and spatial verification. Additional material is available at www.xrce.xerox. com/Deep-Image-Retrieval.

Keywords: Deep learning · Instance-level retrieval

1 Introduction

Since their ground-breaking results on image classification in recent ImageNet challenges [29,50], deep learning based methods have shined in many other computer vision tasks, including object detection [14] and semantic segmentation [31]. Recently, they also rekindled highly semantic tasks such as image captioning [12,28] and visual question answering [1]. However, for some problems such as *instance-level image retrieval*, deep learning methods have led to rather underwhelming results. In fact, for most image retrieval benchmarks, the state of the art is currently held by conventional methods relying on local descriptor matching and re-ranking with elaborate spatial verification [30,34,58,59].

Recent works leveraging deep architectures for image retrieval are mostly limited to using a pre-trained network as local feature extractor. Most efforts have been devoted towards designing image representations suitable for image retrieval on top of those features. This is challenging because representations for

retrieval need to be compact while retaining most of the fine details of the images. Contributions have been made to allow deep architectures to accurately represent input images of different sizes and aspect ratios [5,27,60] or to address the lack of geometric invariance of convolutional neural network (CNN) features [15,48].

In this paper, we focus on *learning* these representations. We argue that one of the main reasons for the deep methods lagging behind the state of the art is the lack of supervised learning for the specific task of instance-level image retrieval. At the core of their architecture, CNN-based retrieval methods often use local features extracted using networks pre-trained on ImageNet for a classification task. These features are learned to distinguish between different semantic categories, but, as a side effect, are quite robust to intra-class variability. This is an undesirable property for instance retrieval, where we are interested in distinguishing between particular objects – even if they belong to the same semantic category. Therefore, learning features for the specific task of instance-level retrieval seems of paramount importance to achieve competitive results.

To this end, we build upon a recent deep representation for retrieval, the regional maximum activations of convolutions (R-MAC) [60]. It aggregates several image regions into a compact feature vector of fixed length and is thus robust to scale and translation. This representation can deal with high resolution images of different aspect ratios and obtains a competitive accuracy. We note that all the steps involved to build the R-MAC representation are differentiable, and so its weights can be learned in an end-to-end manner. Our **first contribution** is thus to use a *three-stream Siamese network* that explicitly optimizes the weights of the R-MAC representation for the image retrieval task by using a triplet ranking loss (Fig. 1).

To train this network, we leverage the public Landmarks dataset [6]. This dataset was constructed by querying image search engines with names of different landmarks and, as such, exhibits a very large amount of mislabeled and false positive images. This prevents the network from learning a good representation. We propose an automatic cleaning process, and show that on the cleaned data learning significantly improves.

Our **second contribution** consists in learning the pooling mechanism of the R-MAC descriptor. In the original architecture of [60], a rigid grid determines the location of regions that are pooled together. Here we propose to predict the location of these regions given the image content. We train a region proposal network with bounding boxes that are estimated for the Landmarks images as a by-product of the cleaning process. We show quantitative and qualitative evidence that region proposals significantly outperform the rigid grid.

The combination of our two contributions produces a novel architecture that is able to encode one image into a compact fixed-length vector in a single forward pass. Representations of different images can be then compared using the dot-product. Our method significantly outperforms previous approaches based on global descriptors. It even outperforms more complex approaches that involve keypoint matching and spatial verification at test time.
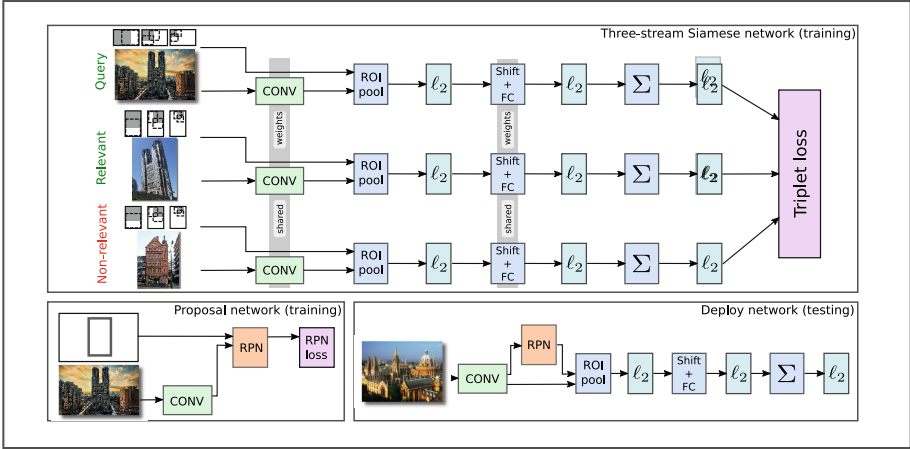
Fig. 1. Summary of the proposed CNN-based representation tailored for retrieval. At training time, image triplets are sampled and simultaneously considered by a *triplet-loss* that is well-suited for the task (top). A *region proposal network* (RPN) learns which image regions should be pooled (bottom left). At test time (bottom right), the query image is fed to the learned architecture to efficiently produce a *compact global image representation* that can be compared with the dataset image representations with a simple dot-product.

Finally, we would like to refer the reader to the recent work of Radenovic *et al.* [47], concurrent to ours and published in these same proceedings, that also proposes to learn representations for retrieval using a Siamese network on a geometrically-verified landmark dataset.

The rest of the paper is organized as follows. Section 2 discusses related works. Sections 3 and 4 present our contributions. Section 5 validates them on five different datasets. Finally Sect. 6 concludes the paper.

2   Related Work

We now describe previous works most related to our approach.

Conventional image retrieval. Early techniques for instance-level retrieval are based on bag-of-features representations with large vocabularies and inverted files [37,44]. Numerous methods to better approximate the matching of the descriptors have been proposed, see *e.g.* [24,35]. An advantage of these techniques is that spatial verification can be employed to re-rank a short-list of results [39,44], yielding a significant improvement despite a significant cost. Concurrently, methods that aggregate the local image patches have been considered. Encoding techniques, such as the Fisher Vector [40], or VLAD [25], combined with compression [22,42,46] produce global descriptors that scale to larger databases at the cost of reduced accuracy. All these methods can be combined with other post-processing techniques such as query expansion [3,8,9].

CNN-based retrieval. After their success in classification [29], CNN features were used as off-the-shelf features for image retrieval [6, 48]. Although they outperform other standard global descriptors, their performance is significantly below the state of the art. Several improvements were proposed to overcome their lack of robustness to scaling, cropping and image clutter. [48] performs region cross-matching and accumulates the maximum similarity per query region. [5] applies sum-pooling to whitened region descriptors. [27] extends [5] by allowing cross-dimensional weighting and aggregation of neural codes. Other approaches proposed hybrid models involving an encoding technique such as FV [41] or VLAD [15, 38], potentially learnt as well [2] as one of their components.

Tolias *et al.* [60] propose R-MAC, an approach that produces a global image representation by aggregating the activation features of a CNN in a fixed layout of spatial regions. The result is a fixed-length vector representation that, when combined with re-ranking and query expansion, achieves results close to the state of the art. Our work extends this architecture by discriminatively learning the representation parameters and by improving the region pooling mechanism.

Fine-tuning for retrieval. Babenko *et al.* [6] showed that models pre-trained on ImageNet for object classification could be improved by fine-tuning them on an external set of Landmarks images. In this paper we confirm that fine-tuning the pre-trained models for the retrieval task is indeed crucial, but argue that one should use a good image representation (R-MAC) and a ranking loss instead of a classification loss as used in [6].

Localization/Region pooling. Retrieval methods that ground their descriptors in regions typically consider random regions [48] or a rigid grid of regions [60]. Some works exploit the center bias that benchmarks usually exhibit to weight their regions accordingly [5]. The spatial transformer network of [21] can be inserted in CNN architectures to transform input images appropriately, including by selecting the most relevant region for the task. In this paper, we would like to bias our descriptor towards interesting regions without paying an extra-cost or relying on a central bias. We achieve this by using a proposal network similar in essence to the Faster R-CNN detection method [49].

Siamese networks and metric learning. Siamese networks have commonly been used for metric learning [55], dimensionality reduction [17], learning image descriptors [53], and performing face identification [7, 20, 56]. Recently triplet networks (i.e. three stream Siamese networks) have been considered for metric learning [19, 62] and face identification [51]. However, these Siamese networks usually rely on simpler network architectures than the one we use here, which involves pooling and aggregation of several regions.

3   Method

This section introduces our method for retrieving images in large collections. We first revisit the R-MAC representation (Sect. 3.1) showing that, despite its handcrafted nature, all of its components consist of differentiable operations.

From this it follows that one can learn the weights of the R-MAC representation in an end-to-end manner. To that aim we leverage a three-stream Siamese network with a triplet ranking loss. We also describe how to learn the pooling mechanism using a region proposal network (RPN) instead of relying on a rigid grid (Sect. 3.2). Finally we depict the overall descriptor extraction process for a given image (Sect. 3.3).

3.1 Learning to Retrieve Particular Objects

R-MAC revisited. Recently, Tolias *et al.* [60] presented R-MAC, a global image representation particularly well-suited for image retrieval. The R-MAC extraction process is summarized in any of the three streams of the network in Fig. 1 (top). In a nutshell, the convolutional layers of a pre-trained network (*e.g.* VGG16 [54]) are used to extract activation features from the images, which can be understood as local features that do not depend on the image size or its aspect ratio. Local features are max-pooled in different regions of the image using a multi-scale rigid grid with overlapping cells. These pooled region features are independently ℓ_2-normalized, whitened with PCA and ℓ_2-normalized again. Unlike spatial pyramids, instead of concatenating the region descriptors, they are sum-aggregated and ℓ_2-normalized, producing a compact vector whose size (typically $256 - 512$ dimensions) is independent of the number of regions in the image. Comparing two image vectors with dot-product can then be interpreted as an approximate many-to-many region matching.

One key aspect to notice is that *all these operations are differentiable.* In particular, the spatial pooling in different regions is equivalent to the *Region of Interest* (ROI) pooling [18], which is differentiable [13]. The PCA projection can be implemented with a shifting and a fully connected (FC) layer, while the gradients of the sum-aggregation of the different regions and the ℓ_2-normalization are also easy to compute. Therefore, one can implement a network architecture that, given an image and the precomputed coordinates of its regions (which depend only on the image size), produces the final R-MAC representation in a single forward pass. More importantly, *one can backpropagate through the network architecture to learn the optimal weights of the convolutions and the projection.*

Learning for particular instances. We depart from previous works on fine-tuning networks for image retrieval that optimize classification using cross-entropy loss [6]. Instead, we consider a ranking loss based on image triplets. It explicitly enforces that, given a query, a relevant element to the query and a non-relevant one, the relevant one is closer to the query than the other one. To do so, we use a three-stream Siamese network in which the weights of the streams are shared, see Fig. 1 top. Note that the number and size of the weights in the network (the convolutional filters and the shift and projection) is independent of the size of the images, and so we can feed each stream with images of different sizes and aspect ratios.

Let I_q be a query image with R-MAC descriptor q, I^+ be a relevant image with descriptor d^+, and I^- be a non-relevant image with descriptor d^-.

We define the ranking triplet loss as

$$L(I_q, I^+, I^-) = \frac{1}{2}\max(0, m + \|q - d^+\|^2 - \|q - d^-\|^2),\qquad(1)$$

where m is a scalar that controls the margin. Given a triplet with non-zero loss, the gradient is back-propagated through the three streams of the network, and the convolutional layers together with the "PCA" layers – the shifting and the fully connected layer – get updated.

This approach offers several advantages. First and foremost, we directly optimize a ranking objective. Second, we can train the network using images at the same (high) resolution that we use at test time[1]. Last, learning the optimal "PCA" can be seen as a way to perform discriminative large-margin metric learning [63] in which one learns a new space where relevant images are closer.

3.2   Beyond Fixed Regions: Proposal Pooling

The rigid grid used in R-MAC [60] to pool regions tries to ensure that the object of interest is covered by at least one of the regions. However, this uniform sampling poses two problems. First, as the grid is independent of the image content, it is unlikely that any of the grid regions accurately align with the object of interest. Second, many of the regions only cover background. This is problematic as the comparison between R-MAC signatures can be seen as a many-to-many region matching: image clutter will negatively affect the performance. Note that both problems are coupled: increasing the number of grid regions improves the coverage, but also the number of irrelevant regions.

We propose to replace the rigid grid with region proposals produced by a Region Proposal Network (RPN) trained to localize regions of interest in images. Inspired by the approach of Ren *et al.* [49], we model this process with a fully-convolutional network built on top of the convolutional layers of R-MAC (see bottom-left part of Fig. 1). This allows one to get the region proposals at almost zero cost. By using region proposals instead of the rigid grid we address both problems. First, the region proposals typically cover the object of interest more tightly than the rigid grid. Second, even if they do not overlap exactly with the region of interest, most of the proposals do overlap significantly with it (see Sect. 5.3), which means that increasing the number of proposals per image not only helps to increase the coverage but also helps in the many-to-many matching.

The main idea behind an RPN is to predict, for a set of candidate boxes of various sizes and aspects ratio, and at all possible image locations, a score describing how likely each box contains an object of interest. Simultaneously, for each candidate box it performs regression to improve its location. This is achieved by a fully-convolutional network consisting of a first layer that uses 3×3 filters, and two sibling convolutional layers with 1×1 filters that predict, for each candidate box in the image, both the *objectness* score and the regressed location. Non-maximum

---

[1] By contrast, fine-tuning networks such as VGG16 for classification using high-resolution images is not straightforward.

suppression is then performed on the ranked boxes to produce k final proposals per image that are used to replace the rigid grid.

To train the RPN, we assign a binary class label to each candidate box, depending on how much the box overlaps with the ground-truth region of interest, and we minimize an objective function with a multi-task loss that combines a classification loss (log loss over object *vs* background classes) and a regression loss (smooth ℓ_1 [13]). This is then optimized by backpropagation and stochastic gradient descent (SGD). For more details about the implementation and the training procedure of the RPNs, we refer the reader to [49].

We note that one could, in principle, learn the RPN *and* the ranking of the images simultaneously. However, preliminary experiments showed that correctly weighting both losses was difficult and led to unstable results. In our experiments, we first learn the R-MAC representation using a rigid grid, and only then we fix the convolutional layers and learn the RPN, which replaces the rigid grid.

3.3   Building a Global Descriptor

At test time, one can easily use this network to represent a high-resolution image. One feeds the image to the network, which produces the region proposals, pools the features inside the regions, embeds them into a more discriminative space, aggregates them, and normalizes them. All these operations happen in a single forward pass (see bottom-right part of Fig. 1). This process is also quite efficient: we can encode approximately 5 high-resolution (*i.e.* 724 pixels for the largest side) images per second using a single Nvidia K40 GPU.

4   Leveraging Large-Scale Noisy Data

To train our network for instance-level image retrieval we leverage a large-scale image dataset, the **Landmarks** dataset [6], that contains approximately 214K images of 672 famous landmark sites. Its images were collected through textual queries in an image search engine without thorough verification. As a consequence, they comprise a large variety of profiles: general views of the site, close-ups of details like statues or paintings, with all intermediate cases as well, but also site map pictures, artistic drawings, or even completely unrelated images, see Fig. 2.

We could only download a subset of all images due to broken URLs. After manual inspection, we merged some classes together due to partial overlap. We also removed classes with too few images. Finally, we meticulously removed all classes having an overlap with the Oxford 5k, Paris 6k, and Holidays datasets, on which we experiment, see Sect. 5. We obtained a set of about 192,000 images divided into 586 landmarks. We refer to this set as **Landmarks-full**. For our experiments, we use 168,882 images for the actual fine-tuning, and the 20,668 remaining ones to validate parameters.

Cleaning the Landmarks dataset. As we have mentioned, the Landmarks dataset present a large intra-class variability, with a wide variety of views and
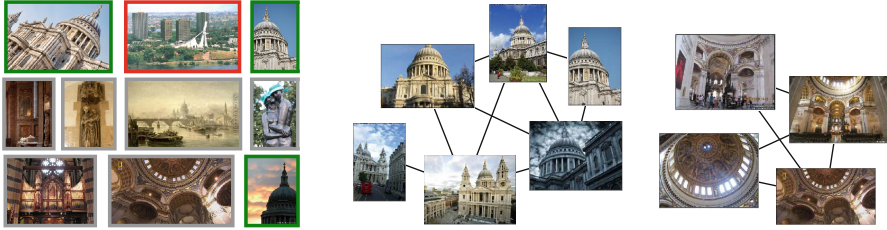
Fig. 2. Left: random images from the "St Paul's Cathedral" landmark. Green, gray and red borders resp. denote prototypical, non-prototypical, and incorrect images. **Right**: excerpt of the two largest connected components of the pairwise matching graph (corresponding to outside and inside pictures of the cathedral). (Color figure online)

profiles, and a non-negligible amount of unrelated images (Fig. 2). While this is not a problem when aiming for classification (the network can accommodate during training for this diversity and even for noise), for instance-level matching we need to train the network with images of the same particular object or scene. In this case, variability comes from different viewing scales, angles, lighting conditions and image clutter. We pre-process the Landmarks dataset to achieve this as follows.

We first run a strong image matching baseline within the images of each landmark class. We compare each pair of images using invariant keypoint matching and spatial verification [32]. We use the SIFT and Hessian-Affine keypoint detectors [32,33] and match keypoints using the first-to-second neighbor ratio rule [32]. This is known to outperform approaches based on descriptor quantization [43]. Afterwards, we verify all matches with an affine transformation model [44]. This heavy procedure is affordable as it is performed offline only once at training time.

Without loss of generality, we describe the rest of the cleaning procedure for a single landmark class. Once we have obtained a set of pairwise scores between all image pairs, we construct a graph whose nodes are the images and edges are pairwise matches. We prune all edges which have a low score. Then we extract the connected components of the graph. They correspond to different profiles of a landmark; see Fig. 2 that shows the two largest connected components for St Paul's Cathedral. In order to avoid any confusion, we only retain the largest connected component and discard the rest. This cleaning process leaves about 49,000 images (divided in 42,410 training and 6382 validation images) still belonging to one of the 586 landmarks, referred to as **Landmarks-clean**.

Bounding box estimation. Our second contribution (Sect. 3.2) is to replace the uniform sampling of regions in the R-MAC descriptor by a learned ROI selector. This selector is trained using bounding box annotations that we automatically estimate for all landmark images. To that aim we leverage the data obtained during the cleaning step. The position of verified keypoint matches is a meaningful cue since the object of interest is consistently visible across the
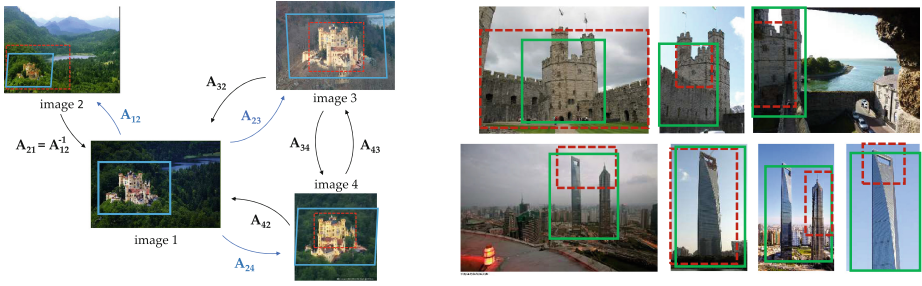
Fig. 3. Left: the bounding box from image 1 is projected into its graph neighbors using the affine transformations (blue rectangles). The current bounding box estimates (dotted red rectangles) are then updated accordingly. The diffusion process repeats through all edges until convergence. **Right**: initial and final bounding box estimates (resp. dotted red and plain green rectangles). (Color figure online)

landmark's pictures, whereas distractor backgrounds or foreground objects are varying and hence unmatched.

We denote the union of the connected components from all landmarks as a graph $\mathcal{S} = \{\mathcal{V}_\mathcal{S}, \mathcal{E}_\mathcal{S}\}$. For each pair of connected images $(i, j) \in \mathcal{E}_\mathcal{S}$, we have a set of verified keypoint matches with a corresponding affine transformation A_{ij}. We first define an initial bounding box in both images i and j, denoted by B_i and B_j, as the minimum rectangle enclosing all matched keypoints. Note that a single image can be involved in many different pairs. In this case, the initial bounding box is the geometric median of all boxes[2], efficiently computed with [61]. Then, we run a diffusion process, illustrated in Fig. 3, in which for a pair (i, j) we predict the bounding box B_j using B_i and the affine transform A_{ij} (and conversely). At each iteration, bounding boxes are updated as: $B'_j = (\alpha - 1)B_j + \alpha A_{ij}B_i$, where α is a small update step (we set $\alpha = 0.1$ in our experiments). Again, the multiple updates for a single image are merged using geometric median, which is robust against poorly estimated affine transformations. This process iterates until convergence. As can be seen in Fig. 3, the locations of the bounding boxes are improved as well as their consistency across images.

5   Experiments

We now present our experimental results. We start by describing the datasets and experimental details (Sect. 5.1). We then evaluate our proposed ranking network (Sect. 5.2) and the region proposal pooling (Sect. 5.3). Finally, we compare our results to the state of the art (Sect. 5.4).

---

[2] Geometric median is robust to outlier boxes compared to *e.g.* averaging.

5.1   Datasets and Experimental Details

Datasets. We evaluate our approach on five standard datasets. We experiment mostly with the **Oxford 5k** building dataset [44] and the **Paris 6k** dataset [45], that contain respectively $5,062$ and $6,412$ images. For both datasets there are 55 query images, each annotated with a region of interest. To test instance-level retrieval on a larger-scale scenario, we also consider the **Oxford 105k** and the **Paris 106k** datasets that extend Oxford 5k and Paris 6k with 100k distractor images from [44]. Finally, the INRIA **Holidays** dataset [23] is composed of 1,491 images and 500 different scene queries.

Evaluation. For all datasets we use the standard evaluation protocols and report mean Average Precision (mAP). As is standard practice, in Oxford and Paris one uses only the annotated region of interest of the query, while for Holidays one uses the whole query image. Furthermore, the query image is removed from the dataset when evaluating on Holidays, but not on Oxford or Paris.

Experimental details. Our experiments use the very deep network (VGG16) of Simonyan *et al.* [54] pre-trained on the ImageNet ILSVRC challenge as a starting point. All further learning is performed on the Landmarks dataset unless explicitly noted. To perform fine-tuning with classification [6] we follow standard practice and resize the images to multiple scales (shortest side in the $[256 - 512]$ range) and extract random crops of 224×224 pixels. This fine-tuning process took approximately 5 days on a single Nvidia K40 GPU. When performing fine-tuning with the ranking loss, it is crucial to mine hard triplets in an efficient manner, as random triplets will mostly produce easy triplets or triplets with no loss. As a simple yet effective approach, we first perform a forward pass on approximately ten thousand images to obtain their representations. We then compute the losses of all the triplets involving those features (with margin $m = 0.1$), which is fast once the representations have been computed. We finally sample triplets with a large loss, which can be seen as hard negatives. We use them to train the network with SGD with momentum, with a learning rate of 10^{-3} and weight decay of $5 \cdot 10^{-5}$. Furthermore, as images are large, we can not feed more than one triplet in memory at a time. To perform batched SGD we accumulate the gradients of the backward passes and only update the weights every n passes, with $n = 64$ in our experiments. To increase efficiency, we only mine new hard triplets every 16 network updates. Following this process, we could process approximately 650 batches of 64 triplets per day on a single K40 GPU. We processed approximately 2000 batches in total, *i.e.*, 3 days of training. To learn the RPN, we train the net for 200k iterations with a weight decay of $5 \cdot 10^{-5}$ and a learning rate of 10^{-3}, which is decreased by a factor of 10 after 100k iterations. This process took less than 24 h.

5.2   Influence of Fine-Tuning the Representation

In this section we report retrieval experiments for the baselines and our ranking loss-based approach. All results are summarized in Table 1. First of all, as can

Table 1. Comparison of R-MAC [60], our reimplementation of it and the learned versions fine-tuned for classification on the full and the clean sets (C-Full and C-Clean) and fine-tuned for ranking on the clean set (R-Clean). All these results use the initial regular grid with no RPN.

Dataset	PCA	R-MAC		Learned R-MAC		
		[60]	Reimp.	C-Full	C-Clean	R-Clean
Oxford 5k	PCA Paris	66.9	66.9	-	-	-
	PCA Landmarks	-	66.2	74.8	75.2	81.1
Paris 6k	PCA Oxford	83.0	83.0	-	-	-
	PCA Landmarks	-	82.3	82.5	83.2	86.0

be seen in the first and second columns, the accuracy of our reimplementation of R-MAC is identical to the one of the original paper. We would also like to highlight the following points:

PCA learning. R-MAC [60] learns the PCA on different datasets depending on the target dataset (*i.e.* learned on Paris when evaluating on Oxford and vice versa). A drawback of this is that different models need to be generated depending on the target dataset. Instead, we use the Landmarks dataset to learn the PCA. This leads to a slight decrease in performance, but allows us to have a single universal model that can be used for all datasets.

Fine-tuning for classification. We evaluate the approach of Babenko *et al.* [6], where the original network pre-trained on ImageNet is fine-tuned on the Landmarks dataset on a classification task. We fine-tune the network with both the complete and the clean versions of Landmarks, denoted by *C-Full* and *C-Clean* in the table. This fine-tuning already brings large improvements over the original results. Also worth noticing is that, in this case, cleaning the dataset seems to bring only marginal improvements over using the complete dataset.

Fine-tuning for retrieval. We report results using the proposed ranking loss (Sect. 3.1) in the last column, denoted by *R-Clean*. We observe how this brings consistent improvements over using the less-principled classification fine-tuning. Contrary to the latter, we found of paramount importance to train our Siamese network using the clean dataset, as the triplet-based training process is less tolerant to outliers. Figure 4 (left) illustrates these findings by plotting the mAP obtained on Oxford 5k at several training epochs for different settings. It also shows the importance of initializing the network with a model that was first fine-tuned for classification on the full landmarks dataset. Even if *C-Full* and *C-Clean* obtain very similar scores, we speculate that the model trained with the full Landmark dataset has seen more diverse images so its weights are a better starting point.

Image size. R-MAC [60] finds important to use high resolution images (longest side resized to 1024 pixels). In our case, after fine-tuning, we found no noticeable
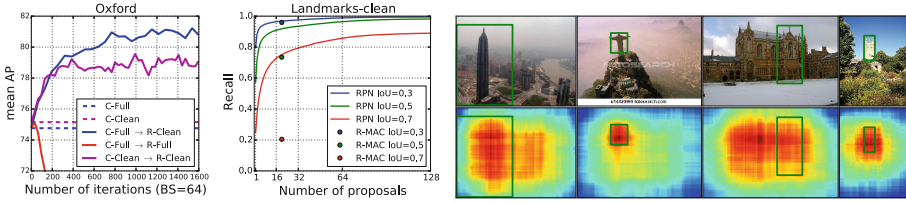
Fig. 4. Left: evolution of mAP when learning with a rank-loss for different initializations and training sets. **Middle**: landmark detection recall of our learned RPN for several IoU thresholds compared to the R-MAC fixed grid. **Right**: heat-map of the coverage achieved by our proposals on images from the Landmark and the Oxford 5k datasets. Green rectangles are ground-truth bounding boxes. (Color figure online)

Table 2. Proposals network. mAP results for Oxford 5k and Paris 6k obtained with a fixed-grid R-MAC, and our proposal network, for an increasingly large number of proposals, before and after fine-tuning with a ranking-loss. The rigid grid extracts, on average, 20 regions per image.

Dataset	Model	**Grid**	**# Region Proposals**					
			16	32	64	128	192	256
Oxford 5k	C-Full	74.8	74.9	75.3	75.3	76.4	76.7	76.8
	R-Clean	81.1	81.5	82.1	82.6	82.8	83.1	83.1
Paris 6k	C-Full	82.5	81.8	82.8	83.4	84.0	84.1	84.2
	R-Clean	86.0	85.4	86.2	86.7	86.9	87.0	87.1

difference in accuracy between 1024 and 724 pixels. All further experiments resize images to 724 pixels, significantly speeding up the image encoding and training.

5.3   Evaluation of the Proposal Network

In this section we evaluate the effect of replacing the rigid grid of R-MAC with the regions produced by the proposal network.

Evaluating proposals. We first evaluate the relevance of the regions predicted by our proposal network. Figure 4 (middle) shows the detection recall obtained in the validation set of Landmarks-Clean for different IoU (intersection over union) levels as a function of the number of proposals, and compares it with the recall obtained by the rigid grid of R-MAC. The proposals obtain significantly higher recall than the rigid grid even when their number is small. This is consistent with the quantitative results (Table 2), where 32–64 proposals already outperform the rigid regions. Figure 4 (right) visualizes the proposal locations as a heat-map on a few sample images of Landmarks and Oxford 5k. It clearly shows that the proposals are centered around the objects of interest. For the Oxford 5k images, the query boxes are somewhat arbitrarily defined. In this case, as expected, our proposals naturally align with the entire landmark in a query agnostic way.

Table 3. Accuracy comparison with the state of the art. Methods marked with an [a] use the full image as a query in Oxford and Paris instead of using the annotated region of interest as is standard practice. Methods with a [b] manually rotate Holidays images to fix their orientation. [c] denotes our reimplementation. We do not report QE results on Holidays as it is not a standard practice.

	Method	Dim.	Datasets				
			Oxf5k	Par6k	Oxf105k	Par106k	Holidays
Global descriptors	Jégou & Zisserman [26]	1024	56.0	-	50.2	-	72.0
	Jégou & Zisserman [26]	128	43.3	-	35.3	-	61.7
	Gordo *et al.* [16]	512	-	-	-	-	79.0
	Babenko *et al.* [6]	128	55.7^a	-	52.3^a	-	$75.9/78.9^b$
	Gong *et al.* [15]	2048	-	-	-	-	80.8
	Babenko & Lempitsky [5]	256	53.1	-	50.1	-	80.2^c
	Ng *et al.* [36]	128	59.3^a	59.0^a	-	-	83.6
	Paulin *et al.* [38]	256K	56.5	-	-	-	79.3
	Perronnin & Larlus [41]	4000	-	-	-	-	84.7
	Tolias *et al.* [60]	512	66.9	83.0	61.6	75.7	$85.2^c/86.9^{c,b}$
	Kalantidis *et al.* [27]	512	68.2	79.7	63.3	71.0	84.9
	Arandjelovic *et al.* [2]	4096	71.6	79.7	-	-	$83.1/87.5^b$
	Previous state of the art		71.6 [2]	83.0 [60]	63.3 [27]	75.7 [60]	84.9 [27]
	Ours	512	**83.1**	**87.1**	**78.6**	**79.7**	$\mathbf{86.7/89.1^b}$
Matching /Spatial verif./QE	Chum *et al.* [8]		82.7	80.5	76.7	71.0	-
	Danfeng *et al.* [10]		81.4	80.3	76.7	-	-
	Mikulik *et al.* [35]		84.9	82.4	79.5	77.3	75.8^b
	Shen *et al.* [52]		75.2	74.1	72.9	-	76.2
	Tao *et al.* [57]		77.8	-	-	-	78.7
	Deng *et al.* [11]		84.3	83.4	80.2	-	84.7
	Tolias *et al.* [58]		86.9	85.1	85.3	-	81.3
	Tolias *et al.* [60]	512	77.3	86.5	73.2	79.8	-
	Tolias & Jégou [59]		**89.4**	82.8	84.0	-	-
	Xinchao *et al.* [30]		73.7	-	-	-	89.2
	Kalantidis *et al.* [27]	512	72.2	85.5	67.8	79.7	-
	Azizpour *et al.* [4]		79.0	85.1	-	-	**90.0**
	Previous state of the art		**89.4** [59]	86.5 [60]	85.3 [58]	79.8 [60]	**90.0** [4]
	Ours + QE	512	89.1	**91.2**	**87.3**	86.8	-

Retrieval results. We now evaluate the proposals in term of retrieval performance, see Table 2. The use of proposals improves over using a rigid grid, even with a baseline model only fine-tuned for classification (*i.e.* without ranking loss). On Oxford 5k, the improvements brought by the ranking loss and by the proposals are complementary, increasing the accuracy from 74.8 mAP with the C-Full model and a rigid grid up to 83.1 mAP with ranking loss and 256 proposals per image.

5.4   Comparison with the State of the Art

Finally we compare our results with the current state of the art in Table 3. In the first part of the table we compare our approach with other methods that also compute global image representations without performing any form of spatial verification or query expansion at test time. These are the closest methods to ours, yet our approach significantly outperforms them on all datasets – in one case by more than 15 mAP points. This demonstrates that a good underlying representation is important, but also that using features learned for the particular task is crucial.

In the second part of Table 3 we compare our approach with other methods that do not necessarily rely on a global representation. Many of these methods have larger memory footprints (*e.g.* [4,10,59,60]) and perform a costly spatial verification (SV) at test time (*e.g.* [30,35,60]). Most of them also perform query expansion (QE), which is a comparatively cheap strategy that significantly increases the final accuracy. We also experiment with average QE [9], which has a negligible cost (we use the 10 first returned results), and show that, despite not requiring a costly spatial verification stage at test time, our method is on equal foot or even improves the state of the art on most datasets. The only methods above us are the ones of Tolias and Jégou [59] (Oxford 5k) and Azizpour *et al.* [4] (Holidays). However, they are both hardly scalable as they require a lot of memory storage and a costly verification ([59] requires a slow spatial verification that takes more than 1 s per query, excluding the descriptor extraction time). Without spatial verification, the approach of Tolias and Jégou [59] achieves 84.8 mAP in 200 ms. In comparison, our approach reaches 89.1 mAP on Oxford 5k for a runtime of 1 ms per query and 2 kB data per image. Other methods such as [10,52,58] are scalable and obtain good results, but perform some learning on the target dataset, while in our case we use a single universal model.

6   Conclusions

We have presented an effective and scalable method for image retrieval that encodes images into compact global signatures that can be compared with the dot-product. The proposed approach hinges upon two main contributions. First, and in contrast to previous works [15,41,48], we *deeply* train our network for the specific task of image retrieval. Second, we demonstrate the benefit of predicting and pooling the likely locations of regions of interest when encoding the images. The first idea is carried out in a Siamese architecture [17] trained with a ranking loss while the second one relies on the successful architecture of region proposal networks [49]. Our approach very significantly outperforms the state of the art in terms of retrieval performance when using global signatures, and is on par or outperforms more complex methods while avoiding the need to resort to complex pre- or post-processing.

References

1. Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Zitnick, C.L., Parikh, D.: VQA: visual question answering. In: ICCV (2015)
2. Arandjelovic, R., Gronat, P., Torii, A., Pajdla, T., Sivic, J.: NetVLAD: CNN architecture for weakly supervised place recognition. In: CVPR (2016)
3. Arandjelovic, R., Zisserman, A.: Three things everyone should know to improve object retrieval. In: CVPR (2012)
4. Azizpour, H., Razavian, A., Sullivan, J., Maki, A., Carlsson, S.: Factors of transferability for a generic convnet representation. TPAMI **PP**(99), 1 (2015)
5. Babenko, A., Lempitsky, V.S.: Aggregating deep convolutional features for image retrieval. In: ICCV (2015)
6. Babenko, A., Slesarev, A., Chigorin, A., Lempitsky, V.: Neural codes for image retrieval. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014, Part I. LNCS, vol. 8689, pp. 584–599. Springer, Heidelberg (2014). doi:10.1007/978-3-319-10590-1_38
7. Chopra, S., Hadsell, R., Lecun, Y.: Learning a similarity metric discriminatively, with application to face verification. In: CVPR (2005)
8. Chum, O., Mikulik, A., Perdoch, M., Matas, J.: Total recall II: Query expansion revisited. In: CVPR (2011)
9. Chum, O., Philbin, J., Sivic, J., Isard, M., Zisserman, A.: Total recall: automatic query expansion with a generative feature model for object retrieval. In: ICCV (2007)
10. Danfeng, Q., Gammeter, S., Bossard, L., Quack, T., Van Gool, L.: Hello neighbor: accurate object retrieval with k-reciprocal nearest neighbors. In: CVPR (2011)
11. Deng, C., Ji, R., Liu, W., Tao, D., Gao, X.: Visual reranking through weakly supervised multi-graph learning. In: ICCV (2013)
12. Frome, A., Corrado, G.S., Shlens, J., Bengio, S., Dean, J., Ranzato, M.A., Mikolov, T.: DeViSE: a deep visual-semantic embedding model. In: NIPS (2013)
13. Girshick, R.: Fast R-CNN. In: CVPR (2015)
14. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: CVPR (2014)
15. Gong, Y., Wang, L., Guo, R., Lazebnik, S.: Multi-scale orderless pooling of deep convolutional activation features. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014, Part VII. LNCS, vol. 8695, pp. 392–407. Springer, Heidelberg (2014). doi:10.1007/978-3-319-10584-0_26
16. Gordo, A., Rodríguez-Serrano, J.A., Perronnin, F., Valveny, E.: Leveraging category-level labels for instance-level image retrieval. In: CVPR (2012)
17. Hadsell, R., Chopra, S., Lecun, Y.: Dimensionality reduction by learning an invariant mapping. In: CVPR (2006)
18. He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014, Part III. LNCS, vol. 8691, pp. 346–361. Springer, Heidelberg (2014). doi:10.1007/978-3-319-10578-9_23
19. Hoffer, E., Ailon, N.: Deep metric learning using triplet network. In: Feragen, A., Pelillo, M., Loog, M. (eds.) SIMBAD 2015. LNCS, vol. 9370, pp. 84–92. Springer, Heidelberg (2015). doi:10.1007/978-3-319-24261-3_7
20. Hu, J., Lu, J., Tan, Y.P.: Discriminative deep metric learning for face verification in the wild. In: CVPR (2014)

21. Jaderberg, M., Simonyan, K., Zisserman, A., Kavukcuoglu, K.: Spatial transformer networks. In: NIPS (2015)
22. Jégou, H., Chum, O.: Negative evidences and co-occurences in image retrieval: The benefit of PCA and whitening. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part II. LNCS, vol. 7573, pp. 774–787. Springer, Heidelberg (2012). doi:10.1007/978-3-642-33709-3_55
23. Jegou, H., Douze, M., Schmid, C.: Hamming embedding and weak geometric consistency for large scale image search. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part I. LNCS, vol. 5302, pp. 304–317. Springer, Heidelberg (2008). doi:10.1007/978-3-540-88682-2_24
24. Jégou, H., Douze, M., Schmid, C.: Improving bag-of-features for large scale image search. IJCV (2010)
25. Jégou, H., Douze, M., Schmid, C., Pérez, P.: Aggregating local descriptors into a compact image representation. In: CVPR (2010)
26. Jégou, H., Zisserman, A.: Triangulation embedding and democratic aggregation for image search. In: CVPR (2014)
27. Kalantidis, Y., Mellina, C., Osindero, S.: Cross-dimensional weighting for aggregated deep convolutional features. In: arXiv preprint arXiv:1512.04065 (2015)
28. Karpathy, A., Joulin, A., Fei-Fei, L.: Deep fragment embeddings for bidirectional image-sentence mapping. In: NIPS (2014)
29. Krizhevsky, A., Sutskever, I., Hinton, G.: ImageNet classification with deep convolutional neural networks. In: NIPS (2012)
30. Li, X., Larson, M., Hanjalic, A.: Pairwise geometric matching for large-scale object retrieval. In: CVPR (2015)
31. Long, J., Shelhamer, E., Darrell, T.: Fully convolutional networks for semantic segmentation. In: CVPR (2015)
32. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV (2004)
33. Mikolajczyk, K., Schmid, C.: Scale & affine invariant interest point detectors. IJCV (2004)
34. Mikulík, A., Perdoch, M., Chum, O., Matas, J.: Learning a fine vocabulary. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part III. LNCS, vol. 6313, pp. 1–14. Springer, Heidelberg (2010). doi:10.1007/978-3-642-15558-1_1
35. Mikulik, A., Perdoch, M., Chum, O., Matas, J.: Learning vocabularies over a fine quantization. IJCV (2013)
36. Ng, J.Y.H., Yang, F., Davis, L.S.: Exploiting local features from deep networks for image retrieval. In: CVPR workshops (2015)
37. Nister, D., Stewenius, H.: Scalable recognition with a vocabulary tree. In: CVPR (2006)
38. Paulin, M., Douze, M., Harchaoui, Z., Mairal, J., Perronin, F., Schmid, C.: Local convolutional features with unsupervised training for image retrieval. In: ICCV (2015)
39. Perdoch, M., Chum, O., Matas, J.: Efficient representation of local geometry for large scale object retrieval. In: CVPR (2009)
40. Perronnin, F., Dance, C.: Fisher kernels on visual vocabularies for image categorization. In: CVPR (2007)
41. Perronnin, F., Larlus, D.: Fisher vectors meet neural networks: a hybrid classification architecture. In: CVPR (2015)
42. Perronnin, F., Liu, Y., Sánchez, J., Poirier, H.: Large-scale image retrieval with compressed fisher vectors. In: CVPR (2010)

43. Philbin, J., Isard, M., Sivic, J., Zisserman, A.: Descriptor learning for efficient retrieval. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part III. LNCS, vol. 6313, pp. 677–691. Springer, Heidelberg (2010). doi:10.1007/978-3-642-15558-1_49

44. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: CVPR (2007)

45. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Lost in quantization: Improving particular object retrieval in large scale image databases. In: CVPR (2008)

46. Radenovic, F., Jegou, H., Chum, O.: Multiple measurements and joint dimensionality reduction for large scale image search with short vectors-extended version. ICMR (2015)

47. Radenovic, F., Tolias, G., Chum, O.: CNN image retrieval learns from BoW: unsupervised fine-tuning with hard examples. In: Leibe, B., et al. (eds.) ECCV 2016, Part I. LNCS, vol. 9905, pp. 3–20. Springer, Heidelberg (2016)

48. Razavian, A.S., Azizpour, H., Sullivan, J., Carlsson, S.: CNN features off-the-shelf: an astounding baseline for recognition. In: CVPR Deep Vision Workshop (2014)

49. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: NIPS (2015)

50. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: Imagenet large scale visual recognition challenge. IJCV (2015)

51. Schroff, F., Kalenichenko, D., Philbin, J.: FaceNet: a unified embedding for face recognition and clustering. In: CVPR (2015)

52. Shen, X., Lin, Z., Brandt, J., Wu, Y.: Spatially-constrained similarity measurefor large-scale object retrieval. TPAMI (2014)

53. Simo-Serra, E., Trulls, E., Ferraz, L., Kokkinos, I., Fua, P., Moreno-Noguer, F.: Discriminative learning of deep convolutional feature point descriptors. In: ICCV (2015)

54. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: ICLR (2015)

55. Song, H.O., Xiang, Y., Jegelka, S., Savarese, S.: Deep metric learning via lifted structured feature embedding. In: CVPR (2016)

56. Sun, Y., Chen, Y., Wang, X., Tang, X.: Deep learning face representation by joint identification-verification. In: NIPS (2014)

57. Tao, R., Gavves, E., Snoek, C.G., Smeulders, A.W.: Locality in generic instance search from one example. In: CVPR (2014)

58. Tolias, G., Avrithis, Y., Jégou, H.: Image search with selective match kernels: aggregation across single and multiple images. IJCV (2015)

59. Tolias, G., Jégou, H.: Visual query expansion with or without geometry: refining local descriptors by feature aggregation. PR (2015)

60. Tolias, G., Sicre, R., Jégou, H.: Particular object retrieval with integral max-pooling of CNN activations. In: ICLR (2016)

61. Vardi, Y., Zhang, C.H.: The multivariate L1-median and associated data depth. In: Proceedings of the National Academy of Sciences (2004)

62. Wang, J., Song, Y., Leung, T., Rosenberg, C., Wang, J., Philbin, J., Chen, B., Wu, Y.: Learning fine-grained image similarity with deep ranking. In: CVPR (2014)

63. Weinberger, K.Q., Saul, L.K.: Distance metric learning for large margin nearest neighbor classification. JMLR (2009)