# Multi-Table Reinforcement Learning for Visual Object Recognition

Monica Piñol*, Angel D. Sappa and Ricardo Toledo

Computer Vision Center and Computer Science Dept.,
Universitat Autònoma de Barcelona,
08193 Bellaterra, Barcelona, Spain
{mpinyol,asappa,ricard}@cvc.uab.es
* corresponding author
www.cvc.uab.es

**Abstract.** This paper presents a bag of feature based method for visual object recognition. Our contribution is focussed on the selection of the best feature descriptor. It is implemented by using a novel multi-table reinforcement learning method that selects among five of classical descriptors (i.e., Spin, SIFT, SURF, C-SIFT and PHOW) the one that best describes each image. Experimental results and comparisons are provided showing the improvements achieved with the proposed approach.

**Keywords:** Object recognition; Artificial intelligence; Reinforcement Learning

## 1 Introduction

Bag of features (BoF) has become one of the most widely used approaches for visual object recognition (e.g., [1], [2], [3], [4]). It consists of four steps. Firstly, it finds the interest points (detectors) and describes them (descriptors) in order to characterize the given object at a higher abstraction level. Secondly, the extracted feature points, from all the images in the training set, are structured in a kind of dictionary of words. This dictionary of words is obtained through a learning process and will be used during in the next step. Thirdly, each of the images from the training set is represented by means of a histogram that count the number of times a given word appears. Finally, the obtained histograms are used to train a support vector machine (SVM), which classifies the given images. The BoF is a flexible architecture and the fourth steps mentioned above can be implemented through the use of different algorithms. Hence, the final result will depend on the selection of the right algorithm for each step.

From the four steps mentioned above, a particular attention should be given to the first one, since it represents the most sensible and results are highly dependent on the right descriptor selection. Since the use of different descriptors implies different results, one could think that the best option for this first step is to concatenate as much descriptors as possible. Hence, a given image will be represented by all possible different descriptors. Unfortunately, such a kind of *brute-force* strategy in most of the cases is not feasible due to the fact that it introduces noise. Recently, in [5], the authors present a method to select the best descriptor for each image using *reinforcement learning* (RL). RL is a simple method that allows to learn the best action under a *trail-and-error* framework based on a set of user defined *states*. Although interesting results have been obtained in [5], and most of the time the approach converges to the best descriptor, the problem now is how to define a reliable state to be used during the RL learning process. The current work tackles this problem by proposing a strategy that helps to select the state that maximize the result.

RL has been largely used in the robotics community during the last two decades. Recently, it has attracted the attention in the computer vision field to address problems such as image segmentation or object recognition, just to mention a few. For instance, in the segmentation domain the RL method is used to select the appropriate threshold (e.g., [6], [7]). In [8], the authors propose a RL based face recognition technique that is able to learn the best feature from each image. Similarly, in the object recognition field, [9] presents a RL technique using first order logic. Finally, RL has also been used for learning interest points [10], [11] or for selecting methods for classification [12].

In this paper we present a BoF based approach for the object recognition. As mentioned above the current work is focussed on the first step by using RL. More precisely, the current work contributes with a novel scheme for selecting the best state in the RL method. This scheme results in a multi-table formulation. Regarding the rest of steps of BoF, in the current implementation we use a kd-tree in the second step and a support vector machine in the fourth step. The reminder of the paper is organized as follow. Section 2 presents a brief summary of the RL. Then, section 3 details the proposed method. Experimental results are provided in section 4 and conclusions and future work are given in section 5.

## 2 Reinforcement learning

As mentioned above the current work proposes the use of a multi-table RL strategy for finding the best descriptor that characterizes a given image. In this section a brief description of RL is presented just to introduce the notations and the definition of the used elements (see [13] for more details).

The RL is a learning method used in those cases where the agent does not have a prior knowledge about which is the correct action to take. The RL is a Markov decision process intended to learn how an *agent* ought to take an *action* in a given *environment* so that a *reward* is maximized. These concepts are defined with the following tuple $\langle S, A, \delta, \tau \rangle$, where: $S$ is is a set of environment states; $A$ is a set of actions; $\delta$ is a transition function $\delta{:}S \times A \rightarrow S$ and $\tau$ is a reward/punishment function, $\tau{:}S \times A \rightarrow \Re$ .

By using the definitions presented above the RL methods works as follow: for a given state $s_z$, the agent selects the action $a_h$ that maximize the expected reward $r$ based on the $\tau$ function. In other words, by applying the action $a_h$ in

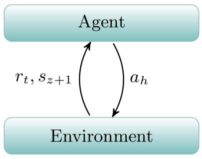


**Fig. 1.** Illustration of interaction between agent and the environment

the state $s_z$ a new state $s_{z+1}$ and a reward $r$ are obtained. Figure 1 illustrates the interaction between the agent and the environment.

The RL can be solved using dynamic programming, Monte Carlo method and temporal difference learning. The temporal difference learning is used in the current work because it does not require a model and it is fully incremental [13]. More concretely, the used framework is based on the *Q-learning* algorithm [14]. In our work, the current state $s_z$ is only affected by the previous visits but not for the future since the Markov decision problem is of first order [8]. The $\delta$ and $\tau$ functions are nondeterministic; hence each element of the Q-table, for an iteration $n$, is computed as follow:

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

$$\alpha_n = \frac{1}{1 + \mathbf{visits}_n(s_z, a_h)}, \tag{2}$$

where $\gamma$ is a discount factor for future reinforcements and is defined as $0 \leq \gamma < 1$. The Eq. 2 is the value of $\alpha_n$ for a nondeterministic world and **visits** is the number of iterations visiting the Q-table at the tuple $(s_z, a_h)$ [15].

## 3 Proposed method

The proposed approach, as mentioned in Section 1, uses the classical BoF ([1], [2]) for object recognition; our work is particularly focussed on the first step of BoF. In other words, we propose a multi-table reinforcement learning based strategy to select the best descriptor for each image from a set that contains the most widely used in the literature. This section presents the definition of the main elements of RL as well as the proposed strategy to combine the Q-table. Figure 2 illustrates the BoF (see $left - side$) with the proposed RL for the first step (see $right - side$). The three remainder steps of BoF are implemented following the state of the art, hence they are not detailed in this section (see [4] for more details).

### 3.1 Tuple definition

This section aims at describing the different elements used in the RL formulation of the current work. The tuple $\langle S, A, \delta, \tau \rangle$ is defined as follows:

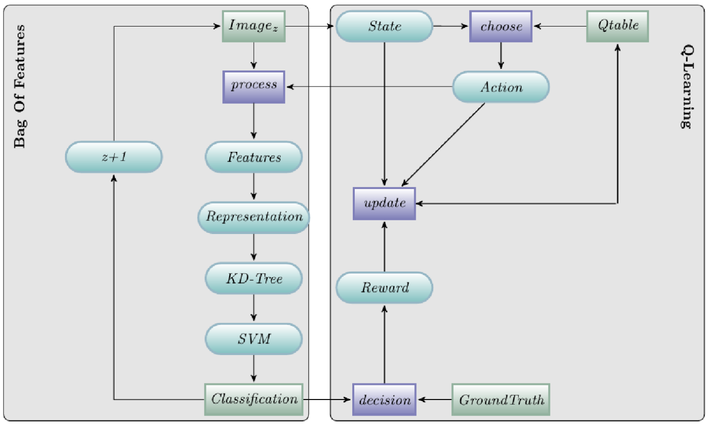


**Fig. 2.** Illustration of learning the best descriptor for each image using Q-learning

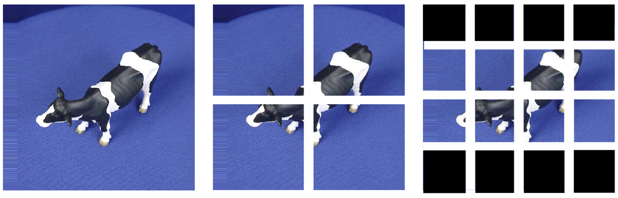

**Fig. 3.** ($left$) Image from ETH database [16]. (*middle*) Image splitted up into four squares. ($right$) Image splitted up into sixteen squares, but only eight of them are used

**State definition:** A state is defined as a set of characteristics from the given image. In order to tackle the challenge of defining a single representative state [5] that could be used with different databases, in the current work we propose the use four different state definitions. In all the cases the states are defined by a set of clusters obtained from the extracted vectors of characteristics—by using k-means. The vector for a given image is defined by extracting the information in a structured way. For instance, in the case of Fig. 3 the vector of characteristics is defined with 13 elements. The first element of that vector is obtained extracting information from Fig. 3($left$); then, Fig. 3(*middle*) contributes with the next four elements of the vector; finally, the last eight elements are obtained from Fig. 3($right$). Once the vectors from all the given training images have been obtained a k-means clustering is used to compute the states that will be used for the Q-learning.

***1) $L^*a^*b^*$ state definition:*** This state uses the $L^*a^*b^*$ color space. This color space is obtained by converting the RGB to XYZ and then, XYZ to $L^*a^*b^*$ (see [17], [18] for more details). The $L^*$ represents the luminance of the image, $a^*$ represents the difference between the red and green colors, and the $b^*$ is the difference between the yellow and blue colors. As mentioned above the given image is split up into 13 squares (see Fig. 3, for each one of them the median value of $L^*a^*b^*$ is computed. Note that since $L^*a^*b^*$ has three components this state definition results in a vector of 39 elements.

*2)* ***Gradient state definition:*** This state definition uses the gradient in x and y directions. The gradient provides edges, but in this case, the state is defined by extracting the median of values in x and y. The partition shown in Fig. 3 is also used here. Hence, this state is defined with a vector of 26 elements.

*3)* ***Entropy state definition:*** The entropy measures the uncertainty of the information. In this case, the information is computed using the same partition shown in Fig. 3, but from the corresponding gray-scale image instead of the RGB color one. For each element of the partition the entropy is computed as follow:

$$E = -\sum_{i=1}^{N} p_i log_2(p_i), \tag{3}$$

where, $p(x)$ is the histogram of image data. In this case the state is defined with a vector of 13 elements.

*4)* ***Histogram of interest point state definition:*** This particular state is defined using all the descriptors of the set of actions. It works as follow; for each image from the training set, it extracts all the interested points and describe them accordingly. After that, similarly to the process of BoF, it constructs a dictionary and find a representation of the image interested points. Finally, a vector with 50 elements (10 elements per descriptor) is extracted to represent the state.

**Actions:** In the current work, the actions are a set of descriptors. In this case, the RL learns the best descriptor for each image. Note that there is a large number of descriptors in the literature [19] the five most representative descriptors are selected for this work: SIFT [20], PHOW [4], C-SIFT [21], SURF [3] and Spin [22].

**$\delta$ function:** Usually, in the RL, the $\delta$ functions is defined as $\delta : s_z \times a_h \rightarrow s_{z+1}$. But, in this work, the $\delta$ function does not give a new state. In the current work, after applying an action $a_h$ to the state $s_z$ it generates a new representation of the image (features). The features obtained in this stage are used in the BoF for classifying the object. After that, the process continues through a new image. Summarizing, given a state $s_z$ and applying an action $a_h$ we obtain a new image from the training set and this new image does not have any similitude with the previous image.

**$\tau$ function:** The$\tau$ functions is defined by $\tau : s_z \times a_h \rightarrow \Re$, when the classification step gives the same label than the given object, the $\tau$ function gives a reward, and when the label does not match with the ground truth, the $\tau$ provides a punishment.

### 3.2 Combination of Q-tables:

Like in [5], the joining process of BoF and RL is used to train the Q-table. Fig. 2 shows this training process, which works as follow. For a given image, the agent extracts the state and applies a descriptor selected from the Q-table using the exploration/exploitation trade off. In the current work the $\varepsilon$-greedy algorithm is used as a strategy for the exploration/exploitation. After applying the descriptor, the agent follow the BoF scheme using the kdtree algorithm and the support vector machine. Once it finishes, the agent obtains a label of the classified image. The agent compares the obtained label with the ground-truth and obtains a reward/punishment. This information is used to update the Q-table according to Eq. 1, using the reward obtained before, the state and the applied action. Finally, after completing a whole iteration the agent extracts a new image from the training set and starts the training process again. In the current work this process is applied four times, one time per state definition. As a result we obtain four Q-tables. Now the question is to define which one should be used for a given image.

In the current work we propose a simple strategy for combining the four Q-tables computed as mentioned above. Actually, the information is not combined; the strategy consists in selecting the action from that Q-table where the reward is maximized. As will be presented in next section this simple strategy allows to improve results with respect to state-of-the-art, which only work with a single state definition.

## 4 Experimental results

The proposed approach has been evaluated using two different databases. Additionally, it has been compared with a recent approach [5] as well as with others two BoF implementations where the first step consists of: *i*) just a single descriptor; and *ii*) a RL based approach with different states definition.

The first experiment is using the ETH database. Figure 4 shows the nine classes we have selected (i.e., apple, car, cow, cowcup, cup, dog, horse, pear and tomato) for testing and comparing the proposed approach. Each of these classes contain 45 images, which were randomly selected. These 45 images are split up into three sets: 15 images for training the BoF; 15 images for training the Q-table and 15 images for testing. The process of training starts creating for each definition of the states the corresponding Q-table. The process is repeated 60.000 times and the values of $\gamma = 0.9$ and $\varepsilon = 0.2$ are used.

In order to have a first comparison of the results obtained with the proposed approach, the performance of BoF for each descriptor is computed. Table 1 shows that the best performance is obtained when BoF uses the PHOW descriptor (74.81% of recognition ratio). In Table 2 the performance of using the BoF with the RL method is presented. The first four rows depict the performance independently obtained for every state definition. In this case (BoF with RL) the best performance is reached with the $L^*a^*b^*$ state definition (82.4%

**Fig. 4.** Some of the objects contained in the nine classes of ETH database

**Table 1.** Performance of BoF using a single descriptor

| Descriptor | Performance |
|---|---|
| Spin | 60.00% |
| SIFT | 61.48% |
| SURF | 62.96% |
| C-SIFT | 68.15% |
| PHOW | 74.81% |

**Table 2.** Performance of BoF with RL

| State Definition | Performance |
|---|---|
| $L^*a^*b^*$ | 82.2% |
| Gradient | 77.8% |
| Entropy | 78.5% |
| Histogram of words | 77.1% |
| App. presented in[5] | 81.4% |
| **Proposed approach (multi-table RL)** | **83.4%** |

of recognition ratio). Additionally, Table 2 shows the result obtained using the proposal presented in [5] (see fifth row). Finally, the performance obtained with the proposed approach (multi-table RL) is depicted in the last row. It can be appreciated that the best performance is obtained by using the strategy proposed in the current work (83.4% of recognition ratio). Some of these results are presented in Fig. 5 by means of the corresponding confusions matrices. Figure 5($left$) shows the confusion matrix resulting when BoF with a single descriptor is used, in this case the PHOW descriptor. The confusion matrix presented in Fig. 5($middle$) corresponds to the BoF using RL and the $L^*a^*b^*$ state definition. Finally, Fig. 5($right$) depicts the confusion matrix resulting from the proposed approach.

A similar comparison to the one presented above has been performed with another database to validate the proposed approach. In this case, the COIL database [23], which contains 100 classes has been selected (Fig. 6 shows some of the objects contained in the COIL database). Each of these classes contain 45 images, which are split up into three sets: 15 images for training the BoF, 15 images for training RL and 15 for testing. In this case the process is repeated 600.000 times and the values of $\gamma$ and $\varepsilon$ are the same as those used in the first experiment ($\gamma = 0.9$ and $\varepsilon = 0.2$)

The performance of BoF for each descriptor is computed and presented Table 3; it can be appreciated that the best performance is again achieved using the PHOW descriptor (98.3% of recognition ratio). In Table 4 the performance obtained when the BoF is used with the RL method is presented. The first four rows present the performance obtained for each of the state definitions introduced in Section 3.1. In this case the best performance corresponds to the BoF with RL and with the gradient state definition (98.8% of recognition ratio). In

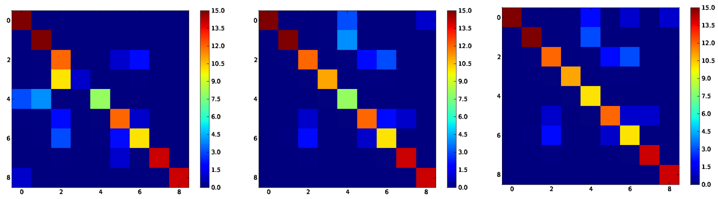

**Fig. 5.** Different confusion matrices for ETH database. (*left*) Using only the PHOW descriptor (74.81%). (*middle*) Using $L^*a^*b^*$ state definition (82.2%). (*right*) Using the proposed method (83.7%).

**Fig. 6.** Some of the objects contained in the COIL database (the whole database contains 100 classes, each class contains 45 images)

order to compare the results with [5], the Table 4 presents in the fifth row the recognition ratio. Finally, in the last row of the Table 4, the performance of the proposed strategy is shown; note that in this case it reaches 99.0% of recognition ratio. The confusion matrices corresponding to three of the examples presented in Tables 3 and 4 are presented in Fig. 7. In the left side the confusion matrix using BoF with a single descriptor (PHOW descriptor) is presented; the middle illustration corresponds to BoF with RL when the best state definition is used (gradient state definition). Finally, the right side illustration depicts the results obtained with the proposed approach. Note that in this case since the is a larger number of objects in the database and the recognition ratios are about 99%, the confusion matrices are almost a diagonal line.

## 5 Conclusions and future work

This paper presents a BoF based approach for visual object recognition. We propose to improve classical BoF by means of the use of a RL strategy for selecting the best descriptor for each image. Our contribution lies on a novel method that allows the use of a multi-table strategy in the RL. This multi-table strategy allows to pick up the best state definition for each image. Experimental results are obtained using the BoF with the RL with two databases: ETH and COIL. In the first database, PHOW is the best descriptor and results in a 74.81% of recognition ratio, the recognition ratios reaches 83.4% using the proposed method. In the second database, the best single descriptor is also PHOW and in this case a recognition ratio of 98.3% is reached, however, with the proposed method it could be also improved up to the 99.0% of recognition ratio. Future

**Table 3.** Performance of BoF using a single descriptor

| Descriptor | Performance |
|---|---|
| Spin | 83% |
| SIFT | 92.2% |
| SURF | 82.27% |
| C-SIFT | 94.47% |
| PHOW | 98.3% |

**Table 4.** Performance of BoF with RL

| State Definition | Performance |
|---|---|
| $L^*a^*b^*$ | 98.53% |
| Gradient | 98.8% |
| Entropy | 98.6% |
| App. presented in[5] | 98.3% |
| **Proposed approach (multi-table RL)** | **99.0%** |

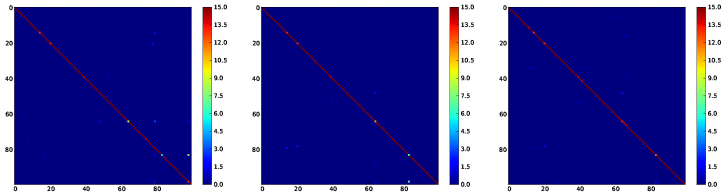

**Fig. 7.** Different confusion matrices for COIL database. (*left*) Using only the PHOW descriptor (98.31%). (*middle*) Using gradient state definition (98.8%). (*right*) Using the proposed method (99.0%).

work will be focused on the combination of descriptors and new state definitions in order to further improve the performance.

**Acknowledgments.** This work was partially supported by the Spanish Government under Research Program Consolider Ingenio 2010: MIPRCV (CSD2007-00018) and Project TIN2011-25606. Monica Piñol was supported by Universitat Autònoma de Barcelona grant PIF 471-01-8/09.

## References

1. Csurka, G., Dance, C.R., Fan, L., Willamowski, J., Bray, C.: Visual categorization with bags of keypoints. In: Workshop on Statistical Learning in Computer Vision, Proc. the European Conference on Computer Vision. Volume 1., Prague, Czech Republic (2004) 1–22
2. Fei-Fei, L., Perona, P.: A bayesian hierarchical model for learning natural scene categories. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition. Volume 2., San Diego, CA, USA (2005) 524–531
3. Bay, H., Tuytelaars, T., Gool, L.V.: Surf: Speeded up robust features. In: Proc. the European Conference on Computer Vision, Graz, Austria, Springer (2006) 404–417
4. Bosch, A., Zisserman, A., Muñoz, X.: Image classification using random forests and ferns. In: Proc. International Conference on Computer Vision, Rio de Janeiro, Brazil, IEEE (2007) 1–8
5. Piñol, M., Sappa, A.D., López, A., Toledo, R.: Feature selection based on reinforcement learning for object recognition. In: Adaptive Learning Agent Workshop, Valencia, Spain (2012) 4–8

6. Shokri, M., R.Tizhoosh, H.: A reinforcement agent for threshold fusion. Applied Soft Computing **8**(1) (2008) 174–181
7. Sahba, F., Tizhoosh, H.R., Salama, M.: Aplication of opposition-based reinforcement learning in image segmentation. In: IEEE Symposium on Computational Intelligence in Image and Signal Processing, Honolulu, HI, USA (2007) 246–251
8. Harandi, M.T., Ahmadabadi, M.N., Araabi, B.N.: Face recognition using reinforcement learning. In: Proc. IEEE International Conference on Image Processing. Volume 4., Singapore, IEEE (2004) 2709–2712
9. Häming, K., Peters, G.: Learning scan paths for object recognition with relational reinforcement learning. In: Proceedings of the 7th IASTED International Conference on Signal Processing, Pattern Recognition and Applications. Volume 678., Innsbruck, Austria (2010) 253
10. Jodogne, S.: Reinforcement learning of perceptual classes using q learning updates. In: Proc. of the 23rd IASTED International Multi-Conference on Artificial Intelligence and Applications, Innsbruck (Austria) (2005) 445–450
11. Jodogne, S., Piater, J.H.: Interactive selection of visual features through reinforcement learning. In: Proc. of the 24th SGAI Internal Conference on Innovative Techniques and Applications of Artificial Intelligence. Volume 21., Cambridge (UK) (2004) 285–298
12. Bianchi, R., Ramisa, A., de Mántaras, R.: Automatic selection of object recognition methods using reinforcement learning. Advances in Machine Learning I **262** (2010) 421–439
13. Sutton, R., Barto, A.: Reinforcement learning: An introduction. 1 edn. MIT Press, Cambridge Univ Press, MA(USA) (1998)
14. Watkins, C.J.C.H.: Learning from delayed rewards. Ph.D. thesis, King's College, Cambridge UK (1989)
15. Mitchell, T.M.: Machine Learning. McGraw-Hill Science/Engineering/Math (1997)
16. Leibe, B., Schiele, B.: Analyzing appearance and contour based methods for object categorization. In: Proc. IEEE Conference on Computer Vision and Pattern Recognition. Volume 2., Los Alamitos, CA, USA, IEEE Computer Society (2003) 409
17. Ruzon, M.A., Tomasi, C.: Edge, junction, and corner detection using color distributions. IEEE Trans. on Pattern Analysis and Machine Intelligence**23**(11) (2001) 1281–1295
18. Martin, D.R., Fowlkes, C.C., Malik, J.: Learning to detect natural image boundaries using local brightness, color, and texture cues. IEEE Trans. on Pattern Analysis and Machine Intelligence**26**(5) (2004) 530–549
19. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. IEEE Trans. on Pattern Analysis and Machine Intelligence**27**(10) (2005) 1615–1630
20. Lowe, D.: Distinctive image features from scale invariant keypoints. International Journal on Computer Vision **60**(2) (2004) 91–110
21. van de Sande, K.E.A., Gevers, T., Snoek, C.G.M.: Evaluating color descriptors for object and scene recognition. IEEE Trans. on Pattern Analysis and Machine Intelligence**32**(9) (2010) 1582–1596
22. Lazebnik, S., Schmid, C., Ponce, J.: A sparse texture representation using local affine regions. IEEE Trans. on Pattern Analysis and Machine Intelligence**27**(8) (2005) 1265–1278
23. Nene, S.A., Nayar, S.K., Murase, H.: Columbia Object Image Library (COIL-100). Technical report (Feb 1996)