Neurocomputing 150 (2015) 106–115

Contents lists available at ScienceDirect

Neurocomputing

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

Adaptive feature descriptor selection based on a multi-table reinforcement learning strategy

Monica Piñol [a,*], Angel D. Sappa [a], Ricardo Toledo [a,b]

[a] Computer Vision Center, Universitat Autònoma de Barcelona, 08193-Bellaterra, Barcelona, Spain
[b] Computer Science Dept., Universitat Autònoma de Barcelona, 08193-Bellaterra, Barcelona, Spain

ARTICLE INFO

Article history:
Received 21 October 2013
Received in revised form
28 January 2014
Accepted 22 March 2014
Available online 14 October 2014

Keywords:
Reinforcement learning
Q-learning
Bag of features
Descriptors

ABSTRACT

This paper presents and evaluates a framework to improve the performance of visual object classification methods, which are based on the usage of image feature descriptors as inputs. The goal of the proposed framework is to learn the best descriptor for each image in a given database. This goal is reached by means of a reinforcement learning process using the minimum information. The visual classification system used to demonstrate the proposed framework is based on a bag of features scheme, and the reinforcement learning technique is implemented through the Q-learning approach. The behavior of the reinforcement learning with different state definitions is evaluated. Additionally, a method that combines all these states is formulated in order to select the optimal state. Finally, the chosen actions are obtained from the best set of image descriptors in the literature: PHOW, SIFT, C-SIFT, SURF and Spin. Experimental results using two public databases (ETH and COIL) are provided showing both the validity of the proposed approach and comparisons with state of the art. In all the cases the best results are obtained with the proposed approach.

© 2014 Elsevier B.V. All rights reserved.

1. Introduction

In the computer vision domain the visual object classification (VOC) has attracted the attention of researchers over the last two decades (e.g., [1–4]). Generally, VOC is based on the representation of the given scene in a space of features, which were extracted and then described by means of some feature descriptors. These feature descriptions are then used as discriminative elements to characterize the given objects. They are computed using information of *interest points* together with their neighborhood; such interest points are pixels with special characteristics (e.g., [5,6]). Hence, given an image, the feature descriptors characterize the objects at a higher abstraction level, where classical learning techniques can be used in order to recognize the target object. More elaborated techniques, such as Bag of Features (BoF), are becoming nowadays popular for visual object recognition (e.g., [7,3,8,9]). The BoF consists of four steps as detailed below:

1. Extract the features from the images of the training set using a given detector and a given descriptor.
2. Build a dictionary of visual words using the features extracted before.
3. Construct a histogram, using (1) and (2), for each image in the training set. Hence, the histogram bins represent the number of times a visual word is in the image.
4. Train a classification algorithm using the histogram obtained before.

The BoF architecture is flexible, so that there are different combinations that can be used to implement the four steps presented above. The final performance of the BoF depends on the correct algorithm selection.

The current work is focused on the first step of the BoF; in particular, the goal is to learn the best algorithm to describe the interest points. From our experience, the performance of the BOF is strongly influenced by the image feature descriptor, so we state that identifying the best image descriptor for each image will improve the classification rate. A naive approach to solve this problem could be the concatenation of all the possible descriptors. However, this solution is not always feasible since on the one hand it could take a large amount of resources (e.g., memory, CPU time) and on the other hand this would introduce noise to the solution [10]. The challenge of the problem and the importance of finding the right solution have been recently addressed. An approach to select the best descriptor for each image is presented in [10,11].

* Corresponding author.
E-mail addresses: mpinyol@cvc.uab.es (M. Piñol), asappa@cvc.uab.es (A.D. Sappa), ricard@cvc.uab.es (R. Toledo).

http://dx.doi.org/10.1016/j.neucom.2014.03.080
0925-2312/© 2014 Elsevier B.V. All rights reserved.

In [10] a method for selecting the best descriptor for every image in the database is proposed. In order to select the best descriptor, several attributes of the image (e.g., colorfulness, roughness, shininess, etc.) are taken into account. Although interesting results are presented, their main drawback is the use of a supervised learning scheme where the authors select the descriptors with a subjective criterion. On the contrary, in [11] a method that learns the best descriptor for each image using a Reinforcement Learning (RL) scheme is presented. The RL is a simple learning method based on a trial and error strategy. This work presents two improvements from [11].

1. We propose to use several state definitions.
2. A multi-table scheme is introduced in order to exploit the best state definition for each image.

In summary, this work proposes a novel method to learn the best descriptor from a given set. In order to improve the performance, multiple state definitions are used. This scheme works with a BoF approach, and in concrete, the implementation uses a kd-tree in the second step and a support vector machine (SVM) in the fourth step. The reminder of the paper is organized as follows. Section 2 presents the state of the art. Section 3 summarizes the RL technique. Then, Section 4 presents in detail the proposed method. Experimental results and comparisons are provided in section 5. Section 6 gives the conclusions and future work.

2. State of the art

Reinforcement learning is a learning technique widely used in the robotics community; recently, some work involving RL have been proposed in the computer vision field. For instance, in image segmentation, the RL technique is used to select the appropriate threshold (e.g., [12,13]). In [14] the authors propose a RL based approach to tackle the face recognition problem. The authors present a method to learn the set of dominant features for each image. An approach that joins an active learning with RL is presented in [15]; in this case it learns the exploration and exploitation trade-off during the sampling process. Finally, there are also some works in visual object recognition using RL (e.g., [16–19]). In [20], the authors propose a bottom-up/top-down recognition system in order to learn a similar model than the human learning. In concrete, the method joins the RL and a first order logic technique to recognize objects. In [17] a method that learns the features of the image in order to recognize objects is presented . A RL based approach is used for selecting the classification algorithm, which is later on embedded in the fourth step of a BoF scheme [19].

In this work, we propose the use of RL to learn the best feature descriptor for VOC with a BoFs approach. The novelty of the work, with respect to the former one [11], is the multi-table framework that enables the use of different states

3. Reinforcement learning

The reinforcement learning, as mentioned before, is a trial-and-error learning process [21] where the agent does not have a prior knowledge about which is the correct action to take. RL can be used as a technique to solve a Markov decision process (MDP) problem, in which the *agent* learns how to take an *action* in a given *environment* in order to maximize the expected *reward*. These concepts are incorporated to the tuple of MDP $\langle S, A, \delta, \tau \rangle$ where:

- S is a set of environment states. In this work the states are characteristics of the image.
- A is a set of actions. This work uses five image feature descriptors as actions: Spin, SIFT, SURF, C-SIFT and PHOW.δ
- is a transition function, $\delta : S \times A \rightarrow S$. τ
- is a reward/punishment function, $\tau : S \times A \rightarrow \Re$.

Using the definitions presented above, the RL method works as follows: the agent interacts with the environment to select an action (a_h). The action is selected to maximize the expected reward (r_t) based on $\tau(s_z, a_h)$. Applying the action (a_h) to the state (s_z), the environment gives a new state (s_{z+1}) according to the δ function, and a reward/punishment (r_t) according to the τ function. An illustration summarizing this process is depicted in Fig. 1.

The RL can be implemented using dynamic programming, Monte Carlo method, and temporal difference learning. In this work, a temporal difference based method has been used because it does not require a model and it is fully incremental [21]. More specifically, we decided to use the Q-learning algorithm [22]. In Q-learning, the agent learns the action policy $\pi : S \longrightarrow A$, where the policy π maps the states and the actions to maximize the expected reward (formulated as $E[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots]$). The value function ($V^\pi$) over the states is generated using the policy π and the states (Eq. (1)) and π^* is the optimal policy that the agent must learn during the training process (see Eq. (2)):

$$V^\pi(s) \equiv E[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots] \equiv E\left[\sum_{i=0}^{\infty} \gamma^i r_{t+i}\right], \tag{1}$$

$$\pi^* \equiv \arg\max_\pi V^\pi(s), \quad \forall (s). \tag{2}$$

The best value function V^{π^*} is learnt with the RL strategy using the δ and τ functions explained before. However, since none of them (δ and τ) are known a priori the Q-learning proposes an algorithm to also learn the optimal policy (π^*). Specifically, the Q-learning defines an evaluation function (Q) as Eq. (3) that is used to find the best action. In this case, the π^* can be defined as in the following equation:

$$Q(s,a) = E[r(s,a) + \gamma V^{\pi^*} \delta(s,a)], \tag{3}$$

$$\pi^*(s) = \mathrm{argmax}_a E[r(s,a) + \gamma V^{\pi^*} \delta(s,a)] = \mathrm{argmax}_a Q(s,a). \tag{4}$$

The δ and τ can be deterministic or nondeterministic. δ is deterministic when, given a state (s_z) and applying an action (a_h) a new state (s_{z+1}) is returned; if we cannot guarantee to reach always the same new state (s_{z+1}) the δ function is nondeterministic. The same happens with the τ function, for a given state (s_z) and action (a_h) it could return different rewards (r_t). When the δ and/or τ functions are nondeterministic the environment is nondeterministic. In the case of nondeterministic Q-learning, the formulation of the Q evaluation is defined as follows:

$$Q_n(s_z, a_h) \longleftarrow (1 - \alpha_n) Q_{n-1}(s_z, a_h) + \alpha_n [r + \gamma \max_{a'} Q_{n-1}(s_{z+1}, a')], \tag{5}$$

$$\alpha_n = \frac{1}{1 + \mathbf{visits}_n(s_z, a_h)}, \tag{6}$$

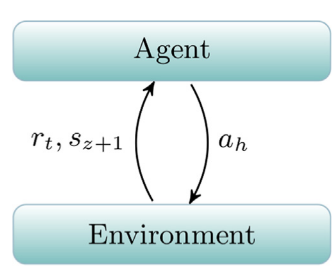

Fig. 1. Illustration of interaction between agent and the environment.

Download English Version:

https://daneshyari.com/en/article/412123

Download Persian Version:

https://daneshyari.com/article/412123

Daneshyari.com