Neurocomputing 257 (2017) 5–15

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Deep binary codes for large scale image retrieval

Song Wu[a], Ard Oerlemans[b], Erwin M. Bakker[a], Michael S. Lew[a,*]

[a] LIACS Media Lab, Leiden University, Niels Bohrweg 1, Leiden, The Netherlands
[b] VDG Security BV, Zoetermeer, The Netherlands

ARTICLE INFO

Article history:
Received 13 July 2016
Revised 5 December 2016
Accepted 18 December 2016
Available online 6 February 2017

Keywords:
Convolutional neural network
Deep binary codes
Late fusion
Large scale image search

ABSTRACT

Recent studies have shown that image representations built upon deep convolutional layers in Convolutional Neural Networks (CNNs) have strong discriminative characteristics. In this paper, we present a novel and effective method to create compact binary codes (deep binary codes) based on deep convolutional features for image retrieval. Deep binary codes are generated by comparing the response from each feature map and the average response across all the feature maps on the deep convolutional layers. Additionally, a spatial cross-summing strategy is proposed to directly generate bit-scalable binary codes. As the deep binary codes on different deep layers can be obtained by passing the image through the CNN and each of them makes a different contribution to the search accuracy, we then present a dynamic, on-the-fly late fusion approach where the top N high quality search scores from deep binary codes are automatically determined online and fused to further enhance the retrieval precision. Two strengths of the proposed methods are that the generation of deep binary codes is based on a generic model, which does not require additional training for new image domains, and that the dynamic late fusion scheme is query adaptive. Extensive experimental results on well known benchmarks show that the performance of deep binary codes are competitive with state-of-the-art approaches for large scale image retrieval. Moreover, it is shown that the dynamic late fusion scheme significantly enhances the search accuracy.

1. Introduction

Content-based image retrieval aims to find relevant images in an image database that share a similar appearance with a given query image. This is a challenging task for large scale visual search, because one must address both the typical appearance transformations such as changes in perspective, rotation and scale; and also minimize memory, computational cost, and response time.

Traditional image retrieval systems based on visual word representations mainly owe their success to locally invariant features and large visual codebooks. The Bag-of-Words (BoW) [1–3] approach is usually employed to encode local features into a histogram according to the occurrence frequency of each visual word. Peronnin et al. proposed the Fisher Vector [4]. The visual words in a Fisher Vector are constructed with a Gaussian mixture model (GMM) where the gradients of local features corresponding to particular parameters in GMM are summed. The Fisher Vector image representation is the concatenation of each accumulated gradient. Jegou et al. proposed a Vector of Locally Aggregated Descriptors (VLAD) [5] to capture more information from the image. VLAD and its variations [6–8] are viewed as a type of simplified Fisher Vector, and it accumulates the difference of each local feature to the visual words and concatenates these accumulated values to describe an image.

Visual word based approaches are challenging to scale to very large image databases, as they have significant computational and memory requirements. Hashing techniques, such as iterative quantization (ITQ) [9], locality-sensitive hashing (LSH) [10], spectral hashing (SH) [11], spherical hashing (SpH) [12], locality-sensitive hashing from shift-invariant kernels (SKLSH) [13], density sensitive hashing (DSH) [14] as well as PCA-random rotation (PCA-RR) [9] focus on learning compact yet powerful image representations for efficient large scale visual search. The basic idea of hashing-based approaches is to construct a hash function to map each visual object into a binary string code such that similar visual objects are mapped into similar binary codes. Unlike the above mentioned hashing approaches, which seek a linear function to project data into a binary vector, recent supervised hashing methods based on convolutional neural network (CNN) architectures [15,16] seek to learn multiple hierarchical non-linear transformations to generate distinctive binary codes. However, most state-of-the-art hash function learning methods require additional training for each new image domain. This can require significant resources

* Corresponding author.
E-mail addresses: s.wu@liacs.leidenuniv.nl, wusongbeckham@gmail.com (S. Wu), a.oerlemans@vdgsecurity.com (A. Oerlemans), e.m.bakker@liacs.leidenuniv.nl (E.M. Bakker), m.s.lew@liacs.leidenuniv.nl (M.S. Lew).

http://dx.doi.org/10.1016/j.neucom.2016.12.070
0925-2312/

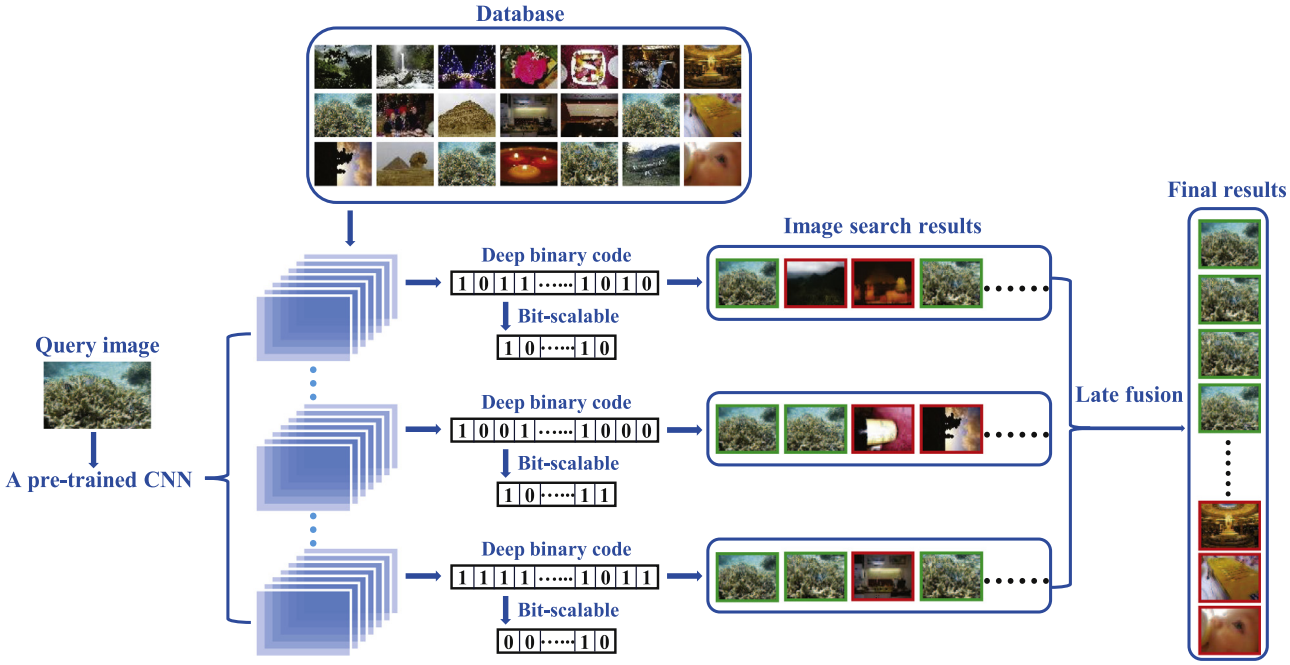

Fig. 1. The proposed image retrieval framework. Our method consists of two main components. The first is the deep binary code generation on each deep convolutional layer of a pre-trained CNN. In the second component, we propose a dynamic late fusion scheme to further increase the search precision. Images with green rectangles are positive results. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

both for assembling the supervised training data and the learning process.

In recent years, deep learning models, particularly deep convolutional neural networks (CNNs), have achieved great success in various computer vision applications [17], such as visual tracking [18], pose estimation [19–22], image segmentation [23–27], person verification [28], image classification [29–32], annotation [33], object detection [34–36] and visual search [37–43]. The semantic image segmentation can be referred to as a problem of CNN based pixel-level classification or labeling. The applications of human pose estimation can be formulated as a CNN based regression problem toward face and body landmarks. The high performance of image classification, object detection and visual search is mainly due to the high-level and powerful representation learning capability of CNN. The CNN based image representation makes use of the transfer property of a CNN architecture that is pre-trained on a large scale dataset. It has been shown to provide a highly discriminative descriptor representing an image and to produce superior performance. Most of these research projects utilize the outputs from the fully connected layers to represent images (directly used, followed by normalization or followed by PCA reduction [41]).

In particular, visual representations from activations of deep convolutional layers have been shown to lead to high accuracy for image retrieval in real world image test sets. This is achieved by processing a max-pooling, spatial max-pooling [42,44] or sum-pooling [43] operation on the deep convolutional layers. Better performance is obtained using deep convolutional features than if the features from the fully connected layers are used. These features have very useful properties: first, they can be efficiently extracted from an image of any size and aspect ratio. Second, features from the convolutional layers have a natural interpretation as descriptors of local image regions corresponding to receptive fields of the particular features. Finally, simple pooling operations can aggregate feature maps from deep convolutional layers into low dimensional and highly distinctive features. Inspired by the advantages of image representation through aggregating activations from deep convolutional layers, we propose a novel and efficient approach to construct bit-scalable binary codes from deep convolutional layers for highly efficient image retrieval (as shown in Fig. 1). This idea is mainly based on the fact that similar visual objects have similar distributions of responses of feature maps on deep convolutional layers. In this work, we propose to generate the binary code on each convolutional layer according to the comparison between the response of each feature map and the average response across all the feature maps on the same deep convolutional layer. Additionally, a strategy of spatial cross-summing is designed to generate bit-scalable deep binary codes. Extensive experiments on well-known image retrieval benchmarks demonstrate the effectiveness of the proposed binary code representation (referred to as deep binary codes) and show competitive results compared to state-of-the-art image retrieval approaches.

The strengths of the proposed deep binary codes are three-fold. First, the deep binary code is highly efficient regarding computational and memory costs. By passing a test image through a pre-trained CNN architecture, the compact binary codes on each deep convolutional layer can easily be generated. Second, the length of a deep binary code can be controlled by the spatial cross-summing operation. Third, available pre-trained CNN architectures (VGGNet [32], AlexNet [29] as well as GoogleNet [45]) can be directly employed to generate deep binary codes.

It is worth to note that during the procedure of passing an image through a pre-trained CNN architecture, all the deep binary codes from lower to higher layers can be obtained. The similarity scores given by deep binary codes from different layers vary largely. As illustrated in Fig. 4, for a specific query image, the average precision score of each deep binary code is different, and it is difficult to determine in advance which deep binary code is the most robust one. Thus, we are motivated to investigate how to fuse the search scores returned by deep binary codes from different layers, to further improve the precision of visual search. Inspired by the idea proposed by Zheng et al. [46] which demonstrates that the score curve returned by a good feature shows an “L” shape, while that returned by a bad feature shows a gradually dropping tendency, the effectiveness of a feature can be estimated, as it is