Towards Good Practices for Image Retrieval Based on CNN Features

Omar Seddati Stéphane Dupont Saïd Mahmoudi Mahnaz Parian
CS - TCTS Lab
Bd Dolez 31, Mons, Belgium
{omar.seddati, stephane.dupont, said.mahmoudi, mahnaaz.pariyaan}@umons.ac.be

Abstract

Recent works have demonstrated that Convolutional Neural Networks (CNNs) achieve state-of-the-art results in several computer vision tasks. CNNs have also shown their ability to provide effective descriptors for image retrieval. In this paper, we focus on CNN feature extraction for instance-level image search. We started by studying in depth several methods proposed to improve the Regional Maximal Activation (RMAC) approach. Then, we selected some of these advances and introduced a new approach that combines multi-scale and multi-layer feature extraction with feature selection. We also propose an approach for local RMAC descriptor extraction based on class activation maps. Our parameter-free approach provides short descriptors and achieves state-of-the-art performance without the need of CNN finetuning or additional data in any way . In order to demonstrate the effectiveness of our approach, we conducted extensive experiments on four well known instance-level image retrieval benchmarks (the INRIA Holidays dataset, the University of Kentucky Benchmark, Oxford$5k$ *and Paris*$6k$*).*

1. Introduction

Over the past decades, available image and video collections have seen consistent growth through the easily accessible devices that we now use on a daily basis. These huge multimedia collections motivated researchers to look for efficient approaches for Content-Based Image Retrieval (CBIR). Especially, instance-level image search (where an image is used as query to retrieve images of the same object) has received a lot of attention and became one of the most active topics in the field of computer vision. Many of the available image retrieval systems are based on basic descriptors (i.e. color histogram, shape, etc.) which provides results that does not match the query image at an instance level. Therefore, it is crucial to improve the existing search methods and use more powerful image descriptors to develop efficient instance level image retrieval solutions.

Conventional search systems were based on aggregation of local and global features achieved via image descriptors such as SIFT [22] or GIST [36] by methods such as Bag of Words (BoW) [37]. These methods have shown good results in nearest neighbors search because of considering the local characteristics of the images and being invariant to local transformations and illumination [37] [42].

Recently, there was numerous advances in the field of information retrieval especially by employing deep learning approaches. Particularly, Convolutional Neural Networks (CNN) [19] showed a significant improvement in search results quality. Most of the CNN-based methods, use internal activations of an ImageNet pre-trained CNN as image representations and a k-nearest neighbor (kNN) approach to build an image search system. To this moment, the majority of these methods discard the fully connected (FC) layers and use the last convolutional layer for feature extraction. Indeed, FC layers provide higher-level semantic features which disregard the local characteristics of the objects, and thus show poor results for instance level search [24]. By using non-FC features we aim to distinguish different objects even though they share the same label or semantic class.

Our work is principally based on the RMAC descriptor [40] (and its extensions MS-RMAC [21] and the work of Gordo et al. [10]), which takes into account the feature vector achieved by sum-aggregation of several image regions. The extensions of the RMAC approach make it more robust to scaling and translation variance and have shown improvements in accuracy of retrieved results for instance search. Our proposed approach combines different advances of the RMAC extensions together with other techniques to achieve state-of-the-art results (without using finetuning approaches that need additional annotated data). We start by replacing approximate max-pooling used in the original RMAC approach with a maxpooling layer. Then, we select a more efficient network architecture [12] for feature extraction, which results in less computational cost while attaining similar results. In addition to that, we use a multi-resolution approach, where features are extracted from three different resolutions of the input. Then, the output feature maps are rescaled and summed into one $3D$ feature tensor that encode activations at different scales. The RMAC features in our approach are extracted from the last two convolutional layers of the network and concatenated to produce more representative feature vectors. Unfortunately, this concatenation increases the dimensionality of the descriptor and add redundancy. In order to deal with these limitations, we use a feature selection approach to reduce dimensionality and enhance image retrieval performance. Our approach achieves state-of-the-art results in four standard benchmarks for instance level image search (INRIA Holidays [13], UKB [25], Oxford$5k$ [29] and Paris$6k$ [30]) compared to state-of-the-art methods (not using additional training data).

The rest of the paper is organized as follows: In Section II, we explore the state-of-the-art image retrieval methods. Section III covers the background of our proposed method, explaining the RMAC feature descriptor, the RMAC extensions, MS-RMAC and the work of Gordo et al. [10]. Then, we present the pipeline of our approach and provide some technical details. Section IV shows our experimental results in different datasets and Section V concludes the paper.

2. Related Work

This section takes a deeper look into the existing methods of image search at instance level. These approaches can be divided into two main groups: Conventional methods and CNN based methods.

2.1. Conventional methods

The common conventional similarity search methods at the instance level, are mostly based on Bag of Words (BoW) [37] representation of local and global features. The CBIR systems initially were based on global features such as color, shape and texture which represents content of the image [41, 42, 43] and are invariant to local transformations and robust to geometrical translation [37]. Among the global descriptors,those based on GIST features [36] became popular. With regard to local descriptors, SIFT[22] became one of the most used representations for image search systems. Multiple [14, 16] and soft assignment [30], spatial matching [29, 3], feature selection[39] and large vocabularies [24] are of the extensions available to improve accuracy of BoW based on local features. However, global representation became popular again later by using local feature aggregation methods such as VLAD [15] and Fischer vector [28], as generative methods which create vector representation for nearest neighbor search task. There are numerous extensions for these approaches such as spatial VLAD [2] and triangulation embedding [17]. Furthermore, post processing techniques are also another way to improve the accuracy of search result and refine them. Geometric verification[52, 13, 29], query expansion [20, 7] and retrieval fusion [45] are amongst the most used approaches.

2.2. CNN Based Methods

In addition to the conventional methods, learning based approaches emerged in image retrieval following the increased popularity of neural networks. Semantic information retrieved from deep neural networks can be efficiently used for image retrieval to increase the accuracy of the search. As a recent successful branch of deep learning, CNN [19] have shown significant improvement in image similarity search tasks. The idea that deep convolutional network can extract high level features in the deeper layers, led the researchers to explore ways to reduce the semantic gap by this method. The extracted features at both fully connected layers [33] and convolutional layers [49] can be used as image representations in search algorithms. The features obtained at the last fully connected are considered as effective high dimensional global features descriptors [6]. In addition to extracting features from whole image, CNN models used tools to to extract features of local image regions [38]. Regional Maximum Activation of Convolutions (RMAC) [40], is a state-of-the-art approach based on CNN that encodes several image regions into compact features and use integral pooling to generate fixed length geometry aware feature vector. After the success of RMAC [40], a few extensions have been proposed to further enhance the descriptor. Moreover, in MS-RMAC [21] the authors proposed a multi-layer approach to increase RMAC robustness to scale and shape variance. In [10], Gordo et al. combined RMAC with triplet networks and proposed also an approach based on region proposal network (RPN) to identify Region of Interest (RoI) and extract local RMAC descriptors. Additionally [9, 6, 4, 34], explored different layers aggregation with a mapping function (except fully connected layers). On the other hand, [9, 4, 33] explored possible aggregation of fully connected and max pooling layers activations via VLAD plus re-ranking schemes which have shown promising results in image retrieval literature.

3. Background

As mentioned before, our approach is based on the RMAC approach [40] and some improvements made by [10, 21]. In this section we discuss these approaches and point the different contributions and how it impacts the retrieval accuracy.

3.1. RMAC

In [40] Tolias et al. reported for the first time a CNN approach competing with traditional methods on challenging retrieval benchmarks. In order to extract features, they discarded the fully connected layers of a pre-trained CNN

(VGG16) and used the resulting fully convolutional CNN for feature extraction. Let's assume we have an input image I of size $(W_I \times H_I)$, the output feature maps will form a $3D$ tensor in the form $C \times W \times H$ (where C is the number of channels, (W, H) the width and height of FMs). If we represent this $3D$ tensor as a set of $2D$ feature maps $\mathcal{X} = \{\mathcal{X}_c\}$, $c = 1...C$, we can compute the MAC (Maximum Activations of Convolutions) using the following equation:

$$f = [f_1...f_c...f_C], with f_c = \max_{x \in \mathcal{X}_c} x \tag{1}$$

In order to compute the RMAC (Regional Maximum Activation of Convolutions) descriptor, Tolias et al. proposed a simple approach to sample $R = \{R_i\}$ a set of square regions within $\mathcal{X}$. The proposed sampling is done at $L = 3$ different scales, at each scale a square kernel is used in a sliding window fashion to compute the approximate integral max-pooling, where the width of the kernel is $k_w = 2 \times min(W, H)/(l + 1), l = 1...L$ and the stride is equal to $60\% \times k_w$. Once the regions are selected, the RMAC descriptor for a region R_i can be computed using the following equation:

$$f_{R_i} = [f_{R_i,1}...f_{R_i,c}...f_{R_i,C}] \tag{2}$$

with $f_{R_i,c} = (\sum\limits_{x \in \mathcal{R}_{i,c}} x^{\alpha})^{\frac{1}{\alpha}} \approx \max_{x \in \mathcal{R}_{i,c}} x$ and $\alpha = 10$.
Then for each region they normalize (l_2 normalization) the resulting vector, do PCA-whitening, normalize again before combining all these vectors and finally normalize to obtain the final RMAC descriptor.
The RMAC approach has several advantages, starting with the use of fully convolutional CNNs, which enables us to keep the aspect ratio of the inputs without using techniques like zero padding (which may harm the retrieval performance). In addition to that, the RMAC descriptor encodes efficiently spatial information while keeping the size of the descriptor not dependent on the resolution of the input but on the number of channels of the selected layer for feature extraction.

3.2. Multi-Scale RMAC (MS-RMAC)

Li et al. [21] point out that the RMAC approach could present some limitations related to the use of single convolutional layer for feature extraction. They believe that the resulting descriptor may not be robust enough for image deformation like scale variation and occlusion. In order to overcome these possible limitations, They proposed the extraction of RMAC descriptors from multiple layers (different depths) and concatenate them into one MS-RMAC vector (they end up with descriptor with 1472 dimensions compared to 512 for the RMAC). Then, they use an iterative approach to attribute a weight to each RMAC descriptor (one RMAC descriptor per layer). These weights, represent the importance of the features extracted by each layer and how they affect the result of descriptors comparison and similarity measure. The MS-RMAC approach (with weighting) shows slight improvements compared to the original RMAC method.
Note: *the term "multi-scale" used by the authors of [21] means that the feature maps have different scales while they use only one single scale for the input image.*

3.3. End-to-end Learning of Deep Visual Representations for Image Retrieval

In [10], Gordo et al. proposed two simple modifications to bring significant improvements to the RMAC representation:

- **ResNet:** in the original RMAC approach [40], the authors tried two CNN architectures (AlexNet and VGG16) for feature extraction. In [10], Gordo et al. used the ResNet101, a more recent and more powerful architecture.

- **Multi-resolution:** unlike MS-RMAC, Gordo et al. proposed an approach where three resolutions of the input image are feeded to the network. The RMAC descriptors are computed separately and l_2-normalized. Then the three vectors are summed and l_2-normalized.

On top of that, Gordo et al. proposed a region proposal network (RPN) pooling to replace the RMAC rigid region pooling mechanism (in order to reduce the impact of background on RMAC representations), but their experiences showed that the use of RPN was not able to improve retrieval performance compared to the basic multi-scale ResNet101 RMAC descriptor.

In order to further improve the ResNet101 representations, Gordo et al. [10] used the landmarks dataset (introduced by Babenko et al. [6]) to finetune the ResNet101 using two strategies:

- **Classification finetuning:** they added a classifier above the ResNet101 feature extractor and did the finetuning in a standard fashion on the landmarks dataset using a cross-entropy loss.

- **Triplet networks:** they trained a triplet network with a ranking loss to enhance the RMAC representation.

Both strategies demonstrated their ability to improve retrieval performance. In the first case (classification finetuning), the improvement is related to the fact that the training data is closer to the evaluation data (compared to ImageNet) and to the higher resolution of the inputs (compared to standard resolution used for ImageNet-based models, i.e.

224×224). In the second case, the ranking training is usually used to learn better embeddings, which explains the significant improvements noticed especially for Oxford 5k and Paris 6k benchmarks since the training data and test data are similar. At the end, the authors combined these two strategies to provide their best performing approach.

The experiments conducted by Gordo et al. [10] bring simple and interesting solutions to improve image retrieval performance. The use of ResNet101 and Multi-resolution is straightforward and brings significant improvements without increasing the length of the descriptor (unlike the MS-RMAC, the sum of multi-resolution vectors instead of concatenating them). But the finetuning solution implies using **additional annotated data**, which comes at a high cost.

3.4. Our approach

As mentioned in the previous section, in this paper, we propose a new approach (Fig. 1) on top of the RMAC method and the improvements proposed in [10, 21]. Amongst the techniques not using instance labeled training data, our approach achieves beyond state-of-the-art results with shorter descriptors, and comes close to the method using instance labeled training data [10] Compared to the original RMAC, we use a ResNet architecture as in [10]. We found that ResNet50 gives the same performances as ResNet101 while reducing the computing cost. We did also replace the approximate max-pooling of the RMAC approach with the classic max-pooling layer. We keep using the same approach for the kernel size selection, but we increase the possible overlap between regions to 60% and replace the sum-pooling with square sum-pooling (RMAC-modified). Additionally, we made the following modifications:

- **Multi-resolution (feature map fusion):** we use multiple resolutions as proposed in [10], but instead of computing the RMAC descriptor separately for each resolution and then sum the resulting vectors, we rescale the output feature maps of the three resolutions to the same resolution (the highest resolution), we sum them and then we compute the modified RMAC descriptor. The motivation behind this modification is to produce one $3D$ tensor (i.e. Fig. 2) that encodes all the spatial activations (even if they belong to different resolutions) before computing the image descriptor. This approach can be seen as an "early fusion" compared to the one proposed in [10] ("late fusion"). We believe that this less aggressive fusion (early fusion) provides more stable representation since it acts more locally in space and gives a preprocessed $3D$ feature tensor that eases the task of encoding efficiently the whole information using the RMAC descriptor.

- **Multi-layer:** in addition to the use of multi-resolution, we found that using a multi-layer can still bring some improvements. But unlike the MS-RMAC, instead of concatenating the RMAC descriptors computed for each convolutional layer, we use only the last two convolutional layers since the first layers are too sensitive to local deformation.

- **Feature selection:** compared to the original RMAC and the MS-RMAC, we did not use PCA-whitening or layer weighting, we have opted for a different strategy, feature selection using the Principal Feature Analysis method [23]. We will show that this enhances instance level image retrieval performance and reduce the dimensionality at the same time. The entire feature selection work-flow can be summarized as follows:

  - We use the extracted RMAC descriptors as samples and compute the covariance matrix.
  - We compute the principal components and eigenvalues of the covariance matrix.
  - We construct the transformation matrix $\mathcal{A}_q$ formed by the principal axes in feature space (the directions of maximum variance).
  - Let's assume that $V = \{V_i\}$ are the rows of the matrix $\mathcal{A}_q$, we use K-Means to cluster the vectors V_i (the number of cluster is equal to the number of principal features, i.e. 768).
  - Finally, we find the closest vector V_i to each cluster center and select the corresponding feature as principal feature.

- **Localization** (optional) **:** in [10], Gordo et al. trained an RPN and tried to improve the retrieval performance by replacing the rigid grid of RMAC. Here, in order to explore this idea while continuing in our current direction (enhancing instance-level search without finetuning our model using additional data), we propose to use the class activation maps (CAM) introduced in [51]. Let assume that we have a classification CNN (some convolutional layers followed by a pooling layer and a fully-connected layer), using this method enables us to compute a heatmap indicating how each region of the input image contributes to the activation of a specific category. The approach is straightforward, first we select one neuron from the fully-connected (FC) layer (e. g. the neuron corresponding to the category giving the highest activation) and we multiply its weight vector with feature maps before the last pooling layer, then we sum all the resulting feature maps. In order to preserve the aspect ratio of the inputs when using this approach, we transform our CNN to a fully-convolutional CNN (by transforming the FC layers into convolutional layers) and we add an adaptive average pooling layer (to

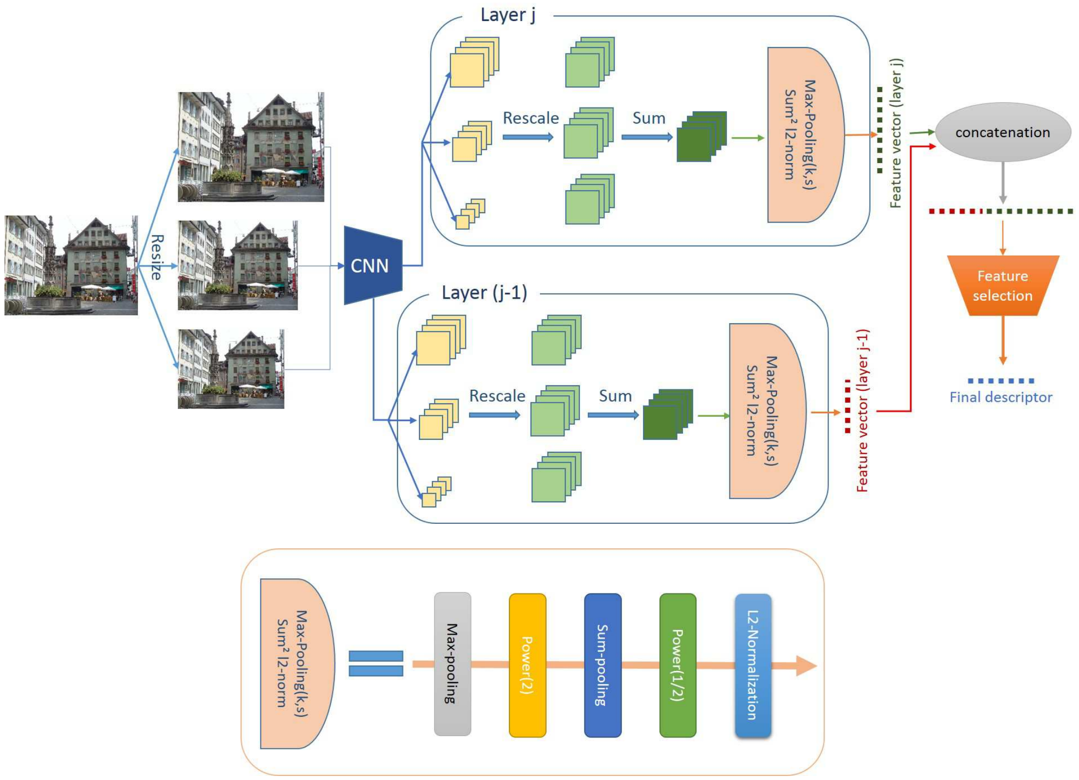

Figure 1. Our approach.

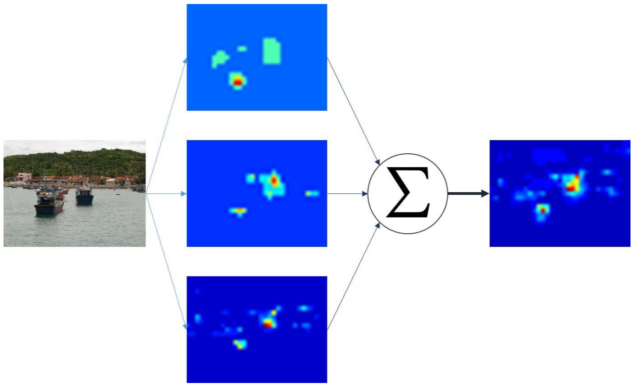

Figure 2. An example showing the heatmap obtained by the sum fusion of three rescaled feature maps.

produce FMs with a spatial resolution of 1×1) followed by a maxpooling to identify the highest score. Then, we use CAM to compute the heatmap and a threshold of $0.9 \times average(heatmap)$ to generate a spatial binary mask that we multiply later with feature maps before computing RMAC in order to extract local RMAC descriptors. In Fig. 3 we show how the CAM approach can be used to generate masks and the kind of results to be expected.

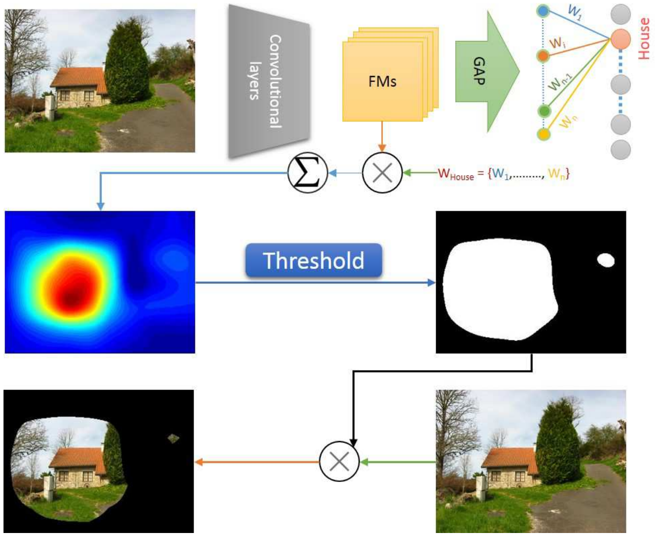

Figure 3. An example of segmentation using the CAM approach.

The complete pipeline of our approach is detailed in Algorithm 1.

4. Experiments

In this section, we first introduce the standard benchmarks used to evaluate our approach. Then, we present the results of our approach and provide a comparison with the state-of-the-art methods.

Note: *In order to extract CNN features, we use the ResNet50 [12] and the publicly available Torch toolbox [8]. For the multi-resolution based approach, we use three different input resolutions for each image, where S is the largest side of the input and $S \in [550, 800, 1050]$.*

4.1. INRIA Holidays

Figure 4. INRIA Holidays Benchmark.

INRIA Holidays consists of 1491 personal holiday images of 500 image groups (each group represent an object or a scene). During the evaluation, the first image of each group is used as query (by the end, 500 images are used as queries and the other 991 are used to retrieve images similar to the query). An example of the Holidays collection is shown in Fig. 4. In table 1, we show our intermediates and final results for the evaluation conducted on the Holidays benchmark. In this table, we used the following annotation:

- Baseline ($S = 800$): we used the ResNet50 to extract features and followed the modified RMAC approach to compute the final feature vectors.
- M-R: we used the ResNet50 and the multi-resolution approach of Gordo et al. [10] (M-R is used for multi-resolution).
- M-R l_j: as explained in the previous section, the approach of Gordo et al. [10] computes RMAC vectors for each resolution before doing the fusion, while in our approach we resize the feature maps and combine them, then we compute RMAC descriptors (l_j indicates that we are using features extracted from the last convolutional layer).
- M-R l_{j-1}: same as the previous one, but we use the penultimate convolutional layer for feature extraction .

Method	Dimensionality	mAP
Baseline ($S = 800$)	2048	90.57
M-R	2048	91.47
M-R l_j	2048	92.16
M-R l_{j-1}	1024	85.04
M-R $l_{j\&j-1}$	2048+1024	91.10
M-R $l_{j\&j-1}$ + FS	768	93.98
M-R $l_{j\&j-1}$+FS+CAM	768	93.97

Table 1. Our intermediate results on the Holidays benchmark

- M-R $l_{j\&j-1}$: same as the previous one, but we concatenate features extracted by both convolutional layers.
- M-R $l_{j\&j-1}$ + FS: same as the previous one, but we use feature selection (FS for feature selection).
- M-R $l_{j\&j-1}$ + FS + CAM: same as the previous one, but we also compute local RMAC descriptors.
  When evaluating, we compute the Manhattan distance between the queries and the 991 other images for both local and global descriptors, then we keep only the minimal distance (global or local) and use it to find the closest images to the query.

In table 2, we compare our approach with state-of-the-art methods on the Holidays benchmark. As we can see, the only method that achieves better results than our approach is the one proposed by Gordo et al. in [10]. But as explained before, our approach provides shorter descriptors and needs no additional data to finetune the feature extractor.

4.2. University of Kentucky Benchmark (UKB)

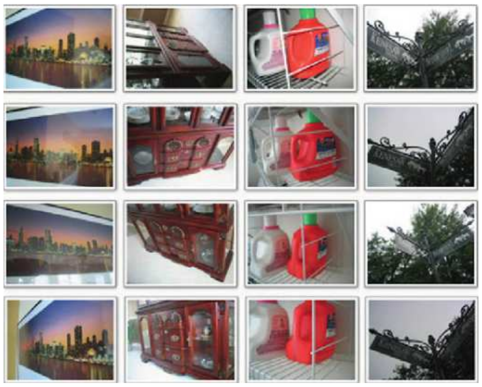

Figure 5. University of Kentucky Benchmark (UKB)

The University of Kentucky retrieval benchmark is a dataset introduced in [25] which consist of 2550 classes, each class with 4 images with JPEG format. The pictures are from diverse categories such as animals, plants, household objects, etc. An example of the dataset images are

Method	Dimensionality	mAP
Triplet Network[10]	2048	94.8
Ours	768	94.0
RMAC (ResNet101)[10]	2048	91.3
Arandjelovic et al [1]	4096	87.5
RMAC [10]	512	86.9
MS-RMAC [21]	1472	86.7
Kalantidis et al [18]	512	84.9
Perronnin and Larlus [27]	4000	84.7
Ng et al [44]	128	83.6
Radenovic et al [32]	512	82.5
Gong et al[9]	2048	80.8
Babenko and Lempitsky [5]	256	80.2
Paulin et al [26]	256K	79.3
Gordoa et al [11]	512	79.0
Babenko et al [6]	128	78.9
Jegou and Zisserman [17]	1024	72.0
Jegou and Zisserman [17]	128	61.7

Table 2. Accuracy comparison with state-of-the-art methods on the Holidays benchmark

illustrated in Fig. 5. To evaluate our approach, we followed the standard evaluation protocol, for each group we select one image as the query and use kNN to retrieve the four nearest images to the query from the whole collection. Then, we compute the recall at four. Our results and a comparison with state-of-the-art approaches are reported in table 3.

Method	UKbench
Our (M-R $l_{j\&j-1}$ + FS + CAM)	3.91
Our (M-R $l_{j\&j-1}$ + FS)	3.91
Zheng et al.[48]	3.85
Gordo et al[10]	3.84
Zheng et al.[49]	3.84
Zheng et al.[47]	3.81
Qin et al.[31]	3.67
Zheng et al.[50]	3.62
Zhang et al.[46]	3.60
Jegou et al.[17]	3.53
Shen et al.[35]	3.52
Wengert et al.[43]	3.42

Table 3. Recall@4 comparison with state-of-the-art methods on UKbench.

4.3. Oxford $5k$ & Paris $6k$

We also evaluate our approach on two additional benchmarks, the Oxford Buildings dataset (Oxford $5k$ contains 5062 images) and the Paris landmarks dataset (Paris $6k$ contains 6412 images). For both benchmarks, images were collected from Flickr by searching for particular landmarks. There are 55 manually annotated queries corresponding to 11 landmarks for each benchmark. The mean average precision (mAP) is used to measure retrieval performance over the 55 queries. In table 4, we report our intermediate results for these two benchmarks.

Method	Dimensionality	Oxford $5k$
M-R l_j	2048	67.77
M-R l_{j-1}	1024	54.05
M-R $l_{j\&j-1}$	2048+1024	66.82
M-R $l_{j\&j-1}$ + FS	320	72.27
M-R $l_{j\&j-1}$+FS+CAM	320	70.54

Method	Dimensionality	Paris $6k$
M-R l_j	2048	82
M-R l_{j-1}	1024	69.48
M-R $l_{j\&j-1}$	2048+1024	81.88
M-R $l_{j\&j-1}$ + FS	384	87.10
M-R $l_{j\&j-1}$+FS+CAM	384	82.55

Table 4. Our intermediate results on Oxford $5k$ and Paris $6k$

Our best results for both Oxford $5k$ and Paris $6k$ and a comparison with state-of-the-art approaches are reported in table 5. The only method that achieves significantly better results than ours is the one proposed by Gordo et al. in [10], where a large-scale landmarks benchmark was used to finetune the feature extractor. These training images are similar to the evaluation images (both Oxford $5k$ and Paris $6k$ contain landmarks images), which enables the finetuned CNN to extract more specific features and significantly improve the performances on Oxford $5k$ and Paris $6k$ compared to the enhancement noticed for the Holidays dataset and the UKbench.

5. Conclusion

The popularity of CNN in many of computer vision tasks increased in recent years. Image retrieval systems, are not exempted of this advances. But instance level image retrieval is not thoroughly explored yet in this domain.

In this paper we tackle the problem of instance image retrieval by modifying the recently introduced RMAC descriptor, along with MS-RMAC and multi-resolution extensions. Our approach achieves state-of-the-art results on four well known image retrieval benchmarks without using additional annotated data, finetuning or very high-dimensional descriptors. To this end, we followed the main ideas proposed on some works based on RMAC to improve the effectiveness of this descriptor. But we replaced several components of these proposed pipelines with others, achieving better or close results without the requirement of finetuning

Method	Dim	O5k	P6k
Triplet Network[10]	2048	86.1	94.5
Ours	384	72.3	87.1
Radenovic et al [32]	512	79.7	83.8
Arandjelovic et al [1]	4096	71.6	79.7
RMAC [40]	512	66.9	83
MS-RMAC [21]	1472	68.9	77.6
Kalantidis et al [18]	512	68.2	79.7
Ng et al [44]	128	59.3	59.0
Paulin et al [26]	256K	56.5	-
Jegou and Zisserman [17]	1024	56.0	-
Babenko et al [6]	128	55.7	-
Babenko and Lempitsky [5]	256	53.1	-
Jegou and Zisserman [17]	128	43.3	-

Table 5. Accuracy comparison with state-of-the-art methods on Oxford $5k$ & Paris $6k$. The methods in italics indicate that the approach uses training data to finetune the CNN.

or using additional data in any way. We start with architecture of the feature extractor, the use of a multi-resolution method and applying RMAC over the sum-aggregated rescaled feature maps, which bring a substantial improvement in retrieval scores. Additionally, we employed a feature selection approach to reduce the dimensionality of our descriptors and further enhance the performance of similarity search. Furthermore, in order to replace the use of RPN for localizing region of interest, we proposed to use the CAM approach which provides a way to achieve segmentation without resorting to any additional training. Results for this last idea however remained inconclusive. In most of state-of-the-art approaches based on CNN, the models are pre-trained on ImageNet with small resolutions (typically 224 × 224), in our future work, we plan to finetune our feature extractor on a high resolution ImageNet. Our aim is to make the model filters work better on higher resolution inputs in order to enhance the image encoding and gain better performance. We will also aim to improve our approach by employing unsupervised techniques (e.g. applying some transformations to an input image to generate positive examples needed to train a triplet networks) and conduct further evaluations on other benchmarks.

Algorithm 1 Our approach

```
procedure $pre_{p}rocess(img)$
    $resolutions = [1050, 800, 550]$
    $imgs = \{\}$
    for $r \leftarrow 1$ to 3 do
        $longest\_side = resolutions[r]$
        $imgs[r] = scale(img, longest\_side)$
    return $imgs$
procedure $ext_{F}M(img)$
    $CNN : forward(img)$
    $FM_j = CNN\{layer[j]\}.output$    ▷ a $3D$ tensor with 2048 channels
    $FM_{j-1} = CNN\{layer[j-1]\}.output$    ▷ a $3D$ tensor with 1024 channels
    return $FM_j, FM_{j-1}$
procedure $OS_{R}MAC(FMs)$
    $C = FMs.channels$
    $W = FMs.width$
    $H = FMs.height$
    $vec_{s}um = vector[C].fill(0)$
    for $l \leftarrow 1$ to 3 do
        $k_w = max(1, ceil(2 \times min(W, H)/(l+1)))$
        $k_s = max(1, floor(0.4 \times k_w))$
        $mp = maxpooling(k_w, k_s)$
        $FMs_{mp} = mp(FMs)$

        $FMs_{mp_2} = power_2(FMs_{mp})$
        $FMsum_{mp_2} = sumpooling(FMs_{mp_2})$
        $FMsum_{mp} = power_{1/2}(FMsum_{mp_2})$
        $vec_{s}um + = FMsum_{mp}$
    return $norm_2(vec_{s}um)$
procedure $Main()$
    $img = load(src)$
    $imgs = pre_{p}rocess(img)$

    $FM_{all,j}, FM_{all,j-1} = ext_{F}M(imgs[1])$

    $S_j, S_{j-1} = sizeOf(FM_{all,j}), sizeOf(FM_{all,j-1})$

    for $r \leftarrow 2$ to 3 do    ▷ $Scale + SUM$
        $FM_{r,j}, FM_{r,j-1} = ext_{F}M(imgs[r])$

        $FM_{r,j} = scale(FM_{r,j}$
        $FM_{all,j} + = FM_{r,j}$

        $FM_{r,j-1} = scale(FM_{r,j-1}$
        $FM_{all,j-1} + = FM_{r,j-1}$

    $vec_j = OS_{R}MAC(FM_{all,j})$
    $vec_{j-1} = OS_{R}MAC(FM_{all,j-1})$
    $vec_{a}ll = concat(vec_j, vec_{j-1})$

    $our_{d}escriptor = feature_{s}election(vec_{a}ll)$
```

References

[1] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5297–5307, 2016.

[2] R. Arandjelovic and A. Zisserman. All about vlad. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1578–1585, 2013.

[3] Y. Avrithis and G. Tolias. Hough pyramid matching: Speeded-up geometry re-ranking for large scale image retrieval. *International journal of computer vision*, 107(1):1–19, 2014.

[4] H. Azizpour, A. Sharif Razavian, J. Sullivan, A. Maki, and S. Carlsson. From generic to specific deep representations for visual recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 36–45, 2015.

[5] A. Babenko and V. Lempitsky. Aggregating local deep features for image retrieval. In *Proceedings of the IEEE international conference on computer vision*, pages 1269–1277, 2015.

[6] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky. Neural codes for image retrieval. In *European conference on computer vision*, pages 584–599. Springer, 2014.

[7] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[8] R. Collobert, K. Kavukcuoglu, and C. Farabet. Torch7: A matlab-like environment for machine learning. In *BigLearn, NIPS Workshop*, number EPFL-CONF-192376, 2011.

[9] Y. Gong, L. Wang, R. Guo, and S. Lazebnik. Multi-scale orderless pooling of deep convolutional activation features. In *European conference on computer vision*, pages 392–407. Springer, 2014.

[10] A. Gordo, J. Almazán, J. Revaud, and D. Larlus. End-to-end learning of deep visual representations for image retrieval. *International Journal of Computer Vision*, pages 1–18, 2016.

[11] A. Gordoa, J. A. Rodríguez-Serrano, F. Perronnin, and E. Valveny. Leveraging category-level labels for instance-level image retrieval. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3045–3052. IEEE, 2012.

[12] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[13] H. Jegou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. *Computer Vision–ECCV 2008*, pages 304–317, 2008.

[14] H. Jégou, M. Douze, and C. Schmid. Improving bag-of-features for large scale image search. *International journal of computer vision*, 87(3):316–336, 2010.

[15] H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3304–3311. IEEE, 2010.

[16] H. Jegou, C. Schmid, H. Harzallah, and J. Verbeek. Accurate image search using the contextual dissimilarity measure. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(1):2–11, 2010.

[17] H. Jégou and A. Zisserman. Triangulation embedding and democratic aggregation for image search. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3310–3317, 2014.

[18] Y. Kalantidis, C. Mellina, and S. Osindero. Cross-dimensional weighting for aggregated deep convolutional features. In *European Conference on Computer Vision*, pages 685–701. Springer, 2016.

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[20] Y.-H. Kuo, K.-T. Chen, C.-H. Chiang, and W. H. Hsu. Query expansion for hash-based image object retrieval. In *Proceedings of the 17th ACM international conference on Multimedia*, pages 65–74. ACM, 2009.

[21] Y. Li, Y. Xu, J. Wang, Z. Miao, and Y. Zhang. Ms-rmac: Multiscale regional maximum activation of convolutions for image retrieval. *IEEE Signal Processing Letters*, 24(5):609–613, 2017.

[22] D. G. Lowe. Object recognition from local scale-invariant features. 2:1150–1157, 1999.

[23] Y. Lu, I. Cohen, X. S. Zhou, and Q. Tian. Feature selection using principal feature analysis. In *Proceedings of the 15th ACM international conference on Multimedia*, pages 301–304. ACM, 2007.

[24] A. Mikulík, M. Perdoch, O. Chum, and J. Matas. Learning a fine vocabulary. *Computer Vision–ECCV 2010*, pages 1–14, 2010.

[25] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *Computer vision and pattern recognition, 2006 IEEE computer society conference on*, volume 2, pages 2161–2168. Ieee, 2006.

[26] M. Paulin, M. Douze, Z. Harchaoui, J. Mairal, F. Perronin, and C. Schmid. Local convolutional features with unsupervised training for image retrieval. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 91–99, 2015.

[27] F. Perronnin and D. Larlus. Fisher vectors meet neural networks: A hybrid classification architecture. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3743–3752, 2015.

[28] F. Perronnin, Y. Liu, J. Sánchez, and H. Poirier. Large-scale image retrieval with compressed fisher vectors. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3384–3391. IEEE, 2010.

[29] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[30] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[31] D. Qin, C. Wengert, and L. Van Gool. Query adaptive similarity for large scale object retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1610–1617, 2013.

[32] F. Radenović, G. Tolias, and O. Chum. Cnn image retrieval learns from bow: Unsupervised fine-tuning with hard examples. In *European Conference on Computer Vision*, pages 3–20. Springer, 2016.

[33] A. S. Razavian, J. Sullivan, S. Carlsson, and A. Maki. Visual instance retrieval with deep convolutional networks. *ITE Transactions on Media Technology and Applications*, 4(3):251–258, 2016.

[34] A. Sharif Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 806–813, 2014.

[35] X. Shen, Z. Lin, J. Brandt, S. Avidan, and Y. Wu. Object retrieval and localization with spatially-constrained similarity measure and k-nn re-ranking. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3013–3020. IEEE, 2012.

[36] C. Siagian and L. Itti. Rapid biologically-inspired scene classification using features shared with visual attention. *IEEE transactions on pattern analysis and machine intelligence*, 29(2):300–312, 2007.

[37] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *null*, page 1470. IEEE, 2003.

[38] S. Sun, W. Zhou, Q. Tian, and H. Li. Scalable object retrieval with compact image representation from generic object regions. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 12(2):29, 2016.

[39] G. Tolias, Y. Kalantidis, Y. Avrithis, and S. Kollias. Towards large-scale geometry indexing by feature selection. *Computer Vision and Image Understanding*, 120:31–45, 2014.

[40] G. Tolias, R. Sicre, and H. Jégou. Particular object retrieval with integral max-pooling of cnn activations. *arXiv preprint arXiv:1511.05879*, 2015.

[41] J. Wang and X.-S. Hua. Interactive image search by color map. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(1):12, 2011.

[42] X.-Y. Wang, B.-B. Zhang, and H.-Y. Yang. Content-based image retrieval by integrating color and texture features. *Multimedia tools and applications*, 68(3):545–569, 2014.

[43] C. Wengert, M. Douze, and H. Jégou. Bag-of-colors for improved image search. In *Proceedings of the 19th ACM international conference on Multimedia*, pages 1437–1440. ACM, 2011.

[44] J. Yue-Hei Ng, F. Yang, and L. S. Davis. Exploiting local features from deep networks for image retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 53–61, 2015.

[45] S. Zhang, M. Yang, T. Cour, K. Yu, and D. N. Metaxas. Query specific fusion for image retrieval. In *Computer Vision–ECCV 2012*, pages 660–673. Springer, 2012.

[46] S. Zhang, M. Yang, X. Wang, Y. Lin, and Q. Tian. Semantic-aware co-indexing for image retrieval. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1673–1680, 2013.

[47] L. Zheng, S. Wang, Z. Liu, and Q. Tian. Lp-norm idf for large scale image search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1626–1633, 2013.

[48] L. Zheng, S. Wang, Z. Liu, and Q. Tian. Packing and padding: Coupled multi-index for accurate image retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1939–1946, 2014.

[49] L. Zheng, S. Wang, L. Tian, F. He, Z. Liu, and Q. Tian. Query-adaptive late fusion for image search and person reidentification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1741–1750, 2015.

[50] L. Zheng, S. Wang, W. Zhou, and Q. Tian. Bayes merging of multiple vocabularies for scalable image retrieval. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1955–1962, 2014.

[51] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2921–2929, 2016.

[52] W. Zhou, Y. Lu, H. Li, Y. Song, and Q. Tian. Spatial coding for large scale partial-duplicate web image search. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 511–520. ACM, 2010.