Co-Occurrence of Deep Convolutional Features for Image Search

Juan I. Forcen

das-Nano — Veridas, 31192, Tajonar, Spain
Dpt. Estadística, Informática y Matemáticas
Institute of Smart Cities
Universidad Pública de Navarra
Campus Arrosadía , 31006, Pamplona, Spain

Miguel Pagola, Edurne Barrenechea

Dpt. Estadística, Informática y Matemáticas
Institute of Smart Cities
Universidad Pública de Navarra
Campus Arrosadía , 31006, Pamplona, Spain

Humberto Bustince

Dpt. Estadística, Informática y Matemáticas
Institute of Smart Cities
Universidad Pública de Navarra
Campus Arrosadía , 31006, Pamplona, Spain
King Abdullazih Universitiy, Jeddah, Saudy Arabia

Abstract

Image search can be tackled using deep features from pre trained Convolutional Neural Networks (CNN). The feature map from the last convolutional layer of a CNN encodes descriptive information from which a discriminative global descriptor can be obtained. We propose a new representation of *co-occurrences* from deep convolutional features to extract additional relevant information from this last convolutional layer. Combining this *co-occurrence* map with the feature map we achieve an improved image representation. We present two different methods to get the *co-occurrence* representation, the first one based on direct aggregation of activations, and the second one, based on a trainable *co-occurrence* representation. The image descriptors derived from our methodology improve

Email address: jiforcen@das-nano.com (Juan I. Forcen)

the performance in very well-known image retrieval datasets as we prove in the experiments.

Keywords: *co-occurrence*, image retrieval, feature aggregation, pooling

---

1. Introduction

Visual image search has rapidly evolved from variants of the bag-of-words model [1] based on local features, typically SIFT [2], to approaches focused on deep Convolutional Neural Network (CNN) features [3]. The first contributions in image retrieval using deep features were proposed by Razavian et al. [4] and Babenko et al. [5]. Basically they established different aggregation strategies for deep features and demonstrated state-of-the-art performance in popular benchmarks. According to these results, taking into account that representations for image retrieval need to be compact, i.e., around a few hundred dimensions, recent contributions have been made, in order to improve the quality of the final image representation. Relevant works as Tolias et. al. [6] or Kalantidis et al. Basically in [7], have focused in the methodology of feature extraction from the layers of the network into compact feature vectors using an off-the-shelf CNN, commonly known as a general feature extractor. [7] established a straightforward way of creating powerful image representations by means of multidimensional aggregation and weighting: an image is feed in a CNN, obtaining a tensor A, i.e. the activation maps of the last convolution layer, then these deep convolutional features are aggregated to derive a final feature vector, i.e., the global image representation. Zheng et al. [3] define this category as *pre-trained single-pass* category of CNN-based approaches.

Other approaches have tried to fine-tune the CNN with training datasets related to test datasets [8]. These approaches improve results in particular datasets, but have the drawback of requiring training datasets with expensive annotations depending on a category of each test set.

In computer vision has been widely used the concept of *co-occurrence* matrix to represent textures among other visual features. A *Co-occurrence* matrix [9]

is defined from an image being the distribution of co-occurring pixel values at a given spatial offset. Recently, this concept of *co-occurrence* has been extended to the *co-occurrence* of features activations in convolutional layers [10]. In this approach, said *co-occurrence* layer calculates the correlation between each pair of feature maps by means of the maximum product of the activations given a set of spatial offsets. This *co-occurrence* representation obtain a 1-dimension vector which contains one correlation value for each possible pair of channels, therefore does not contain spatial information. Our proposed *co-occurrences* representation is a tensor with the same dimension of the original activation tensor, which contains for each location their *co-occurrences* information. In this paper we applied the *co-occurrence* representation to obtain an improved image representation for image retrieval applications. The contributions of this paper can be summarized as follows:

- We propose a new definition for *co-occurrence* representation of convolutional deep features.
- We introduce a new concept of *co-occurrence* filter. This filter is able to capture the dependencies between channels activations to obtain an improved *co-occurrence* representation.
- We propose a linear and a bilinear pooling approach based on *co-occurrence*, over off-the-shelf CNN convolutional features, for image retrieval.
- We demonstrate in the experimental results the effectiveness of our method to well-known image retrieval datasets, and we compare our method with the state-of-the-art techniques.

The rest of this paper is organized as follows. In Section 2, related work with our proposal is recalled. In Section 3, our new definition of *co-occurrence* and its implementation by means of a *co-occurrence* filter is proposed. Next, in Section 4 are explained two pooling methodologies used to obtain a final image descriptor from our deep *co-occurrences* tensor, followed by Section 5 where a methodology to learn the best *co-occurrence* representation is presented. Finally

in Section 6 our method is compared with other *co-occurrence* representation and with state of the art image retrieval methods. Finally, the conclusions and future work are highlighted in Section 7.

2. Related work

CNN-based retrieval methods have been emerged in recent years and are replacing the classical local detectors and descriptors methods. Several CNN models pretrained in giant datasets like Imagenet [11], serve as good choices for extracting features, including VGG [12] or ResNet [13]. Based on the transfer learning principle, the first idea was to extract an image descriptor from a fully-connected layer of the network, however, it has been observed that the pooling layer after the last convolutional layer (e.g., pool5 in VGGNet), usually yields superior accuracy than the fully-connected descriptors and other convolutional layers [6]. Basically, [4] and [6] proposed a feature aggregation pipeline using max-pooling that, in combination with normalization and whitening, obtained state-of-the-art results for low dimensional image codes. Following these results, research efforts have been focused on the aggregation of the features from the pre-trained CNNs. This means, to identify proper spatial regions or weighting functions to obtain a low dimensional image representation. For example, Babenko et al. [5] used a global sum pooling with a center priority, and Kalantidis et al. [7] proposed a non-parametric spatial weighting method focusing on activation regions and a channel weighting related to activation sparsity with global sum pooling. Similarly, Cao et al. [14] proposed a method to derive a set of base regions directly from the activations of the convolutional layer. Jimenez et al. [15] studied the class activation maps for spatial weighting and Mohedano et al. [16] and Simeoni et al.[17] proposed to use different human-based saliency measures for spatial pooling. However, all of these methods does not take into account the correlation between the features of the convolutional layers. This concept of correlation between image features was introduced by Yang et al. [18], named as feature *co-occurrence*, characterizes the spatial dependency of

the visual features in a given image. Recently, Shih et al. [10] introduce a new *co-occurrence* representation for deep convolutional networks, demonstrating its effectiveness to exploit the information of visual features in the field of visual recognition. But, in the wrong side this method loses spatial information and its execution is very slow.

To overcome these problems, in this paper we consider that the correlation or interdependence between feature maps contains useful information, so in order to improve the final accuracy in the image search problem, we propose to add the *co-occurrence* information to the final image descriptor using a new representation of deep *co-occurrence*.

3. Deep *co-occurrence* Tensor of Deep Convolutional Features

In convolutional neural networks when an image is feed in the CNN, the result after the last convolutional layer is an activation map A of size $M \times N$ and D channels, $A \in \mathbb{R}^{M \times N \times D}$. This activations map represents how much is activated each feature (represented in channels) for a given spatial position.

The goal of *co-occurrence* representation is to characterize the spatial dependency of the image features. Yang et al. [18] call it *"Spatial co-occurrence Kernel"* and considered it as a count of the times that two visual features satisfy a spatial condition. Shih et al. [10] present a new idea behind the *co-occurrence* representation, recording the spatial correlation c between a pair of feature maps k and w, seeking the maximal correlation response for a set of spatial offsets $o_{ij} = [o_{ij,x}, o_{ij,y}]^\top \in \mathbb{R}^2$.*i.e.*

$$c(k, w) = \max_{o_{ij}} \sum_{p \in [1,m] \times [1,n]} a_p^k a_{p+o_{ij}}^w \tag{1}$$

where a_p^k is the k_{th} channel of A at location p. And $a_{p+o_{ij}}^w$ is the w_{th} channel of a at location $p + o_{ij}$.

In this approach the spatial information is lost in the resultant *co-occurrence* vector c of size D^2, due to the *co-occurrence* between two channels is a single value. Moreover, this high dimensionality makes it unaffordable in deep tensors

like VGG with 512 channels. For this reason, Shih et al. [10] add a $1 \times 1 \times N$ convolution filter to reduce the number of channels before the *co-occurrence* layer. As a side effect this channel reduction causes a reduction of performance as demonstrated in [10]; this representation was also used in [19].

3.1. Deep-Co-ocurrence-Tensor

In this section we propose a new method to obtain a *co-occurrence* representation from an activation tensor of a deep convolutional layer. This *co-occurrence* representation is a tensor with equal dimensions than the activation tensor which encodes the correlation between feature maps for each tensor location.

We define that a *co-occurrence* happens when the value of two different activations, $a^k_{i,j}$ and $a^w_{u,v}$, are greater than a threshold, t, and both are spatially located inside of a given region. (graphical interpretation is depicted in Figure 1.).

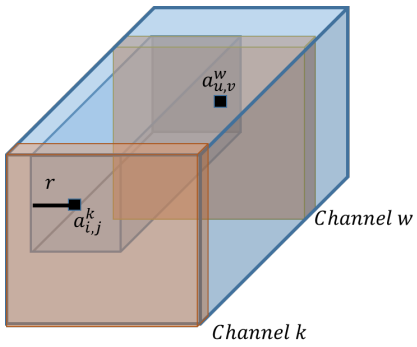


Figure 1: A *co-occurrence* occurs when the activation of two different activations $a^k_{i,j}$ and $a^w_{u,v}$ are greater than a threshold t and both are spatially located inside of a given region.

Given a convolutional map $A \in \mathbb{R}^{M \times N \times D}$, containing a set of activations, given a distance r and a threshold t, we define a positive *co-occurrence* between

two activations $a^k_{i,j}$ and $a^w_{u,v}$ as:

$$\rho(a^k_{i,j}, a^w_{u,v}) = \begin{cases} 1, & \text{if } |i-u| \leq r \text{ and } |j-v| \leq r \text{ and} \\ & a^k_{i,j} > t \text{ and } a^w_{u,v} > t \\ 0, & \text{otherwise.} \end{cases} \quad (2)$$

The resultant elements of the *co-occurrence* tensor are the sum of all the activations at the positive *co-occurrences* divided by the number of channels. So, the *co-occurrence* tensor is represented as $C_T \in \mathbb{R}^{M \times N \times D}$, calculated by:

$$C_T(i,j,k) = \sum_{u=1}^{M} \sum_{v=1}^{N} \frac{1}{D-1} \sum_{w=1}^{D} \rho(a^k_{i,j}, a^w_{u,v}) \cdot a^w_{u,v} \quad (3)$$

Remark: The *co-occurrence* of a channel with itself is not considered for the calculation of the *co-occurrence*, for this reason all the activations aggregated are divided by $D-1$.

3.1.1. Co-occurrence and image representativeness

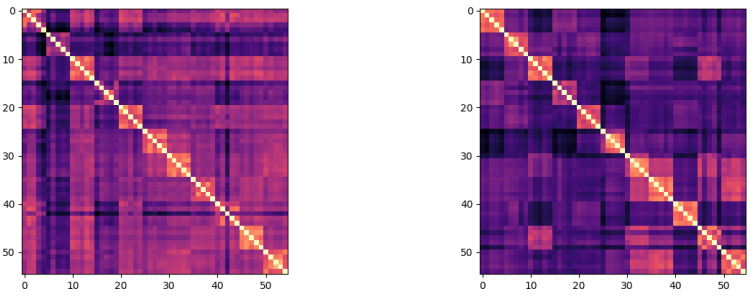
In order to study the representativeness of our proposed *co-occurrence* tensor we visualize the pair-wise correlation of the query images of Oxford [20] and Paris [21] datasets. We calculate the total *co-occurrence* vector $C_V \in \mathbb{R}^{1 \times D}$ as the sum of all the *co-occurrences* per channel:

$$C_V(k) = \sum_{i=1}^{M} \sum_{j=1}^{N} C_T(i,j,k) \quad (4)$$

We use these vectors C_V of dimension $1 \times D$ to compute the pair-wise correlation between images. The query-sets for both datasets contain 55 images in total, with 5 images of 11 classes of landmarks. We calculate the C_T for each query image with $r = 4$ and threshold t as the average mean of all the activations at A (being A the last pooling layer called 'pool5' obtained from a VGG16 pre-trained network in Imagenet dataset).

In Figure 2 we observe that the *co-occurrence* representation is highly correlated for images of the same landmark and less correlated for images of different

landmarks. These figures, evidence that the *co-occurrence* tensor contains discriminative information and therefore could be useful to obtain a representative vector applied to image search.



(a) Oxford (b) Paris

Figure 2: Correlation of *co-occurrence* vector CV for the 55 images in the query-set of Oxford (a) and Paris (b) datasets. Images are sorted by landmark.

3.1.2. Co-occurrence filter representation

An important advantage of our *co-occurrence* representation is that it can be implemented using convolutional filters. We define the *co-occurrence* filter as a convolutional filter: $F \in \mathbb{R}^{D\times D\times S\times S}$, where D is de number of channels in the activation tensor A, S the window size being $S = 2\cdot r+1$, with r the radius that defines the *co-occurrence* region (see Figure 1).

Note that activations do not compute to their channel *co-occurrence* calculation. So all filters elements are initialized to one except the related with itself channel, that are initialized with zero or a small value ε, for example i.e. $1e-10$.

$$F_{a,b,c,d} \in \mathbb{R}^{D\times D\times S\times S} = \begin{cases} 1, & \text{if } a = b \\ 0. & \text{otherwise} \end{cases} \tag{5}$$

Given an activation tensor, $A \in \mathbb{R}^{M\times N\times D}$, of last convolution operator in a neural network, the *co-occurrence* tensor $C_T \in \mathbb{R}^{M\times N\times D}$ can be obtained as a convolution between a thresholded activation tensor and the *co-occurrence*

filter:

$$C_T = (A_{\rho_A} * F) \cdot \rho_A \tag{6}$$

where $A_{\rho_A} = A \cdot \rho_A$ and $\rho_A \in \mathbb{R}^{M \times N \times D}$, $\rho_A = A > \overline{A}$, with $\overline{A}$ the average mean of the activation map, i.e.:

$$\rho_A(i,j,k) = \begin{cases} 1, & \text{if } a^k_{i,j} > \frac{1}{M \cdot N \cdot D} \sum_{i=1}^{M} \sum_{j=1}^{N} \sum_{k=1}^{D} a^k_{i,j} \\ 0. & \text{otherwise} \end{cases} \tag{7}$$

The pseudo-code of the implementation is shown in Algorithm 1 and in figure 3 is depicted an illustration of the *co-occurrence* tensor calculation.

Algorithm 1: *Co-occurrence* tensor.

calcCooc (A, S);

Input : A : Tensor of activations with shape $D \times M \times N$

S : window size

Output: C_T : Tensor of *co-occurrences*

$filters = ones(D, D, S, S)$

For $i = 1$ *to* D :

    $filters[i, i, :, :] =$ 1e-10

$\rho_A = A > mean(A)$

$A_{\rho_A} = A \cdot \rho_A$

$C_T = conv2d(A_{\rho_A}, filters, padding = r)$ / $(D - 1)$

$C_T = C_T \cdot \rho_A$

return C_T

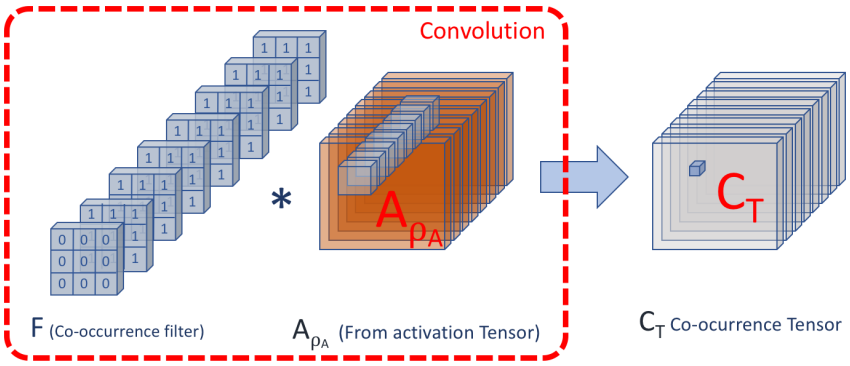


Figure 3: Example of the *co-occurrences* filter $F_0 \in \mathbb{R}^{D \times S \times S}$, convolved with A_{ρ_A} to obtain a value in position (i, j) in the *co-occurrences* tensor (Best viewed in color).

This *co-occurrence* implementation based on convolutions, makes possible to calculate the *co-occurrence* representation with no channel reduction. Moreover, it is simple and straightforward to learn an improved *co-occurrence* representation by adding a *co-occurrence* layer in a trainable architecture.

In the next sections are introduced two different ways to use the *co-occurrence* tensor applied to image retrieval. The first one, called *Direct co-occurrences* uses a straightforward approach to calculate *co-occurrence* (see Algorithm 1) and in the second one called *Learnable co-occurrences* a *co-occurrence* filter is learned to obtain an improved *co-occurrence* representation.

Implementation of the proposed method and some examples, in PyTorch are publicly available: `https://github.com/jiforcen/co-occurrence`.

4. Direct *co-occurrences*

Off-the-self methods of image retrieval obtain a compact representation of the images and queries to perform the query search, typically in three steps: (1) to feed the image in a pre-trained CNN to extract its activation tensor, (2) to apply a pooling function to obtain a compact representation and (3) ro apply l2-normalization and pca / whitenning to reduce the dimensionality and increment the discriminative power.

In our method the compact representation, the second step, is obtained by pooling the activation tensor with the *co-occurrence* tensor (eee Figure 4). We have used two different methods, linear and bilinear pooling, to perform this aggregation and demonstrate the effectiveness of the *co-occurrence* representation. Next, these two approaches are explained in detail.

4.1. Linear weighted pooling

Following the lineal weighted pooling methodology proposed by Kalantidis et al. [7], we transform the original tensor of activations into a new weighted tensor:

$$A'_{i,j,k} = \alpha_{i,j}\beta_k A_{i,j,k} \tag{8}$$

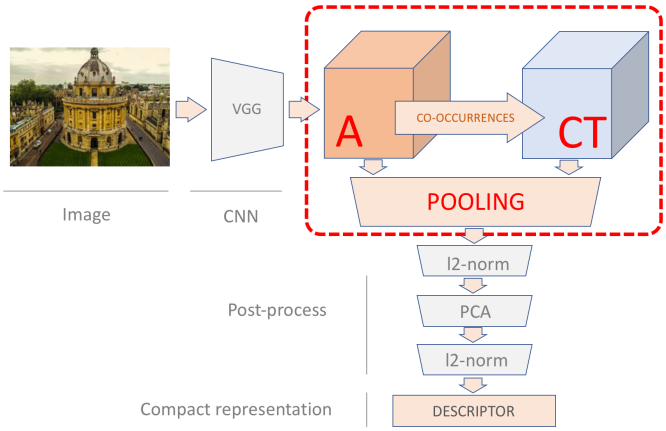


Figure 4: Aggregation pipeline to obtain the final compact image representation

Where $A \in \mathbb{R}^{M \times N \times D}$ is the tensor of activations from the last convolutional layer, $\alpha_{i,j}$ are the spatial *co-occurrences* obtained from the *co-occurrence* tensor and β_k are channel *co-occurrences*, also obtained from the *co-occurrence* tensor. The final step of lineal pooling is the sum the new A' tensor of activations by channel to obtain a single vector of dimension $1 \times D$.

The spatial *co-occurrences* $\alpha_{i,j}$ basically are the normalized total *co-occurrence* across all channels for every image location (i, j), taking into account that spatial locations with large *co-occurrences* across channels should correspond to discriminative locations of the image.

We apply a power normalization to obtain the final spatial *co-occurrences* matrix:

$$\alpha_{i,j} = \left(\frac{S(i,j)}{\left(\sum_{i=0}^{i=M} \sum_{j=0}^{j=N} S(i,j)^a \right)^{1/a}} \right)^{1/b} \tag{9}$$

where $S \in \mathcal{R}^{M \times N}$ is the matrix of aggregated *co-occurrences* from all channels per spatial location:

$$S(i,j) = \sum_{k=1}^{D} CT(i,j,k) \tag{10}$$

Figure 5 depicts spatial *co-occurrences* $\alpha_{i,j}$ for several images of the Oxford dataset (*co-occurrence* tensor C_T is obtained by Equation 3). We visualize that

our spatial *co-occurrences* tends to give large values to locations with salient visual content and reject background information.

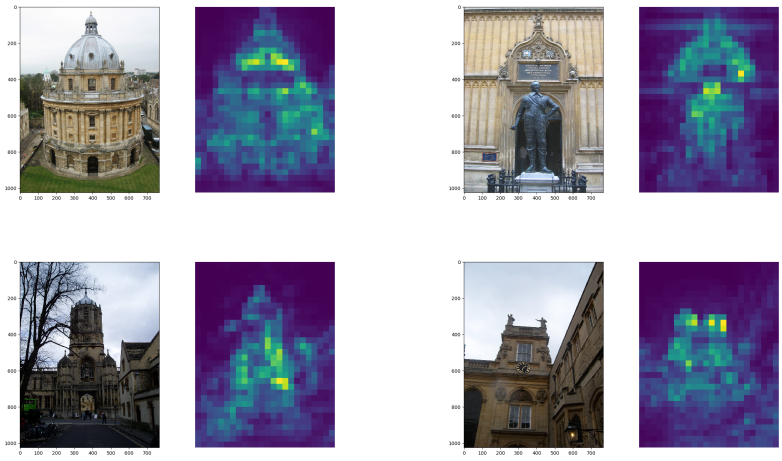


Figure 5: Spatial *co-occurrences* obtained from Eq. (9) for images of Oxford dataset.

To calculate the importance of each channel the inverse of its *co-occurrence* value is used, such a way we boost the contribution of rare features (in a similar way as term frequency inverse document frequency):

$$\beta_k = log\left(\frac{\sum_{l=0}^{D} VC(l)}{\epsilon + VC(k)}\right) \tag{11}$$

where ϵ is a constant to avoid division by zero.

4.2. Bilinear pooling

Bilinear, or second order pooling, introduced by Tenenbaum et al. in [22], has the ability to capture pairwise correlations between channels of a descriptor. It have been successfully applied for different tasks, like semantic segmentation [23], fine grained visual recognition [24] or face recognition [25]. The bilinear pooling of a single dimension descriptor x can be calculated as the outer product of x an its transpose x^T. The outer product captures pairwise correlations between the feature channels and can model feature interactions. In our approach

we combine the tensor of activations with the *co-occurrence* tensor:

$$B(A, C_T) = \sum_{i}^{M} \sum_{j}^{N} A_{ij} \times C^T_{Ti,j} \tag{12}$$

Where $A_{i,j} \in \mathbb{R}^{1\times D}$ is the vector of activations and $C_{T_{i,j}} \in \mathbb{R}^{1\times D}$ the vector of *co-occurrences* at position (i, j). The main disadvantage of bilinear pooling is the large size of the resultant descriptor $B = \mathbb{R}^{D\times D}$. For example an activation tensor with $D = 512$ channels will produce a final descriptor with about $512 \times 512 \approx 250K$ values, which is excessive in most cases. To tackle with that problem, Y. Gao et al. [26] proposed compact bilinear pooling which is a kernelized view of bilinear pooling, achieving almost equal results of bilinear pooling with a final vector of 8K values which means a reduction of two orders of magnitude. We will use the compact bilinear pooling implementation in the experiments to reduce the $D \times D$ bilinear matrix into a $1 \times 8K$ vector.

4.3. Experiments

In this section we present the results obtained with our proposal of direct *co-ocurrences* combined with linear and bilinear pooling. To perform the experiments we use common image retrieval datasets: Oxford [20], Paris [21], ROxford [27], RParis [27] and Holidays [28]. The Oxford Buildings Dataset consists of 5062 images with 11 different landmarks, each represented by 5 possible queries, Paris dataset is similar, created with 6412 Paris images. ROxford and RParis are a review of Oxford and Paris with three different difficulty levels. The Holidays dataset has 1491 images and 500 queries of holiday photos.

In addition we have used image retrieval common strategies, like spatial masks and query expansion techniques to reach improved performance results.

4.3.1. Spatial masks

Yandex et al. [5] supposed that the objects tend to be located close to the geometrical center of an image. So, they incorporate such centering prior with a simple heuristic, which assigns larger weights to the features from the center of the feature map. In the experimental study we evaluate this mask and we also

try a new mask configuration, Top-Down, with larger weights to the features in the top of the feature map. This can be useful in datasets like Oxford and Paris which contains discriminant features in the top of the image such as high buildings and less discriminant objects in the bottom like floor, grass, trees, roads or even persons.

4.3.2. Query expansion

Query expansion repeats the search process based on a new query composed by the top ranked results on the first query [29], [30], [8], [7]. We use two different query expansion methods in our experiments. The simplest method, called average query expansion $_{\text{A}}QE$, is the average of the n nearest descriptors after the first query. The second method, denoted alpha query expansion αQE, presented by Radenovic et al. [31], which is a weighted aggregation of the nearest descriptors.

4.3.3. Linear weighted pooling results

For each image, the tensor of activations A is the result of the last pooling layer "pool5" obtained from a VGG16 pretrained network in Imagenet dataset (similar to [6], [7], [15] and [32]). The tensor A has 512 channels, and its dimensions (height and width) are proportional to the input image size, due to we pass input images through the network with their original size. The tensor is processed following the scheme presented in Subsection 4.1 resulting in a 512 dimensions single vector per image. In Oxford experiments, Paris dataset is used for PCA purposes, whilst in Paris and Holidays dataset Oxford dataset is used. For each image query, all of the images of the dataset are sorted according to their euclidean distance. Also a simple query expansion $_{\text{A}}QE$ method is applied in Oxford and Paris dataset. To evaluate the performance in the given queries of these datasets we use the mean Average Precision metric (mAP), which is the standard procedure used in the literature.

In Table 1 we present the comparison of our *co-occurrence* based linear weighting pooling with other linear weighting methods. Ucrow [7] is the simplest

way to aggregate the activation tensor in a compact descriptor, being calculated as the average of $A \in \mathbb{R}^{M \times N \times D}$ over the dimensions M and N. Also, we compare with the method proposed in Kalantidis et al. [7] called crow.

These two methods are compared with out method ChCO-SC_T which is a combination of channel weighting $ChCO$ and spatial weighting [1] SC_T to perform the linear weighted pooling aggregation of the activation tensor and *co-occurrence* tensor.

Regarding the spatial masks (Subsection 4.3.1) we have test a Top-Down weighting matrix, in which pixels of the top rows have higher weight than pixels in the bottom ($SpTD$) and a center prior weighting matrix [5], in which pixels of the center have higher value than pixels in the borders ($SpCt$).

The *co-occurrence* tensor C_T was calculated with $r = 4$ [2] and the threshold t is the average mean of all the activations of the tensor A.

Analysing Table 1, we can appreciate that the *co-occurrence* based pooling is very helpful to obtain representative image vectors (ChCO-SC_T), because weighted vectors improve uniform aggregation (ucrow). Moreover, using ChCO-SC_T our performance is similar in Paris and Oxford and better in Holidays than the crow[7] method, which is a state-of-the-art in pre-trained single pass methodologies. Therefore, the *co-occurrence* tensor captures feature correlations and is able to provide better image representations. We have found that we get a large improvement in our results if the feature tensor A is multiplied by the spatial masks, top-down $SpTD$ or center prior $SpCt$. Basically, top-down mask assumes upright images, which is the case of Oxford and Paris datasets due to all of the queries are buildings, and confirms that the spatial structure of the image is relevant in image retrieval.

[1] As in [7] the parameters used in the power normalization were $a = 2$ and $b = 2$.

[2] Previous experiments with $r = 4$, $r = 6$ and $r = 8$ showed us that size influence was low, with the best case $r = 4$.

	Oxford	ROxford			Paris	RParis			Holidays
		Easy	Medium	Hard		Easy	Medium	Hard	
Method	mAP	mAP	mAP	mAP	mAP	mAP	mAP	mAP	mAP
ucrow	66.0	60.53	41.15	11.98	75.8	74.75	57.69	30.20	81.1
crow[7]	67.2	61.92	44.66	17.94	78.7	76.13	60.17	33.38	82.5
ChCO-SC_T	67.05	61.31	44.52	18.71	79.17	77.16	60.69	33.89	83.22
ChCO-SC_T + $SpTD$	**71.68**	**63.37**	**47.63**	**21.96**	**80.89**	**78.79**	**62.59**	**36.84**	83.04
ChCO-SC_T + $SpCt$	66.82	61.67	43.05	14.91	80.18	77.32	61.05	33.79	**83.96**
ucrow + $\text{A}QE$	70.5	55.32	39.94	13.08	82.7	81.32	65.29	38.72	-
crow + $\text{A}QE$ [7]	71.5	55.75	42.11	17.89	85.5	81.82	67.93	43.16	-
ChCO-SC_T + $\text{A}QE$	71.30	56.41	42.46	18.93	85.31	83.59	69.03	44.12	-
ChCO-SC_T + $SpTD$ + $\text{A}QE$	**77.48**	60.27	**46.58**	**21.19**	**87.13**	85.19	**71.01**	**46.43**	-
ChCO-SC_T + $SpCt$ + $\text{A}QE$	72.99	**61.83**	44.57	16.79	86.08	**86.04**	70.93	44.93	-

Table 1: Results of linear weighted pooling aggregation based on the *co-occurrence* tensor for the following datasets: Oxford, Paris, Holidays, ROxford and RParis. (Coocurrences are calculated with r=4)

4.3.4. Bilinear pooling results

In this experiment are evaluated the results obtained using the bilinear pooling method (Section 4.2). All parameters and measures are equal to the previous experiments (4.3.3). In Table 2 we compare the results of the bilinear pooling of activations A and *co-occurrences* C_T ($BP(AC_T)$) with the performance of bilinear pooling with itself A ($BP(AA)$). So, we can evaluate if the *co-occurrence* provides useful information, also combined with the spatial masks. We conclude that the combination of *co-occurrences* and activations is better in almost all the cases than the multiplication of the activations by itself ($BP(AA)$). Therefore, as in the previous experiment, we have proved that the *co-occurrences* reflects the correlations of the features and provides better final image representations. The performance of the bilinear pooling is similar than the previous linear combination for Oxford and Paris datasets but worse in Holidays, and the Top-Down spatial mask influences in a similar way than the previous experiment.

Compact bilinear pooling aggregation of 512 depth vectors result in a vector of 8192 features, in experiments showed in Table 2 we have reduced said vector in the PCA step to 512 features, in order to compare it with linear weighted pooling.

In Table 3 are shown the results with larger final representation size. As we

	Oxford	ROxford			Paris	RParis			Holidays
		Easy	Medium	Hard		Easy	Medium	Hard	
Method	mAP	mAP	mAP	mAP	mAP	mAP	mAP	mAP	mAP
$BP(AA)$	65.94	60.410	45.99	21.66	75.88	73.66	57.80	30.67	80.42
$BP(AC_T)$	64.23	59.76	44.12	19.75	77.77	77.18	60.09	33.080	**82.62**
$BP(AC_T) + SpTD$	**71.00**	**63.48**	**47.62**	**22.83**	**79.56**	**77.72**	**62.14**	**37.09**	79.51
$BP(AA) + \text{A}QE$	67.66	48.670	37.76	15.76	81.00	77.71	63.76	37.87	-
$BP(AC_T) + \text{A}QE$	66.08	51.24	38.22	15.9	82.97	83.03	67.06	40.85	-
$BP(AC_T) + SpTD + \text{A}QE$	**77.25**	**56.01**	**43.69**	**19.33**	**84.84**	**83.17**	**68.88**	**44.38**	-

Table 2: Results of bilinear weighted pooling aggregation based on the *co-occurrence* tensor for the following datasets: Oxford, Paris, Holidays, ROxford and RParis. (Final vector size used is 512.)

can observe larger vectors produce a great improvement in the mAP results, being much better than the linear approach.

		Oxford	ROxford			Paris	RParis			Holidays
			Easy	Medium	Hard		Easy	Medium	Hard	
Method	FV Size	mAP	mAP	mAP	mAP	mAP	mAP	mAP	mAP	mAP
$BP(AC_T) + SpTD$	512	71.00	63.48	47.62	22.83	79.56	77.72	62.14	37.09	79.51
$BP(AC_T) + SpTD$	1024	72.74	65.22	48.83	23.93	82.19	79.88	64.15	39.34	
$BP(AC_T) + SpTD$	2048	73.76	65.98	50.14	26.14	83.08	81.46	65.49	40.76	81.10
$BP(AC_T) + SpTD$	4096	**76.56**	**67.86**	**51.92**	**27.15**	**83.68**	**81.93**	**64.95**	**39.92**	**81.75**
$BP(AC_T) + SpTD + \text{A}QE$	512	77.25	56.01	43.69	19.33	84.84	83.17	68.88	44.38	-
$BP(AC_T) + SpTD + \text{A}QE$	1024	77.49	58.67	46.21	21.99	87.45	85.40	71.28	47.24	-
$BP(AC_T) + SpTD + \text{A}QE$	2048	79.03	59.38	46.68	22.19	88.65	86.94	72.99	49.75	-
$BP(AC_T) + SpTD + \text{A}QE$	4096	**81.71**	**64.33**	**51.28**	**26.64**	**89.16**	**88.11**	**73.74**	**50.23**	-

Table 3: Comparison of bilinear accuracy incrementing the final vector size.

5. Learning *co-occurrences*

In this section is explained how a *co-occurrence* filter, F, can be learned within a trainable architecture to obtain a better *co-occurrence*, C_T, representation. We have used a Siamese architecture due to it is known to produce highly discriminative embeddings [33] [34], [35].

5.1. Siamese learning

The Siamese architecture is trained with paired samples, maximizing the similarity of positive pairs representation and also maximizing the dissimilarity

of negative pairs. In Figure 6 is presented the Siamese architecture adopted. It contains two equal branches that share parameters; each branch is represented as CoOc-NET. The output of the CoOc-NET is the l2-normalization of the resultant vector after the bilinear pooling operation between the activation tensor and its *co-occurrence* tensor. In our experiments only the *co-occurrence* filter is trained, and the rest of VGG layers are freeze, so the *co-occurrence* filter weights change to obtain more discriminative representations.

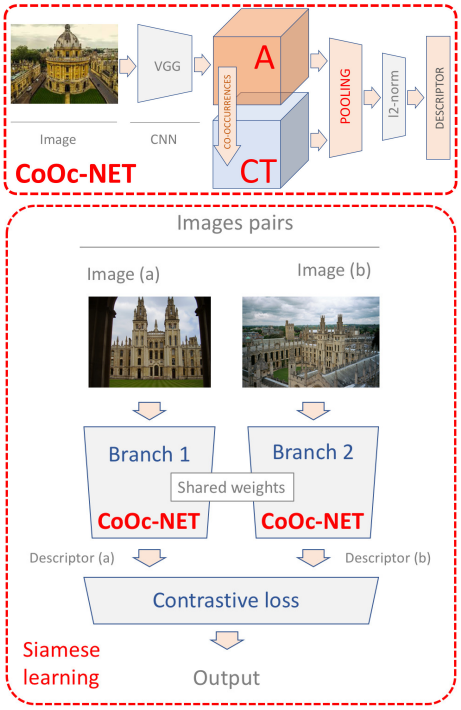


Figure 6: In the top the CoOcNET pipeline used to obtain a compact representation from an image combining the *co-occurrences* tensor. In the bottom the siamese architecture based in two equal branches sharing weights and contrastive loss.

This Siamese architecture is trained in combination with contrastive loss [33] using as train input a pair of images $P_{Im} = [Im_a, Im_b]$ which produces descriptors f_a and f_b. Label $Y(P_{Im})$ will be 1 if the images of the pair corresponds to the same class or 0 if not. Following this notation contrastive loss will be

calculated as:

$$d = \|f_a - f_b\|_2 \tag{13}$$

$$Ł(P_{Im}) = Y(P_{Im}) * d^2 + (1 - Y(P_{Im})) * \max(\tau - d, 0)^2 \tag{14}$$

where τ is a parameter to establish a distance margin where dissimilar pairs influence the loss or not.

5.2. Experiments

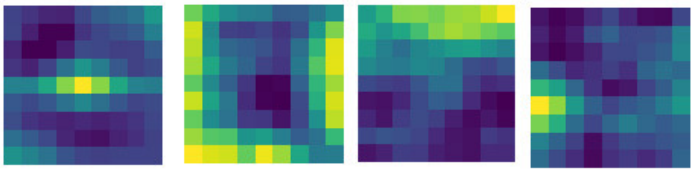
In this section we present the results obtained using the trainable *co-occurrence* filter, the experiments done are equivalent to section 4.3.

5.2.1. Training Step

In the training process of Siamese architecture we have used an image dataset called retrieval-SfM published by Radenovic et al. [31]. This dataset contains 163k images grouped in 713 clusters of images. We have used the pairing images selection procedure of [31]. Other parameters are: batch size equal to 5 and Adam [36] optimizer with momentum 0.85 and learning rate 1e-9.

As we have explained previously only the *co-occurrence* filter is learned, freezing the rest of the network. This *co-occurrence* filter fine-tunning process takes only around 30 epochs to achieve the best result in the validation set. In Figure 7 are shown four 9×9 filters after the training process. Each filter represents the correlation between a pair of channels.

In the Figure 7, direct and learned *co-occurrence* representations are compared. After the learning process the *co-occurrence* representation shows the ability to put more emphasis in representative regions to discern between queries, for example buildings have higher activation values and a irrelevant objects or persons are dismissed from the *co-occurrence* representation. In cases 7 (h) and 7 (i) we can see that the *co-occurrence* between features representing humans are avoided, and only the *co-occurrence* of the features representing buildings are taken into account.



(a) 5 - 5 (b) 12 - 8 (c) 5 - 9 (d) 1 - 9

Figure 7: Filter examples of size $9x9$ pixels where each spatial offset is weighted to obtain the best *co-occurrence* representation. Each image represents the filter between a pair of channels. (Best viewed in color)

5.2.2. Evaluation

In this section is evaluated the *co-occurrence* representation after the *co-occurrence* filter training process in a CoOcNET pipeline. The evaluation is performed similar to [31], with its same whitening procedure, alpha query expansion method αQE, and testing also each query in multiscale, ms, $(1, \sqrt{\frac{1}{2}}, \frac{1}{2})$.

In table 4 are shown the results comparing bilinear pooling with the itself activation tensor $BP(AA)$, bilinear pooling combining the activation tensor and the *co-occurrence* tensor with fix-weights $BP(AC_T)$ and with the learned weights $BP(AC_T)_{learn}$. It is easy to observe that learning *co-occurrences* $BP(AC_T)_{learn}$ obtains the best result, showing its ability to capture even better the relations between the features of the activation map.

6. Comparison with State-of-the-art results

In this section we compare the results of our method with the Shih et al. [10] *co-occurrence* interpretation method and also with other state-of-the-art methods in image retrieval.

6.1. Co-occurrence representation comparison

Shih et al. [10] define *co-occurrences* as the maximal correlation between a pair of feature maps, for a set of spatial offsets, whilst we define *co-occurrences*

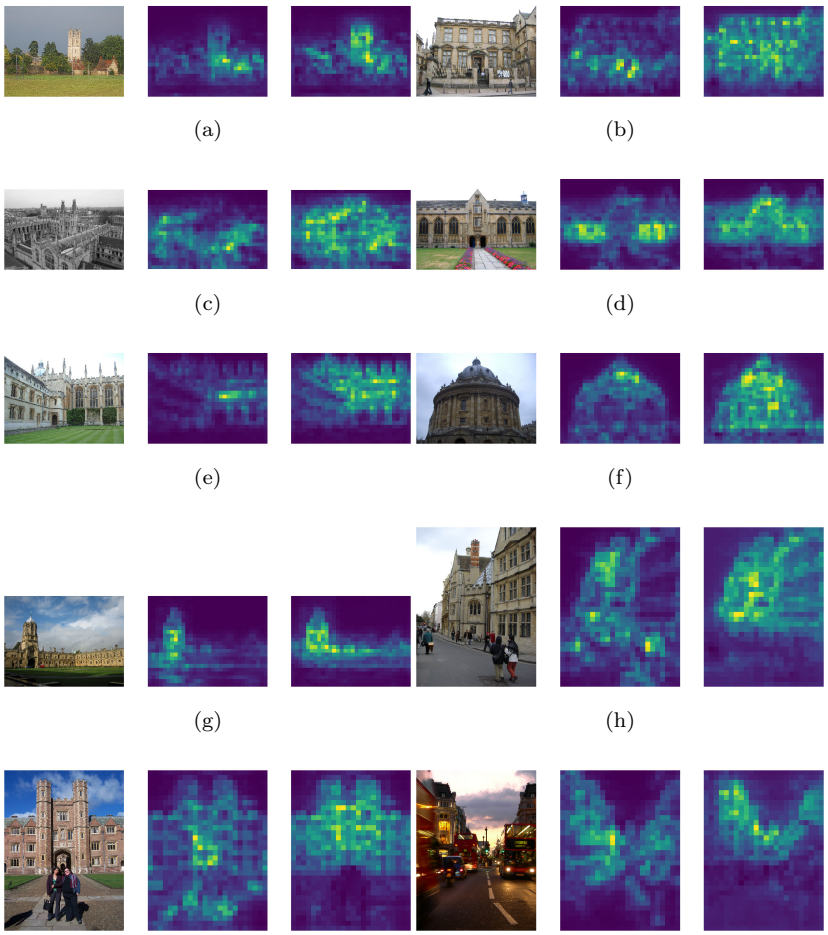


Figure 8: *co-occurrence* representation of the *co-occurrence* tensor (sum over channels) for some example images, showing for each one its *co-occurrence* representation with fixed weights (left) and with trainable weights (right), (Best viewed in color).

as the sum of the activations inside a region, being the activations value above a threshold. (Section 3.1).

In order to compare these two different interpretations, we have modified Shih et al. [10] method to return a *co-occurrence* tensor instead of a *co-occurrences* vector. This modification consists in picking the summed maxi-

	Oxford	ROxford			Paris	RParis		
		Easy	Medium	Hard		Easy	Medium	Hard
Method	mAP	mAP	mAP	mAP	mAP	mAP	mAP	mAP
$BP(AA)$ + ms	76.67	72.97	55.67	31.24	90.31	89.69	72.22	48.0
$BP(AC_T)$ + ms	76.79	72.53	56.34	32.93	89.55	90.13	71.86	48.22
$BP(AC_T)_{learn}$ + ms	**80.75**	**74.65**	**57.94**	**33.68**	**91.02**	**90.18**	**72.44**	**48.37**
$BP(AA)$ + αQE + ms	80.11	76.08	59.65	33.95	93.42	93.80	80.17	59.36
$BP(AC_T)$ + αQE + ms	81.13	74.53	60.86	38.50	93.23	93.93	80.35	60.44
$BP(AC_T)_{learn}$ + αQE + ms	**85.76**	**82.25**	**67.04**	**42.23**	**94.84**	**94.21**	**80.97**	**61.42**

Table 4: Results of bilinear weighted pooling aggregation using the pipeline proposed in 5 for learnable coocurrences in the following datasets: Oxford, Paris, Holidays, ROxford and RParis. (Final vector size used is 8192.)

mum correlation map instead of the own value. This produces a tensor 3D tensor $C_T^{'} = \mathbb{R}^{M \times N \times D^2}$ with D^2 channels. Each channel of this tensor represents the correlation between a pair of channels. The aggregation of all the correlations of each channel with the rest produce a tensor $C_T^{''} = \mathbb{R}^{M \times N \times D}$, with the same size than the original activation tensor A as in our method.

In Table 5 we present the comparison for linear and bilinear aggregation schemes using both *co-occurrence* representation methods. The mAP results obtained with the lineal aggregation are quite similar for both methods, but when bilinear aggregation is used our method outperforms significantly Shih et al. method. Furthermore Shih et al. method has an efficiency drawback in its implementation, because it is necessary to find the maximum correlation value for each pair of channels.

For this reason we have compared the execution time of the *co-occurrence* tensor C_T generation of both methods, with a subset of one hundred images of Paris6k. The feature map of each image was extracted with two pre-trained networks VGG [12] and ResNet [13], resulting in tensors of 32x24 (width x height) with 512 channels (VGG) and 2048 channels (ResNet). Also, we have studied the performance with a smaller tensor of 32 channels depth (because is the depth of tensors used in the Shih et al. implementation).

Table 6 shows the average time after fifty executions of each experiment

ROxford and RParis datasets							
		ROxford			RParis		
		Easy	Medium	Hard	Easy	Medium	Hard
Method	size	mAP	mAP	mAP	mAP	mAP	mAP
Shih ChCO-SC_T + $SpTD$	512	64.3	47.64	20.86	79.45	62.92	37.15
ChCO-SC_T + $SpTD$	512	64.28	47.02	20.30	79.01	62.18	35.79
ChCO-SC_T + $SpCt$	512	62.06	42.81	14.52	77.52	61.06	33.39
Shih $BP(AC_T)$ + $SpTD$	512	56.27	44.51	21.94	73.15	61.25	37.95
$BP(AC_T)$ + $SpTD$	512	63.48	47.62	22.83	77.72	62.14	37.09
Shih $BP(AC_T)$ + $SpTD$	4096	63.15	48.61	24.5	76.1	62.84	39.78
$BP(AC_T)$ + $SpTD$	4096	**67.86**	**51.92**	**27.15**	**81.93**	**64.95**	**39.92**
		mAP (AQE)	mAP (AQE)	mAP (AQE)	mAP (AQE)	mAP (AQE)	mAP (AQE)
Shih ChCO-SC_T + $SpTD$	512	56.7	44.29	18.2	85.04	71.56	47.55
ChCO-SC_T + $SpTD$	512	57.98	44.32	16.56	84.57	70.37	45.45
ChCO-SC_T + $SpCt$	512	61.23	44.34	17.44	86.7	71.24	44.91
Shih $BP(AC_T)$ + $SpTD$	512	53.1	43.66	19.31	79.49	68.04	45.55
$BP(AC_T)$ + $SpTD$	512	56.01	43.69	19.33	83.17	68.88	44.38
Shih $BP(AC_T)$ + $SpTD$	4096	59.82	49.33	25.34	85.36	74.45	52.37
$BP(AC_T)$ + $SpTD$	4096	**64.33**	**51.28**	**26.64**	**88.11**	**73.74**	**50.23**

Table 5: Results comparison betwen our proposed coocurrences and an adaptation os Shih et al method.

Tensor size	ours (single)	DeepCooc[10] (single)	ours (batch 5)	DeepCooc[10] (batch 5)
32x24x512 (VGG)	0.977 ms	464.699 ms	0.695 ms	495.66 ms
32x24x2048 (ResNet)	16.678 ms	2417.895 ms	5.578 ms	*Out-of-memory*
32x24x32	0.258 ms	29.321 ms	0.318 ms	35.227 ms

Table 6: Performance comparison between *co-occurrence* methods.

using a computer with a CPU i7-7700K@4.20GHz, GPU GeForce GTX1080Ti, and 32GB of RAM.

As we can see our implementation is more than hundred times faster than the Shih et al. method. Therefore, we have demonstrated that our method allows the use of *co-occurrences* representations, breaking the performance barrier that made *co-occurrences* calculation out of the reach for many applications.

6.2. Comparison with State-of-the-art results

In Table 7, we present the results in ROxford and RParis datasets of state-of-the-art methods which uses VGG as feature extractor. In the pre-trained single pass category we improve the state-of-the-art performance with ChCO-SC_T + $SpTD$ based in the linear aggregation of *co-occurrences* against well known image retrieval methods like crow [7], SPoC [5], MAC and R-MAC [6]

and GeM [31]. Moreover, with bilinear pooling we can obtain a final vector representation with higher dimensions than the number of channels of the last VGG layer. Using Off-The-Self VGG and $BP(AC_T)$ with a final vector size of 8192 a great mAP improvement is achieved.

Finally, we compare our *co-occurrence* representation based on trainable *co-occurrence* filter against GeM pooling [37] using the same training and fine-tunning procedure for both methods. Again we demonstrate a huge improvement as consequence of adding *co-occurrence* information to the final vector representation, even when only the *co-occurrence* filter is trained.

		ROxford			RParis		
		Easy	Medium	Hard	Easy	Medium	Hard
Method	size	mAP	mAP	mAP	mAP	mAP	mAP
Pre-trained single pass							
ucrow	512	60.53	41.15	11.98	74.75	57.69	30.20
ucrow + αQE	512	55.32	39.94	13.08	81.32	65.29	38.72
crow [7]	512	61.92	44.66	17.94	76.13	60.17	33.38
crow + αQE [7]	512	55.75	42.11	17.89	81.82	67.93	43.16
SPoC [5]	512		38.0	11.4		59.8	32.4
MAC [6]	512		37.8	14.6		59.2	35.9
R-MAC [6]	512		42.5	12.0		66.2	40.9
GeM [31]	512		40.5	15.7		63.2	38.8
ChCO-SC_T + $SpTD$ **(ours)**	512	63.37	47.63	21.96	78.79	62.59	36.84
ChCO-SC_T + $SpTD$ + αQE **(ours)**	512	60.27	46.58	21.19	85.19	71.01	46.43
$BP(AC_T)$ + ms **(ours)**	8192	72.53	56.34	32.93	90.13	71.86	48.22
$BP(AC_T)$ + αQE + ms **(ours)**	8192	**74.53**	**60.86**	**38.50**	**93.93**	**80.35**	**60.44**
Fine-Tunning							
Radenovic VGG16-GeM [31] + ms	512		61.9	33.7		69.3	44.3
Radenovic VGG16-GeM [31] + αQE + ms	512		66.6	38.9		74.0	51.0
$BP(AC_T)_{learn}$ + ms **(ours)**	8192	74.65	57.94	33.68	90.18	72.44	48.37
$BP(AC_T)_{learn}$ + αQE + ms **(ours)**	8192	**82.25**	**67.04**	**42.23**	**94.21**	**80.97**	**61.42**

Table 7: Comparison with state-of-the-art results for ROxford and RParis datasets.

7. Conclusions

In this work we have presented a new definition for *co-occurrence* tensor of deep convolutional features. This *co-occurrence* representation embeds relevant information of the image, allowing us to add discriminative information to the

compact final image representations for image retrieval. In addition, our *co-occurrence* implementation allows to learn the *co-occurrence* filter to have better *co-occurrence* representations.

In our approach we combine the proposed *co-occurrence* tensor by means of weighted linear pooling and bilinear pooling with the original tensor of activations in a simple a straight forward pipeline. In the experimental results we have evidenced that the *co-occurrence* tensor improve the results over the standard procedure, so the ability of *co-occurrences* to capture additional information and create powerful image representations was demonstrated.

For future research, we plan to study other aggregation and normalization schemes for *co-occurrence* and adapt our methodology on multi regional image representation [5].

References

References

[1] S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2 (2006) 2169–2178. doi:10.1109/CVPR.2006.68.

[2] D. G. Lowe, Object recognition from local scale-invariant features, in: Proceedings of the Seventh IEEE International Conference on Computer Vision, Vol. 2, 1999, pp. 1150–1157 vol.2. doi:10.1109/ICCV.1999.790410.

[3] L. Zheng, Y. Yang, Q. Tian, SIFT Meets CNN: A Decade Survey of Instance Retrieval, IEEE Transactions on Pattern Analysis and Machine Intelligence 40 (5) (2018) 1224–1244. doi:10.1109/TPAMI.2017.2709749.

[4] S. J. Razavian A. S., Azizpour H., C. S., CNN features off-the-shelf: an astounding baseline for recognition, 2014 Proceedings of the IEEE International Conference on Computer Vision.

[5] A. B. Yandex, V. Lempitsky, Aggregating local deep features for image retrieval, Proceedings of the IEEE International Conference on Computer Vision 2015 Inter (2015) 1269–1277. doi:10.1109/ICCV.2015.150.

[6] G. Tolias, R. Sicre, H. Jégou, Particular object retrieval with integral max-pooling of CNN activations, ICL 2016 - International Conference on Learning Representations, May 2016, San Juan, Puerto Rico. (2015) 1–12.

[7] Y. Kalantidis, C. Mellina, S. Osindero, Cross-dimensional weighting for aggregated deep convolutional features, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 9913 LNCS (2016) 685–701. doi:10.1007/978-3-319-46604-048.

[8] A. Gordo, J. Almazán, J. Revaud, D. Larlus, End-to-End Learning of Deep Visual Representations for Image Retrieval, International Journal of Computer Vision (2017)doi:10.1007/s11263-017-1016-8.

[9] R. M. Haralick, K. Shanmugam, I. Dinstein, Textural features for image classification, IEEE Transactions on Systems, Man, and Cybernetics SMC-3 (6) (1973) 610–621. doi:10.1109/TSMC.1973.4309314.

[10] Y. F. Shih, Y. M. Yeh, Y. Y. Lin, M. F. Weng, Y. C. Lu, Y. Y. Chuang, Deep co-occurrence feature learning for visual object recognition, Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017 2017-January (2017) 7302–7311. doi:10.1109/CVPR.2017.772.

[11] J. Deng, W. Dong, R. Socher, L. Li, and, Imagenet: A large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 248–255. doi:10.1109/CVPR.2009.5206848.

[12] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-

scale image recognition, in: International Conference on Learning Representations, 2015.

[13] A. Krizhevsky, I. Sutskever, G. E. Hinton., AlexNet, Advances in neural information processing systems (2012)doi:10.1016/B978-008046518-0.00119-7.

[14] J. Cao, L. Liu, P. Wang, Z. Huang, C. Shen, H. T. Shen, Where to Focus: Query Adaptive Matching for Instance Retrieval Using Convolutional Feature Maps (2016) 1–10.

[15] A. Jimenez, J. M. Alvarez, X. Giro-i Nieto, Class-Weighted Convolutional Features for Visual Instance Search, in: In 28th British Machine Vision Conference (BMVC), 2017.

[16] E. Mohedano, K. McGuinness, X. Giro-I-Nieto, N. E. O'Connor, Saliency weighted convolutional features for instance search, in: Proceedings - International Workshop on Content-Based Multimedia Indexing, Vol. 2018-Septe, 2018. doi:10.1109/CBMI.2018.8516500.

[17] O. Simeoni, A. Iscen, G. Tolias, Y. Avrithis, O. Chum, Unsupervised object discovery for instance recognition, Proceedings - 2018 IEEE Winter Conference on Applications of Computer Vision, WACV 2018 2018-January (2018) 1745–1754. doi:10.1109/WACV.2018.00194.

[18] Y. Yang, S. Newsam, Bag-of-visual-words and spatial extensions for land-use classification, Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems - GIS '10 (2010) 270doi:10.1145/1869790.1869829.

[19] S. Elkerdawy, N. Ray, H. Zhang, Fine-grained vehicle classification with unsupervised parts co-occurrence learning, Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 11132 LNCS (2019) 664–670. doi:10.1007/978-3-030-11018-5_54.

[20] J. Philbin, O. Chum, M. Isard, J. Sivic, A. Zisserman, Object retrieval with large vocabularies and fast spatial matching, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2007.

[21] J. Philbin, O. Chum, M. Isard, J. Sivic, A. Zisserman, Lost in quantization: Improving particular object retrieval in large scale image databases, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2008.

[22] J. B. Tenenbaum, W. T. Freeman, Separating style and content with bilinear models, Neural Computation 12 (6) (2000) 1247–1283. doi:10.1162/089976600300015349.

[23] J. Carreira, R. Caseiro, J. Batista, C. Sminchisescu, Semantic Segmentation with Second-Order Pooling, in: A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, C. Schmid (Eds.), Computer Vision – ECCV 2012, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012, pp. 430–443.

[24] C. Guillot-Soulez, S. Soulez, L'analyse conjointe : présentation de la méthode et potentiel d'application pour la recherche en GRH, Revue de gestion des ressources humaines 80 (2) (2014) 33. doi:10.3917/grhu.080.0033.

[25] A. R. Chowdhury, T. Lin, S. Maji, E. Learned-Miller, One-to-many face recognition with bilinear cnns, in: 2016 IEEE Winter Conference on Applications of Computer Vision (WACV), 2016, pp. 1–9. doi:10.1109/WACV.2016.7477593.

[26] Y. Gao, O. Beijbom, N. Zhang, T. Darrell, Compact bilinear pooling, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 317–326. doi:10.1109/CVPR.2016.41.

[27] F. Radenović, A. Iscen, G. Tolias, Y. Avrithis, O. Chum, Revisiting oxford and paris: Large-scale image retrieval benchmarking, in: Proceedings of

the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 5706–5715.

[28] H. Jegou, M. Douze, C. Schmid, Hamming embedding and weak geometric consistency for large scale image search, in: Proceedings of the 10th European Conference on Computer Vision: Part I, ECCV '08, Springer-Verlag, Berlin, Heidelberg, 2008, pp. 304–317. doi:10.1007/978-3-540-88682-224.

[29] R. Arandjelovi, A. Zisserman, Three things everyone should know to improve object retrieval, in: 2012 IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 2911–2918. doi:10.1109/CVPR.2012.6248018.

[30] G. Tolias, H. Jgou, Visual query expansion with or without geometry: Refining local descriptors by feature aggregation, Pattern Recognition 47 (10) (2014) 3466 – 3476. doi:https://doi.org/10.1016/j.patcog.2014.04.007.

[31] F. Radenovic, G. Tolias, O. Chum, Fine-tuning CNN Image Retrieval with No Human Annotation, IEEE Transactions on Pattern Analysis and Machine Intelligence (2018) 1–14doi:10.1109/TPAMI.2018.2846566.

[32] E. Mohedano, K. McGuinness, X. Giro-i Nieto, N. E. O'Connor, Saliency Weighted Convolutional Features for Instance Search.

[33] S. Chopra, R. Hadsell, Y. LeCun, Learning a similarity metric discriminatively, with application to face verification, in: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), Vol. 1, 2005, pp. 539–546 vol. 1. doi:10.1109/CVPR.2005.202.

[34] G. R. Koch, Siamese neural networks for one-shot image recognition, 2015.

[35] L. Bertinetto, J. Valmadre, J. F. Henriques, A. Vedaldi, P. H. S. Torr, Fully-convolutional siamese networks for object tracking, in: G. Hua, H. Jégou

(Eds.), Computer Vision – ECCV 2016 Workshops, Springer International Publishing, Cham, 2016, pp. 850–865.

[36] D. Kingma, J. Ba, Adam: A method for stochastic optimization, International Conference on Learning Representations.

[37] F. Radenović, G. Tolias, O. Chum, Cnn image retrieval learns from bow: Unsupervised fine-tuning with hard examples, in: B. Leibe, J. Matas, N. Sebe, M. Welling (Eds.), Computer Vision – ECCV 2016, Springer International Publishing, Cham, 2016, pp. 3–20.