Deep Convolutional Image Retrieval: A General Framework

Maria Tzelepi, Anastasios Tefas

Aristotle University of Thessaloniki, Department of Informatics

Abstract

In this paper a Convolutional Neural Network framework for Content Based Image Retrieval is proposed. We employ a deep CNN model to obtain the feature representations from the activations of the deepest layers and we retrain the network in order to produce more efficient image descriptors, 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 dataset are available, and the Relevance Feedback based Retraining, if feedback from users is available. We propose these approaches independently or in a pipeline, where each retraining approach operates as a pretraining step to the subsequent one. We also apply a query expansion method with spatial reranking on top of these approaches in order to boost the retrieval performance. The experimental evaluation on six 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.

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

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

1. Introduction

Information Retrieval (IR) refers to the process of obtaining material (text documents, images, audio etc.) that satisfies a certain information need from large databases [1]. Over the long history of IR, numerous works emerged in the field of text retrieval [2], audio [3], video [4], and image retrieval [5]. Image retrieval is a research area of IR of great scientific interest since 1970s. Earlier studies include manual annotation of images using keywords and searching by text [6]. Due to 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, in 1990s Content Based Image Retrieval (CBIR) has been proposed [7].

The objective of CBIR is to retrieve images that are relevant to a query image from a large collection based on their visual content [8]. A key issue concerning CBIR is to extract meaningful information from raw data in order to eliminate the so-called semantic-gap [9]. 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 [10], Vector of Locally Aggregated Descriptors (VLAD) [11] or combine bag-of-words models [12] with local descriptors such as Scale-Invariant Feature Transform (SIFT) [13].

Several recent studies introduce Deep Learning algorithms [14] against the shallow aforementioned approaches to a wide range of computer vision tasks, including image retrieval [15, 16, 17, 18]. 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), [19, 20], 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

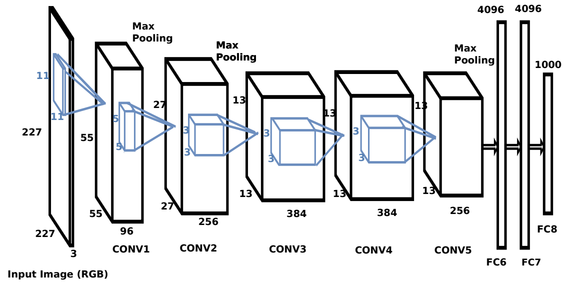

Figure 1: Overview of the CaffeNet Architecture

activations, followed by fully connected layers (an overview of the utilized network is provided in Fig. 1). 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 [21, 22], face recognition [23, 24], digit recognition [25, 26], pose estimation [27], object and pedestrian detection [28, 29], and action recognition [30]. 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, [31]. Inspired 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 usually drawn from the last layers, while several recent works are directed at utilizing the convolutional layers for the feature extraction. 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 approaches in order to enhance the deep CNN descriptors. We employ a pretrained model to derive feature representations from the activations of the deepest layers and we retrain the model, 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. This information can consist in only the dataset to be searched, the labels of the dataset or of a part of the dataset, and finally information acquired from users' feedback, that is, relevant or irrelevant images as deemed by multiple users.

In this paper we propose a general framework for CNN model retraining in the retrieval domain. The contributions of our study can be summarized as follows:

- To the best of our knowledge this is the first work that is able to exploit any kind of available information about the retrieval task. The proposed retraining approaches of our method can be categorized as follows:

  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.

- We deploy combinatory schemes, where all the above approaches can be employed in a pipeline. In this fashion each retraining approach operates as a pretraining step to the subsequent one.

- We suggest a query expansion technique with a spatial verification step applicable to all the above cases.
- This is the first approach that 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.

In Fig. 2 we schematically describe the proposed framework.

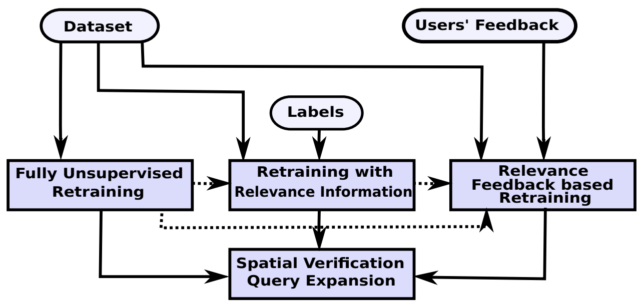

Figure 2: The proposed retraining approaches of our method based on the available information

The remainder of the manuscript is structured as follows. Section 2 discusses prior work. The proposed framework is described in detail in Section 3. The proposed spatial verification and query expansion technique is presented in Section 4. The experiments are provided in Section 5. Finally, conclusions are drawn in Section 6.

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 [32]. In [33] 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 [34, 35], CNN activations at multiple scale levels are combined with the VLAD representation. In [36], a feature aggregation pipeline is presented using sum pooling. while in [37] 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 [38]. In [39], a three-stream Siamense network is proposed to optimize the weights of the so-called R-MAC representation, proposed in [38], for the retrieval task, using a triplet ranking loss. The public Landmarks dataset, that is also used in [33], is utilized for the model training. In [40] a pipeline that uses the convolutional CNN-features and the bag-of-Words aggregation scheme is proposed. Finally, in [41], the bilinear CNN-based architectures [42] 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.

Subsequently, in [43] an online learning method to learn a similarity function between heterogeneous data modalities by preserving relative similarity constraints from two directions is proposed. In general, considerable research attention has been focused over the past few years on the cross-modal retrieval [44, 45], while another research direction in the retrieval domain, which has attracted intensive attention, concerns deep hashing-based techniques [46, 45, 47]. Under the hashing view, where the goal is to map the data points into a Hamming space of binary codes preserving the similarity in the original space, in [48] a novel unsupervised hashing approach is proposed by integrating feature aggregating and hash function learning into a joint optimization framework. In [45] an end-to-end deep learning framework which can perform feature learning and hash-code learning simultaneously is proposed. Finally, in [46] a two stage hashing framework for cross-modal retrieval tasks which can work in multiple settings like single label, multi-label, and both paired and unpaired scenario, while preserving the structure and semantic relationships that exists within the data is proposed. We should note that the proposed approach can be combined with deep hashing methods to increase the retrieval performance even more, which constitutes a main direction of our future work.

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 [49]. 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, [50], while a relevant idea recently successfully introduced in the cross-modal retrieval domain [51]. 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 [50]. 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. Finally, 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.

3. Proposed Method

In this paper we propose a CNN model retraining framework for CBIR, capable of exploiting any kind of available information. The core idea is to utilize the ability of a deep CNN to modify its internal structure, in order to produce better image representations for the retrieval task.

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 1,000 ImageNet classes,

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

[21]. 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 non-linearity ($f(x) = max(0, x)$) 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 1,000 ImageNet classes. The softmax loss is used during the training. An overview of the CaffeNet architecture is provided in Fig. 1.

We employ the CaffeNet model to directly extract feature representations from a certain hidden 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 layer utilized for the feature extraction, we build the target representations for each image, and subsequently we retrain the neural network.

Based on the available information our method suggests three basic retraining approaches: The FU retraining, if no information is available, the RRI, in the case that the labels of the dataset are available, and the RF, if feedback from users is available. Each of them can be applied independently or in a pipeline, where each approach operates as a pretraining step to the following retraining process. The three basic proposed retraining approaches are presented in detail in the following subsection.

3.1. Model Retraining Approaches

3.1.1. Fully Unsupervised Retraining

In the FU approach, we aim to amplify the primary retrieval presumption that the relevant images to a certain query are meant to be closer to the query 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 [52]. That is, we retrain the pretrained CNN model on the given dataset, aiming at minimizing the squared distance between each image representation and its n nearest representations. A schematic description is provided in Fig. 3.

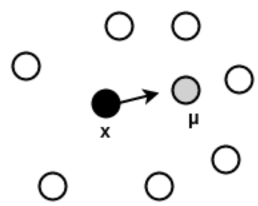

Figure 3: Schematic description of the Fully Unsupervised approach. ○ denote the neighbors of the sample $\boldsymbol{x}$, and $\boldsymbol{\mu}$ the mean vector of the nearest neighbors of $\boldsymbol{x}$

Let us denote by $\mathcal{I} = \{\mathbf{I}_i, i = 1, \ldots, N\}$ the set of N images to be searched, and by $\mathbf{x} = F_L(\mathbf{I})$ 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. We compute the mean vector of the n nearest representations to $\mathbf{x}_i$ and we denote it by $\boldsymbol{\mu}_i$. The new target representations for the images of $\mathcal{I}$ can be determined by solving the following optimization problem:

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

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

$$\begin{aligned} \frac{\partial \mathcal{J}}{\partial \mathbf{x}_i} &= \frac{\partial}{\partial \mathbf{x}_i} \left(\sum_{i=1}^{N} \|\mathbf{x}_i - \boldsymbol{\mu}_i\|_2^2 \right) \\ &= \frac{\partial}{\partial \mathbf{x}_i} \left((\mathbf{x}_i - \boldsymbol{\mu}_i)^\intercal (\mathbf{x}_i - \boldsymbol{\mu}_i) \right) \\ &= 2(\mathbf{x}_i - \boldsymbol{\mu}_i), \end{aligned} \tag{2}$$

Consequently, the update rule for the n-th iteration for each image can be formulated as:

$$\mathbf{x}_i^{(n+1)} = \mathbf{x}_i^{(n)} - 2\eta(\mathbf{x}_i^{(n)} - \boldsymbol{\mu}_i), \quad \mathbf{x}_i \in \mathcal{X} \tag{3}$$

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

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 modified model and obtaining the new representations.

3.1.2. Retraining with Relevance Information

Samples Provided 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 labeled 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 deepest neural layers of the CNN model used for the feature extraction, aiming to minimize the squared distance between $\mathbf{x}_i$ and the m nearest relevant representations, and simultaneously to maximize the squared distance between $\mathbf{x}_i$ and the n nearest irrelevant representations. A schematic description is provided in Fig. 4.

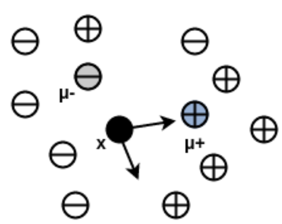

Figure 4: Schematic description of the Supervised approach. $\oplus$ denotes a relevant image to the sample $\boldsymbol{x}$, while $\ominus$ denotes an irrelevant one. We indicate the mean vector of relevant images to $\boldsymbol{x}$ by $\boldsymbol{\mu}_{+}$, and the mean vector of irrelevant ones as $\boldsymbol{\mu}_{-}$.

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 $C^i = \{\mathbf{c}_l, l = 1, \ldots, L^i\}$

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 n 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:

$$\min_{\mathbf{x}_i \in \mathcal{X}} \mathcal{J}^+ = \min_{\mathbf{x}_i \in \mathcal{X}} \sum_{i=1}^{N} \|\mathbf{x}_i - \boldsymbol{\mu}^i_+\|_2^2, \tag{4}$$

and

$$\max_{\mathbf{x}_i \in \mathcal{X}} \mathcal{J}^- = \max_{\mathbf{x}_i \in \mathcal{X}} \sum_{i=1}^{N} \|\mathbf{x}_i - \boldsymbol{\mu}^i_-\|_2^2. \tag{5}$$

We solve the above optimization problems using gradient descent.

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

$$\mathbf{x}_i^{(n+1)} = \mathbf{x}_i^{(n)} - 2\zeta(\mathbf{x}_i^{(n)} - \boldsymbol{\mu}^i_+), \quad \mathbf{x}_i \in \mathcal{X} \tag{6}$$

and

$$\mathbf{x}_i^{(n+1)} = \mathbf{x}_i^{(n)} + 2\beta(\mathbf{x}_i^{(n)} - \boldsymbol{\mu}^i_-), \quad \mathbf{x}_i \in \mathcal{X} \tag{7}$$

Consequently, the combinatory update rule, deriving by adding the equations (6) and (7) can be formulated as:

$$\mathbf{x}_i^{(n+1)} = \mathbf{x}_i^{(n)} - (1-\beta)(\mathbf{x}_i^{(n)} - \boldsymbol{\mu}^i_+) + \beta(\mathbf{x}_i^{(n)} - \boldsymbol{\mu}^i_-), \quad \mathbf{x}_i \in \mathcal{X} \tag{8}$$

where the parameter $\beta = 1 - \zeta, \in [0, 1]$ controls the desired distance both from relevant and irrelevant representations. Plainly, $\beta = 0$ sets as target representation for each image the mean vector of its m relevant representations, while as $\beta \rightarrow 1$ the new target representations are more affected by the irrelevant contribution.

Distractors. In the case where there are images in the dataset that do not belong to a certain class and serve as distractors in the retrieval, we can introduce them to the model retraining procedure. Thus, granted that the distractors are close to the training samples, that is, their representations are among the n aforementioned irrelevant representations of each image, we concurrently retrain the pretrained model so that the squared distance between the distractor representations and each certain image representation be maximized.

Denoting by $\mathcal{D} = \{\mathbf{d}_j, j = 1, \ldots, P\}$ the set of feature representations of the P distractors gathered from all the training images, our goal for each corresponding distractor can be formulated as follows:

$$\max_{\mathbf{d}_j \in \mathcal{D}} \mathcal{J} = \max_{\mathbf{d}_j \in \mathcal{D}} \sum_{j=1}^{P} \|\mathbf{d}_j - \mathbf{x}_i^j\|_2^2. \tag{9}$$

Consequently, following the gradient, the update rule for the n-th iteration for a distractor image can be formulated as:

$$\mathbf{d}_j^{(n+1)} = \mathbf{d}_j^{(n)} + 2\theta(\mathbf{d}_j^{(n)} - \mathbf{x}_i^j), \quad \mathbf{d}_j \in \mathcal{D} \tag{10}$$

where the parameter $\theta \in [0, 0.5]$ controls the desired distance from the certain image representation.

Thus, as in the previous approach, using the above target representations we retrain the neural network on the images provided with relevance information and on distractors (if any) using back-propagation.

3.1.3. 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 [53]. 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 bring the relevant images closer to the specific query and move away from it the irrelevant ones. Towards this end, we retrain the pretrained model by training on relevant and irrelevant images so that the corresponding relevant representations come closer in terms of Euclidean distance to the query representation, while the irrelevant ones move further away. We provide a schematic description in Fig. 5.

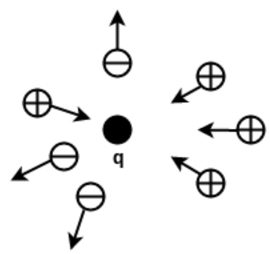

Figure 5: Schematic description of the Relevance Feedback based approach. ⊕ denote the relevant images to the query $\boldsymbol{q}$, while ⊖ denote the irrelevant ones, as they are given by the users

Let us denote by $\mathcal{Q} = \{\mathbf{Q}_k, k = 1, \ldots, K\}$ a set of queries, $\mathcal{I}_+^k = \{\mathbf{I}_i, i = 1, \ldots, N^k\}$ a set of relevant images to a certain query, by $\mathcal{I}_-^k = \{\mathbf{I}_j, j = 1, \ldots, M^k\}$ 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 $\mathcal{X}_+^k = \{\mathbf{x}_i, i = 1, \ldots, N^k\}$ the set of feature representations emerged in L layer of N images that have been qualified as relevant by a user, and by $\mathcal{X}_-^k = \{\mathbf{x}_j, j = 1, \ldots, M^k\}$ the set of M irrelevant feature representations. The new target representations for the relevant and irrelevant images can be respectively determined by solving the following optimization problems:

$$\min_{\mathbf{x}_i \in \mathcal{X}_+^k} \mathcal{J}^+ = \min_{\mathbf{x}_i \in \mathcal{X}_+^k} \sum_{i=1}^{N} \|\mathbf{x}_i - \mathbf{q}^k\|_2^2, \tag{11}$$

and

$$\max_{\mathbf{x}_j \in \mathcal{X}_-^k} \mathcal{J}^- = \max_{\mathbf{x}_j \in \mathcal{X}_-^k} \sum_{j=1}^{M} \|\mathbf{x}_j - \mathbf{q}^k\|_2^2. \tag{12}$$

We solve the above optimization problems using gradient descent. The update rules for the n-th iteration can be formulated as:

$$\mathbf{x}_i^{(n+1)} = \mathbf{x}_i^{(n)} - 2\alpha(\mathbf{x}_i^{(n)} - \mathbf{q}^k), \quad \mathbf{x}_i \in \mathcal{X}_+^k \tag{13}$$

and

$$\mathbf{x}_j^{(n+1)} = \mathbf{x}_j^{(n)} + 2\alpha(\mathbf{x}_j^{(n)} - \mathbf{q}^k), \quad \mathbf{x}_j \in \mathcal{X}_-^k \tag{14}$$

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

Similarly 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. We note that the above methodology can be implemented in iterative steps as well as in order to improve a certain's user information need, following the basic relevance feedback concept. More information and experiments are provided in Section 5.5.

3.2. Layer-wise training

The above approaches can be applied on several hidden layers. As mentioned before, several works utilize the fully connected layers [33, 32, 49, 34], since these layers are meant to capture high-level semantic information, while there are also works that utilize the convolutional layers exploiting the spatial information of these layers, using either sum-pooling techniques [36, 37] or max-pooling [38]. In our experiments we apply them on the 7th fully connected layer (FC7), and on the 6th fully connected layer (FC6). The dimension of both the FC6 and FC7 layers is 4096 features. Firstly, in the case of the FC7 layer, we employ the CaffeNet model and we adapt it by discarding the FC8 layer and by replacing the ReLU7 layer (that is the ReLU layer following the FC7 layer) with a PReLU layer, [54], which is initialized randomly, and then we retrain it using the appropriate target representations according to the retraining strategy for the CaffeNet's FC7 features. We note that we consider the responses after the ReLU layer. Since the first layers of CaffeNet trained on ImageNet learned more generic feature representations, all the convolutional layers remain unchanged, and we slightly update the FC6 layer using a small learning rate and the FC7 layer with a bigger learning rate, restricting the training cost. In the case of the FC6 modification, we remove the FC7 and FC8 layers, and we replace the ReLU6 layer with a PReLU layer which is initialized randomly, and then we retrain the FC6 layer using proper target representations from the FC6 activation layer of the CaffeNet model.

4. Spatial Verification and Query Expansion

Query Expansion is a standard, in most cases of negligible cost, technique for accomplishing better retrieval results [55]. The majority of CBIR methods include a query expansion step that boosts the retrieval performance. On top of the aforementioned approaches we also introduce a simple query expansion method by re-issuing the top ten retrieved corresponding image representations to the initial query as a new query representation, following the average query expansion scheme.

Let $\mathbf{Q}$ be a certain query, with a CNN representation $\mathbf{q}$. We consider the top ten retrieved images $\mathbf{R}_i, i = 1, \ldots, 10$ of $\mathbf{Q}$ and their corresponding CNN representations $\mathbf{x}_i, i = 1, \ldots, 10$. Then, the new query representation $\mathbf{q}_{new}$ is as follows:

$$\mathbf{q}_{new} = \frac{1}{11}(\mathbf{q} + \sum_{i=1}^{10} \mathbf{x}_i). \tag{15}$$

Furthermore, we suggest an additional spatial verification step as follows: We consider a shortlist of N top initially retrieved images for each query $\mathbf{Q}$, denoted as $\mathbf{R}_{i,0}, i = 1, \ldots, N$. Each of these images is cropped into nine equal-sized overlapping regions, $\mathbf{R}_{i,1}, \ldots, \mathbf{R}_{i,9}$. An example of the cropping approach is presented in Fig. 6. Subsequently, we extract the CNN features of the cropped images and we perform query to the dataset of $N \times 10$ images formed by both the initial image and the cropped ones. 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. That is, we rank the $N \times 10$ representations $\mathbf{x}_{i,l}$, $i = 1, \ldots, N, l = 0, \ldots, 9$ of the formed dataset in a list, and we perform query expansion considering the first ten unique corresponding full image representations of the aforementioned list, $\mathbf{x}_{i,0}$.

Figure 6: Spatial Verification and Query Expansion - An example of image cropping: The first retrieved image, $R_{1,0}$, for a certain query is cropped into 9 overlapping regions denoted as $R_{1,1}, \ldots, R_{1,9}$. The height and the width of each region are equal to the half-height and the half-width respectively of the full image

5. 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 we demonstrate the experimental results. Finally, we present the experiments on the proposed relevance feedback technique for a certain's user information need in iterative steps.

5.1. Evaluation Metrics

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

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

$$Recall = \frac{n.\ of\ Relevant\ Retrieved\ Images}{n.\ of\ Relevant\ Images} \tag{17}$$

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:

$$AP_i = \frac{1}{Q_i} \sum_{n=1}^{N} \frac{R_i^n}{n} t_n^i, \tag{18}$$

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.

5.2. Datasets

Inria Holidays [56]: 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. [33, 36]. We measure the retrieval performance in terms of mAP. Sample images are shown in Fig. 7.

Figure 7: Sample images of the Inria Holidays dataset

Paris 6k [57]: consists of 6,392 images (20 of the 6,412 provided images are corrupted) collected from Flickr by searching for particular Paris landmarks and provides 55 queries. Following the standard evaluation protocol we measure the retrieval performance in mAP. Like in most CNN-based works [32, 33, 49, 35, 36] 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. 8.

Figure 8: Sample images of the Paris 6k dataset

UKBench [58]: contains 10,200 images of objects divided into 2,550 classes. Each class consists of 4 images. All 10,200 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. 9.

Figure 9: 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 7,650 images and a set of 2,550 queries. As in UKBench, we use the top-3 score for the evaluation, which is a number between 0 and 3.

NUS-WIDE [59]: contains nearly 270,000 images collected from Flickr. It is a multi-label dataset in which each image is annotated with one or multiple concepts from 81 semantic concepts. However, we should note that NUS-WIDE provides links for downloading the images that are not valid, and thus there are differences with the datasets used in previous works. NUS-WIDE dataset is widely used for evaluating hashing techniques for image retrieval, where most of the works (e.g. [60, 61]) utilize the 21 most frequent concepts consisting of at least 5,000 images, and the supervised methods use 500 images per concept to form a training set of 10,500 images. In our work, we follow the setting of the 21 most frequent concepts, demanding each image to be associated with only one concept. Thus, we form a database of 40,000 images, with at least 81 images per concept. For each of the 21 concepts we randomly select 100 images, to build the test set of 2,100 queries. We measure the retrieval performance in terms of mAP for the entire used database of 40,000 images. We also report the mAP within the top 50 retrieved images. Finally, we use 40,000 additional images that do not belong to any concept and serve as distractors, to test the retrieval performance of our models in the formed database of 80,000 images. Samples are provided in Fig. 10.

Figure 10: Sample images of the NUS-WIDE dataset

CIFAR-10 [62]: contains 60,000 images of size 32×32, divided into 10 classes. Each class contains 6,000 images. Following other works, like [60], we use 50,000 images as the dataset to be searched, and we randomly select 1,000 images from the remaining 10,000 images to perform queries. The retrieval performance is measured in terms of mAP, for the entire dataset of 50,000 images. We also report the mAP within the top 50 retrieved images. Sample images are provided in Fig. 11.

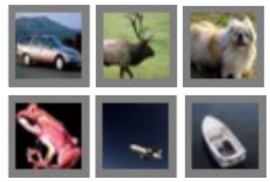

Figure 11: Sample images of the Cifar-10 dataset

5.3. Experimental Setup

The proposed method was implemented using the Caffe Deep Learning framework, [63]. In our work we use the adaptive moment estimation algorithm (Adam) [64], instead of the simple gradient descent for the network optimization since it is more stable, with the default parameters $\beta_1 = 0.9, \quad \beta_2 = 0.999, \quad \epsilon = 1e-08$, while the learning rate is set to $1e-05$. The batch size is set to 64, and the models are trained for 50 epochs. The models are trained on an NVIDIA GeForce GTX 1080 with 8GB of GPU memory. All results obtained using Euclidean distance. In the following we present the selected parameters for each of the proposed approaches.

5.3.1. Fully Unsupervised Retraining

In this set of experiments, we consider the 2 nearest representations of each image for the model retraining in all the used datasets, except for the Inria Holidays dataset, where we obtain the new target representations with respect to the 1 nearest representation. The parameter η in (3) is set to 0.5.

5.3.2. 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 parameter β in (8) is set to 0.2. In Paris 6k dataset, we retrain the network considering relevance information for images annotated either as good or as ok. Furthermore, for the Paris 6k dataset, where we

Table 1: Inria Holidays

	Scheme	Feature Representation	mAP
1	CaffeNet	CaffeNet ⇒ FC6	0.6184
2		CaffeNet ⇒ FC7	0.6988
3	FU	CaffeNet → FU(FC6; **FC6**) ⇒ FC6	0.6608
4		CaffeNet → FU(FC6,FC7; **FC7**) ⇒ FC7	0.7307
5	RRI	CaffeNet → RRI(FC6; **FC6**) ⇒ FC6	0.6649
6		CaffeNet → RRI(FC6,FC7; **FC7**) ⇒ FC7	0.74
7	RF	CaffeNet → RF(FC6; **FC6**) ⇒ FC6	0.7557
8		CaffeNet → RF(FC6,FC7; **FC7**) ⇒ FC7	0.7556
9	FU+RF	CaffeNet → FU(FC6,FC7; **FC7**) → RF(FC6,FC7; **FC7**) ⇒ FC7	0.7942
10	FU+RRI	CaffeNet → FU(FC6,FC7; **FC7**) → RRI(FC6,FC7; **FC7**) ⇒ FC7	0.7687
11	FU+RRI+RF	CaffeNet → FU(FC6,FC7; **FC7**) → RRI(FC6,FC7; **FC7**) → RF(FC6,FC7; **FC7**) ⇒ FC7	**0.8497**

utilize information deriving from the available distractors, we note that the number of the utilized distractors varies through datasets and employed approaches. Thus, in Paris dataset, we use information obtained from 846 distractor images for the model retraining, in the FC7 approach (6^{th} row of Table 2), while only 38 distractors used in the FC6 one (5^{th} row of Table 2). The parameter θ in (10) for the distractors target formulation is set to 0.5.

5.3.3. Relevance Feedback Based Retraining

In the experiments that conducted to validate the performance of the Relevance Feedback based approach, we consider for each of 500 different users 1 relevant and 5 irrelevant images for the Inria Holidays dataset, which forms a training set of 3,000 images. In Paris 6k dataset, 40 relevant and 20 irrelevant images are considered for each of 55 different users, while in UKBench-2 dataset we use 1 relevant and 1 irrelevant images for the 2,550 different users. In CIFAR-10 dataset we use 12 relevant and 1 irrelevant images for the 1,000 different users, while in NUS-WIDE we use 5 relevant and 1 irrelevant images for the 2,100 different users. The parameter α in (13), (14) is set to 0.5.

5.4. Experimental Results

The three proposed retraining approaches can be applied on several hidden layers. Several works utilize the fully connected layers [33, 32, 49, 34], while there

Table 2: Paris 6k

	Scheme	Feature Representation	mAP
1	CaffeNet	CaffeNet $\Rightarrow$ FC6	0.4621
2		CaffeNet $\Rightarrow$ FC7	0.5388
3	FU	CaffeNet $\rightarrow$ FU(FC6; **FC6**) $\Rightarrow$ FC6	0.6855
4		CaffeNet $\rightarrow$ FU(FC6,FC7; **FC7**) $\Rightarrow$ FC7	0.6984
5	RRI	CaffeNet $\rightarrow$ RRI(FC6; **FC6**) $\Rightarrow$ FC6	0.9794
6		CaffeNet $\rightarrow$ RRI(FC6,FC7; **FC7**) $\Rightarrow$ FC7	**0.9808**
7	RF	CaffeNet $\rightarrow$ RF(FC6; **FC6**) $\Rightarrow$ FC6	0.6418
8		CaffeNet $\rightarrow$ RF(FC6,FC7; **FC7**) $\Rightarrow$ FC7	0.6547
9	FU+RF	CaffeNet $\rightarrow$ FU(FC6,FC7; **FC7**) $\rightarrow$ RF(FC6,FC7; **FC7**) $\Rightarrow$ FC7	0.7714

Table 3: UKBench

	Scheme	Feature Representation	Score
1	CaffeNet	CaffeNet $\Rightarrow$ FC6	3.1308
2		CaffeNet $\Rightarrow$ FC7	3.3501
3	FU	CaffeNet $\rightarrow$ FU(FC6; **FC6**) $\Rightarrow$ FC6	3.48
4		CaffeNet $\rightarrow$ FU(FC6,FC7; **FC7**) $\Rightarrow$ FC7	3.5559
5	RRI	CaffeNet $\rightarrow$ RRI(FC6; **FC6**) $\Rightarrow$ FC6	**3.9927**
6		CaffeNet $\rightarrow$ RRI(FC6,FC7; **FC7**) $\Rightarrow$ FC7	3.9371

Table 4: UKBench-2

	Scheme	Feature Representation	Score
1	CaffeNet	CaffeNet ⇒ FC6	2.2086
2		CaffeNet ⇒ FC7	2.3996
3	FU	CaffeNet → FU(FC6; **FC6**) ⇒ FC6	2.4345
4		CaffeNet → FU(FC6,FC7; **FC7**) ⇒ FC7	2.5878
5	RRI	CaffeNet → RRI(FC6; **FC6**) ⇒ FC6	2.6996
6		CaffeNet → RRI(FC6,FC7; **FC7**) ⇒ FC7	2.7769
7	RF	CaffeNet → RF(FC6; **FC6**) ⇒ FC6	2.3400
8		CaffeNet → RF(FC6,FC7; **FC7**) ⇒ FC7	2.5020
9	FU+RRI	CaffeNet→ FU(FC6,FC7; **FC7**) → RRI(FC6,FC7; **FC7**) ⇒ FC7	**2.8251**
10	FU+RF	CaffeNet → FU(FC6,FC7; **FC7**) → RF(FC6,FC7; **FC7**) ⇒ FC7	2.6396

Table 5: NUS-WIDE 40k

	Scheme	Feature Representation	mAP	mAP@50
1	CaffeNet	CaffeNet ⇒ FC6	0.0962	0.1608
2		CaffeNet ⇒ FC7	0.1276	0.1734
3	FU	CaffeNet → FU(FC6; **FC6**) ⇒ FC6	0.114	0.247
4		CaffeNet → FU(FC6,FC7; **FC7**) ⇒ FC7	0.1606	0.326
5	RRI	CaffeNet → RRI(FC6; **FC6**) ⇒ FC6	0.1532	0.2806
6		CaffeNet → RRI(FC6,FC7; **FC7**) ⇒ FC7	0.2242	0.3521
7	RF	CaffeNet → RF(FC6; **FC6**) ⇒ FC6	0.1439	0.3537
8		CaffeNet → RF(FC6,FC7; **FC7**) ⇒ FC7	0.1856	0.4255
9	FU+RF	CaffeNet → FU(FC6,FC7; **FC7**) → RF(FC6,FC7; **FC7**) ⇒ FC7	0.2095	**0.4561**
10	FU+RRI	CaffeNet → FU(FC6,FC7; **FC7**) → RRI(FC6,FC7; **FC7**) ⇒ FC7	**0.2350**	0.3599

Table 6: NUS-WIDE 80k

	Scheme	Feature Representation	mAP	mAP@50
1	CaffeNet	CaffeNet ⇒ FC6	0.0539	0.1403
2		CaffeNet ⇒ FC7	0.0772	0.2155
3	FU	CaffeNet → FU(FC6; **FC6**) ⇒ FC6	0.064	0.1621
4		CaffeNet → FU(FC6,FC7; **FC7**) ⇒ FC7	0.094	0.2285
5	RRI	CaffeNet → RRI(FC6; **FC6**) ⇒ FC6	0.092	0.2005
6		CaffeNet → RRI(FC6,FC7; **FC7**) ⇒ FC7	0.1395	0.2529
7	RF	CaffeNet → RF(FC6; **FC6**) ⇒ FC6	0.1098	0.32
8		CaffeNet → RF(FC6,FC7; **FC7**) ⇒ FC7	0.1364	0.3777
9	FU+RF	CaffeNet → FU(FC6,FC7; **FC7**) → RF(FC6,FC7; **FC7**) ⇒ FC7	**0.1489**	**0.4017**
10	FU+RRI	CaffeNet → FU(FC6,FC7; **FC7**) → RRI(FC6,FC7; **FC7**) ⇒ FC7	0.1434	0.2627

Table 7: CIFAR-10 - CaffeNet Initialization

	Scheme	Feature Representation	mAP	mAP@50
1	CaffeNet	CaffeNet ⇒ FC6	0.2153	0.4613
2		CaffeNet ⇒ FC7	0.2533	0.5210
3	FU	CaffeNet → FU(FC6; **FC6**) ⇒ FC6	0.2423	0.4707
4		CaffeNet → FU(FC6,FC7; **FC7**) ⇒ FC7	0.2862	0.5393
5	RRI	CaffeNet → RRI(FC6; **FC6**) ⇒ FC6	0.332	0.5212
6		CaffeNet → RRI(FC6,FC7; **FC7**) ⇒ FC7	0.4297	0.5942
7	RF	CaffeNet → RF(FC6; **FC6**) ⇒ FC6	0.2444	0.5676
8		CaffeNet → RF(FC6,FC7; **FC7**) ⇒ FC7	0.2766	**0.6232**
9	FU+RRI	CaffeNet → FU(FC6,FC7; **FC7**) → RRI(FC6,FC7; **FC7**) ⇒ FC7	**0.4585**	0.6044

Table 8: CIFAR-10 - KevinNet Initialization

	Scheme	Feature Representation	mAP	mAP@50
1	KevinNet	KevinNet ⇒ FC6	0.2922	0.5988
2		KevinNet ⇒ FC7	0.6024	0.847
3	FU	CaffeNet → FU(FC6; **FC6**) ⇒ FC6	0.3756	0.6897
4		CaffeNet → FU(FC6,FC7; **FC7**) ⇒ FC7	0.6379	0.8466
5	RRI	KevinNet → RRI(FC6; **FC6**) ⇒ FC6	0.53	0.72
6		KevinNet → RRI(FC6,FC7; **FC7**) ⇒ FC7	0.6989	0.8519
7	RF	KevinNet → RF(FC6; **FC6**) ⇒ FC6	0.3882	0.76
8		KevinNet → RF(FC6,FC7; **FC7**) ⇒ FC7	0.6377	**0.8542**
9	FU+RRI	CaffeNet → FU(FC6,FC7; **FC7**) → RRI(FC6,FC7; **FC7**) ⇒ FC7	**0.7285**	0.8532

are also works that utilize the convolutional layers with pooling techniques to produce the image descriptors [36, 37, 38]. In our work we use the fully-connected layers, since these layers are meant to capture high level semantic information. Thus, experiments conducted for each of the three proposed retraining approaches on the FC6 and the FC7 layers (we did not considered the FC8 layer, since it is a distribution over the 1,000 ImageNet class labels). We note that by utilizing the FC7 layer, we produce richer descriptors which can lead to better retrieval performance, since the FC7 layer captures higher level concepts as compared to the FC6 layer, however it comes with additional computational cost. More information can be found in the section 5.4.1 that discusses the computational cost. Furthermore, based on the available information, the retraining approaches can be applied in a pipeline. In this fashion, each retraining approach operates as a pretraining step to the subsequent one. For example, the Fully Unsupervised approach can be applied as pretraining step to both the Retraining with Relevance Information and the Relevance Feedback based approaches, since it requires no additional information except for the dataset itself. Therefore, we have conducted indicative experiments building combinatory retraining schemes, investigating the assumption that the combinatory schemes can improve the single-step training approaches.

In the following we denote by FC6 and FC7 the feature representations obtained from the FC6 and FC7 layer of the CNN model respectively. We also abbreviate the applied query expansion technique to QE, and the spatial verification and query expansion to SVQE. Finally, we denote by FU($L_1, L_2, ...; \boldsymbol{L_T}$) the fully unsupervised retraining on the layers $L_1, L_2, ...$ with target representations obtained from the $\boldsymbol{L_T}$ layer, by RRI($L_1, L_2, ...; \boldsymbol{L_T}$) the retraining with relevance information on the layers $L_1, L_2, ...$ with target representations obtained from the $\boldsymbol{L_T}$ layer, and correspondingly by RF($L_1, L_2, ...; \boldsymbol{L_T}$) 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* $\Rightarrow$ *FC7* implies that we obtain the FC7 representations directly from the CaffeNet model and we use them for the retrieval procedure, while $\textit{CaffeNet} \rightarrow \textit{RRI(FC6,FC7; }\boldsymbol{FC7}\textit{)} \Rightarrow \textit{FC7}$ denotes that we formulate the target representations using the features emerged in the FC7 CaffeNet layer and we retrain both the FC6 and FC7 layers of the CaffeNet, then we extract the FC7 representations of the modified model, and we use them for the retrieval.

Tables 1 - 8 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. We also see that the other proposed methodologies applied on the modified via the FU approach model yield better retrieval results, as compared to the CaffeNet's employment, in any considered case. In some cases this sequential strategy can lead to outstanding performance, as in the UKBench-2 dataset, where the refined with relevance information model on the fully unsupervised model outperforms any other approach. Hence, we mainly suggest the FU retraining as a pretraining step that can be utilized to boost the performance of the other retraining approaches. Additionally, we observe that the modified in RRI fashion descriptors enhance significantly the baseline results of CaffeNet in all the datasets on both the FC6 and FC7 approaches, and in Paris 6k dataset we can accomplish state-of-the-art performance by a single training step. Furthermore it is shown that by applying the proposed approaches in pipelines we can achieve outstanding performance. In the Inria Holidays dataset (Table 1), we notice that the RF approach is more effective than both the FU and RRI. This is reasonable since the training set of the RF approach (consisting of 3,000 images) is considerably larger than the one of the other two approaches (consisting of 991 images). Furthermore, we can observe in the case of the UKBench-2 dataset (Table 3) that he 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 (5,100 against 10,200 images).

Regarding the NUS-WIDE dataset, we first examine the impact of the number of training samples to the retrieval performance. That is, we apply our Fully Unsupervised retraining approach using 2,000, 5,000, 13,000 26,000 and 40,000 (that is the entire database) training samples. The experimental results are illustrated in Fig. 12 . As it is shown, the obtained mAP utilizing 13,000 images is 0.1606, while it reaches up to 0.1678 utilizing the entire database. Thus, since the number of training samples also comes with computational cost, a good compromise is to set the number of training samples to 13,000 images. Therefore, in the following, we utilize 13,000 images from the database to train the proposed models, and we use the entire database in the retrieval stage, for the evaluation. Furthermore, we test the performance each of the proposed approaches in the extended version of the dataset with 80,000 images, where we use 40,000 additional images, that do not belong to any concept and serve as distractors. In Table 5 we illustrate the experimental results on the NUS-WIDE dataset of 40,000 images, and in Table 6 we illustrate the experimental results on the NUS-WIDE dataset of 80,000 images. As we can observe in both cases each of the proposed approaches improves notably the baseline results. The RRI approach applied on the FC7 layer achieves the best performance in the single-step retraining, improving the baseline CaffeNet's results by 10 mAP points. Regarding the RF approach, we see that the improvement is more significant for the top-50 retrieved images, achieving outstanding performance against the other retraining strategies. This is reasonable since we use the top 6 retrieved images for each of the queries to build the dataset for the model retraining, and hence we expect to better improve the top retrieved images. Concerning the combinatory schemes, as noticed in the other datasets, we observe that we can achieve enhanced performance, as compared to the single-step retraining, by applying the retraining approaches in a pipeline.

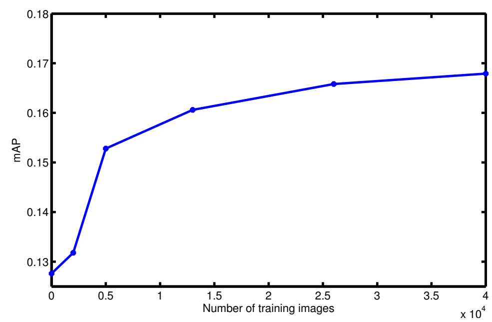

Figure 12: mAP using the FU retraining approach, for various numbers of training samples

In the case of the CIFAR-10 dataset apart from the CaffeNet pretrained model, we also use the KevinNet model [47], which is trained on the CIFAR-10 dataset for producing binary hash codes for the retrieval task, validating our claim that the proposed method is applicable to various model architectures, as well as to finetuned weights for different tasks. We use either the FC7 or the FC6 representations, in order to maintain the computational complexity of the proposed approach, however we could also use the representations produced by the subsequent encoding layer, the so called fc8 kevin encode layer. We also tested the performance of the proposed RRI approach in the aforementioned layer, achieving a considerable improvement from 0.7907 to 0.8369 in terms of mAP. This also confirms that the proposed approach can be applied in combination with other approaches for image retrieval. Furthermore, following this direction, we also evaluated the hashing codes produced by the RRI optimized fc8 kevin encode layer, and we report a significant improvement from 0.7863 to 0.8466 in terms of mAP. As it is shown in Tables 7 and 8 the KevinNet model achieves notably better baseline results, which is reasonable since it is finetuned on CIFAR-10 dataset. Furthermore, the observations drawn in the previous datasets, are also confirmed in CIFAR-10. That is, each of the proposed approaches improve the baseline results of CaffeNet and KevinNet correspondingly. Additionally, concerning the RF approach, similarly to the NUS-WIDE dataset, we see that achieves significantly better results for the top 50 retrieved images, as expected since we use the top 13 retrieved images of each query, and thus the top retrieved images are better

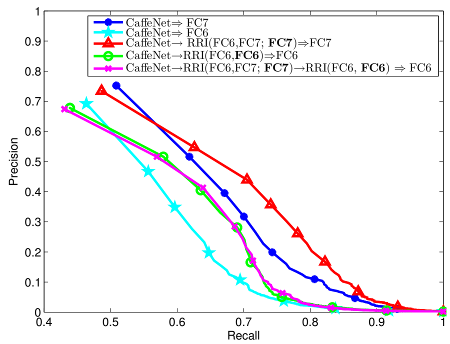

Figure 13: Inria Holidays: Precision-Recall curves of RRI

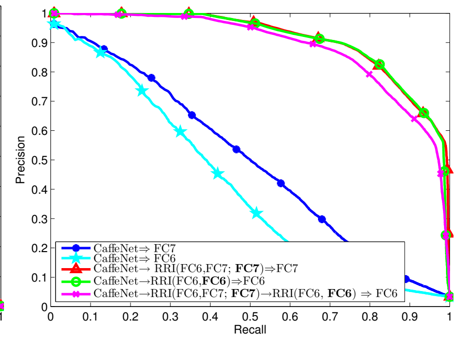

Figure 14: Paris 6k: Precision-Recall curves of RRI

improved. Finally, we can observe that the proposed approaches on the KevinNet initialization, while improve notably the mAP over all the dataset, achieve comparatively poorer improvement over the top 50 retrieved images in the case of the FC7 representations. This is attributed to the fact that the optimized weights for the binary hashing retrieval task, achieve already enhanced performance, and thus the proposed method can slightly boost them. On the contrary, we observe that we can achieve more significant improvement for the 50 retrieved images on the CaffeNet initialization.

In Fig. 13, 14 we provide the Precision-Recall curves of the considered approaches of the RRI scheme for the Iniria Holidays and Paris 6k datasets respectively. In both the datasets, the FC7 modification yields better performance.

In Fig. 15, 16 we illustrate the the Precision-Recall curves for the combinatory schemes on Inria Holidays, and Paris 6k datasets respectively. It is shown that we can indeed achieve significantly enhanced results by applying our retraining approaches in a pipeline as compared to the independent ones.

In Fig. 17, 18, 19 we provide some examples of the top retrieved images for certain queries, using the baseline CaffeNet's features and features obtained from our retrained models, in Paris 6k, Inria Holidays and UKBench-2 datasets, respectively.

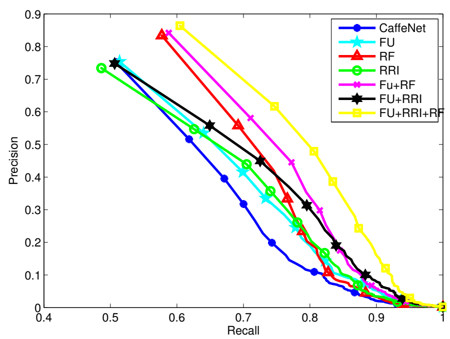

Figure 15: Inria Holidays: Combinatory Schemes

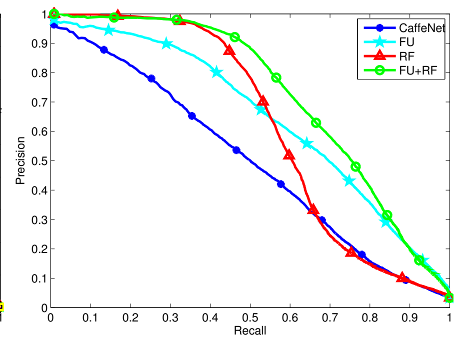

Figure 16: Paris 6k: Combinatory Schemes

Figure 17: Paris: The query image is the first one of the top row and the images that follow in the top row are the first 6 retrieved using the baseline FC7 representation. The top 6 retrieved images using the RRI approach on the FC7 layer are shown in the second row for the same query

Figure 18: Inria Holidays: The query image is the first one of the top row and the images that follow in the top row are the first 5 retrieved using the baseline FC7 representation. The top 5 retrieved images using the RRI approach on the FC7 layer are shown in the second row for the same query

Figure 19: UKBench-2: 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 FC6 representation. The top 3 retrieved images using the RRI approach on the FC6 layer are shown in the second row for the same query

5.4.1. Computational Cost

The proposed method requires a CNN pretrained model, ideally trained on the ImageNet dataset composed of 1.2 million images divided into 1,000 classes, since it produces a rich description of the physical world. Training such a model, depending on the available GPUs, requires roughly a few days. However, a common practice in CNN-based works in the retrieval domain is to utilize a pretrained CNN model, and hence in our work, as previously stated, we utilize the CaffeNet model. Subsequently, applying each of the proposed methods, requires a certain training time. Once the models are trained based on the available information, no additional time is required for the retrieval procedure. That is, the testing complexity is exactly the same as the baseline models (e.g. CaffeNet, or KevinNet). Regarding the training time, the experiments conducted on an NVIDIA GeForce GTX 1080 with 8GB of GPU memory, where the average backward pass time for an input image of the fixed size of 227×227 is 3.32 ms, while the forward pass takes 2.73 ms, for the model which produces output at the FC7 layer, and correspondingly, the average backward pass time is 2.53 ms, and the forward one is 1.97 ms, for the model which produces output at the FC6 layer. Furthermore we also performed experiments on an NVIDIA GeForce GTX 1060 with 6GB of GPU memory, as well as on an NVIDIA Quadro K4000 with 3GB of GPU memory, to measure the training time. The results are illustrated in Table 9.

Table 9: Training time for an input image on various GPUs (FC7 model)

GPU	Backward Pass	Forward Pass
NVIDIA GeForce GTX 1080	3.32 ms	2.73 ms
NVIDIA GeForce GTX 1060	5.86 ms	2.79 ms
NVIDIA Quadro K4000	13.23 ms	9.07 ms

In order to improve the deploy speed of the proposed models, we utilize the NVIDIA TensorRT[2] tool. TensorRT is a high-performance learning inference library, which automatically optimizes trained neural networks for run-time performance. Thus, using TensorRT we achieve a significant speed up in both the proposed model architectures. That is, for the FC7 model the forward pass takes 1.43 ms, while for the FC6 model it takes 1.18 ms.

5.4.2. Impact of the probabilistic factors

In this work, we propose a model retraining framework, which is overall able to exploit any kind of available information. The core idea is that we utilize a pretrained CNN model, in order to derive the feature representations of a deep layer, and we retrain the weights of the model, 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. Hence, the utilization of fixed weights as the model initialization for the retraining task, leads to deterministic results, in the retrieval performance. In this section, we investigate the impact of the probabilistic factors in the performance of the proposed method. That is, the ordering of input data and the different test images to perform queries. We choose the CIFAR-10 dataset, to explore the impact of the aforementioned factors, since the test set of the rest of the datasets is fixed, not allowing to straightforwardly perform queries with different images. We use as weights initialization the CaffeNet model. Thus, we repeat each of the FU, RRI,

[2] `https://developer.nvidia.com/tensorrt`

and RF experiments 5 times, using different random shuffling of input images, and we evaluate the retrieval performance of the corresponding retrained models on the FC7 layer, using 5-fold cross validation on 5 different test sets of 1,000 queries, randomly selected from the provided test set of 10,000 images. We also compute the mAP for the 5 five different test sets using the CaffeNet model. The experimental results for the mean value and the standard deviation of the mAP for the five runs are illustrated in Table 10. It is evident that the probabilistic factors do no affect the results significantly, giving quite stable performance among the runs.

Table 10: CIFAR-10: 5-fold Cross Validation

Retraining method	**mAP**
CaffeNet	0.2553 ± 0.0051
FU	0.2836 ± 0.0057
RRI	0.4379 ± 0.0087
RF	0.2759 ± 0.0036

In Table 11 we provide the experimental evaluation of our spatial verification and query expansion technique on the best approach of each dataset. In the UKBench-2 dataset we use 100 queries. From the demonstrated results we can notice that indeed the query expansion improves the retrieval results, while the spatial reranking step slightly boosts the initial performance. This is reasonable, since the spatial verification is more useful on the region based image retrieval, where we perform queries with a specified region of interest. To this aim, we employ the cropped-queries versions of Paris 6k dataset, and we apply our RRI method on the initial CaffeNet's features. The baseline mAP is 0.5345 for Paris 6k dataset. We note that in this version, the corresponding full images of the cropped queries are included in the search set. Subsequently, we apply our spatial verification and query expansion approach on the modified representations. Table 12 illustrates the experimental results.

Table 11: mAP & Score - Spatial Verification & Query Expansion on our best approaches

	Paris 6k	**UKBench-2**
Best Result	0.9808	2.80
QE	0.9915	2.82
SVQE	0.9916	2.85

Table 12: mAP - Spatial Verification & Query Expansion on Refined with Relevance Information FC7 approach

	Paris 6k Cropped
Initial Result	0.9742
QE	0.9890
SVQE	0.9932

Finally, in Table 13 we compare our method against other CNN-based, as well as hand-crafted feature-based methods, on image retrieval. MAP measures the retrieval performance in the Inria Holidays and Paris 6k datasets, while top-4 score is used in the case of the UKBench dataset. Since the proposed RF approach is novel, and the competitive methods do not utilize information derived from users' feedback, the RF results are reported only in Tables 1-8 and we do not include them in the comparisons. Methods marked with * use the cropped queries in Paris 6k dataset. To the best of our knowledge, the proposed approach outperforms every other competitive method, in two out of three datasets. We should note that in Inria Holidays dataset, we can accomplish competitive results only with the RF approach and the combinatory retraining schemes. This is attributed to the nature of the dataset. The Inria Holidays dataset is composed of 991 train images belonging to 500 classes, with each class consisting of between 1 to 12 images. That is, we have less than 2 images per class, on average. Therefore, since a key factor of success of the RRI approach is the number of relevant images per class for the new targets formulation in the retraining process, Inria dataset can not benefit from it. Furthermore, the number of the train images constitutes in general an important factor in the deep CNN learning. Thus, the train dataset, consisting of 991 images, in both FU and RRI approaches, is small for achieving competitive results against other methods. On the contrary, as we can observe in Table 1, the RF retraining approach, which builds a dataset of 3,000 images, outperforms the aforementioned proposed approaches, and we can achieve competitive results to the state-of-the-art methods by applying combinatory retraining strategies.

The trained models of the proposed framework are available at: `https://github.com/mtzelepi/framework`

Table 13: Comparison against other methods

Method	**Inria Holidays**	**Paris 6k**	**UKBench**
CVLAD* [65]	0.827	-	3.62
VLAD* [11]	0.653	-	-
T-embedding* [66]	0.781	-	-
BOW 200k-D* [67]	0.54	0.46	2.81
Neural Codes [33]	0.793	-	3.56
CNNaug-ss [32]	0.843	0.795	3.644
ReDSL.FC1 [49]	-	0.9474	-
Spoc [36]	0.808	-	3.65
CNN-VLAD [35]	0.84	0.694	-
CRB-CNN-16 [41]	0.854	-	3.56
Deep Image Retrieval [39]	0.907	0.912	-
Deep Image Retrieval & QE [39]	-	0.938	-
R-MAC* [38]	-	0.83	-
R-MAC & QE [38]	-	0.865	-
CroW* [37]	0.849	0.796	-
CroW & QE [37]	-	0.83	-
Ours	0.74	**0.9808**	**3.9927**
Ours & QE	-	**0.9916**	-

5.5. Relevance Feedback

As mentioned before, the relevance feedback based retraining approach can be materialized in order to improve a certain's user information need in iterative steps [68]. Thus, in each feedback round, the user marks either as relevant or as irrelevant the retrieved images, and the system uses this information to retrain the CNN model according to the methodology described in Section 3.1.3 for multiple users. The relevance feedback procedure is integrated by feeding the images of the given dataset and the query image into the input layer of the modified model and obtaining the new representations. The above process is performed in each relevance feedback round, by initializing the CNN model with the parameters of the previous round and retraining on the new set of relevant and irrelevant images with their corresponding updated targets. Given a new query from the user, the system executes the procedure from the beginning.

In order to evaluate the proposed approach, we perform experiments on the Inria Holidays dataset. We obtain the FC7 representations from the CaffeNet model. We consider as search set 991 images and we perform 100 queries from the residue. Each class in the search set consists of between 1 and 12 images. We execute 3 relevance feedback rounds for each query. At each relevance feedback round we use 3 relevant and 3 irrelevant images for the model retraining. The model is trained for 2 epochs at each relevance feedback round. We measure the performance at each feedback round in terms of Precision, and we compute the average precision obtained over all the performed queries. Average precision is measured for the top 3 retrieved images. Experimental results are illustrated in Fig. 20. We can observe that the proposed methodology improves the retrieval performance by the first feedback round.

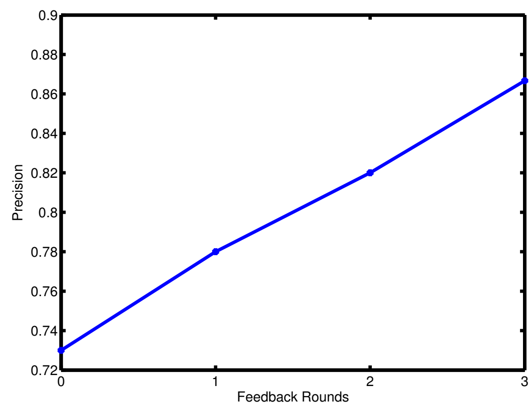

Figure 20: Inria Holidays: Relevance Feedback

6. Conclusions

In this paper we proposed a model retraining framework for enhancing deep CNN representations in the retrieval domain. The proposed method is able to exploit any kind of 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 feature representations and build the target representations according to each approach, and then we retrain appropriately the network's weights. We also proposed combinatory retraining strategies, where each of the retraining approaches can be utilized as a pretraining step in order to boost the performance of the following one. A query expansion technique with a spatial verification step applied on top of the best stated approaches provides further boosting of the retrieval performance. We should note that all the proposed approaches are applicable to any other CNN-based works for image retrieval that utilize a CNN model to directly extract feature representations. Experimental results indicate the effectiveness of our method, performing 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] G. Salton, C. Buckley, Term-weighting approaches in automatic text retrieval, Information processing & management 24 (5) (1988) 513–523.

[3] G. Guo, S. Z. Li, Content-based audio classification and retrieval by support vector machines, IEEE Transactions on Neural Networks 14 (1) (2003) 209–215.

[4] Y.-G. Jiang, C.-W. Ngo, J. Yang, Towards optimal bag-of-features for object categorization and semantic video retrieval, in: Proceedings of the 6th ACM International Conference on Image and Video Retrieval, ACM, 2007, pp. 494–501.

[5] Y. Rui, T. S. Huang, S.-F. Chang, Image retrieval: Current techniques, promising directions, and open issues, Journal of visual communication and image representation 10 (1) (1999) 39–62.

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

[7] 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.

[8] 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.

[9] 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.

[10] 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.

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

[12] G. Csurka, C. Dance, L. Fan, J. Willamowski, C. Bray, Visual categorization with bags of keypoints, in: Workshop on statistical learning in computer vision, European Conference on Computer Vision (ECCV), Vol. 1, Prague, 2004, pp. 1–2.

[13] 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.

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

[15] 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.

[16] 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.

[17] 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.

[18] Y. Wei, Y. Zhao, C. Lu, S. Wei, L. Liu, Z. Zhu, S. Yan, Cross-modal retrieval with cnn visual features: A new baseline, IEEE transactions on cybernetics 47 (2) (2017) 449–460.

[19] 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.

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

[21] 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.

[22] 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.

[23] 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.

[24] D. Wang, J. Yang, J. Deng, Q. Liu, Facehunter: A multi-task convolutional neural network based face detector, Signal Processing: Image Communication 47 (2016) 476–481.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

[29] D. Tomè, F. Monti, L. Baroffio, L. Bondi, M. Tagliasacchi, S. Tubaro, Deep convolutional neural networks for pedestrian detection, Signal Processing: Image Communication 47 (2016) 482–489.

[30] S. Yan, J. S. Smith, B. Zhang, Action recognition from still images based on deep vlad spatial pyramids, Signal Processing: Image Communication 54 (2017) 118–129.

[31] 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.

[32] 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.

[33] 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.

[34] 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.

[35] 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.

[36] 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.

[37] 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.

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

[39] 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.

[40] 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.

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

[42] 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.

[43] Y. Wu, S. Wang, Q. Huang, Online asymmetric similarity learning for cross-modal retrieval, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4269–4278.

[44] K. Chen, T. Bui, F. Chen, Z. Wang, R. Nevatia, Amc: Attention guided multi-modal correlation learning for image search, arXiv preprint arXiv:1704.00763.

[45] Q.-Y. Jiang, W.-J. Li, Deep cross-modal hashing.

[46] D. Mandal, K. N. Chaudhury, S. Biswas, Generalized semantic preserving hashing for n-label cross-modal retrieval, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4076–4084.

[47] K. Lin, H.-F. Yang, J.-H. Hsiao, C.-S. Chen, Deep learning of binary hash codes for fast image retrieval, in: Proceedings of the IEEE conference on computer vision and pattern recognition workshops, 2015, pp. 27–35.

[48] T.-T. Do, D.-K. L. Tan, T. T. Pham, N.-M. Cheung, Simultaneous feature aggregating and hashing for large-scale image search, arXiv preprint arXiv:1704.00860.

[49] 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.

[50] 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.

[51] 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.

[52] 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.

[53] 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.

[54] 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.

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

[56] H. Jégou, M. Douze, C. Schmid, Hamming embedding and weak geometric consistency for large scale image search, in: A. Z. David Forsyth, Philip Torr (Ed.), European Conference on Computer Vision (ECCV), Vol. I of LNCS, Springer, 2008, pp. 304–317.

[57] 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.

[58] 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.

[59] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, Y. Zheng, Nus-wide: a real-world web image database from national university of singapore, in: Proceedings of the ACM international conference on image and video retrieval, ACM, 2009, p. 48.

[60] H. Lai, Y. Pan, Y. Liu, S. Yan, Simultaneous feature learning and hash coding with deep neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3270–3278.

[61] W.-J. Li, S. Wang, W.-C. Kang, Feature learning based deep supervised hashing with pairwise labels, arXiv preprint arXiv:1511.03855.

[62] A. Krizhevsky, G. Hinton, Learning multiple layers of features from tiny images.

[63] 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.

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

[65] 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.

[66] H. Jégou, A. Zisserman, Triangulation embedding and democratic aggregation for image search, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2014, pp. 3310–3317.

[67] 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.

[68] 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.