Noname manuscript No.
(will be inserted by the editor)

Exploiting tf-idf in deep Convolutional Neural Networks for Content Based Image Retrieval

Nikolaos Kondylidis · Maria Tzelepi · Anastasios Tefas

Received: date / Accepted: date

Abstract In this paper, a novel term frequency-inverse document frequency (tf-idf) based method that utilizes deep Convolutional Neural Networks (CNN) for Content Based Image Retrieval (CBIR) is proposed. That is, we treat the learned filters of the convolutional layers of a CNN model as detectors of visual words. Each of these filters has been trained to be activated in different visual patterns. Thus, since the activations of each filter provide information about the degree of presence of the visual pattern that the filter has learned during the training procedure, we consider the activations of these filters as the tf part. Subsequently, we propose three approaches of computing the idf part. Finally, we propose a query expansion technique on top of the formulated descriptors. The proposed approach interconnects the standard tf-idf method with the modern CNN analysis for visual content, providing a very powerful image retrieval technique with improved results as it is highlighted by extensive experiments in four challenging image datasets.

Nikolaos Kondylidis
Department of Informatics
Aristotle University of Thessaloniki
Thessaloniki, Greece
E-mail: kondilidisnikos@gmail.com

Maria Tzelepi
Department of Informatics
Aristotle University of Thessaloniki
Thessaloniki, Greece
E-mail: mtzelepi@csd.auth.gr

Anastasios Tefas
Department of Informatics
Aristotle University of Thessaloniki
Thessaloniki, Greece
E-mail: tefas@aiia.csd.auth.gr

1 Introduction

Content Based Image Retrieval (CBIR) refers to the task of retrieving relevant images to a query from a large image collection based on their visual content [1,2]. The query can be an example image, an image region, multiple example images, a visual sketch, or a multimodal query [3]. Query by example image is the most common paradigm of CBIR, also known as Query by Example. Given the feature representations of the images to be searched and the query image, the output of the CBIR procedure includes a search in the feature space, in order to retrieve a ranked set of images in terms of similarity (*e.g.* cosine similarity) to the query representation. Query by image region, also known as Region Based Image Retrieval (RBIR) is a promising variation of CBIR, which has received much attention over the past years [4]. Instead of searching by global features that describe the entire image, RBIR proposes the extraction of features from a specific region of interest to perform the query. A key issue concerning CBIR is to extract meaningful information from raw data in order to bridge the so-called semantic-gap [5]. The semantic-gap refers to the difference between the low level representations of images (i.e. pixels) and their higher level concepts (e.g. persons, objects, actions, etc.). Among the most effective approaches in this direction are those that use the Fisher Vector descriptors [6], Vector of Locally Aggregated Descriptors (VLAD) representations [7,8] or combine bag-of-words models [9] with local descriptors such as Scale-Invariant Feature Transform (SIFT) [10].

Deep Convolutional Neural Networks (CNN), [11,12], are the most efficient Deep Learning architectures [13] for image analysis and recognition. A common CNN architecture comprises of a number of convolutional and subsampling (pooling) layers with nonlinear neural activations, usually followed by fully connected layers. That is, the input image is introduced to the neural network as a three dimensional tensor with dimensions (i.e., width and height) equal to the dimensions of the image and depth equal to the number of color channels (usually three in RGB images). Three dimensional filters are learned and applied in each layer where convolution is performed and the output is passed to the neurons of the next layer for nonlinear transformation using appropriate activation functions. Usually, after multiple convolution and subsampling layers the structure of the deep architecture changes to fully connected layers and single dimensional signals. These activations of the fully connected layers, or the convolutional ones followed by a pooling method, are commonly used as deep descriptors for classification, clustering or retrieval. Note that different CNN architectures can be deployed using combinations of the three main building blocks (convolutional layers, subsampling, and fully connected layers).

Over the last few years, deep CNN have been established as one of the most promising research directions in the computer vision area due to their outstanding performance in a series of vision recognition tasks, such as image classification [14], face recognition [15], digit recognition [16], pose estimation [17], object and pedestrian detection [18,19]. It has also been demonstrated that features extracted from the activations of a CNN trained in a fully supervised

fashion on a large, fixed set object recognition task can be re-purposed to novel generic recognition tasks, [20].

Motivated by these results, deep CNN have been introduced in the CBIR research field. The primary approach of applying deep CNN in the retrieval domain is to obtain the feature representations from a pretrained model by feeding images in the input layer of the model and taking activation values usually drawn from the last layers, while several recent works are directed at utilizing the convolutional layers for the feature extraction.

Tf-idf is a widely used term-weighting technique in document-retrieval [21]. That is, the tf part measures the number of occurrences of a word in a document, while the idf part measures how important a word is for the retrieval task. The weighting is the product of the two terms. The tf-idf weights reflect the importance of the words of a document in a corpus. Thus, a document is represented by a n-dimensional vector of the weighted word frequencies using a vocabulary of n words. Over the past years, tf-idf has also been proven to be efficient in image and particular object retrieval, [22,23].

In this paper we propose a novel tf-idf based method utilizing the deep CNN for CBIR. We treat the learned filters of the convolutional layers of a pretrained CNN model as the detectors of the visual words. Each of these filters has been trained to be activated in different visual patterns. Thus, since the activations of each filter provide information about the degree of presence of the visual pattern that the filter has learned during the training procedure, we consider the activations of these filters as the tf part. We refer to the visual patterns as *convolutional words*, and to the filters as *convolutional word detectors*. Subsequently, as second step towards the tf-idf scheme, we propose three approaches of computing the idf part.

The main contributions of this work can be summarized as follows:

- We introduce the tf-idf weighting scheme into deep CNN for CBIR.
- We propose a novel image desctription method using as description vector the weighted convolutional word frequencies.
- We propose a query expansion technique on top of the formulated descriptors

The remainder of the manuscript is structured as follows: The prior CNN-based works in image retrieval are discussed in Section 2. The proposed method is presented in Section 3. Section 4 presents the utilized CNN models. The proposed visualization method is presented in Section 5. The proposed query expansion approach using pseudo relevance feedback is provided in Section 6. The experimental results are provided in Section 7. Finally conclusions are drawn in Section 8.

2 Prior Work

In this Section we survey previous CNN-based research works in image retrieval domain. First, several works have focused on the aggregation of the

convolutional features. For example, in [27], a feature aggregation pipeline is presented using sum pooling, while in [26], CNN activations at multiple scale levels are combined with the VLAD representation. An approach that produces compact feature vectors derived from the convolutional layer activations that encode several image regions is proposed in [28]. In [30] a pipeline that uses the convolutional CNN-features and the bag-of-Words aggregation scheme is proposed, while in [31] a multi-scale scheme for extracting local features that take geometric invariance into account for the task of visual instance retrieval, is proposed.

Subsequently, several works proposed model retraining methods for improving the retrieval performance. Specifically, in [24] an image retrieval method, where a CNN pretrained model is retrained on a different dataset with relevant image statistics and classes to the dataset considered at the test time and achieves improved performance, is proposed. A deep CNN is retrained with similarity learning objective function, considering triplets of relevant and irrelevant instances obtained from the fully connected layers of the pretrained model, in [25]. In [29], a three-stream Siamense network is proposed to optimize the weights of the so-called R-MAC representation, proposed in [28], for the retrieval task, using a triplet ranking loss. The public Landmarks dataset, that is also used in [24], is utilized for the model training. In [32] a model retraining method which exploits supervised learning, using the fully connected layers is proposed, while in [33] a novel method capable of learning improved image representation towards image retrieval, based on the available information proposes supervised, fully unsupervised, and relevance feedback based model retraining.

From a different viewpoint, in [34] the authors propose to exploit complementary strengths of CNN features of different layers outperforming the concatenation of multiple layers, while in [35] the authors focus on diffusion, proposing a regional diffusion mechanism, which handles one or more query vectors at the same cost.

In [36] a new distance metric learning algorithm, namely weakly-supervised deep metric learning, is proposed, for social image retrieval by exploiting knowledge from community contributed images associated with user-provided tags. The learned metric can well preserve the semantic structure in the textual space and the visual structure in the original visual space simultaneously, which can enable to learn a semantic-aware distance metric. Finally, in [37], a Robust Structured Subspace Learning algorithm which integrates image understanding and feature learning into a joint learning framework is proposed. The learned subspace is adopted as an intermediate space to reduce the semantic gap between the low-level visual features and the high-level semantics. The method is evaluated in different image understanding tasks, including image search.

Consequently, the proposed method innovatively introduce the well established tf-idf scheme of text-domain into image retrieval, proposing as feature representation the description vector of the weighted convolutional word frequencies against the direct extraction of the feature representations. That is, the proposed work can be mainly related with previous description based works such as [34],

however we should emphasize that the proposed method can also be combined with various CNN-based works, as the aforementioned ones.

3 Proposed Method

3.1 The tf-idf weighting scheme

In this paper we propose a tf-idf based weighting technique utilizing the deep CNN for CBIR. Firstly, the tf-idf weighting scheme for document retrieval is described as follows: Considering a vocabulary of n terms, the tf-idf is a n-dimensional vector that is calculated as the element-wise product of the tf and idf vectors. That is, a document is represented by a n-dimensional vector, $\mathbf{x}_d$, where each of its n dimensions is given by:

$$[\mathbf{x}_d]_t = F_{tf}(t, d) \times F_{idf}(d, t, D), \tag{1}$$

where d is the index of a document, t is the index of a certain term of the vocabulary of n terms, D is the corpus, $F_{tf}(t, d)$ is the function which computes the frequency of term t in the document d, and $F_{idf}(d, t, D)$ is the function which computes the inverse document frequency.

The term frequency (tf) part produces a n-dimensional vector where each dimension is given by the frequency of occurrence of each term t in the document d, that is how many times the term t appears in the document d. The inverse document frequency (idf) part provides information about the importance of each term, in the sense that terms that appear in many different documents are less informative, and hence of less importance, as compared to those that appear rarely. Thus, the idf part produces a n-dimensional vector, $\mathbf{r}$, where each dimension is given by:

$$[\mathbf{r}]_t = \log \frac{\mid D \mid}{\mid d \in D : F_{tf}(t, d) \neq 0 \mid}, \tag{2}$$

where $\mid \cdot \mid$ is the cardinality of a set.

3.2 Preliminary Concepts

A common deep CNN comprises of a number of convolutional and subsampling (pooling) layers with non-linear neuron activations, usually followed by fully connected layers. The convolutional layer parameters are the parameters of the corresponding filters or kernels. These filters have certain width and height and extend through the full depth of the input volume. For example, a filter of the first convolutional layer of the utilized CNN model has size $3 \times 3 \times 3$ i.e. width and height 3, and depth 3, since the input images have depth 3, which is the number of the color channels.

Every neuron in a convolutional layer is connected to a local region in the input volume, (as opposed to the regular neural networks where the neuron

is connected to all the neurons in the previous layer) that is the so-called receptive field of the neuron. We denote as depth slice each 2-dimensional slice of depth, in each convolutional layer, for example the first convolutional volume of the utilized model with size $224 \times 224 \times 64$ has 64 depth slices, each of size 224×224. The neuron, constitutes the fundamental computational unit of the neural networks, which receives an input and performs a dot product with its weights. All the neurons in each depth slice share the same weights (e.g. in the first convolutional layer of the used model there are 64 unique sets of weights or filters), and hence we can also refer to the neurons of a single depth slice as a convolutional neuron, which corresponds to a unique filter.
Each convolutional layer computes the output of the neurons, which is a dot product between its weights (filter) and its receptive field. In a forward pass, the aforementioned output for a single filter is a 2-dimensional activation map, consisting of the activations of the filter at every spatial position when we slide the filter with stride equal to one. Note that bigger stride produces smaller activation map spatially. The activations of all the filters of a convolutional layer produce a volume of k 2-dimensional activation maps, where k is the number of the filters.

3.3 Tf-idf in deep CNNs

The filters of the convolutional layers of a pretrained CNN model have been trained to recognize specific visual features in the input image, and hence in our approach we consider these features as the *convolutional words*, and the learned filters as the *convolutional word detectors*. For an input image each filter is activated when it sees a certain visual feature, e.g. edges, blobs, shapes, etc., in the first layers or a visual concept, e.g, face, car, etc., in the last layers, and thus the activations of the filters reveal the degree of presence of the convolutional word that learned during the training procedure. To translate this to the tf-idf technique, the activations of each filter correspond to the tf part. Since the activations of each filter output a 2-dimensional activation map of the responses of the filter at every position of the input image, we take the maximum activation value [28], in order to assign a unique activation value to every filter, considering that the degree of presence of the specific visual pattern is equal to the maximum degree of presence. This leads to loss of the spatial information that capture the convolutional layers. Thus, the vocabulary size is equal to the number of the learned filters (or the convolutional neurons) of the utilized convolutional layer or the number of the neurons if we obtain the responses of the fully connected layers. Additionally, in [38] it has been shown that the CNN features have a hierarchical nature. Thus, drawing analogy to the document domain, based on the utilized layer each descriptor provides a description of frequencies of words, on the first layers (i.e., simple visual patterns such as corners, lines, etc.), pairs of words on the following layers (i.e., simple objects like faces, cars, etc.), and so on, leading to more complex descriptions on the fully connected layers (e.g., landscapes, buildings).

Since the range of the activation value of a neuron varies through the network layers, we apply a normalization step in order to ensure that the activation value of each neuron of the network lies in the range of $[0,1]$. That is, for each image of the dataset, we divide each activation value of the neurons of a certain layer by the maximum activation value of the neurons of this layer over the entire dataset. In the following, we refer to the resulting normalized activation value as normalized activation of a neuron.

3.4 Idf Computation

An issue arising on materializing the idf part is that the utilized normalized activations take values between 0 and 1, while in the standard tf-idf technique in text domain a word either exists or not. Thus, we need to investigate when a neuron is considered as activated or not. To this aim, we suggest three approaches. The first approach defines a threshold T with value between 0 and 1. If the activations are over this threshold value, the neurons are considered as activated. That is:

$$x_n^i = \begin{cases} 1, & \text{if } a_n^i > T \\ 0, & otherwise \end{cases} \tag{3}$$

where a_n^i is the normalized activation of the neuron n, for an image i.

The second approach defines a percentage threshold value for the entire network activation and in each layer we consider as activated the neurons with the maximum activations in a cumulative way until we reach the predefined activation percentage threshold. That is,

$$x_n^i = \begin{cases} 1, & \text{if } a_n^i \text{ in top K\% of activations} \\ 0, & otherwise \end{cases} \tag{4}$$

where a_n^i is the normalized activation of the neuron n, for an image i. For instance, if we define the percentage value to 10% and the utilized layer consists of 100 neurons, we consider as activated the ten neurons with the highest activation values.

3.4.1 Statistical idf

Finally, the third approach estimates the importance weight of each neuron based on the standard deviation of the normalized activation values of the specific neuron to the entire dataset. More specifically, the weight w_n of each neuron n is computed as follows:

$$w_n = \sqrt{\sum_i (a_n^i - \mu_n)^2}, \tag{5}$$

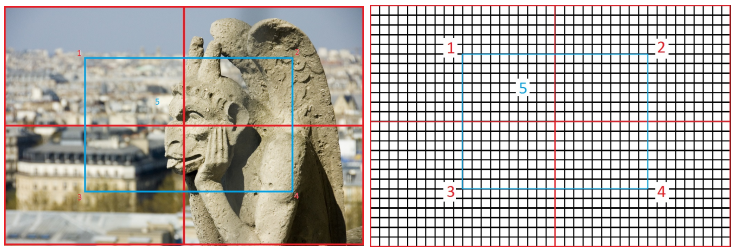
where $\mu_n = \sum_i a_n^i$ is the mean value of the activations of the neuron n in the entire dataset I, and a_n^i is the normalized activation of the neuron n, for an image $i \in I$. Thus, the weighted description is given by the following equation:

$$x_n^i = w_n \times a_n^i \tag{6}$$

The proposed weighting scheme tries to use the statistical behaviour of the neurons in order to estimate its importance. That is, a neuron that is consistenly activated with similar values throughout the entire dataset, and thus it has a very small standard deviation in the activation value, has limited importance in the retrieval task. The standard deviation of the activation resamples the idf value of the neuron. On the contrary, a neuron that has a large standard deviation is more informative since it has different behaviour across the dataset and thus it can be used for discriminating the images. As previously, the tf term is estimated by the normalized activation of the specific neuron n to the specific image i.

3.5 Pyramid-based Approach

In order to partially preserve spatial information from the activations of the convolutional layers we propose the following approach. We divide the activation map into S sections. Then we treat each of the resulting sections as a separate image. That is, we apply the proposed tf-idf approach to each of them. Thus, instead of considering the maximum activation value over the whole image, we consider the maximum activation value over each section. This value informs us about the presence of the convolutional words on the corresponding section of the input image.
Regarding the idf part, which provides information about the importance of the certain convolutional word, we maintain the same weights calculated for the whole image. That is, we extract more than one value for a convolutional neuron for an input image in order to partially preserve the position of the detected convolutional word. Hence, we produce S times bigger description for the whole image. This allows us to decide not only whether two images contain the same convolutional words, but also if these words are approximately in the same position on the image. It is clear that the more sections lead to more detailed detections on the image, however this also renders the comparison between two images more difficult.
The above procedure is illustrated in Fig. 1 for five sections. More specifically, the initial image is divided into five sections (four non overlapping sections whose width and height are equal to the half-width and half-height of the full image, respectively, while the fifth section positioned in the center of the image is of equal dimensions, and has 25% overlap with each of the other four sections), and correspondingly the activation map for a certain neuron is divided into the same sections. Then, the maximum activation values for each of the five sections is derived, and the tf-idf scheme is applied. The final representation is derived by concatenating the five representations.



(a) Five sections in an input image (b) Five sections in the activation map

Fig. 1: Division of the activation maps into sections

4 CNN Model

In this work, we first utilize a commonly used CNN model, that is the VGG-16, to apply the proposed tf-idf scheme, and subsequently we utilize our recently published fully convolutional model, optimized towards image retrieval in a fully unsupervised fashion. The two models are described below.

4.1 VGG-16

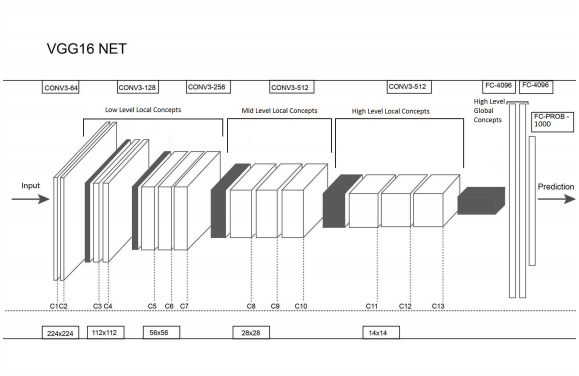


Fig. 2: Overview of the VGG-16 Architecture.

We first utilize the VGG 16-layer model [39], trained on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2014 to classify 1,000 ImageNet classes, since it is a commonly used baseline model. The model consists of sixteen trained neural layers; the first thirteen are convolutional and the remaining three are fully connected. Max-pooling layers follow the second, forth, seventh, tenth, and thirteenth convolutional layers, while the ReLU non-linearity

($f(x) = max(0, x)$) is applied to every convolutional and fully connected layer, except the last fully connected layer. The output of the last fully connected layer is a distribution over 1000 ImageNet classes. The softmax loss is used during the training. We use the VGG 16 layer model to directly extract the representations from certain layers in order to apply the proposed tf-idf approach.
It has been shown in [38] that the CNN features have a hierarchical nature. That is, convolutional neurons at the first levels are meant to detect low level local concepts like edges and corners. The next levels are able to detect more complicated patterns like basic shapes, which are based on the detections of previous convolutional levels, which can be considered as combination of convolutional words; the mid level local concepts. The last convolutional layers are able to detect even more complicated patterns, close to ones that humans detect, like hands, faces or even cars; the high level local concepts. Finally, the activations of neurons of the fully connected layers are based on the detection of combinations of patterns and the reason they activate cannot be pointed directly on some pattern on the input image. Neurons of fully connected layers produce only one output for the whole image, and they are able to detect high level global concepts. In Fig.2, we illustrate an overview of the VGG-16 model, providing information about the number and the size of the filters of each layer, which also depicts the aforementioned observations on the level of detail of the detected patterns.

4.2 Fully Unsupervised Model

The Fully Unsupervised (FU) [33] model, is a recent state-of-the-art fully convolutional model, which is retrained on the modified CaffeNet model[1], in order to produce more efficient and compact image representations for the retrieval task. More specifically, in the Fully Unsupervised (FU) approach, the goal is to amplify the primary retrieval presumption that the relevant image representations are closer to the certain query representation in the feature space. The rationale behind this approach is rooted to the cluster hypothesis which states that documents in the same cluster are likely to satisfy the same information need [40]. That is, the pretrained CNN model is retrained on the given dataset, aiming at maximizing the cosine similarity between each image representation and its n nearest representations, in terms of cosine distance.

The set of N images to be searched is denoted by $\mathcal{I} = \{\mathbf{I}_i, i = 1, \ldots, N\}$, their corresponding feature representations emerged in the L layer by $\mathcal{X} = \{\mathbf{x}_i, i = 1, \ldots, N\}$, and by $\boldsymbol{\mu}^i$ the mean vector of the $n \in \{1, \ldots, N-1\}$ nearest representations to $\mathbf{x}_i$, denoted as $\mathcal{X}^i = \{\mathbf{x}_l^i, l = 1, \ldots, N-1\}$. That is,

$$\boldsymbol{\mu}^i = \frac{1}{n} \sum_{l=1}^{n} \mathbf{x}_l^i \tag{7}$$

The optimization problem to be solved is:

[1] https://github.com/BVLC/caffe/tree/master/models/bvlc_reference_caffenet

$$\max_{\mathbf{x}_i \in \mathcal{X}} \mathcal{J} = \max_{\mathbf{x}_i \in \mathcal{X}} \sum_{i=1}^{N} \frac{\mathbf{x}_i^{\mathsf{T}} \boldsymbol{\mu}^i}{||\mathbf{x}_i||\ ||\boldsymbol{\mu}^i||} \tag{8}$$

The above optimization problem is solved using gradient descent. The first-order gradient of the objective function $\mathcal{J}$ is given by:

$$\frac{\partial \mathcal{J}}{\partial \mathbf{x}_i} = \frac{\partial}{\partial \mathbf{x}_i} \left(\sum_{i=1}^{N} \frac{\mathbf{x}_i^{\mathsf{T}} \boldsymbol{\mu}^i}{||\mathbf{x}_i||\ ||\boldsymbol{\mu}^i||} \right) = \frac{\boldsymbol{\mu}^i}{||\mathbf{x}_i||\ ||\boldsymbol{\mu}^i||} - \frac{\mathbf{x}_i^{\mathsf{T}} \boldsymbol{\mu}^i}{||\mathbf{x}_i||^3\ ||\boldsymbol{\mu}^i||} \mathbf{x}_i \tag{9}$$

The update rule for the v-th iteration for each image can be formulated as:

$$\mathbf{x}_i^{(v+1)} = \mathbf{x}_i^{(v)} + \eta \left(\frac{\boldsymbol{\mu}^i}{||\mathbf{x}_i^{(v)}||\ ||\boldsymbol{\mu}^i||} - \frac{\mathbf{x}_i^{(v)\mathsf{T}} \boldsymbol{\mu}^i}{||\mathbf{x}_i^{(v)}||^3\ ||\boldsymbol{\mu}^i||} \mathbf{x}_i^{(v)} \right), \quad \mathbf{x}_i \in \mathcal{X} \tag{10}$$

A normalization step has been introduced as:

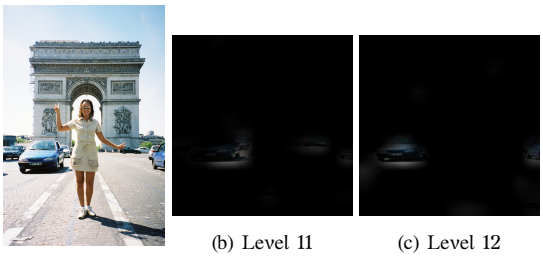
$$\mathbf{x}_i^{(v+1)} = \mathbf{x}_i^{(v)} + \eta ||\mathbf{x}_i^{(v)}||\ ||\boldsymbol{\mu}^i|| \left(\frac{\boldsymbol{\mu}^i}{||\mathbf{x}_i^{(v)}||\ ||\boldsymbol{\mu}^i||} - \frac{\mathbf{x}_i^{(v)\mathsf{T}} \boldsymbol{\mu}^i}{||\mathbf{x}_i^{(v)}||^3\ ||\boldsymbol{\mu}^i||} \mathbf{x}_i^{(v)} \right), \quad \mathbf{x}_i \in \mathcal{X} \tag{11}$$

Hence, using the above representations as targets in the layer of interest, a regression task is formulated for the neural network, which is initialized on the CaffeNet's weights and is trained on the utilized dataset, using back-propagation. The Euclidean loss is used during training for the regression task. Thus, the procedure is integrated by feeding the entire dataset into the input layer of the retrained adapted model and obtaining the new representations.

5 Visualization

In this work we aim to study the usefulness of the hidden convolutional layers of a CNN for the image retrieval task. More specifically, we study the usefulness of the patterns that are detected by the aforementioned layers. Towards this aim, we also propose a novel visualization method, which reveals in which parts of the input image, a convolutional filter was activated, and thus, what patterns has been trained to recognize. The technique of filter visualization is described below. Every convolutional neuron takes as input many regions of the input image and outputs an activation value for each of them, forming the activation map. The values of the activation maps are normalized as mentioned above in order to belong to the interval $[0, 1]$. We multiply the RGB values of all pixels of one region with the produced activation value. This produces a new image where every region that did not activate the neuron is shaded. We use bicubic interpolation to resize the activation map so that the visual result appears more uniform.

Since the presentation of the visualization results through all the network layers, for various neurons is impractical, in Figs. 3-9, we will present specific indicative image examples illustrating the image regions that activate specific neurons of

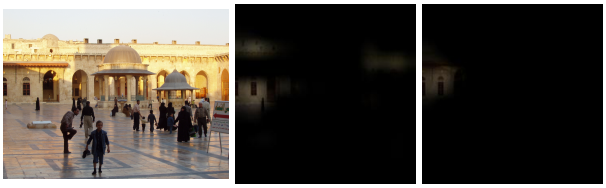


(a) Input Image (b) Level 11 (c) Level 12

Fig. 3: Example of tracing cars

(a) Level 9 (b) Level 10 (c) Level 11 (d) Level 13

Fig. 4: Example of tracing human parts for the input image of Fig. 3



(a) Input Image (b) Level 12 (c) Level 13

Fig. 5: Example of tracing windows

specific layers. It is evident that the convolutional filter responses can be, indeed, considered as visual words since they are usually activated in semantically similar regions like humans, human parts, cars, domes, buildings, etc. Thus, the proposed method claim that we can consider images as documents of visual words and work with tf-idf strategies for retrieval is also supported by the visualization of the filter activations as given in Figs.3-9. The VGG-16 model is utilized in the visualization experiments.

6 Query Expansion using Pseudo Relevance Feedback

Query Expansion is a standard, in most cases of negligible cost, technique for accomplishing better retrieval results [41]. Concisely, the idea is to re-formulate

(a) Level 9 (b) Level 10 (c) Level 11 (d) Level 12 (e) Level 13

Fig. 6: Example of tracing human parts for the input image of Fig. 5

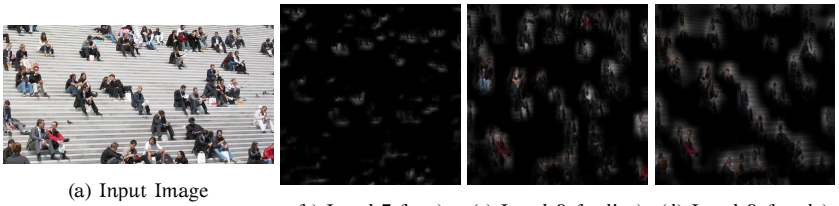
(a) Level 8 (b) Level 9 (c) Level 10 (d) Level 11

(e) Level 12 (f) Level 13

Fig. 7: Example of tracing arch like patterns for the input image of Fig. 5

(a) Level 9 (b) Level 10 (c) Level 11 (d) Level 12 (e) Level 13

Fig. 8: Example of tracing domes for the input image of Fig. 5



(a) Input Image (b) Level 7 (legs) (c) Level 9 (bodies) (d) Level 9 (heads)

Fig. 9: Example of tracing Human parts

the initial query, utilizing information deriving from the evaluation of the initial query. The majority of CBIR methods include a query expansion step that boosts the retrieval performance. On the top of the proposed descriptors we also introduce a simple query expansion method using Pseudo Relevance Feedback. That is, we propose to re-issue the the top n ranked results from the initial query as a new query, in order to enhance the original query representation with additional relevant representations, following either the average query expansion scheme or the max one.

The average scheme can be described as follows:

Let $\mathbf{Q}$ be a certain query image, and $\mathbf{q}$ the corresponding CNN representation using the proposed method. We consider the top k retrieved images and their corresponding CNN representations $\mathbf{x}_j$, $j = 1, \ldots, k$. Then, the new query representation $\mathbf{q}_{new}$ is as follows:

$$\mathbf{q}_{new} = \frac{1}{k+1}(\mathbf{q} + \sum_{j=1}^{k} \mathbf{x_j}). \tag{12}$$

The max scheme considers as the new query representation the max values of the top k retrieved images in each dimension.

Furthermore, in the case of region based image retrieval, where we perform queries with a specified region of interest, we suggest a spatial verification step as follows: We consider a shortlist of N top initially retrieved images for each query. Each of these images is cropped into l overlapping regions. Subsequently, we extract the CNN features and apply the proposed tf-idf weighting on the dataset consisting of the cropped images and the initial full images, and we perform the query again. Then, we rerank the shortlist of the initially retrieved images based on the similarity of the images of the formed dataset to the query, and we expand the initial query representation as described above with respect to the reranked list.

7 Experimental Results

7.1 Evaluation Metrics

Throughout this work we use mean Average Precision (mAP), and top-N score in order to evaluate the performance of the proposed method. The definitions of the above metrics follow below:

Mean average precision is the mean value of the Average Precision (AP) of all the queries. The definition of AP for the i-th query is formulated as follows:

$$AP_i = \frac{1}{Q_i} \sum_{n=1}^{N} \frac{R_i^n}{n} t_n^i, \tag{13}$$

where Q_i is the total number of relevant images for the i-th query, N is the total number of images of the search set, R_i^n is the number of relevant retrieved

images within the n top results; t_n^i is an indicator function with $t_n^i = 1$ if the n-th retrieved image is relevant to the i-the query, and $t_n^i = 0$ otherwise.

Top-N score refers to the average number of same-object images, within the top-N ranked images.

7.2 Datasets

Inria Holidays [42]: consists of 991 images divided into 500 classes, and 500 discrete queries. Each class in the search set consists of between 1 and 12 images. Some images of the dataset are not in a natural orientation. We note that we have not proceeded to any preprocessing step of these images, as in other CNN-based works, e.g. [24,27] . We measure the retrieval performance in terms of mAP.

Oxford 5k [43]: consists of 5062 images collected from Flickr by searching for particular Oxford landmarks. The collection has been manually annotated to generate a comprehensive ground truth for 11 different landmarks, each represented by 5 possible queries. Images are assigned one of four possible queries: Good, Ok, Junk and Absent. Good and ok images are considered as positive examples, absent as negative examples while junk images as null examples. Following the standard evaluation protocol we measure the retrieval performance in mAP. We use both the full queries and cropped queries versions of the dataset.

Paris 6k [44]: similar to Oxford Buildings dataset, consists of 6392 images (20 of the 6412 provided images are corrupted) collected from Flickr by searching for particular Paris landmarks and provides 55 queries. The retrieval performance is also measured in terms of mAP, using both the full queries and cropped queries versions of the dataset.

UKBench [45]: contains 10200 images of objects divided into 2550 classes. Each class consists of 4 images. All 10200 images are used as queries. The performance is reported as Top-4 Score, which is a number between 0 and 4.

7.3 Experimental Setup

In our experiments, we first apply the proposed tf-idf approach in the VGG model, on certain layers and we also experiment with combinations of different layers, in order to show that the proposed scheme can improve the baseline results of directly extracted feature representations using a commonly used CNN pretrained model. More specifically, we perform experiments with all the convolutional, with all the fully connected layers (except for the last fully connected which is a distribution over the ImageNet 1000 classes, and is usually used in hashing techniques), and with the last convolutional and the fully connected independently. Experiments also conducted with the combination of the last convolutional layer and the fully connected ones, as well as with the combination of all the convolutional and all the fully connected. Additionally, in order to partially preserve spatial information from the convolutional layers, in the

Feature Representation	Oxford 5k Cropped	Oxford 5k Full	Paris 6k Cropped	Paris 6k Full
VGG Baseline	0.411	0.475	0.358	0.676
TF-IDF(VGG)	0.473	0.543	0.699	0.705
TF-IDF(VGG) & QE	0.52	0.563	0.757	0.746

Table 1: mAP of our method in the Oxford 5k and Paris 6k datasets (VGG-16)

Feature Representation	Inria Holidays	UKBench
VGG Baseline	0.772	3.184
TF-IDF(VGG)	0.8	3.668
TF-IDF(VGG) & QE	-	3.779

Table 2: mAP and Top-4 Score of our method in the Inria Holidays and UK-Bench datasets (VGG-16)

Feature Representation	Oxford 5k	Oxford 105k	Paris 6k	Paris 106k	UKBench
FU Baseline	0.5509	0.5302	0.8107	0.7468	3.8094
TF-IDF(FU)	0.5742	0.5476	0.8292	0.7481	3.8421

Table 3: mAP and Top-4 Score of our method (FU retrained model)

VGG case, we apply the aforementioned method of division into five sections in the last convolutional layer.

Subsequently, we apply the proposed tf-idf based representation scheme on the FU retrained model, in order to show that the proposed method is applicable to different network architectures, and also can be combined with other state-of-the-art CNN-based works, improving their performance even more. In this case, we perform experiments utilizing the last two optimized convolutional layers, where the max pooling operator outputs 384-dimensional, and 256-dimensional feature representations respectively. All results are obtained using the cosine distance. In the following we abbreviate the proposed tf-idf scheme utilizing the VGG model as *TF-IDF(VGG)*, and correspondingly *TF-IDF(FU)* the proposed tf-idf scheme on the FU optimized model. We also abbreviate as *QE* the proposed query expansion method.

7.4 Experimental Results

Idf Approach	UKBench	Oxford 5k	Oxford 105k	Paris 6k	Paris 106k
No idf	3.8094	0.5509	0.5302	0.8107	0.7468
Threshold Value	3.8336	0.5658	**0.5476**	0.8270	0.7461
Percentage Threshold	3.8294	**0.5742**	0.5475	**0.8292**	**0.7481**
Statistical idf	**3.8421**	0.5636	0.5415	0.8287	0.73

Table 4: mAP and Top-4 Score for different idf computation approaches (FU retrained model)

In order to validate the performance of the proposed method we present the experiments conducted in the four datasets. Table 1 and Table 2 illustrate the experimental results utilizing the VGG-16 model, for the Oxford 5k and the Paris 6k datasets, and for the Inria Holidays and the UKBench, respectively. We compare the proposed method against the baseline VGG utilizing the same layers, and extracting directly the feature representations from them, using the common max-pooling operation in the case of the convolutional layers [28]. As we can observe the proposed method can achieve notably improved results against the baseline, in all the used datasets, validating our claim that the tf-idf weighting method can successfully applied in the deep CNNs enhancing the information captured from the CNN layers. Furthermore we can observe that the query expansion gives another boost in the performance. We should also note that the query expansion cannot be applied to the Inria Holidays dataset, since in many cases there is only one relevant to the query sample in the dataset.

Subsequently, in Table 3, we provide the experimental results for the UKBench, Paris 6k, and Oxford 5k datasets, for the tf-idf scheme on the FU retrained model. For the Oxford and Paris datasets, we also provide the evaluation results utilizing the 100k distractors (the two augmented datasets are abbreviated as Oxford 105k and Paris106k, respectively). Note that the published paper provides the retrained models only for the UKBench and Paris 6k datasets, however we also trained a CNN model on the Oxford 5k dataset, utilizing the FU method. We compare the proposed scheme utilizing the last two optimized convolutional layers, against the direct concatenation of their feature representations, using the max-pooling operation, as proposed in the original work, [33]. As we can see, the proposed tf-idf based description technique utilizing the two optimized convolutional layers, is superior over the description derived directly from these layers. Consequently, the proposed method can be successfully combined with other state-of-the-art works yielding improved performance.

We should note that combining FU with QE, as well as with other proposed variants (*e.g.*, pyramid-based approach) is beyond the scope of the current manuscript. Thus, we have indicatively combined only VGG with QE as a proof of concept example in order to show that the proposed tf-idf approach can be easily combined with other methodological improvements (like the QE) that have been proposed as elements of an image retrieval system pipeline.

Regarding the utilizing approach for computing the idf term, different approaches have been proven the best through the datasets. For example, in the Inria Holidays, on the VGG model, the statistical idf was utilized, in the UKBench dataset on the VGG model, the second approach of the percentage threshold value, while in the Paris 6k dataset, utilizing the FU retrained model, the second approach of the percentage threshold value also used. In Table 4 we present the Top-4 Score and mAP for the different idf computation approaches in the UKBench, Oxford and Paris datasets, utilizing the FU retrained model. It can be observed that the proposed tf-idf schemes outperform the baseline without tf-idf in all the utilized datasets.

Table 5 provides a comparison of the proposed tf-idf scheme on the FU optimized model, against other CNN-based as well as hand-crafted techniques

for CBIR. Since the proposed method does not utilize supervised learning, we only compare it with unsupervised methods. We also note that we provide the results of the comparisons against other methods regardless the dimension of the utilized features of each one, however, whenever it is possible we compare with the ones with the closest feature dimension.
For fair comparisons we do not include region-based methods, like R-MAC [28] or Regional Diffusion [35], since the proposed compared approach on the FU model, does not exploit spatial information. Furthermore, we should note that we have included a method for partially preserving spatial information for completeness purposes. That is, the proposed representation method is combined with a pyramid based approach, however there are more sophisticated works in this direction (e.g. [28]), and it is out of the principal scope of the proposed work to explore such incremental combinations. Therefore, even the proposed pyramid based approach (that has been indicatively applied only in the VGG case) is not comparable with the aforementioned methods, since it aims at partially preserving spatial information (i.e. the low resolution input image is divided into five sections). On the contrary, the R-MAC representation, for example, using images of high resolution can utilize even 30 regions at different scales for each image, while in the [35] each image has on average 21 regions.
As we can see the proposed image description method which uses the weighted convolutional word frequencies can achieve state-of-the-art performance utilizing the FU retrained model, against other unsupervised approaches.
We finally highlight that the proposed representation can be combined with supervised approaches too, in the same way combined with the FU approach. For example, if we apply the proposed tf-idf method on our supervised retrained model [33], we can achieve top-4 score 3.9740 (against 3.96 of the direct feature extraction) in the UKBench dataset which, to the best of our knowledge, is superior over state-of-the-art supervised methods. Furthermore, in the Paris dataset we can achieve mAP 0.9757 (against 0.9730 of the direct feature extraction), that is also superior over other supervised methods. However, such an investigation is beyond the scope of the manuscript, and constitutes one of the directions of our future work. That is, in this work, we propose a novel image description method, based on the tf-idf scheme, against directly extracting the feature representations of a CNN model. Methods marked with * use only the cropped queries versions.

Method	Dim	Oxford 5k	Oxford 105k	Paris 6k	Paris 106k	UKBench
CVLAD* [46]	64k	0.514	-	-	-	3.62
VLAD* [7]	128	0.448	-	-	-	-
T-embedding* [47]	512	0.528	0.461	-	-	-
Fine-residual VLAD [8]	256	-	-	-	-	3.43
BOW* [48]	200k	0.364	-	0.46	-	2.81
SPoC [27]	256	**0.589**	**0.578**	-	-	3.65
Multi-layer [34]	4k	0.567	-	-	-	3.243
MAC* [28]	256	0.522	-	-	-	-
Small Memory Footprint Regimes [31]	256	0.533	0.489	0.67	-	3.368
TF-IDF(FU)	640	0.5742	0.5476	**0.8292**	**0.7481**	**3.8425**

Table 5: Comparison against other unsupervised methods

8 Conclusions

In this paper we proposed a novel method that introduces the well-established tf-idf weighting scheme of the text domain, into deep CNN for CBIR. That is, we proposed a novel image description method using as description vector the weighted convolutional word frequencies. Thus, we proposed to treat the learned filters of the convolutional layers of a pretrained CNN model as the detectors of the visual words. Each of these filters has been trained to be activated in different visual patterns. Hence, since the activations of each filter provide information about the degree of presence of the visual pattern that the filter has learned during the training procedure, we consider the activations of these filters as the tf part. Subsequently, as second step towards the tf-idf scheme, we proposed three approaches of computing the idf part. Thus, exploiting this rendering we can benefit from tf-idf scheme, and enhance the CNN representations. We also proposed a query expansion technique using pseudo relevance feedback on top of the formulated descriptors. Experimental results on four challenging image retrieval datasets demonstrated the improved performance of the proposed approach. It should also be noted that the proposed approach can be easily combined with more sophisticated approaches that have been recently proposed to give a new perspective on treating convolutional image retrieval. To this aim, we also conducted experiments utilized our fully unsupervised model towards image retrieval enhancing even more the retrieval performance, leading also to state-of-the-art results against other unsupervised approaches. Future work includes further investigation on the idf computation, as well as the combination of the proposed method with other ones which include learning.

Acknowledgment

Maria Tzelepi was supported by the General Secretariat for Research and Technology (GSRT) and the Hellenic Foundation for Research and Innovation (HFRI) (PhD Scholarship No. 2826).

References

1. Ritendra Datta, Jia Li, and James Z Wang. Content-based image retrieval: approaches and trends of the new age. In *Proceedings of the 7th ACM SIGMM international workshop on Multimedia information retrieval*, pages 253–262. ACM, 2005.
2. Toshikazu Kato. Database architecture for content-based image retrieval. In *SPIE/IS&T 1992 symposium on electronic imaging: science and technology*, pages 112–123. International Society for Optics and Photonics, 1992.
3. Liam M Mayron. *Image retrieval using visual attention*. Florida Atlantic University, 2008.
4. Ryota Hinami, Yusuke Matsui, and Shin'ichi Satoh. Region-based image retrieval revisited. *arXiv preprint arXiv:1709.09106*, 2017.
5. Arnold WM Smeulders, Marcel Worring, Simone Santini, Amarnath Gupta, and Ramesh Jain. Content-based image retrieval at the end of the early years. *IEEE Transactions on pattern analysis and machine intelligence*, 22(12):1349–1380, 2000.

6. Florent Perronnin, Yan Liu, Jorge Sánchez, and Hervé 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.
7. Relja Arandjelovic and Andrew Zisserman. All about vlad. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1578–1585, 2013.
8. Ziqiong Liu, Shengjin Wang, and Qi Tian. Fine-residual vlad for image retrieval. *Neurocomputing*, 173:1183–1191, 2016.
9. Gabriella Csurka, Christopher Dance, Lixin Fan, Jutta Willamowski, and Cédric Bray. Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, volume 1, pages 1–2. Prague, 2004.
10. David G Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999.
11. B Boser Le Cun, John S Denker, D Henderson, Richard E Howard, W Hubbard, and Lawrence D Jackel. Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems*. Citeseer, 1990.
12. Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
13. Li Deng. A tutorial survey of architectures, algorithms, and applications for deep learning. *APSIPA Transactions on Signal and Information Processing*, 3:e2, 2014.
14. Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.
15. Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1701–1708, 2014.
16. Dan Ciresan, Ueli Meier, and Jurgen Schmidhuber. Multi-column deep neural networks for image classification. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3642–3649. IEEE, 2012.
17. Alexander Toshev and Christian Szegedy. Deeppose: Human pose estimation via deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1653–1660, 2014.
18. Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.
19. Pierre Sermanet, Koray Kavukcuoglu, Sandhya Chintala, and Yann LeCun. Pedestrian detection with unsupervised multi-stage feature learning. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3626–3633. IEEE, 2013.
20. Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. *arXiv preprint arXiv:1310.1531*, 2013.
21. Ricardo Baeza-Yates, Berthier Ribeiro-Neto, et al. *Modern information retrieval*, volume 463. ACM press New York, 1999.
22. James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2007.
23. Josef Sivic and Andrew Zisserman. Video google: A text retrieval approach to object matching in videos. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 1470–1477. IEEE, 2003.
24. Artem Babenko, Anton Slesarev, Alexandr Chigorin, and Victor Lempitsky. Neural codes for image retrieval. In *Computer Vision–ECCV 2014*, pages 584–599. Springer, 2014.
25. Ji Wan, Dayong Wang, Steven Chu Hong Hoi, Pengcheng Wu, Jianke Zhu, Yongdong Zhang, and Jintao Li. Deep learning for content-based image retrieval: A comprehensive study. In *Proceedings of the ACM International Conference on Multimedia*, pages 157–166. ACM, 2014.
26. Joe Ng, Fan Yang, and Larry 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.
27. Artem Babenko and Victor Lempitsky. Aggregating deep convolutional features for image retrieval. *arXiv preprint arXiv:1510.07493*, 2015.

28. Giorgos Tolias, Ronan Sicre, and Hervé Jégou. Particular object retrieval with integral max-pooling of cnn activations. *arXiv preprint arXiv:1511.05879*, 2015.
29. Albert Gordo, Jon Almazán, Jerome Revaud, and Diane Larlus. Deep image retrieval: Learning global representations for image search. In *European Conference on Computer Vision*, pages 241–257. Springer, 2016.
30. Eva Mohedano, Amaia Salvador, Kevin McGuinness, Ferran Marques, Noel E O'Connor, and Xavier Giro-i Nieto. Bags of local convolutional features for scalable instance search. *arXiv preprint arXiv:1604.04653*, 2016.
31. Ali S Razavian, Josephine Sullivan, Stefan Carlsson, and Atsuto Maki. Visual instance retrieval with deep convolutional networks. *ITE Transactions on Media Technology and Applications*, 4(3):251–258, 2016.
32. Maria Tzelepi and Anastasios Tefas. Exploiting supervised learning for finetuning deep cnns in content based image retrieval. In *Pattern Recognition (ICPR), 2016 23rd International Conference on*, pages 2918–2923. IEEE, 2016.
33. Maria Tzelepi and Anastasios Tefas. Deep convolutional learning for content based image retrieval. *Neurocomputing*, 275:2467–2478, 2018.
34. Wei Yu, Kuiyuan Yang, Hongxun Yao, Xiaoshuai Sun, and Pengfei Xu. Exploiting the complementary strengths of multi-layer cnn features for image retrieval. *Neurocomputing*, 237:235–241, 2017.
35. Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, Teddy Furon, and Ondrej Chum. Efficient diffusion on region manifolds: Recovering small objects with compact cnn representations. *arXiv preprint arXiv:1611.05113*, 2016.
36. Zechao Li and Jinhui Tang. Weakly supervised deep metric learning for community-contributed image retrieval. *IEEE Transactions on Multimedia*, 17(11):1989–1999, 2015.
37. Zechao Li, Jing Liu, Jinhui Tang, and Hanqing Lu. Robust structured subspace learning for data representation. *IEEE transactions on pattern analysis and machine intelligence*, 37(10):2085–2098, 2015.
38. Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*, pages 818–833. Springer, 2014.
39. Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–9, 2015.
40. Ellen M Voorhees. The cluster hypothesis revisited. In *Proceedings of the 8th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 188–196. ACM, 1985.
41. Ondrej Chum, James Philbin, Josef Sivic, Michael Isard, and Andrew Zisserman. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8. IEEE, 2007.
42. Hervé Jégou, Matthijs Douze, and Cordelia Schmid. Hamming embedding and weak geometric consistency for large scale image search. In Andrew Zisserman David Forsyth, Philip Torr, editor, *European Conference on Computer Vision*, volume I of *LNCS*, pages 304–317. Springer, oct 2008.
43. James Philbin, Ondřej Chum, Michael Isard, Josef Sivic, and Andrew 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.
44. James Philbin, Ondřej Chum, Michael Isard, Josef Sivic, and Andrew 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.
45. David Nister and Henrik 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.
46. Wan-Lei Zhao, Hervé Jégou, and Guillaume Gravier. Oriented pooling for dense and non-dense rotation-invariant features. In *BMVC-24th British Machine Vision Conference*, 2013.
47. Hervé Jégou and Andrew 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.

48. Hervé Jégou, Florent Perronnin, Matthijs Douze, Javier Sanchez, Pablo Perez, and Cordelia Schmid. Aggregating local image descriptors into compact codes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(9):1704–1716, 2012.