Fast Object Recognition Methods for the UJI Online Robot*

P.J. Sanz
Computer Science Department
Jaume-I University
Castellón, Spain
sanzp@ieee.org

R. Marín
Computer Science Department
Jaume-I University
Castellón, Spain
rmarin@icc.uji.es

J.S. Sánchez
Computer Science Department
Jaume-I University
Castellón, Spain
sanchez@uji.es

Abstract - *Within the context of online robots, a considerable amount of research has traditionally focused on the global system functionality, including the way of interaction between the user and the robot. Recent results in different robotics areas have demonstrated the potential of a number of techniques from the Pattern Recognition and Machine Learning domains, although very few work has been specifically addressed to online robots, where the object recognition is directly performed by the user. In this paper, we investigate the feasibility of using a neural network approach to object recognition in the context of online robots, and discuss the main advantages over the application of statistical learning methods. Some experiments with the UJI (Universitat Jaume I) online robot evaluate the performance of different neural network implementations, comparing it to that of some distance-based object recognition algorithms.*

Keywords: Object recognition; Online robots; Neural networks.

1 Introduction

Remote-controlled robots were developed in the 1940s and were used by trained experts. A new class of remotely controlled robots are now accessible on the Internet: the online robots. These allow users from all over the world to visit museums, tend gardens, navigate undersea, float in blimps, or handle protein crystals.

The first generation of online robots came into existence in 1994. This generation of Internet robots [3, 6-9, 14, 15] are mainly based on robotic arms or simple mobile robots that are directly controlled by human operators. In other words, a human is in the control loop. These telerobots operate within a well-structured environment with little uncertainty and have no local intelligence.

In contrast, research on the second generation of Internet robots [18, 20, 21] has recently begun to focus on autonomous mobile robots that navigate in a dynamic and uncertain environment. The key features of this generation of Internet robotic projects are their autonomy and reactive behaviors which enable them to navigate and cope with uncertainty in the real world.

Online robots involves controlling robots or devices from a web browser remotely and differs from traditional teleoperation in several aspects. In addition to the problems associated with time delay, supervisory control, and stability, online robots must be designed to be operated by non-specialists through intuitive user interfaces and to be accessible 24 hours a day. New methods are needed for coordinating simultaneous users, coping with large variations in demand and time delay, and for detecting and recovering from unsupervised errors. In fact, crucial issues like the interface usability [16] still remain unsolved.

New capabilities arise frequently with the introduction of well-known techniques from other research domains, such as Pattern Recognition and Machine Learning. For example, in most of existing online robots, there is not an automatic object recognition module because input to the robot is simply performed by filling in forms or selecting commands. In this context, some research has been recently addressed to make easier the interaction between the user and the robot. In particular, some statistical learning algorithms have been used in the UJI online robot [12, 13] to automatically recognize the objects appearing in a scene.

In this paper, we extend a previous work [12] by considering the application of neural networks as an alternative to the statistical learning techniques. In general, these statistical methods have shown a high performance (in terms of recognition accuracy), although the main deficiency refers to the response time. Correspondingly, the aim of using a neural network classifier, instead of a statistical approach, in the object recognition module of an online robot will be to improve the response time maintaining the efficiency.

From now on, the paper is organized as follows. Section 2 briefly describes the UJI online robot and in particular, its object recognition module. Section 3 compares the statistical learning techniques with the neural networks. Section 4 introduces the neural network realizations considered in the present work and describes the experiments. Section 5 provides the experimental results and discusses the main differences with respect to those obtained by the statistical learning approaches in our previous work [12]. Finally, conclusions and further work are outlined in Section 6.

2 A short description of the UJI online robot

The UJI online robot [12, 13] allows the manipulation of objects located on a board by means of mouse interactions and also by using a subset of the natural language. As the program is able to learn new object characteristics through the user interaction, the system becomes more robust as time goes by. Such a capability has not been reported in the frame of telerobotics, yet.

The system is organized in a modular manner, involving different tasks: object recognition, natural language understanding, image processing, grasping and camera operations, etc. As can be seen in Figure 1, the software architecture is organized into several modules connected through the CORBA and HTTP standards. The robot knowledge is organized into a database managed by the server side and accessed by the multiple Java clients running over the Internet. This means that the robot knowledge is common to the multiple users and even more important, it is robot independent.

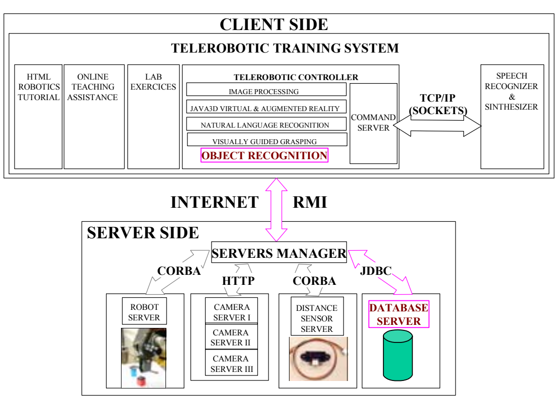

Figure 1. System software architecture

In particular, the robot server is in charge of setting up both the robot and the camera, and also of controlling them. This subsystem runs on a PC and interacts with a SCARA robot via the parallel port. On the other hand, the user interface consists of a Java-based application that allows to obtain images of the robot workspace and sends commands to the robot server in order to control the robot and also to update the object database.

Unlike most of existing web-based robotic systems [3, 6-9, 14, 15, 18, 20, 21], the UJI online robot provides an object recognition module (see Figure 1), which allows the user to interact in a higher level. This avoids the well-known cognitive fatigue, associated with the majority of online robots. In fact, as far as we know, apart from the UJI online robot, the system developed in the University of Columbia is the only with automatic object recognition capabilities.

3 Statistical and neural object recognition approaches

Numerous taxonomies for classification methods in Pattern Recognition [2] have been presented. One commonly-stated division separates neural and more traditional statistical techniques [19] and in this sense, a lot of benchmark and comparison studies [1, 10] have been published to illustrate the main differences between both classification perspectives.

From a practical point of view, the performance of an object recognition module in online robots must be measured in terms of two competing goals: recognition accuracy (or efficiency) and computing time. While the accuracy represents a check on the ability of the algorithms to correctly recognize objects, the computing time becomes a crucial requirement for real-time applications (this is the case, for instance, of online robots).

The statistical learning algorithms show, in general, a good performance in terms of efficiency, although they also present some important drawbacks related to the large response time. For example, the well-known *k*-Nearest Neighbor decision rule [4] constitutes a widely used classifier because it does not rely on a parametric model, is extremely simple to implement, and asymptotically errs with a bounded rate that is at most twice the Bayesian error [2]. Nevertheless, the time required to classify a new object is high, since this time is proportional to the number of features and the number of examples in a training set.

Taking into account the aforementioned disadvantage of most statistical learning methods (e.g., the *k*-Nearest Neighbor), the neural approach appears to be a good alternative because of its good performance in both efficiency and response time. In general, neural networks are superior in classification time to statistical techniques with similar or even better recognition accuracy.

Accordingly, in this paper, we study the employment of neural networks for object recognition in online robots. Despite the high efficiency of the statistical approaches used in our previous work [12], neural networks may provide similar recognition accuracy rates, but significant lower response time.

4 Experiments with neural networks

After a satisfactory experience with different statistical classification techniques [12], now a more elaborated contribution based on neural networks is presented. The ultimate aim of this is to evaluate the possibilities of using an alternative to those statistical schemes for object recognition in the online robots context where the response time is in general, as important as the recognition rate.

Thus, the first step consists of calculating 100 examples for each object that will be used in the recognition phase. In our experiments, we have picked up four different objects (allen, cylinder, cube, and wheel), giving a total of 400 samples. After this, the set of 400 samples is divided into three different groups: training (200 samples), validation (100 samples), and test (100 samples).

```
If ((n1 > 0.6) and (n2, n3, n4 < 0.4)) Then
  Allen wrench
Else
  If ((n2 > 0.6) and (n1, n3, n4 < 0.4)) Then
    Cylinder
  Else
   If ((n3 > 0.6) and (n1, n2, n4 < 0.4)) Then
     Cube
   Else
    If ((n4 > 0.6) and (n1, n2, n3 < 0.4)) Then
      Wheel
    Else
      Indetermination
```

Figure 2. Algorithm to interpret the output from the neural network

Because of the existence of these fur classes (allen, cylinder, cube, and wheel), the network will be provided with four neurons (s1, s2, s3 & s4) in its last layer. The s1 neuron will classify the sample as belonging to the allen object, the s2 as circle, the s3 as cube, and the s4 as wheel. The range of values that will be offered by each one of them is [0.1-0.9]. Thus, the outputs of the neural network will be (n1, n2, n3, n4), being n1 the result of the s1 neuron, n2 of the s2, and so on. Once we get an output from the neural network in the form (n1, n2, n3, n4), our interpretation can be observed in Figure 2.

The procedure for training the neural network calculates the neuron weights according to the particular algorithm that is being used. The algorithms that we have tested in the experiments are the following:

1. trainrp: Resilient backpropagation. [17]
2. trainscg: Scaled conjugate gradient backpropagation.
3. traincgf Conjugate gradient backpropagation with Fletcher-Reeves updates.
4. traincgb Conjugate gradient backpropagation with Powell-Beale restarts.
5. trainoss One step secant backpropagation.

By using these five algorithms, the neural network training will be accomplished when any of the following conditions is fulfilled:

1. We have reached the maximum number of iterations.
2. We have exceeded the time limit.
3. The efficiency has been minimized to the objective established.
4. The efficiency gradient is less than the minimum gradient.
5. The efficiency of the validation has been increased more than the maximum number of validation errors from the last time that was decreased.

On the other hand, related with the kind of descriptors used, note that we need to select a set of those mathematical descriptors that will enable the system to decide to which class of objects a real sample (as detected by the camera) belongs. The idea is that the values of these descriptors are to be quite similar when the samples belong to the same class, and quite different when the samples belong to distinct classes. Thus, the selected descriptors are the invariant Hu descriptors (both surface and border alternatives) [11], Thinness ratio (Tr), Shape Elongation (Lv), and Spreading (Rv) [5].

5 Results and discussion

Firstly, it is necessary to remark that all the results have been obtained by using the *Neural Networks Toolbox of Matlab 6.1* [22], as a way to prototype in a very fast manner, with the end of isolating a software module in a further stage, after conveniently analyzing these preliminary results.

A first experiment employs the whole set of object descriptors (surface and border Hu, Thinness ratio, etc.) in order to train and test the corresponding performance by the neural network classifier. These results are shown in Table 1.

Table 1. Efficiency taken into account the whole set of descriptors

Topology	[10 4]	[50 4]	[10 10 4]	[100 100 4]	Average
Trainrp	92,20	94,20	93,00	95,70	93,78
Trainscg	94,40	94,20	92,90	92,20	93,43
Traincgf	84,30	87,00	84,00	89,20	86,13
Traincgb	92,20	88,40	92,00	92,90	91,38
Trainoss	91,60	89,40	93,80	93,60	92,10
Average	90,94	90,64	91,14	92,72	

We can see that some combinations of algorithm and topology obtain an efficiency rate higher than 95% (i.e., Trainrp and 100;100;4). Besides this, we have realized the following situations:

1. When increasing the number of neurons in the intermediate layers, performance increases too.
2. For those applications where the response time is very important (e.g., online robots), we could select a partially complex configuration that yields good enough efficiency (i.e. 10;4, Trainrp). Moreover, for those applications where the time is not critical, we could select a more complex configuration (e.g., 100;100;4, Trainrp), which improves performance very much.

Table 2. Efficiency when only Hu surface descriptors are considered

Topology	[10 4]	[50 4]	[10 10 4]	[100 100 4]	Average
Trainrp	75,2	75,7	74,1	76,5	75,38
Trainscg	73,7	73,4	72,9	78,2	74,55
Traincgf	72,1	66	66,5	74,7	69,83
Traincgb	70,1	64,9	58,1	76,1	67,30
Trainoss	73,1	74,7	74,6	76,1	74,63
Average	72,84	70,94	69,24	76,32	

From a second experiment, Table 2 reports the results when only the set of Hu surface descriptors are considered. In such a situation, one can observe that these descriptors are not sufficient to correctly discriminate the four problem classes. Thus, it is possible to conclude that this set of Hu surface descriptors are not good enough to identify a class in our scenario.

Now, we consider only the set of Hu border descriptors, and the results are shown in Table 3. Looking at the averages in topologies and training algorithms in Table 3, it is obvious to conclude a worse performance by using the Hu border descriptors than by means of the Hu surface ones.

Table 3. Efficiency (%) when only Hu border descriptors are considered

Topology	[10 4]	[50 4]	[10 10 4]	[100 100 4]	Average
Trainrp	60,8	66,4	64,4	63,3	63,73
Trainscg	60,5	60,3	55,8	59,4	59,00
Traincgf	52,1	50,2	53,3	49,9	51,38
Traincgb	58,2	53,5	59	60,8	57,88
Trainoss	54,2	60,3	57	59,4	57,73
Average	57,16	58,14	57,90	58,56	

The preliminary conclusion would be the following: if neither Hu surface descriptors nor Hu border ones are good descriptors, the set of descriptors that must be good are the rest, that is, those that are related with Tr, Lv, and Rv descriptors. Thus, we can see these new results, using this set of three descriptors in Table 4.

Table 4. Efficiency using the descriptors set: Tr, Lv, Rv.

Topology	[10 4]	[50 4]	[10 10 4]	[100 100 4]	Average
Trainrp	90,7	92,4	90,8	93,5	91,85
Trainscg	86	92,4	91,4	92,1	90,48
Traincgf	80,2	86,1	85,2	92,5	86,00
Traincgb	86,3	91,9	91,9	93	90,78
Trainoss	90,4	90,7	91,4	95,2	91,93
Average	86,72	90,70	90,14	93,26	

To complete the experiments, some results related to the response time will be shown in Table 5, taking into account that all the times, independently on the training algorithms and the set of descriptors, are inside the interval (33;94) msec, depending basically on the topology used, because more complex ones require more computation time. This times have been obtained with an Intel Celeron to 400 Mhz Processor.

Related to the response time required for the training period, we can conclude that all the times are inside the interval (20;70) seconds, including the more complex topology used. More specifically, the lower time corresponds to the Trainrp algorithm, while the higher is for the Trainoss one.

Table 5. Response time for descriptors: Tr, Lv, Rv

Topology	[10 4]	[50 4]	[10 10 4]	[100 100 4]
Trainrp	44	49	44	94
Trainscg	39	50	44	87
Traincgf	33	50	44	88
Traincgb	44	49	44	88
Trainoss	44	49	44	88

6 Conclusions and further work

In this paper, we have studied the possibility of using a neural network approach to object recognition in online robots, and discuss the main advantages over the application of statistical learning methods. Several experiments have been carried out with the UJI online robot in order to evaluate the performance of different neural network implementations.

After analyzing the results shown in the corresponding tables, we can draw some interesting conclusions. The best algorithm performance is for Trainrp, developed by Martin Riedmiller. This algorithm achieves more than 95% of efficiency when a topology of (100,100,4) is used jointly with the complete set of descriptors, and more than 93% when the set of descriptors is limited to (Tr, Lv, Rv) for the same topology. On the other hand, another important issue is that, when time is critical, an interesting possibility will be employ this set of only three descriptors (Tr, Lv, Rv), jointly with the Trainrp algorithm, because the computational time to extract only these three descriptors is always lower than when the complete set is processed. Finally, another important result is that the border and surface Hu descriptors have not important discrimination capabilities for this case.

In our previous work with statistical learning methods [12], the best performance was achieved by a *k*-Nearest Neighbor classifier (93 % of efficiency, and response time of 814 msec). Note that, under the same conditions (set descriptors Tr, Lv, Rv), the results from a neural networks, using Trainrp with the most complex topology (100;100;4) are even better (93.5 % of efficiency, and response time of 94 msec). Thus, these results aim at following to investigate in neural networks applied to online robots.

Related to possible application of the object recognition techniques to online robots in real scenarios, it is interesting to say that recently we are trying to extend our work to the humanitarian demining context. A first stage in consideration would be the possibility of including any kind of object reconstruction (i.e., the landmine), that is, partially occluded by the forest, ground, etc., assisting to the user in his identification tasks. This could be a good example of augmented reality in real life applications, that is, supported from object recognition capabilities.

In fact, some conversations have been initiated in order to analyze the feasibility of adapting and enhancing the UJI online robot system in order to participate within the EUREKA program, by means of the project "Advanced Global System To Eliminate Anti-Personnel Landmines (ANGEL)".

Acknowledgements

This work has partially been funded by the Spanish CICYT (Ministry of Science and Technology) under Grant TIC2000-1703-C03-03, by the Generalitat Valenciana under Grant GV01-244, and by the Fundació Caixa Castelló – Bancaixa under Grant P1-1B2002-07.

References

[1] B. Cheng, and D. Titterington, Neural networks. A review from a statistical perspective, *Statistical Science* Vol. 9, No. 1, pp. 2-54, 1994.

[2] T.M. Cover, and P.E.Hart, "Nearest neighbor pattern classification", *IEEE Trans. on Information Theory*, Vol. 13, No. 1, pp. 21-27, 1967.

[3] M.J. Cox, and J.E.F. Baruch, Robotic Telescopes: An Interactive Exhibit on the World-Wide Web, Proc. 2nd Int. Conference of the World-Wide Web, Chicago, 1994.

[4] B.V. Dasarathy, *Nearest neighbor (NN) norms: NN pattern classification techniques*, IEEE Computer Society Press, Los Alamitos, CA, 1990.

[5] R.O. Duda, and P.E. Hart. *Pattern Classification and Scene Analysis*, John Wiley & Sons, New York, 1973.

[6] A. Ferworn, R. Roque, and I. Vecchia, MAX: Wireless teleoperation via the World Wide Web, Proc. IASTED Conference on Robotics & Applications, Santa Barbara, pp. 158-162, 1999.

[7] K. Goldberg, M. Mascha, S. Gentner, N. Rothenberg, C. Sutter, and J. Wiegley, Desktop teleoperation via the World Wide Web, Proc. IEEE Int. Conference on Robotics and Automation, pp. 654-659, 1995.

[8] K. Goldberg, J. Santarromana, G. Bekey, S. Gentner, R. Morris, J. Wiegley, and E. Berger, The Telegarden, Proc. of ACM SIGGRAPH, pp. 135-1140, 1995.

[9] S. Goldberg, et al., DIGIMUSE: An interactive telerobotic system for remote viewing of 3D art objects, IROS'98: Workshop on Web Robots, Victoria, Canada, pp. 55-60, 1998.

[10] L. Holmström, P. Koistinen, J. Laaksonen, and E. Oja, Neural and statistical classifiers – taxonomy and two case studies, *IEEE Trans. on Neural Networks*, Vol. 8, No. 1, pp 5–17, 1997.

[11] M. K. Hu, Visual pattern recognition by moment invariants, *IRE Trans. on Information Theory*, vol. 8, pp. 179-187, 1962.

[12] R. Marín, J.S. Sánchez, and P.J. Sanz, Object recognition and incremental learning algorithms for a web-based telerebotic system, Proc. of IEEE Int. Conference on Robotics and Automation, Washington D.C., pp. 2719-2724, 2002.

[13] R. Marín, P.J. Sanz., and J.S. Sánchez, A very high level interface to teleoperate a robot via web including augmented reality. Proc. of IEEE Int. Conference on Robotics and Automation, Washington D.C., pp. 2725-2730, 2002.

[14] G.T. McKee, and R. Barson, NETROLAB: providing access to robotics technology using the Internet. *Robotics and Machine Perception. Special issue: Networked Robotics*, Vol. 5, No. 1, 1996.

[15] E. Paulo, and J. Canny, Delivering real reality to the world wide web via telerobotics, Proc. of IEEE Int. Conference on Robotics and Automation, pp. 1250-1256, 1996.

[16] J. Preece, Y. Rogers, H. Sharp, D. Benyon, S. Holland, and T. Carey, *Human-Computer Interaction*, Addison-Wesley, Reading, UK, 1994.

[17] M. Riedmiller, and H. Braun. A direct adaptive method for faster backpropagation learning: the RPROP algorithm. Proc of the IEEE Int. Conference on Neural Networks, San Francisco, pp. 586-591, 1993.

[18] P. Saucy, and F. Mondana, KhepOnTheWeb: open access to a mobile robot on the Internet, *IEEE Robotics and Automation Magazine*, pp. 41-47, 2000.

[19] R.J. Schalkoff, *Pattern Recognition: Statistical, Structural and Neural Approaches*. John Wiley & Sons Inc., 1992.

[20] D. Schulz, W. Burgard, D. Fox, S. Thrun, and A.B. Cremers, Web interface for mobile robots in public places, *IEEE Robotics and Automation Magazine*, pp. 48-56, 2000.

[21] R. Simmons, XAVIER: An autonomous mobile robots on the Web, Proc. of IROS'98 Workshop on Web Robots, Victoria, Canada, pp. 43-48, 1998.

[22] http://www.mathwoks.com