Pattern Recognition Letters 85 (2017) 1–7

Contents lists available at ScienceDirect

Pattern Recognition Letters

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

How could a subcellular image, or a painting by Van Gogh, be similar to a great white shark or to a pizza?

Loris Nanni*, Stefano Ghidoni

DEI, University of Padua, viale Gradenigo 6, Padua, Italy

ARTICLE INFO

Article history:
Received 24 December 2015
Available online 21 November 2016

Keywords:
Deep convolutional neural networks
Transfer learning
Texture descriptors
Texture classification
Ensemble of descriptors

ABSTRACT

In this work, we propose an unorthodox approach for describing a given image. Each image is represented by a feature vector whose elements are the scores assigned to object classes by deep convolutional neural networks that were not related to those that built the given image classification problem. The deep neural networks are trained using 1000 classes; therefore, each image is described by 1000 scores, which are fed to a support vector machine. The proposed approach could be considered a transfer learning method, where, instead of repurposing the learned features to a second classification problem, we use the scores obtained by trained convolutional neural networks. Methods based on state of the art handcrafted descriptors, and the novel approach presented here are compared, together with selected ensembles of such methods. The fusion between a standard approach and the new unorthodox method boosts the performance of the standard approach. The Wilcoxon signed rank test is used to compare the different methods. The novel method is applied to 21 different datasets to demonstrate its generality. The MATLAB source code to replicate our experiments will be available at (https://www.dei.unipd.it/node/2357+Pattern Recognition and Ensemble Classifiers).

© 2016 Elsevier B.V. All rights reserved.

1. Introduction

COMPUTER vision algorithms for recognition tasks often rely on the detection and classification of some local characteristics of the image. A huge variety of features, with different trade-offs between accuracy and computational efficiency, have been proposed in the literature, each one focusing on specific cues of the image to be processed. The SIFT features [1] are widely known and exploited in the robotics community, as they are robust to object rotations and scale variations but at the cost of a rather high computational load. Some more efficient variants also exist [2], which are more suitable when real-time constraints are needed or when low computational power is available.

SIFT, as well as many other feature extraction methods, are often computed in two steps. A keypoint detector is first employed to find the most suitable regions of the image to be analyzed, following the idea explained in [3]. In the second step, such regions are characterized by a vector, often called a descriptor, which results from the measurement of a set of cues, depending on the specific feature being evaluated. The descriptor is often used to feed a classifier, like the Support Vector Machine (SVM) [4] that provides the system's final output.

It is important to note that each feature evaluation algorithm focuses on a specific set of visual cues that are analyzed and measured to evaluate the descriptor. The deep learning paradigm [5] starts from a different perspective. Instead of choosing a predefined set of features (directly connected to the visual cues to be analyzed), which could be unsuitable for the type of images to be analyzed, deep learning proposes to choose the set of features based on the observation of the input images. A crucial aspect to be considered when dealing with deep learning is the training set: it should be properly chosen, since it has a direct and strong influence over the feature selection phase. A safe solution is often to choose huge training sets, made up of ideally millions of images [6]. However, this requirement has a bad impact on both the selection (by human beings) of such a high number of sample images and on the training phase, which requires a huge amount of computational power. This last issue can be overcome by exploiting parallel computing, but this links the benefits of deep learning to the availability of expensive computing systems. The huge amount of data and computation needed represent the main drawbacks of the deep learning paradigm, which can be partially solved thanks to semi-supervised and unsupervised techniques [7].

Our approach can be considered a transfer learning method [8]. In transfer learning, we transfer the learned features in a given problem (base problem) to another task (target problem). This approach works well if the features are general, and it is useful when the target training set is smaller than the base training

* Corresponding author.
E-mail addresses: loris.nanni@unipd.it, loris.nanni@unibo.it (L. Nanni).

http://dx.doi.org/10.1016/j.patrec.2016.11.011
0167-8655/© 2016 Elsevier B.V. All rights reserved.

Table 1
Descriptive summary of the dataset.

Paper	Features extracted using CNN
here	Scores obtained by CNN
[42]	Images are represented as strings of top layers of pretrained CNN
[43]	The convolutional layers of a CNN are used as a filter bank
[44]	The seventh (fully connected) layer of a CNN on ImageNet is used to represent the images
[45]	Images are represented by the last convolutional layer of a CNN
[46]	Five convolutional layers and two fully connected layers are used to describe the images
[47]	Lower-layer features
[48]	First few layers from a pre-trained CNN model
[49]	Top-layer activations are used as feature representation

set. In [8] it is shown that transfer learning can be coupled with deep neural networks. In this work, instead of transferring the features, we use the scores obtained by the pre-trained deep neural networks for training a Support Vector Machine.

The key idea is to solve a given classification problem A by exploiting classifier f_B that was already trained on a different classification problem B and then map the output of this base classification problem to solve the target problem A. The mapping is performed by using a second classifier, as it will be detailed in section III.

The proposed structure requires two stages, classification and mapping to a different problem, and it is therefore more complex with respect to a single classifier; however, it offers the advantage of exploiting a classifier that is already trained. This becomes meaningful given the specific combination of classifiers proposed, as the one which is already trained is a deep learning network, while the mapping classifier is a support vector machine. This structure is able to exploit the feature extraction techniques provided by the deep learning for solving a classification problem that is different from the one for which the deep learner had originally been trained.

We have tested the system using 21 different image classification problems and achieved very good results. To the best of our knowledge, works in the literature that combine transfer learning with the deep learning approach have never been tested on so many datasets as we did, thereby providing statistically significant results. In Table 1 we report several recent deep transfer learning methods. To the best of our knowledge, this work is the first approach based on using the scored of a CNN for training SVM; all the other approaches are based on feature extracted using other layers of the CNN.

In order to let other researchers reproduce our results, the MATLAB code developed for this work will be available at (https://www.dei.unipd.it/node/2357+Pattern Recognition and Ensemble Classifiers).

2. Deep convolutional neural network

Deep learning is one of the most recent machine learning paradigms employed in computer vision. Since its introduction [9], it revolutionized the field, thanks to the superior performance level achieved. Deep learning architectures are characterized by a cascade of several processing layers organized in a hierarchical structure. The way each layer extracts information from its input is the feature extraction algorithm employed by the network, and it is learned during the training phase, as it commonly happens for machine learning algorithms: this means that the deep learner chooses the features to be extracted based on the observation of the training set.

Given the hierarchical organization of the learning network, layers closer to the input (i.e., to the input image being processed) are considered to deal with low-level features (e.g. edges, lines, and corners). Higher layers of the structure process information that can be considered high-level features. It should be noted that a feature extracted by a deep learning algorithm is different from a classical feature (e.g. a SIFT), as the former does not focus on image characteristics that can be explicitly described, which is the case of the latter type of descriptor.

The approach to deep learning chosen in this paper is based on Convolutional Neural Networks (CNNs) [10], available through several software packages and toolboxes. This work is based on the MatConvNet toolbox for Matlab,[1] whose naming convention is also adopted. CNNs are built by choosing a set of *computational blocks* (CBs) that implement data processing functions; such blocks are connected by means of a Directed Acyclic Graph (DAG). Each CB takes an input vector $\boldsymbol{x}$ and provides an output vector $\boldsymbol{y}$, which depends on $\boldsymbol{x}$ and on a set of parameters $\boldsymbol{w}$, that are defined during the training phase. The MatConvNet toolbox provides a number of CBs that can be rearranged in order to form many different networks, thus exploiting the modularity of the CNN structure.

3. Proposed approach

The classification system proposed in this paper is based on the idea of remapping a given deep CNN to solve a target problem which is different from the base problem for which it was trained. Let $A = \{a_0, a_1, \ldots, a_N\}$ be a classification problem, namely a set of classes $\{a_0, \ldots, a_N\}$ into which a given input $\boldsymbol{x}$ should be classified. Then let $y = f_A(x)$ be the classification function implemented by the deep CNN. Such a function takes the input vector and provides an output vector $\boldsymbol{y}$ of length M, which contains the scores assigned to all the possible outcomes considered in the classification problem.

As an example, consider the classification problem *Vehicle* that has three possible outcomes: $Vehicle = \{bike, car, truck\}$. A deep CNN trained to solve such a problem will provide, for each input image, three values that are the scores assigned to the event of the image containing a bike, a car, or a truck, respectively. This is the common way of employing deep CNNs under the (implicit) assumption that the input image will contain a bike, a car or a truck (or none of them), and this is the interesting information that should be extracted.

The approach proposed here moves from the last example, and investigates the idea of solving a different classification problem, say, $Device = \{smartphone, tablet, computer\}$ by considering the scores assigned to the classification problem *Vehicle*.

The above discussion leads to the system proposed in this paper, whose structure is outlined in Fig. 1. The aim of the system is to solve a given target problem Q. To do this, the input image is sent to several different deep CNNs, three in the example, that solve three classification problems: A, B, C. Each CNN is characterized by a different number of labels (M, N, P, respectively). The three deep networks provide three output vectors: $\{a_0, ..., a_M\}$, $\{b_0, ..., b_N\}$, $\{c_0, ..., c_P\}$, which are the inputs to three SVMs, that are trained in order to provide a suitable result to the problem Q. In

[1] The toolbox is freely available at http://www.vlfeat.org/matconvnet/.

Download English Version:

https://daneshyari.com/en/article/4970328

Download Persian Version:

https://daneshyari.com/article/4970328

Daneshyari.com