Deep Convolutional Learning for Content Based Image Retrieval

Maria Tzelepi, Anastasios Tefas

Aristotle University of Thessaloniki, Department of Informatics

Abstract

In this paper we propose a model retraining method for learning more efficient convolutional representations for Content Based Image Retrieval. We employ a deep CNN model to obtain the feature representations from the activations of the convolutional layers using max-pooling, and subsequently we adapt and retrain the network, in order to produce more efficient compact image descriptors, which improve both the retrieval performance and the memory requirements, relying on the available information. Our method suggests three basic model retraining approaches. That is, the Fully Unsupervised Retraining, if no information except from the dataset itself is available, the Retraining with Relevance Information, if the labels of the training dataset are available, and the Relevance Feedback based Retraining, if feedback from users is available. The experimental evaluation on three publicly available image retrieval datasets indicates the effectiveness of the proposed method in learning more efficient representations for the retrieval task, outperforming other CNN-based retrieval techniques, as well as conventional hand-crafted feature-based approaches in all the used datasets.

Keywords: Content Based Image Retrieval, Convolutional Neural Networks, Deep Learning.

Email addresses: mtzelepi@csd.auth.gr (Maria Tzelepi), tefas@aiia.csd.auth.gr (Anastasios Tefas)

1. Introduction

Image retrieval is a research area of Information Retrieval [1] of great scientific interest since 1970s. Earlier studies include manual annotation of images using keywords and searching by text [2]. Content Based Image Retrieval (CBIR), [3], has been proposed in 1990s, in order to overcome the difficulties of text-based image retrieval, deriving from the manual annotation of images, that is based on the subjective human perception, and the time and labor requirements of annotation.

CBIR refers to the process of obtaining images that are relevant to a query image from a large collection based on their visual content [4]. 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. A key issue associated with CBIR is to extract meaningful information from raw data in order to eliminate the so-called semantic-gap [5]. The semantic-gap refers to the difference between the low level representations of images and their higher level concepts. While earlier works focus on primitive features that describe the image content such as color, texture, and shape, numerous more recent works have been elaborated on the direction of finding semantically richer image representations. Among the most effective are those that use the Fisher Vector descriptors [6], Vector of Locally Aggregated Descriptors (VLAD) [7, 8] or combine bag-of-words models [9] with local descriptors such as Scale-Invariant Feature Transform (SIFT) [10].

Several recent studies introduce Deep Learning algorithms [11] against the shallow aforementioned approaches to a wide range of computer vision tasks, including image retrieval [12, 13, 14, 15]. The main reasons behind their success are the availability of large annotated datasets, and the GPUs computational power and affordability. Deep Convolutional Neural Networks (CNN), [16, 17], are considered the more efficient Deep Learning architecture for visual information analysis. CNNs comprise of a number of convolutional and subsampling layers with non-linear neural activations, 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 non-linear transformation using appropriate activation functions. After multiple convolution layers and subsampling the structure of the deep architecture changes to fully connected layers and single dimensional signals. These activations are usually used as deep representations for classification, clustering or retrieval.

Over the last few years, deep CNNs have been established as one of the most promising avenues of research in the computer vision area due to their outstanding performance in a series of vision recognition tasks, such as image classification [18, 19], face recognition [20], digit recognition [21, 22], pose estimation [23], and object and pedestrian detection [24, 25]. It has also been demonstrated that features extracted from the activation of a CNN trained in a fully supervised fashion on a large, fixed set of object recognition tasks can be re-purposed to novel generic recognition tasks, [26]. Motivated by these results, deep CNNs introduced in the vivid research area of CBIR. The primary approach of applying deep CNNs in the retrieval domain is to extract the feature representations from a pretrained model by feeding images in the input layer of the model and taking activation values drawn either from the fully connected layers [27, 28, 29, 30] which are meant to capture high-level semantic information, or from the convolutional layers exploiting the spatial information of these layers, using either sum-pooling techniques [31, 32] or max-pooling [33]. Current research also includes model retraining approaches, which are more relevant to our work, while other studies focus on the combination of the CNN descriptors with conventional descriptors like the VLAD representation. The existing related works are discussed in the following section.

Our work investigates model retraining (also known as finetuning) approaches in order to enhance the deep CNN descriptors for the retrieval task. We employ a pretrained model to extract feature representations from the activations of the convolutional layers using max-pooling, we properly adapt the model, and we subsequently retrain it. By retraining we mean that we use the weights of a model pretrained for classification, and we finetune them for a different task, instead of training from scratch with randomly initialized weights, exploiting the idea that a deep neural architecture can non-linearly distort the feature space in order to modify the feature representations, with respect to the available information.

Based on the available information we propose three retraining approaches, which are overall able to exploit any kind of available information:

- Fully Unsupervised Retraining (FU): if no information is available, except for the dataset itself.
- Retraining with Relevance Information (RRI): if the labels of the dataset or of a part of the dataset are available.
- Relevance Feedback-based Retraining (RF): if feedback from users is available.

Furthermore, since the FU approach can be applied in any case, we deploy combinatory schemes, where the RRI and RF approaches can be applied on the FU modified model, in a pipeline. In this fashion the FU retraining approach operates as a pretraining step to the subsequent one.

Finally, this method uses retargeting for the learning phase, instead of triplet loss, allowing for single sample training which is very fast and can be easily parallelized and implemented in a distributed manner.

The remainder of the manuscript is structured as follows. Section 2 discusses prior work. The proposed method is described in detail in Section 3. Experiments are provided in Section 4. Finally, conclusions are drawn in Section 5.

2. Prior Work

In this Section we present previous CNN-based works for image retrieval. Firstly, an evaluation of CNN features in various recognition tasks, including image retrieval that improve the baseline performance using spatial information is presented in [28]. In [27] 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. From a different viewpoint, in [30, 34], CNN activations at multiple scale levels are combined with the VLAD representation. In [31], a feature aggregation pipeline is presented using sum pooling. while in [32] a cross-dimensional weighting and aggregation of deep convolutional neural network layer output is proposed. An approach that produces compact feature vectors derived from the convolutional layer activations that encode several image regions is proposed in [33]. In [35], a three-stream Siamense network is proposed to optimize the weights of the so-called R-MAC representation, proposed in [33], for the retrieval task, using a triplet ranking loss. The public Landmarks dataset, that is also used in [27], is utilized for the model training. In [36] a pipeline that uses the convolutional CNN-features and the bag-of-Words aggregation scheme is proposed, while in [37] the authors propose to exploit complementary strengths of CNN features of different layers outperforming the concatenation of multiple layers. In [38], the bilinear CNN-based architectures [39] are introduced in the CBIR domain where a bilinear root pooling is proposed to project the features extracted from the two parallel CNN models into a small dimension and the resulting model is trained on image retrieval datasets using unsupervised training. In [40] 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. In [41], a Weakly-supervised Deep Matrix Factorization framework is proposed for social image tag refinement, tag assignment and image retrieval, that uncovers the latent image representations and tag representations embedded in the latent subspace by collaboratively exploiting the weakly-supervised tagging information, the visual structure and the semantic structure.

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 [29]. A related approach has also been proposed in the face recognition task which, using a triplet-based loss function, achieves state-of-the-art performance, [42], while a relevant idea recently successfully introduced in the cross-modal retrieval domain [43]. These approaches are using triplet sample learning which is difficult to be implemented in large scale, and usually active learning is used in order to select meaningful triplets that can indeed contribute to learning [42]. In our approach we extend these methodologies by considering multiple relevant and multiple irrelevant samples in the training procedure for each training sample. Additionally, we boost the training speed by defining representation targets for the training samples and regression on the hidden layers, instead of defining more complex loss functions that need three samples for each training step. That is, our approach uses single sample training allowing for very fast and distributed learning. Furthermore, the proposed method is also able to exploit the geometric structure of the data using unsupervised learning, as well as to exploit the user's feedback using relevance feedback. Finally, since our focus is to produce low-dimensional descriptors, which improve both the retrieval time and the memory requirements, we apply our method on convolutional layers using max-pooling techniques, as opposed to the previous methodologies which utilize the fully-connected layers.

3. Proposed Method

In this work we consider image and video retrieval applications that should be employed in machines with restricted resources in terms of memory and computational power, such as drones, robots, smartphones and other embedded systems. In these cases, there are restrictions in terms of memory (*e.g.* only 2 Gb of RAM in current state of the art GPUs for embedded systems) and in terms of computational power (*e.g.* restricted number of processing cores in GPUs) since energy efficiency and compactness constitutes a major issue. For the above reasons current deep learning architectures that use a huge number of parameters are inappropriate to be used in such applications even if the training procedure is performed offline. For example, in the context of the media coverage of a certain event with drones, a desirable operation would be to retrieve and show relevant images to the ones captured from the drones of points of particular interest (*e.g.* landmark buildings, monuments). This application would impose smaller and faster architectures that could be deployed easier on-drone.

Towards this end, we exploit the ability of a deep CNN to modify its internal structure, and we propose a model retraining method that suggests three approaches relying on the available information, aiming at producing efficient low-dimensional image representations for the retrieval task, which improve both the retrieval performance and the memory requirements.

We utilize the BVLC Reference CaffeNet model[1], which is an implementation of the AlexNet model trained on the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) 2012 to classify 1.3 million images to 1000 ImageNet classes, [18]. The model consists of eight trained neural network layers; the first five are convolutional and the remaining three are fully connected. Max-pooling layers follow the first, second and fifth convolutional layers, while the ReLU nonlinearity ($f(x) = max(0, x)$) is applied to every convolutional and fully connected layer, except the last fully connected layer (denoted as FC8). The output of the FC8 layer is a distribution over 1000 ImageNet classes. The softmax loss is used during the training. An overview of the CaffeNet architecture is provided in Fig. 1.

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

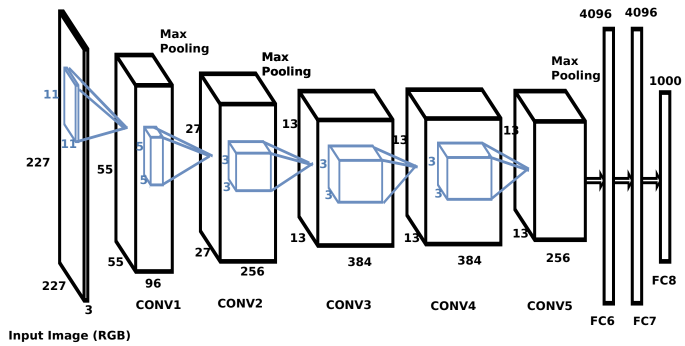

Figure 1: Overview of the CaffeNet Architecture

In general, the neural network accepts an RGB image as a three dimensional tensor of dimensions $W_1 \times H_1 \times D_1$. Subsequently three dimensional filters are learned and applied in each layer where convolution is performed, and output a three dimensional tensor of dimensions $W_2 \times H_2 \times D_2$, where D_2 is equal to the number of filters. The two-dimensional feature maps $W_2 \times H_2$, contain the responses of each filter at every spatial position. We employ the CaffeNet model to directly extract feature representations from a certain convolutional layer. We consider the activations after the ReLU layer. Since the representations obtained from a CNN model for a set of input images are adjustable by modifying the weights of the model, we retrain the parameters of the layer of interest relying on the available information. To this aim, we adapt the pretrained model by removing the layers following the convolutional layer utilized for the feature extraction, and we add an extra pooling layer, the so-called Maximum Activations of Convolutions (MAC) layer, which implements the max-pooling operation over the width and height of the output volume, for each of the D_2 feature maps, [33]. Subsequently, we use the representations obtained from the MAC layer in order to build the new target representations for each image according to the retraining scheme, and we retrain the neural network using the Euclidean Loss for the formulated regression task. The retargeting procedure for each of the proposed approaches is described in the following subsections.

As mentioned previously, the proposed method utilizes the convolutional layers for the feature extraction, against the fully-connected ones [44]. The underlying reasons behind this follow below. First, by definition the convolutional layers preserve spatial information due to the spatial arrangement of the activations, as opposed to the fully-connected ones which discard it since they are connected to all the input neurons. Furthermore, usually the fully-connected layers of CNNs occupy the most of the parameters, for instance, the fully-connected layers of the utilized network contain 59M parameters out of a total of 61M parameters, whereas in VGG [45] the fully connected layers contain 102M parameters out of a total of 138M parameters. Thus, by discarding the fully-connected portion of the network we drastically reduce the amount of the parameters and consequently we restrict the storage requirements and the computational cost. Furthermore, this modification also allows arbitrary-sized input images, since the fixed-length input requirement concerns the fully-connected layers, and hence this allows for using low-resolution images, which can be very useful in order to make our application to comply with the limitations of various embedded systems, since it can further restrict the computational cost. The advantages of the fully convolutional neural networks are also discussed in [46]. Finally, we should also note that state-of-the-art algorithms in the object detection task, like YOLO9000 [47] and SSD [48], also use fully convolutional architectures, in order to improve the detection speed.

More specifically, in our experiments we use either the last convolutional layer, denoted as CONV5, or the forth convolutional layer denoted as CONV4. The dimension of the CONV5 layer is $13 \times 13 \times 256$ features, while the dimension of the CONV4 layer is $13 \times 13 \times 384$ features. Thus, the MAC layer outputs either a 256-dimensional coarse detailed feature representation, or a 384-dimensional fine-detailed one, for each image, based on the utilized convolutional layer.

The proposed retraining method is schematically described in Fig. 2.

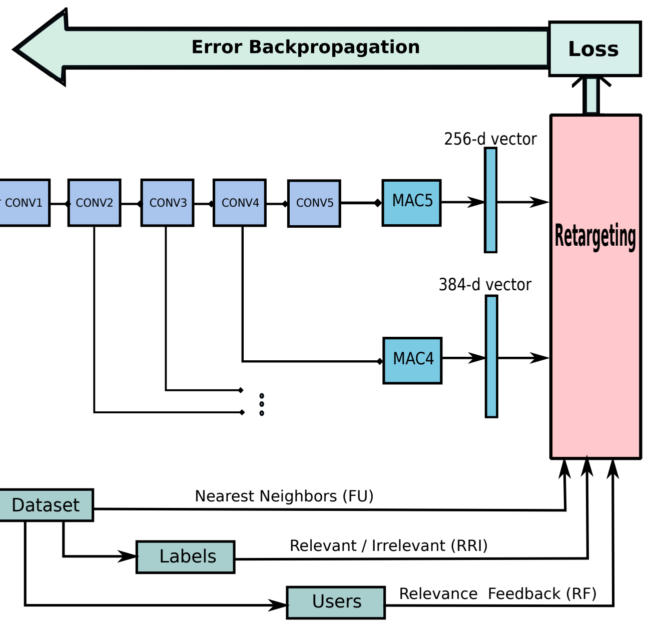

Figure 2: The proposed retraining method.

We should note that various pooling methods could also be used in the proposed approach. Some works in the literature utilize sum-pooling for aggregating the convolutional features to compact descriptors (*e.g.* [31]), while other use max-pooling (*e.g.* [33]). In our investigation we found that max-pooling is superior over sum and stochastic pooling. For example, in Table 1 we show the baseline CaffeNet's results on the CONV5 layer for different pooling methods, in the UKBench-2 dataset. This is consistent with [31], which states that max-pooling achieves better performance, as compared to sum-pooling, while sum-pooling performs better only when the feature descriptors are PCA-whitened. These observations are also drawn in [32, 33].

The three basic proposed retraining approaches are presented in detail in the following subsection.

3.1. Fully Unsupervised Retraining

In the FU approach, we aim 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 [49]. That is, we retrain the pretrained CNN model on the given dataset, aiming at maximizing the cosine similarity between each image representation and its n nearest representations, in terms of cosine distance.

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

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

The new target representations for the images of $\mathcal{I}$ can be determined by solving the following optimization problem:

$$\max_{\mathbf{x}_i \in \mathcal{X}} \mathcal{J} = \max_{\mathbf{x}_i \in \mathcal{X}} \sum_{i=1}^{N} \frac{\mathbf{x}_i^{\intercal} \boldsymbol{\mu}^i}{\|\mathbf{x}_i\| \; \|\boldsymbol{\mu}^i\|} \tag{2}$$

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

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

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

$$\mathbf{x}_i^{(v+1)} = \mathbf{x}_i^{(v)} + \eta\left(\frac{\boldsymbol{\mu}^i}{\|\mathbf{x}_i^{(v)}\| \; \|\boldsymbol{\mu}^i\|} - \frac{\mathbf{x}_i^{(v)\intercal} \boldsymbol{\mu}^i}{\|\mathbf{x}_i^{(v)}\|^3 \; \|\boldsymbol{\mu}^i\|}\mathbf{x}_i^{(v)}\right), \quad \mathbf{x}_i \in \mathcal{X} \tag{4}$$

Finally, we introduce a normalization step, in order to control better the learning rate, as follows:

$$\mathbf{x}_i^{(v+1)} = \mathbf{x}_i^{(v)} + \eta\|\mathbf{x}_i^{(v)}\| \; \|\boldsymbol{\mu}^i\|\left(\frac{\boldsymbol{\mu}^i}{\|\mathbf{x}_i^{(v)}\| \; \|\boldsymbol{\mu}^i\|} - \frac{\mathbf{x}_i^{(v)\intercal} \boldsymbol{\mu}^i}{\|\mathbf{x}_i^{(v)}\|^3 \; \|\boldsymbol{\mu}^i\|}\mathbf{x}_i^{(v)}\right), \quad \mathbf{x}_i \in \mathcal{X} \tag{5}$$

Using the above representations as targets in the layer of interest, we formulate a regression task 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.

3.2. Retraining with Relevance Information

In this approach we propose to enhance the performance of the deep CNN descriptors exploiting the relevance information deriving from the available class labels. To achieve this goal, considering a labelled representation $(\mathbf{x}_i, y_i)$, where $\mathbf{x}_i$ is the image representation and y_i is the corresponding image label, we adapt the convolutional neural layers of the CNN model used for the feature extraction, aiming to maximize the cosine similarity between $\mathbf{x}_i$ and the m nearest relevant representations, and simultaneously to minimize the cosine similarity between $\mathbf{x}_i$ and the l nearest irrelevant representations, in terms of cosine distance. We define as relevant the images belonging to same class, while as irrelevant the images belonging to different classes.

Let $\mathcal{I} = \{\mathbf{I}_i, i = 1, \ldots, N\}$ be a set of N images of the search set provided with relevance information, and $\mathbf{x} = F_L(\mathbf{I})$ the output of the L layer of the pretrained CNN model on an input image $\mathbf{I}$. Then we denote by $\mathcal{X} = \{\mathbf{x}_i, i = 1, \ldots, N\}$ the set of N feature representations emerged in the L layer, by $\mathcal{R}^i = \{\mathbf{r}_k, k = 1, \ldots, K^i\}$ the set of K^i relevant representations of the i-th image and by $C^i = \{\mathbf{c}_j, j = 1, \ldots, L^i\}$ the set of L^i irrelevant representations. We compute the mean vector of the m nearest representations of R^i to the certain image representation $\mathbf{x}_i$, and the mean vector of the l nearest representations of C^i to $\mathbf{x}_i$, and we denote them by $\boldsymbol{\mu}_+^i$ and $\boldsymbol{\mu}_-^i$, respectively. Then, the new target representations for the images of $\mathcal{I}$ can be determined by solving the following optimization problems:

$$\max_{\mathbf{x}_i \in \mathcal{X}} \mathcal{J}^+ = \max_{\mathbf{x}_i \in \mathcal{X}} \sum_{i=1}^{N} \frac{\mathbf{x}_i^\intercal \boldsymbol{\mu}_+^i}{\|\mathbf{x}_i\| \ \|\boldsymbol{\mu}_+^i\|}, \tag{6}$$

$$\min_{\mathbf{x}_i \in \mathcal{X}} \mathcal{J}^- = \min_{\mathbf{x}_i \in \mathcal{X}} \sum_{i=1}^{N} \frac{\mathbf{x}_i^\intercal \boldsymbol{\mu}_-^i}{\|\mathbf{x}_i\| \ \|\boldsymbol{\mu}_-^i\|}, \tag{7}$$

The normalized update rules for the ν-th iteration can be formulated as:

$$\mathbf{x}_i^{(\nu+1)} = \mathbf{x}_i^{(\nu)} + \zeta_1 \|\mathbf{x}_i^{(\nu)}\| \, \|\boldsymbol{\mu}_+^i\| \left(\frac{\boldsymbol{\mu}_+^i}{\|\mathbf{x}_i^{(\nu)}\| \, \|\boldsymbol{\mu}_+^i\|} - \frac{\mathbf{x}_i^{(\nu)\intercal} \boldsymbol{\mu}_+^i}{\|\mathbf{x}_i^{(\nu)}\|^3 \, \|\boldsymbol{\mu}_+^i\|} \mathbf{x}_i^{(\nu)} \right), \quad \mathbf{x}_i \in \mathcal{X} \tag{8}$$

and

$$\mathbf{v}_i^{(\nu+1)} = \mathbf{x}_i^{(\nu)} - \beta_1 \|\mathbf{x}_i^{(\nu)}\| \, \|\boldsymbol{\mu}_-^i\| \left(\frac{\boldsymbol{\mu}_-^i}{\|\mathbf{x}_i^{(\nu)}\| \, \|\boldsymbol{\mu}_-^i\|} - \frac{\mathbf{x}_i^{(\nu)\intercal} \boldsymbol{\mu}_-^i}{\|\mathbf{x}_i^{(\nu)}\|^3 \, \|\boldsymbol{\mu}_-^i\|} \mathbf{x}_i^{(\nu)} \right), \quad \mathbf{x}_i \in \mathcal{X} \tag{9}$$

Consequently, the combinatory normalized update rule, deriving by adding the equations (8) and (9) can be formulated as:

$$\begin{aligned} \mathbf{x}_i^{(\nu+1)} = \mathbf{x}_i^{(\nu)} &+ \zeta \|\mathbf{x}_i^{(\nu)}\| \, \|\boldsymbol{\mu}_+^i\| \left(\frac{\boldsymbol{\mu}_+^i}{\|\mathbf{x}_i^{(\nu)}\| \, \|\boldsymbol{\mu}_+^i\|} - \frac{\mathbf{x}_i^{(\nu)\intercal} \boldsymbol{\mu}_+^i}{\|\mathbf{x}_i^{(\nu)}\|^3 \, \|\boldsymbol{\mu}_+^i\|} \mathbf{x}_i^{(\nu)} \right) \\ &- \beta \|\mathbf{x}_i^{(\nu)}\| \, \|\boldsymbol{\mu}_-^i\| \left(\frac{\boldsymbol{\mu}_-^i}{\|\mathbf{x}_i^{(\nu)}\| \, \|\boldsymbol{\mu}_-^i\|} - \frac{\mathbf{x}_i^{(\nu)\intercal} \boldsymbol{\mu}_-^i}{\|\mathbf{x}_i^{(\nu)}\|^3 \, \|\boldsymbol{\mu}_-^i\|} \mathbf{x}_i^{(\nu)} \right), \quad \mathbf{x}_i \in \mathcal{X} \end{aligned} \tag{10}$$

Thus, as in the previous approach, using the above target representations we retrain the neural network on the images provided with relevance information using backpropagation.

3.3. Relevance Feedback Based Retraining

The idea of this proposed approach is rooted in the relevance feedback philosophy. In general, relevance feedback refers to the ability of users to impart their judgement regarding the relevance of search results to the system. Then, the system can use this information to ameliorate its performance [50, 51]. In this proposed retraining approach we consider information from different users' feedback. This information consists of queries and relevant and irrelevant images to these queries. Then, our goal is to modify the model parameters in order to maximize the cosine similarity between a specific query and its relevant images and minimize the cosine similarity between it and its irrelevant ones.

Let us denote by $\mathcal{Q} = \{\mathbf{Q}_k, k = 1, \ldots, K\}$ a set of queries, $\mathcal{I}_+^k = \{\mathbf{I}_i, i = 1, \ldots, Z\}$ a set of relevant images to a certain query, by $\mathcal{I}_-^k = \{\mathbf{I}_j, j = 1, \ldots, O\}$ a set of irrelevant images, by $\mathbf{x} = F_L(\mathbf{I})$ the output of the L layer of the pretrained CNN model on an input image $\mathbf{I}$, and by $\mathbf{q} = F_L(\mathbf{Q})$ the output of the L layer on a query. Then we denote by $\mathcal{X}_+^k = \{\mathbf{x}_i, i = 1, \ldots, Z\}$ the set of feature representations emerged in L layer of Z images that have been qualified as relevant by a user, and by $\mathcal{X}_-^k = \{\mathbf{x}_j, j = 1, \ldots, O\}$ the set of O irrelevant feature representations.

The new target representations for the relevant and irrelevant images can be respectively determined by solving the following optimization problems:

$$\max_{\mathbf{x}_i \in \mathcal{X}_+^k} \mathcal{J}^+ = \max_{\mathbf{x}_i \in \mathcal{X}_+^k} \sum_{i=1}^{Z} \frac{\mathbf{x}_i^\intercal \boldsymbol{q}^k}{||\mathbf{x}_i||\ ||\boldsymbol{q}^k||}, \tag{11}$$

$$\min_{\mathbf{x}_j \in \mathcal{X}_-^k} \mathcal{J}^- = \min_{\mathbf{x}_j \in \mathcal{X}_-^k} \sum_{j=1}^{O} \frac{\mathbf{x}_j^\intercal \boldsymbol{q}^k}{||\mathbf{x}_j||\ ||\boldsymbol{q}^k||}, \tag{12}$$

The normalized update rules for the v-th iteration can be formulated as:

$$\mathbf{x}_i^{(v+1)} = \mathbf{x}_i^{(v)} + \alpha ||\mathbf{x}_i^{(v)}||\ ||\boldsymbol{q}^k|| \left(\frac{\boldsymbol{q}^k}{||\mathbf{x}_i^{(v)}||\ ||\boldsymbol{q}^k||} - \frac{\mathbf{x}_i^{(v)\intercal} \boldsymbol{q}^k}{||\mathbf{x}_i^{(v)}||^3\ ||\boldsymbol{q}^k||} \mathbf{x}_i^{(v)} \right), \quad \mathbf{x}_i \in \mathcal{X}_+^k \tag{13}$$

and

$$\mathbf{x}_j^{(v+1)} = \mathbf{x}_j^{(v)} - \alpha ||\mathbf{x}_j^{(v)}||\ ||\boldsymbol{q}^k|| \left(\frac{\boldsymbol{q}^k}{||\mathbf{x}_j^{(v)}||\ ||\boldsymbol{q}^k||} - \frac{\mathbf{x}_j^{(v)\intercal} \boldsymbol{q}^k}{||\mathbf{x}_j^{(v)}||^3\ ||\boldsymbol{q}^k||} \mathbf{x}_j^{(v)} \right), \quad \mathbf{x}_j \in \mathcal{X}_-^k \tag{14}$$

Similar to the other approaches, using the above representations as targets in the layer of interest, we retrain the neural network on the set of relevant and irrelevant images.

4. Experiments

In this Section we present the experiments conducted in order to assess the performance of the proposed method. Firstly, a brief description of the evaluation metrics and the datasets is provided. Subsequently, we describe the experimental details of each approach, and finally we demonstrate the experimental results.

4.1. Evaluation Metrics

Throughout this paper we use 4 evaluation metrics: precision, recall, mean Average Precision (mAP), and top-N score. The definitions of the above metrics follow below:

$$Precision = \frac{n.\ of\ Relevant\ Retrieved\ Images}{n.\ of\ Retrieved\ Images} \tag{15}$$

$$Recall = \frac{n.\ of\ Relevant\ Retrieved\ Images}{n.\ of\ Relevant\ Images} \tag{16}$$

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:

$$AP_i = \frac{1}{Q_i}\sum_{n=1}^{N}\frac{R_i^n}{n}t_n^i, \tag{17}$$

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.

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

4.2. Datasets

Paris 6k [52]: consists of 6392 images (20 of the 6412 provided images are corrupted) collected from Flickr by searching for particular Paris 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. Like in most CNN-based works [28, 27, 29, 34, 31] we use the full queries for the retrieval. The query images are not considered in the search set in the retrieval procedure, and neither used in the phase of model retraining. We show some example images in Fig. 3.

Figure 3: Sample images of the Paris 6k dataset

UKBench [53]: 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. Samples are provided in Fig. 4.

Figure 4: Sample images of the UKBench dataset

UKBench-2: since our method performs learning and the UKBench dataset does not provide a discrete set of queries, we hold out one image per class, forming a search set of 7650 images and a set of 2550 queries. As in UKBench, we use the top-3 score for the evaluation, which is a number between 0 and 3.

4.3. Experimental Setup

The proposed method was implemented using the Caffe Deep Learning framework, [54]. As mentioned before, in our experiments we utilize either the CONV5 or the CONV4 layer for the feature extraction. Additionally, in the model retraining phase we replace the ReLU layer, that follows the utilized convolutional layer with a PRELU layer [55] which is initialized randomly. Furthermore, since the first layers of CaffeNet trained on ImageNet learned more generic feature representations, all the previous convolutional layers remain unchanged, and we train only the layer of interest, restricting significantly the training cost. Finally, we use the adaptive moment estimation algorithm (Adam) [56], instead of the simple gradient descent for the network optimization, with the default parameters. All results obtained using cosine distance.

In Table 1 we present the results of our investigation regarding the pooling methods. That is, we report the top-3 Score for UKBench-2 dataset on the CONV5 layer, using different pooling methods. As it is shown the max-pooling attains superior performance over the sum and stochastic pooling.

Pooling Method	**Score**
Max	2.615
Sum	2.50
Stochastic	2.572

Table 1: UKBench-2: Top-3 Score for various pooling methods

We note that we can also utilize other distance metrics. Existing CBIR approaches usually use either cosine distance, *e.g.* [29, 36], or Euclidean distance [27, 28]. We also conducted experiments using the Euclidean distance. The choice of the distance metric, affects the optimization objective for the retargeting procedure. That is, if we consider the Euclidean distance *e.g.* in the FU approach, the optimization problem of (2), is replaced by the following one:

$$\min_{\mathbf{x}_i \in \mathcal{X}} \mathcal{J} = \min_{\mathbf{x}_i \in \mathcal{X}} \sum_{i=1}^{N} ||\mathbf{x}_i - \boldsymbol{\mu}_i||_2^2 \tag{18}$$

Hence, following the gradient, the update rule for the v-th iteration for each image can then be formulated as:

$$\mathbf{x}_i^{(v+1)} = \mathbf{x}_i^{(v)} - 2\eta(\mathbf{x}_i^{(v)} - \boldsymbol{\mu}_i), \quad \mathbf{x}_i \in \mathcal{X} \tag{19}$$

where the parameter $\eta \in [0, 0.5]$ controls the desired distance from the n nearest representations.

Correspondingly, the update rule for the ν-th iteration for each image, for the RRI approach is given by the equation:

$$\mathbf{x}_i^{(\nu+1)} = \mathbf{x}_i^{(\nu)} - (1-\beta)(\mathbf{x}_i^{(\nu)} - \boldsymbol{\mu}_+^i) + \beta(\mathbf{x}_i^{(\nu)} - \boldsymbol{\mu}_-^i), \quad \mathbf{x}_i \in \mathcal{X} \tag{20}$$

where the parameter $\beta = 1 - \zeta, \in [0, 1]$ controls the desired distance both from the relevant and the irrelevant representations.

Finally, the update rules for the ν-th iteration for each image, for the RF approach are given by the following equations:

$$\mathbf{x}_i^{(\nu+1)} = \mathbf{x}_i^{(\nu)} - 2\alpha(\mathbf{x}_i^{(\nu)} - \mathbf{q}^k), \quad \mathbf{x}_i \in \mathcal{X}_+^k \tag{21}$$

and

$$\mathbf{x}_j^{(\nu+1)} = \mathbf{x}_j^{(\nu)} + 2\alpha(\mathbf{x}_j^{(\nu)} - \mathbf{q}^k), \quad \mathbf{x}_j \in \mathcal{X}_-^k \tag{22}$$

where the parameter $\alpha \in [0, 0.5]$ controls the desired distance from the query representation.

The baseline CaffeNet's results on the CONV5 layer utilizing the Euclidean distance is 0.5227 against 0.5602 in Paris 6k dataset, and 2.5286 against 2.6154 in UKBench-2 dataset. We also applied the proposed FU approach on the CONV5 layer, setting the same parameters, on both the UKBench-2 and Paris 6k datasets. The experimental results are illustrated in Fig. 5, 6. As we can observe the cosine similarity attains superior performance over the Euclidean distance in both the considered cases.

In the following we present the selected parameters for each of the proposed approaches.

4.3.1. Fully Unsupervised

First, in the UKBench-2 dataset, we fix the number of nearest representations, n, in (1) to 1 and the retargeting step to 2000 iterations, and we examine the effect of the parameter η in (5). Thus, in Fig. 7 we illustrate the top-3 Score at

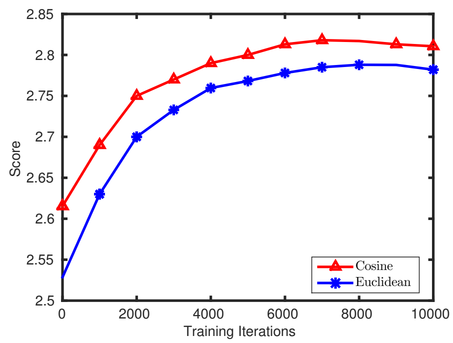

Figure 5: UKBench-2: Comparison of Euclidean and Cosine distances, on the FU approach on CONV5 layer

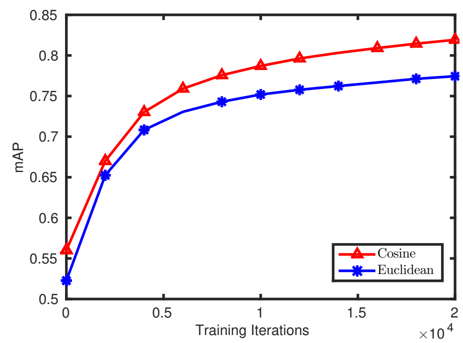

Figure 6: Paris 6k: Comparison of Euclidean and Cosine distances, on the FU approach on CONV5 layer

each iteration of the training process for different values of η. Next, we fix the parameter η to 0.6, and we perform experiments for different numbers of nearest representations, n. Experimental results are shown in Fig. 8. Finally, for fixed values of η and nearest representations, we vary the step of retargeting. That is, we re-determine the targets for the model retraining, (5), with a certain step of iterations. The experimental results are illustrated in Fig. 9. Thus, we set the value η to 0.6, the number of nearest representations to 1, and the retargeting step to 1000 iterations. The same parameters are also used in the UKBench dataset. Finally, in the Paris 6k dataset, we also set the parameter η in (5) to 0.6 and for fixed retargeting step set of 2000 iterations, we examine the appropriate number of nearest representations. Experimental results are shown in Fig. 10. Then, for the optimal number of nearest representations, we examine the retargeting step. Experimental results are shown in Fig. 11. Hence, in Paris 6k dataset we set the value η to 0.6, the number of nearest representations to 20, and the retargeting step to 1000 iterations. Regarding the number of the nearest representations, n, in many datasets it is bounded by the number of samples that are available. For example, in the UKBench-2 the limit for the value of n is 2, since there are only three samples per class. Thus, in Fig. 8, we observe that when the value n exceeds the number of images per class, the performance drops. In the case of

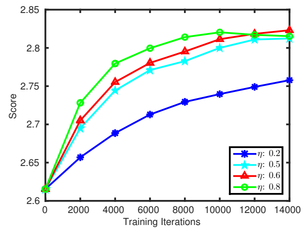

Figure 7: UKBench-2: Score for different values of η in (5)

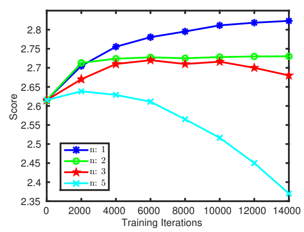

Figure 8: UKBench-2: Score for different numbers of nearest representations, n, in (1)

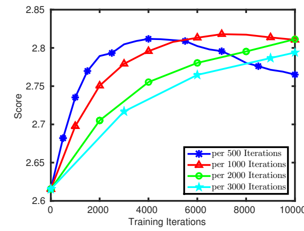

Figure 9: UKBench-2: Score for different retargeting steps

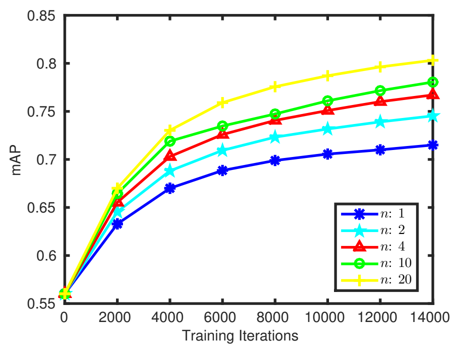

Figure 10: Paris 6k: mAP for different numbers of nearest representations, n, in 1

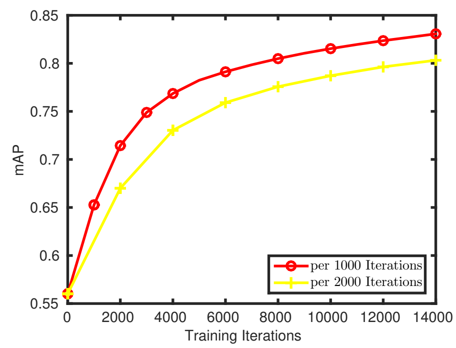

Figure 11: Paris 6k: mAP for different retargeting steps

Paris 6k dataset, where there are more samples available, we see in Fig.10 that the performance improves, as the value of n increases. However, an increased value of the parameter n comes with the cost of finding the n nearest neighbors of each training sample. For a big dataset this cost is critical, but it can be reduced using approximate nearest neighbor techniques. However, this research direction is beyond the scope of this work. Consequently, for a totally unknown dataset an investigation for the value of n between 5 and 10 is a good compromise, however there is also the most safe choice of setting the value 1, which improves the performance in any case.

4.3.2. Retraining with Relevance Information

In the experiments of this approach, since the number of relevant representations varies meaningfully across datasets, we formulate the new target representations for the model retraining with respect to each relevant and 5 nearest irrelevant images of each image. The retargeting step is set to 2000 iterations, the parameter ζ in (10) is set to 0.8, and the parameter β is set to 0.2.

4.3.3. Relevance Feedback based Retraining

In the experiments that conducted to validate the performance of the Relevance Feedback based approach, we consider for each of 2550 different users 1 relevant and 1 irrelevant images for the UKBench-2 dataset, which forms a training set of 5100 images. In Paris 6k dataset, 40 relevant (or equal to the number of relevant, if less) and 40 irrelevant images are considered for each of 55 different users. The parameter α in (13), (14) is set to 0.5.

4.4. Experimental Results

	Feature Representation	**Dimension**	**Score**
1	CaffeNet $\Rightarrow$ CONV4	384	3.3608
2	CaffeNet $\rightarrow$ FU(CONV4) $\Rightarrow$ CONV4	384	3.6999
3	CaffeNet $\rightarrow$ RRI(CONV4) $\Rightarrow$ CONV4	384	3.9122
4	CaffeNet $\rightarrow$ FU(CONV4) $\rightarrow$ RRI(CONV4) $\Rightarrow$ CONV4	384	3.9511
5	CaffeNet $\Rightarrow$ CONV5	256	3.5595
6	CaffeNet $\rightarrow$ FU(CONV5) $\Rightarrow$ CONV5	256	3.8323
7	CaffeNet $\rightarrow$ RRI(CONV5) $\Rightarrow$ CONV5	256	3.8941
8	CaffeNet $\rightarrow$ FU(CONV5) $\rightarrow$ RRI(CONV5) $\Rightarrow$ CONV5	256	**3.9710**

Table 2: UKBench

We illustrate the evaluation results for the three basic model retraining approaches, as well as for the combinatory ones, where the RRI and RF approaches are applied on the FU optimized model.

In the following we denote by CONV5 and CONV4 the feature representations obtained from the CONV5 and CONV4 layer of the CNN model respectively. We denote by FU(L_T) the fully unsupervised retraining on the layer L_T with target representations obtained from the L_T layer, by RRI(L_T) the retraining with relevance information on the layer L_T with target representations obtained from the

	Feature Representation	Dimension	Score
1	CaffeNet $\Rightarrow$ CONV4	384	2.4389
2	CaffeNet $\rightarrow$ FU(CONV4) $\Rightarrow$ CONV4	384	2.70
3	CaffeNet $\rightarrow$ RRI(CONV4) $\Rightarrow$ CONV4	384	2.8624
4	CaffeNet $\rightarrow$ RF(CONV4) $\Rightarrow$ CONV4	384	2.4792
5	CaffeNet $\rightarrow$ FU(CONV4) $\rightarrow$ RRI(CONV4) $\Rightarrow$ CONV4	384	2.9058
6	CaffeNet $\rightarrow$ FU(CONV4)$\rightarrow$ RF(CONV4) $\Rightarrow$ CONV4	384	2.7627
7	CaffeNet $\Rightarrow$ CONV5	256	2.6154
8	CaffeNet $\rightarrow$ FU(CONV5) $\Rightarrow$ CONV5	256	2.8106
9	CaffeNet $\rightarrow$ RRI(CONV5) $\Rightarrow$ CONV5	256	2.8831
10	CaffeNet $\rightarrow$ RF(CONV5) $\Rightarrow$ CONV5	256	2.72
11	CaffeNet $\rightarrow$ FU(CONV5) $\rightarrow$ RRI(CONV5) $\Rightarrow$ CONV5	256	**2.9086**
12	CaffeNet $\rightarrow$ FU(CONV5) $\rightarrow$ RF(CONV5) $\Rightarrow$ CONV5	256	2.8361

Table 3: UKBench-2

L_T layer, and correspondingly by RF(L_T) the relevance feedback based retraining. We use consecutive arrows to describe the retraining pipeline of our approaches, and the implication arrow to show the final feature representation employed for the retrieval procedure. Thus, *CaffeNet* $\Rightarrow$ *CONV5* implies that we obtain the CONV5 representations from the CaffeNet model and we use them for the retrieval procedure, while *CaffeNet* $\rightarrow$ *RRI(CONV4)* $\Rightarrow$ *CONV4* denotes that we formulate the target representations using the features emerged in the CONV4 CaffeNet layer and we retrain with relevance information the CONV4 layer of the CaffeNet, then we extract the CONV4 representations of the modified model, and we use them for the retrieval.

Tables 2 - 4 summarize the experimental results on all the datasets. The best performance is printed in bold. From the provided results several remarks can be drawn. Firstly, we observe that each retraining approach improves the baseline results of CaffeNet in all the used datasets. Furthermore, we can notice that in all the datasets the CONV5 retraining achieves better performance.

	Feature Representation	Dimension	mAP
1	CaffeNet $\Rightarrow$ CONV4	384	0.4589
2	CaffeNet $\rightarrow$ FU(CONV4) $\Rightarrow$ CONV4	384	0.7337
3	CaffeNet $\rightarrow$ RRI(CONV4) $\Rightarrow$ CONV4	384	0.9837
4	CaffeNet $\rightarrow$ RF(CONV4) $\Rightarrow$ CONV4	384	0.6325
5	CaffeNet $\rightarrow$ FU(CONV4) $\rightarrow$ RRI(CONV4) $\Rightarrow$ CONV4	384	0.9715
6	CaffeNet $\rightarrow$ FU(CONV4) $\rightarrow$ RF(CONV4) $\Rightarrow$ CONV4	384	0.8030
7	CaffeNet $\Rightarrow$ CONV5	256	0.5602
8	CaffeNet $\rightarrow$ FU(CONV5) $\Rightarrow$ CONV5	256	0.8347
9	CaffeNet $\rightarrow$ RRI(CONV5) $\Rightarrow$ CONV5	256	0.9854
10	CaffeNet $\rightarrow$ RF(CONV5) $\Rightarrow$ CONV5	256	0.7101
11	CaffeNet $\rightarrow$ FU(CONV5) $\rightarrow$ RRI(CONV5) $\Rightarrow$ CONV5	256	**0.9859**
12	CaffeNet $\rightarrow$ FU(CONV5) $\rightarrow$ RF(CONV5) $\Rightarrow$ CONV5	256	0.9023

Table 4: Paris 6k

Additionally, we observe that the FU approach accomplishes remarkable results, while in UKBench dataset this approach leads to state-of-the-art performance. We also see that the other proposed methodologies applied on the modified via the FU approach model indeed yield better retrieval results, as compared to the CaffeNet's employment, in any considered case except for the CONV4 modification in Paris 6k dataset. Finally, we can observe that refined with relevance information model accomplishes state-of-the-art performance in all the datasets, while the relevance-feedback based model achieves considerably improved results in all the used datasets.

More specifically, in Table 2 we show the experimental results of the proposed retraining approaches in the UKBench dataset. First, we see that the baseline CaffeNet's performance of the CONV5 representations is superior over the CONV4 one. Furthermore we observe that both the RRI and FU approaches improve significantly the baseline performance, and also the RRI achieves better results than the FU one, which is reasonable since the FU approach utilizes no information for the model retraining. Finally we can see that the FU pretraining step boosts the performance of the RRI approach on both the CONV5 and CONV4 layers.

Similar remarks can be drawn for the UKBench-2 dataset, in Table 3. Regarding the RF approach, we can see in the 4^{th} and 10^{th} rows that the method indeed improves the CaffeNet retrieval results on both the CONV5 and CONV4 layers, but we observe that the improvement of the RF approach is not as notable as the FU and RRI ones. We attribute this to the comparatively small training set of the RF approach (5100 against 10200 images). In general, the number of the relevant and irrelevant images that create the new dataset for the model retraining, appears to be the key factor of the RF improvement.

Finally, in Table 4 we illustrate the experimental results on the Paris 6k dataset. As previously, it is shown that the proposed approaches improve the CaffeNet retrieval results. It is also shown, that the RRI approach in a single training step can accomplish state-of-the-art performance (9^{th} row). The FU retraining scheme boosts the RF results, while in the case of the RRI retraining on the FU modified model, the results are marginally improved for the CONV5 layer (9^{th} and 11^{th} rows), and are slightly inferior for the CONV4 (3^{rd} and 5^{th} rows). Finally, we observe that the RF approach performs comparatively poorly.

In Fig. 12 we provide the Precision-Recall curves of all the considered approaches for the Paris 6k datasets, utilizing the CONV5 layer. It is shown that the proposed approaches can indeed achieve significantly enhanced results against the baseline. It is also shown that the RF approach applied on the FU modified model can accomplish considerably improved performance as compared to the RF approach on the CaffeNet model, while this is not confirmed in the case of the RRI approach on the FU retrained model, where the performance is almost identical.

In Fig. 13,14 we provide some examples of the top three retrieved images for certain queries of UKBench-2 dataset, using the baseline CONV5 CaffeNet's features, and features obtained from our FU and RRI on FU retrained models respectively. As it is illustrated, the proposed approaches improve the retrieval results. Additionally we can see in the third example of the two figures that

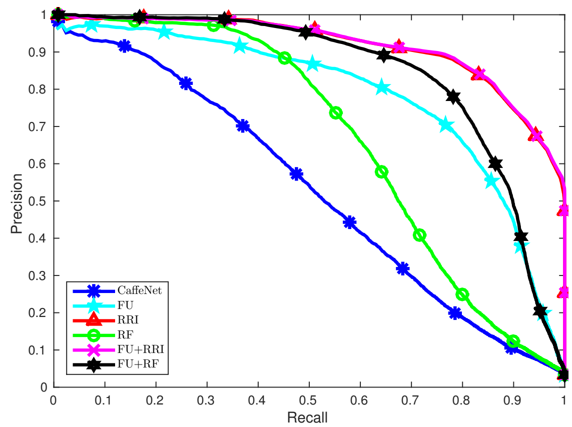

Figure 12: Paris 6k: Precision Recall

the FU retrained model returns two out of three relevant images, while the RRI approach applied on the FU one, returns all the relevant images to the specific query.

Finally, we compare our method against other CNN-based, as well as hand-crafted feature-based methods, on image retrieval. First, we provide a comparison against methods that utilize supervised learning with the proposed RRI approach, which utilizes supervised learning too, in Table 5. Second, we compare the proposed FU approach against other methods that do not utilize supervised learning in Table 6. Since the proposed RF approach is novel, and the competitive methods do not utilize information derived from users' feedback, the results are reported only in Tables 3-4, and we do not include it in the comparisons. We compare our method with the competitive ones, regardless the dimension of the compared feature representations. We also note that among the provided results, there are methods, that use information from multiple regions of the image, as in the case of R-MAC, [33], and Deep Image Retrieval [35]. To the best of our knowledge, the proposed method outperforms every other competitive method. Methods marked with * use the cropped queries in Paris 6k dataset.

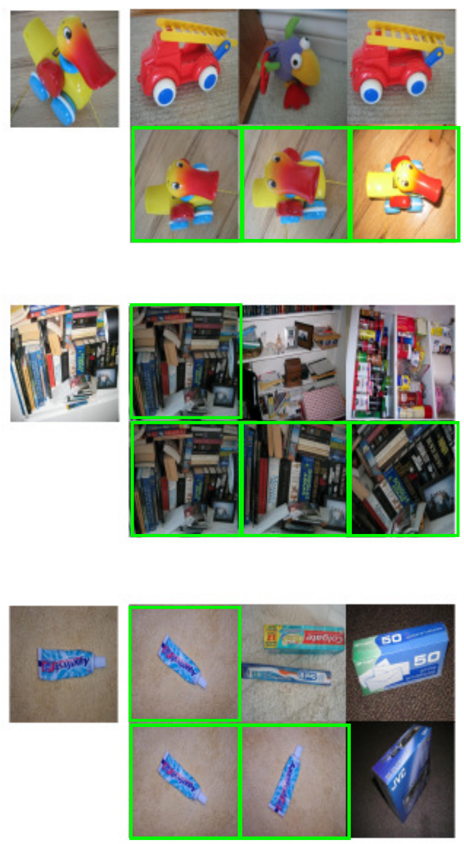

Figure 13: For each of the three sets of images the query image is the first one of the top row and the images that follow in the top row are the first 3 retrieved using the baseline CONV5 representation. The top 3 retrieved images using the FU approach on the CONV5 layer are shown in the second row for the same query

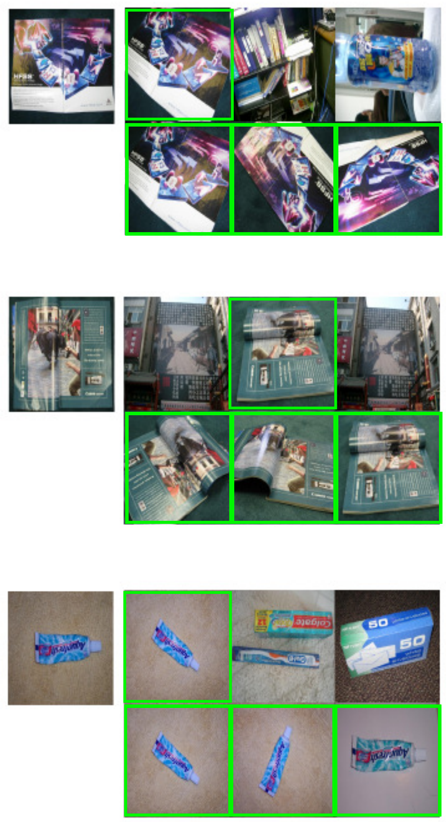

Figure 14: For each of the three sets of images the query image is the first one of the top row and the images that follow in the top row are the first 3 retrieved using the baseline CONV5 representation. The top 3 retrieved images using the FU→RRI approach on the CONV5 layer are shown in the second row for the same query

Method	Dim	Paris 6k	UKBench
Neural Codes [27]	4096	-	3.56
Neural Codes [27]	256	-	3.35
ReDSL.FC1 [29]	4096	0.9474	-
Deep Image Retrieval [35]	512	0.871	-
Ours	256	**0.9859**	**3.9710**

Table 5: Comparison against other supervised methods

Method	Dim	Paris 6k	UKBench
CVLAD* [57]	64k	-	3.62
BOW * [58]	200k	0.46	2.81
CNNaug-ss [28]	4k-15k	0.795	3.644
Spoc [31]	256	-	3.65
Fine-residual VLAD [8]	256	-	3.43
Multi-layer [37]	100k	-	3.69
CNN-VLAD [34]	128	0.694	-
R-MAC* [33]	512	0.83	-
R-MAC* [33]	256	0.729	-
CroW* [32]	256	0.765	-
CRB-CNN-16 [38]	512	-	3.56
Ours	256	**0.8347**	**3.8323**

Table 6: Comparison against other unsupervised methods

5. Conclusions

In this paper we proposed a model retraining methodology for enhancing the deep convolutional representations in the retrieval domain. The proposed method suggests three retraining approaches relying on the available information. Thus, if no information is available, the Fully Unsupervised retraining approach is proposed, if the labels are available the Retraining with Relevance Information, and finally if users' feedback is available the Relevance Feedback based retraining is proposed. We utilize a deep CNN model to obtain the convolutional representations and build the target representations according to each approach, and then we retrain appropriately the network's weights. We also proposed a combinatory retraining strategy, where the FU retraining approach can be utilized as a pretraining step in order to boost the performance of the RRI and RF approaches. We note that all the proposed approaches are applicable to the fully connected layers too, as well as to other CNN architectures. We should also note that the proposed methodology is applicable to any other CNN-based image retrieval method that utilizes a CNN model to directly extract feature representations. Experimental results indicate the effectiveness of our method, with superior performance over the state of the art approaches, either via a single retraining approach, or by utilizing successive retraining processes.

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] C. D. Manning, P. Raghavan, H. Schutze, Introduction to information retrieval, Cambridge University Press, 2008.

[2] N.-S. Chang, K. S. Fu, A relational database system for images, in: Pictorial Information Systems, Springer, 1980, pp. 288–321.

[3] T. Kato, Database architecture for content-based image retrieval, in: SPIE/IS&T 1992 symposium on electronic imaging: science and technology, International Society for Optics and Photonics, 1992, pp. 112–123.

[4] R. Datta, J. Li, J. 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, ACM, 2005, pp. 253–262.

[5] A. W. Smeulders, M. Worring, S. Santini, A. Gupta, R. Jain, Content-based image retrieval at the end of the early years, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (12) (2000) 1349–1380.

[6] F. Perronnin, Y. Liu, J. Sánchez, H. Poirier, Large-scale image retrieval with compressed fisher vectors, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2010, pp. 3384–3391.

[7] R. Arandjelovic, A. Zisserman, All about vlad, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2013, pp. 1578–1585.

[8] Z. Liu, S. Wang, Q. Tian, Fine-residual vlad for image retrieval, Neurocomputing 173 (2016) 1183–1191.

[9] F. Jiang, H.-M. Hu, J. Zheng, B. Li, A hierarchal bow for image retrieval by enhancing feature salience, Neurocomputing 175 (2016) 146–154.

[10] J. Sivic, A. Zisserman, Video Google: A text retrieval approach to object matching in videos, in: Proceedings of the International Conference on Computer Vision, Vol. 2, 2003, pp. 1470–1477.

[11] L. Deng, A tutorial survey of architectures, algorithms, and applications for deep learning, APSIPA Transactions on Signal and Information Processing 3 (2014) e2.

[12] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, F. E. Alsaadi, A survey of deep neural network architectures and their applications, Neurocomputing 234 (2017) 11–26.

[13] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, M. S. Lew, Deep learning for visual understanding: A review, Neurocomputing 187 (2016) 27–48.

[14] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, et al., Deep neural networks for

acoustic modeling in speech recognition: The shared views of four research groups, IEEE Signal Processing Magazine 29 (6) (2012) 82–97.

[15] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, A. Y. Ng, Multimodal deep learning, in: Proceedings of the 28th International Conference on Machine Learning (ICML), 2011, pp. 689–696.

[16] B. B. Le Cun, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, L. D. Jackel, Handwritten digit recognition with a back-propagation network, in: Advances in neural information processing systems 2, Morgan Kaufmann Publishers Inc., 1990, pp. 396–404.

[17] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proceedings of the IEEE 86 (11) (1998) 2278–2324.

[18] A. Krizhevsky, I. Sutskever, G. E. Hinton, Imagenet classification with deep convolutional neural networks, in: Advances in neural information processing systems, 2012, pp. 1097–1105.

[19] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2015, pp. 1–9.

[20] Y. Taigman, M. Yang, M. Ranzato, L. Wolf, Deepface: Closing the gap to human-level performance in face verification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2014, pp. 1701–1708.

[21] D. Ciresan, U. Meier, J. Schmidhuber, Multi-column deep neural networks for image classification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2012, pp. 3642–3649.

[22] Y. LeCun, L. Jackel, L. Bottou, C. Cortes, J. S. Denker, H. Drucker, I. Guyon, U. Muller, E. Sackinger, P. Simard, et al., Learning algorithms for classifica-

tion: A comparison on handwritten digit recognition, Neural networks: the statistical mechanics perspective 261 (1995) 276.

[23] A. Toshev, C. Szegedy, Deeppose: Human pose estimation via deep neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2014, pp. 1653–1660.

[24] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2014, pp. 580–587.

[25] P. Sermanet, K. Kavukcuoglu, S. Chintala, Y. LeCun, Pedestrian detection with unsupervised multi-stage feature learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2013, pp. 3626–3633.

[26] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, T. Darrell, Decaf: A deep convolutional activation feature for generic visual recognition., in: ICML, 2014, pp. 647–655.

[27] A. Babenko, A. Slesarev, A. Chigorin, V. Lempitsky, Neural codes for image retrieval, in: European Conference on Computer Vision (ECCV), Springer, 2014, pp. 584–599.

[28] A. Razavian, H. Azizpour, J. Sullivan, S. Carlsson, Cnn features off-the-shelf: an astounding baseline for recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2014, pp. 806–813.

[29] J. Wan, D. Wang, S. C. H. Hoi, P. Wu, J. Zhu, Y. Zhang, J. Li, Deep learning for content-based image retrieval: A comprehensive study, in: Proceedings of the ACM International Conference on Multimedia, ACM, 2014, pp. 157–166.

[30] Y. Gong, L. Wang, R. Guo, S. Lazebnik, Multi-scale orderless pooling of deep convolutional activation features, in: European Conference on Computer Vision (ECCV), Springer, 2014, pp. 392–407.

[31] A. Babenko, V. Lempitsky, Aggregating local deep features for image retrieval, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1269–1277.

[32] Y. Kalantidis, C. Mellina, S. Osindero, Cross-dimensional weighting for aggregated deep convolutional features, in: European Conference on Computer Vision (ECCV) Workshops, Springer, 2015, pp. 685–701.

[33] G. Tolias, R. Sicre, H. Jégou, Particular object retrieval with integral max-pooling of cnn activations, CoRR abs/1511.05879.

[34] J. Ng, F. Yang, L. Davis, Exploiting local features from deep networks for image retrieval, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2015, pp. 53–61.

[35] A. Gordo, J. Almazán, J. Revaud, D. Larlus, Deep image retrieval: Learning global representations for image search, in: European Conference on Computer Vision, Springer, 2016, pp. 241–257.

[36] E. Mohedano, K. McGuinness, N. E. O'Connor, A. Salvador, F. Marques, X. Giro-i Nieto, Bags of local convolutional features for scalable instance search, in: Proceedings of the 2016 ACM on International Conference on Multimedia Retrieval, ACM, 2016, pp. 327–331.

[37] W. Yu, K. Yang, H. Yao, X. Sun, P. Xu, Exploiting the complementary strengths of multi-layer cnn features for image retrieval, Neurocomputing 237 (2017) 235–241.

[38] A. Alzu'bi, A. Amira, N. Ramzan, Content-based image retrieval with compact deep convolutional features, Neurocomputing 249 (2017) 95–105.

[39] T.-Y. Lin, A. RoyChowdhury, S. Maji, Bilinear cnn models for fine-grained visual recognition, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1449–1457.

[40] Z. Li, J. Tang, Weakly supervised deep metric learning for community-contributed image retrieval, IEEE Transactions on Multimedia 17 (11) (2015) 1989–1999.

[41] Z. Li, J. Tang, Weakly supervised deep matrix factorization for social image understanding, IEEE Transactions on Image Processing 26 (1) (2017) 276–288.

[42] F. Schroff, D. Kalenichenko, J. Philbin, Facenet: A unified embedding for face recognition and clustering, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2015, pp. 815–823.

[43] L. Wang, Y. Li, S. Lazebnik, Learning deep structure-preserving image-text embeddings, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 5005–5013.

[44] M. Tzelepi, A. Tefas, Exploiting supervised learning for finetuning deep cnns in content based image retrieval, in: 23nd International Conference on Pattern Recognition (ICPR), IEEE, 2016.

[45] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556.

[46] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3431–3440.

[47] J. Redmon, A. Farhadi, Yolo9000: better, faster, stronger, arXiv preprint arXiv:1612.08242.

[48] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A. C. Berg, Ssd: Single shot multibox detector, in: European conference on computer vision, Springer, 2016, pp. 21–37.

[49] E. M. Voorhees, The cluster hypothesis revisited, in: Proceedings of the 8th annual international ACM SIGIR conference on Research and development in information retrieval, ACM, 1985, pp. 188–196.

[50] Y. Rui, T. S. Huang, M. Ortega, S. Mehrotra, Relevance feedback: a power tool for interactive content-based image retrieval, IEEE Transactions on Circuits and Systems for Video Technology 8 (5) (1998) 644–655.

[51] M. Tzelepi, A. Tefas, Relevance feedback in deep convolutional neural networks for content based image retrieval, in: Proceedings of the 9th Hellenic Conference on Artificial Intelligence, SETN '16, ACM, 2016, pp. 27:1–27:7.

[52] 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, IEEE, 2008, pp. 1–8.

[53] D. Nister, H. Stewenius, Scalable recognition with a vocabulary tree, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Vol. 2, IEEE, 2006, pp. 2161–2168.

[54] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: Convolutional architecture for fast feature embedding, in: Proceedings of the 22nd ACM international conference on Multimedia, ACM, 2014, pp. 675–678.

[55] K. He, X. Zhang, S. Ren, J. Sun, Delving deep into rectifiers: Surpassing human-level performance on imagenet classification, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1026–1034.

[56] D. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980.

[57] W.-L. Zhao, H. Jégou, G. Gravier, Oriented pooling for dense and non-dense rotation-invariant features, in: BMVC-24th British Machine Vision Conference, 2013.

[58] H. Jégou, F. Perronnin, M. Douze, J. Sanchez, P. Perez, C. Schmid, Aggregating local image descriptors into compact codes, IEEE Transactions on Pattern Analysis and Machine Intelligence 34 (9) (2012) 1704–1716.