

Exploiting “Mental” Images in Artificial Neural Network Computation

Massimo De Gregorio and Maurizio Giordano

Abstract In Artificial Neural Network (ANN) computing the learned knowledge about a problem domain is “implicitly” used by ANN-based system to carry on Machine Learning, Pattern Recognition and Reasoning in several application domains. In this work, by adopting a Weightless Neural Network (WNN) model of computation called DRASiW, we show how the knowledge of a problem, internally stored in a data representation called “Mental” Image (MI), can be made “explicit” both to perform additional and useful tasks in the same domain, and to better tune and adapt WNN behavior in order to improve its performance in the target domain. In this paper, three case studies of MI processing in the realm of WNN applications are discussed with the aim of proving the viability and the potentialities of exploiting internal knowledge of WNNs to self-adapt and improve their performance.

Keywords Weightless systems - Mental images

1 Introduction

In traditional ANNs the knowledge about a problem domain is coded in the configuration of synaptic weights between neurons. The goal of the ANN training phase is to find the optimal configuration of weights that allows the network to properly generate the expected outputs in the classification/recognition phase. The configuration of weights can be considered as the internal state of the network. How it is obtained and changed during the network operation is a matter of the particular

M. De Gregorio (✉)
Istituto di Scienze Applicate e Sistemi Intelligenti “Eduardo Caianiello” – CNR, Via Campi
Flegrei 34, 80078 Pozzuoli, NA, Italy
e-mail: massimo.degregorio@cnr.it

M. Giordano
Istituto di Calcolo e Reti ad Alte Prestazioni – CNR, Via Pietro Castellino 111,
80131 Naples, Italy
e-mail: maurizio.giordano@cnr.it

ANN model adopted. What is important is that, once the network architecture (i.e., layers, number of neurons per layer, and connection paths) is set, the ANN configuration of weights fully characterizes its behavior.

RAM-based neural networks are alternative models of ANNs in which the learned knowledge about the problem domain is coded inside RAM-neuron contents rather than on their interconnections. As in the classical weight-based ANNs, the particular configuration of RAM cells is obtained in the training phases, either if they are carried out in super-, semi- or unsupervised manner. At any time during a RAM-based neural network operation, the configuration of RAM contents represents the internal state of the network. Once we have set the RAM-based ANN architecture (i.e., layers, number of neurons per layer, RAM bit address, type of data stored, etc.), the “image” (snapshot) of RAM contents fully characterizes the internal state of the ANN and, as a consequence, the image represents the knowledge and the behavior of the ANN functioning.

Generally in ANN models this internal state is *implicit*. Although the internal state of the learning process is coded by the information stored in the ANN data structures (either weights or RAM contents), this information may not be accessible by the neural-based system to be exploited in a computational meta-level. In the RAM-based model of the ANN adopted in this work, the DRASiW weightless model, this is possible thanks to a particular feature: the contents of RAM-neurons not only characterize the network behavior, but they are also an additional information explicitly available to the neural-based system, in such a way that the ANN can process this information in a computing meta-level in order to adapt and to tune its future behavior.

As in [10], our approach tries to make explicit the internal representation of knowledge of an ANN with the aim of facilitating an interpretation (that can be geometrical, physical, symbolic, etc.) of the learning process and of discovering its correlation to the input. While authors do not suggest applications of the ANN inner knowledge processing, in our work we prove with real case studies how to exploit this knowledge to adapt to domain changes as well as to improve ANN performance in the target domain.

Works like [13, 18] propose methods to interpret and to make explicit the ANN internal knowledge by extracting the knowledge in form of rules (either symbolic or fuzzy) with the only aim of using such rules to simulate the ANN behavior. On the contrary, in our approach we exploit learned knowledge of an ANN to improve and/or to adapt the performance of the same ANN, automatically and/or with the user feedback, to a data domain which may change in time or may contain incomplete and/or ambiguous information.

The fact that we start from an already trained ANN and we refine its performance, by extracting and exploiting its internal knowledge, makes our approach also different from others, like [21], in which the knowledge of an ANN trained on a problem domain is used to extract a set of concise and intelligible symbolic rules that can be used to “refine” an already existing rule-based system, which may have an incomplete or even incorrect initial knowledge of the target problem.

Another close topic is how to integrate explicit and implicit knowledge in neuro-symbolic (hybrid) processing [15]. In this perspective the solution presented in our work can be considered hybrid too. Regardless of how we explicitly represent the mental images of ANNs, our intent is to exploit high-order characteristics of the learned knowledge (macro quantities, invariants, etc.) as an additional information to the neural-based system to improve its performance.

This chapter is organised as follows. Sections 2 and 3 are devoted to the introduction of the DRASiW model and its internal knowledge representation (“Mental” Images). Section 4 shows and discusses three different applications in which the use of “Mental” Images in the computational process improves the performance of the DRASiW systems. Finally, Sect. 5 sums up concluding remarks and perspectives.

2 DRASiW Model

Weightless Neural Networks (WNNs) [1, 12], differently from classical ANNs, adopt a RAM-based model of neuron by which learning information about a data domain is stored into RAM contents instead of computed weights of neuron connections. A RAM-neuron receives an n -bit input that is interpreted as a unique address (stimulus) of a RAM cell, and used to access it either in writing (learning) or reading (classification) mode. WNNs have proved to provide fast and flexible learning algorithms [2].

WiSARD systems are a particular type of WNN, that can be developed directly on reprogrammable hardware [3]. A WiSARD is composed by a set of classifiers, called *discriminators*, each one assigned to learn binary patterns belonging to a particular class. The WiSARD, also called *multi-discriminator architecture*, has as many discriminators as the number of classes it should be able to distinguish.

Each discriminator consists of a set of RAM-neurons, which store the information of occurrences of binary patterns during the learning stage. Given a binary pattern of size s , the so-called *retina*, it can be classified by a set of WiSARD discriminators, each one having m RAMs with 2^n cells such that $s = m \times n$. Since each RAM cell is uniquely addressed by an n -tuple of bits, the input pattern can be partitioned into a set of n -tuples of bits, each one addressing one cell of a RAM. n -tuples of bits are pseudo-randomly selected and biunivocally mapped to RAMs (see right part of Fig. 1), in such a way that the input binary pattern is completely covered.

The WiSARD training phase works as follows:

1. *Initialization*: all RAMs cells for each discriminator are set to 0.
2. *Training set selection*: a training set of binary patterns, all with the same size, is selected; each pattern is known to belong to (and to represent) only one class.

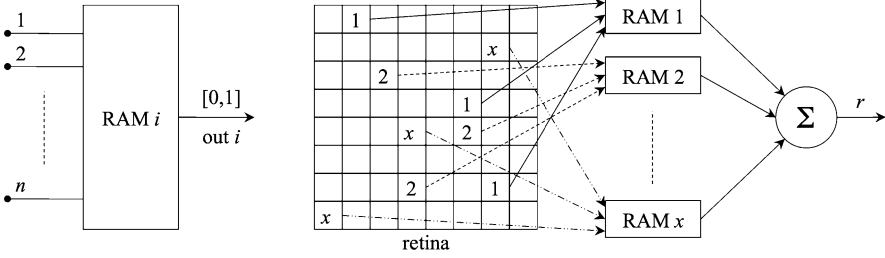


Fig. 1 RAM-neuron (left) and WiSARD discriminator (right)

3. *Training*: for each training pattern the discriminator assigned to the belonging class is selected; the pseudo-random mapping is used to define, from the binary pattern, all n -tuples; each n -tuple forms a unique address of a RAM cell of the discriminator, whose content is set to 1.

After training, if a RAM cell is set to 0 then the n -tuple of bits in the retina, corresponding to physical address (in binary notation) of that memory cell, never occurred across all samples in the training set, otherwise it occurred at least in one sample.

The WiSARD classification phase works as follows:

1. *Test set selection*: a test set of binary patterns, all with the same size, is selected; for each sample of the test set we want to know which category it belongs to.
2. *Classification*: the pseudo-random mapping is used to extract, from each test pattern, the n -tuples of bits in such a way to identify RAM cells to be accessed across all discriminators; contents of accessed cells are summed by an adder (Σ) so giving the number r of RAMs that output 1; r is called *discriminator response*.

It is easy to see that $r = m$ if the input pattern belongs to the training set. While $r = 0$ if no n -tuple of bits in the input pattern appears in the training set. Intermediate values of r express a “similarity measure” of the input pattern with respect to training patterns. The adder enables a network of RAM-neurons to exhibit (like ANN models based on synaptic weights) generalization and noise tolerance [2].

DRASiW [8] is an extension of WiSARD: instead of having RAM cells set to 1 once accessed during training, they are incremented by 1 at each access. Thus, after training, RAM contents store the number of occurrences (frequency) of a specific n -tuple of bits across training patterns. The new domain of memory cells contents (non negative integers) produces the same classification capability of a WiSARD provided that Σ counts the number of addressed non-zero memory cells.

The DRASiW model augments the WiSARD model adding a backward classification capability by which it is possible to generate *prototypes* (i.e., representative samples) of classes learned from training patterns [11, 19]. In DRASiW, RAM-neuron cells act as access counters, whose contents can be reversed to an internal “retina” storing a “Mental” Image (MI). Memory cell contents of DRASiW

discriminators can hence be interpreted as sub-pattern frequencies in the training set. The MI and the internal “retina” metaphors were originally explored and discussed with respect to their cognitive plausibilities in [6].

3 Mental Images

There are two different ways a DRASiW system can produce MIs: statically and dynamically. Static MIs are generated after the training phase and do not change anymore. They represent a pictorial representation of the discriminator internal information. Let consider the 12 instances of black “7”s, reported in Fig. 2a, as the training set for the class “seven”. An example of static MI produced by a DRASiW system trained on this training set, is reported in Fig. 2b. This gray level, non-crisp example of class “seven” is the result of how the sub-patterns appear in the training set. In fact, the gray levels are generated taking into account the sub-pattern frequencies.

Another way of producing MIs is to update them each time the system receives a new training set pattern. This mode, also called *online training*, is by far one of the more interesting operation mode of a DRASiW system. There are many applications in which the system has to adapt to the new and changing appearance of the pattern to classify. The only way to face this problem is to update and store the new information in the MIs. The system updates the MIs each time it receives a new pattern. In Fig. 2c, the reader can notice how the MI changes with respect to the input of patterns. The first MI is produced just with the first “7”. The second one is produced by increasing the gray level of those pixels in common with the previous pattern (more frequent pixels). All the other MIs are the result of applying this procedure each time a new pattern is presented to the system. The MI in Fig. 2d is the result at the end of the process. To sum up, RAM contents corresponding to sub-patterns of the binary input on the retina are increased by one (*reinforcement*), while RAM contents corresponding to those sub-patterns which were not present in the binary input image on the retina are decremented by one (*forgetting*). In other words,

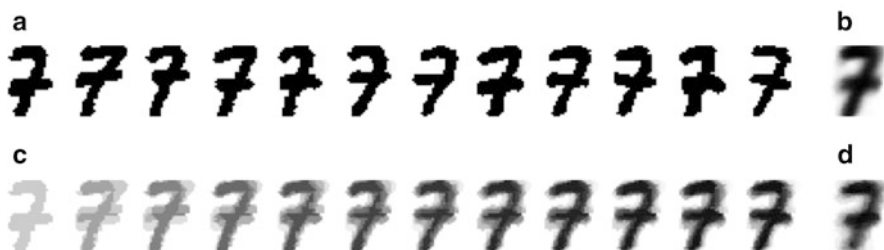


Fig. 2 Static and dynamic mental images: (a) training patterns; (b) static MI; (c) dynamic MI after each training pattern; (d) dynamic MI at the end of training

the *Reinforcement & Forgetting* strategy (*RF*) allows to store the frequency of sub-patterns occurrences during training time. In this way the MI stored and updated in time represents a sort of dynamic prototype (history) of the corresponding class.

4 Improving DRASiW Performance

The first two applications reported in the following subsections deal with the problems of tracking deformable objects and of isolating the background in video sequences. Both of these applications take advantage of dynamic MIs. The third application faces the problem of classification through features. In this case, the information coded in MIs is exploited by the system to identify a set of “metaclasses” used to better refine and improve the classification process.

4.1 Tracking Deformable Objects

In the realm of object tracking systems [22], many real life scenarios, which span from domestic interaction to industrial manufacturing processes, pose hard challenges. In particular, when the object is non-rigid, deformable, and/or manipulated during the tracking, both its position and deformation have to be followed.

In [20] we present a DRASiW system designed and implemented for tracking deformable objects. It supports online training on texture and shape of the object, with the aim of adapting in real-time to changes and of coping with occlusions. This object tracking system deals with *Pizza Making* problem. Pizza is a non-rigid deformable object that can assume whatever shape we want. Hence, it is not possible to define a model for the tracking. In this context, the system should be able to dynamically identify the pizza dough and robustly track it without prior knowledge.

At the beginning, the tracking system is fed with an image representing the object to follow with its initial shape and position. This image is used to train a set of DRASiW discriminators: one discriminator is placed at the target position, the remaining discriminators are placed all around the target position with increasing displacements in the *XY* directions. The configured set of discriminators forms the so called *prediction window* of the tracking system. When the object starts moving, the DRASiW-based tracking system tries to localize the object through the discriminator responses. The higher is the response the more probable the object is in that part of the prediction window processed by that discriminator. Once the system localizes the object in a new position, it uses this image to train again the set of discriminators in the prediction window which is also displaced jointly to the target. So doing, the MI of the object is updated and, hence, it will represent the more recent object shapes.

Figure 3 shows snapshots of pizza making actions: manipulation, dough stretching, seasoning, and backing. The outputs of the DRASiW system are represented

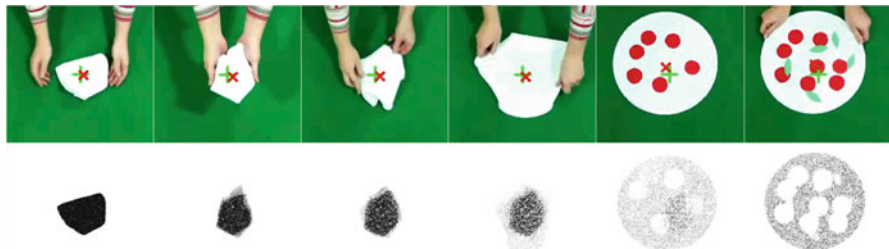


Fig. 3 Sketches of the DRASiW tracking system results in a frame sequence (*top row*), and corresponding MIs (*bottom row*)

by colored crosses. The green cross represents the retina center of the discriminator with the higher response; while the red cross is the mass center of the current MI.

As one can notice, the tracking results improve if the DRASiW system takes into account the information given by the updated MI. We could not reach the same performance if the information contained in the current MI had not been exploited by the system tracking procedure.

4.2 Generating Background Models

Change Detection (CD) is the problem of separating foreground objects from background areas in a video sequence. Several techniques and solutions have been proposed to face the CD problem. Evaluation and comparison surveys of existing techniques can be found in [5, 14, 16, 17]. Regardless of the specific applied method,¹ most approaches share the basic idea of insulating moving objects from the background by comparing image areas of new video frames with respect to either a *background model* or a *model of the target moving objects*. Background models can also be classified as pixel-based or region-based depending on whether computation is based on only the pixel color or a neighborhood of pixels.

In [9] a CD method based on DRASiW is proposed. It exploits a pixel-based background model built around the notion of MI. In the approach, pixel processing is carried out by a DRASiW discriminator. The information stored in neurons is related to the evolution of changeable pixel color in the video timeline. The dynamic MI associated to each pixel represents the *dynamic background model* of it, that is the storing in time of more frequent and up-to-date RGB values assumed by that pixel in video frames. The *RF* mechanism (see Sect. 3) allows to dynamically adapt the MI in such a way that, during the video timeline, not up-to-date RGB values gradually disappear from the background model while new and stable colors of recent frames

¹Just to mentions a fews: physical models, statistical methods with Gaussian mixtures, pixel clustering, image filtering (Kalman, Grabcut, etc.), particle filters and neuron network modeling.

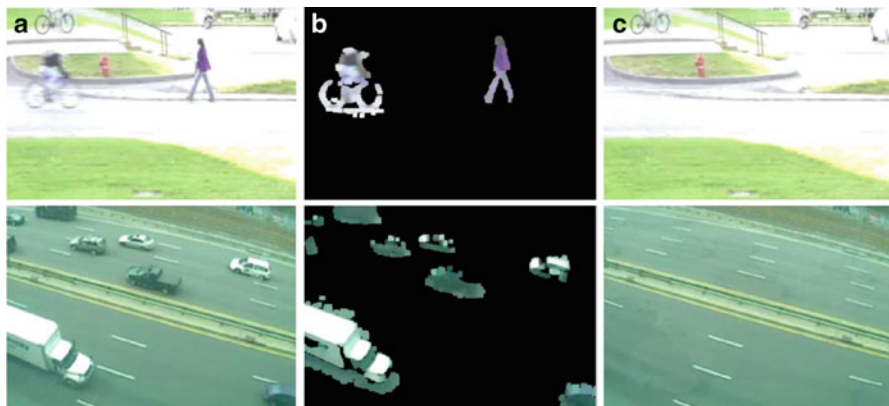


Fig. 4 Outputs of the DRASiW-based CD method: (a) original frame; (b) moving objects highlighting; and (c) MI background model

will contribute more in the background model. The dynamic MI of pixels allows to better adapt the background model to gradual changes in brightness of lights and shadows as well as to natural background noise. Foreground object detection is carried out by evaluating whether the difference between the current pixel color and the stored MI model of the background overcomes a certain threshold. A queue of more recent foreground samples is used to control the time the pixel stays in the foreground. When the queue is full it means that an object was moved to a position of the scene and it has become part of the background.

In Fig. 4 snapshots showing the outputs of the DRASiW-based CD method² are reported. As one can notice, MIs are not only used to fully control the change detection process, but also to filter the input video in order to accomplish two important tasks in video surveillance: 1) moving objects highlighting (see Fig. 4b); 2) subtracting changeable areas from video frames (see Fig. 4c).

4.3 Improving Classification

Activity Recognition aims at identifying the actions carried out by a person given a set of observations of itself and the surrounding environment [7]. Recognition can be accomplished, for example, by exploiting the information retrieved from inertial sensors, such as accelerometers. In some smartphones these sensors are embedded by default and one can benefit from this to classify a set of physical activities (standing, sitting, laying, walking, walking upstairs and walking downstairs) by

²The proposed method participated in the international competition of CD methods on the video repository ChangeDetection.net in 2014, reporting the 3rd best score.

processing inertial body signals through a supervised Machine Learning algorithm for hardware with limited resources [4].

We tried to classify this set of physical activities with DRASiW trained and tested on the HAR³ (Human Activity Recognition) data set of the UCI Machine Learning Repository. The data set consists of 10,299 instances: 7352 for the training set, and 2947 for the test set. Each instance is formed by 561 features with time and frequency domain variables.

The confusion matrix obtained with the best DRASiW system configuration (16-bit addressing for RAM cells) performing an *F-measure* of 89.7, is shown in Table 1a. The confusion matrix of the Table 1b reports the *F-measure* obtained by the same DRASiW system configuration but exploiting the information content of the static MIs. The DRASiW system automatically analyses the MIs to identify features with a very high discriminating power. The analysis outcome is that the six classes can be grouped in three different “metaclasses”: walking (classes 1, 2, and 3), vertical activity (classes 4 and 5), horizontal activity (class 6). This is automatically discovered by the DRASiW system finding out MI overlappings. For the above metaclasses, the MIs have no intersection (no confusion). At this point, when the DRASiW system has to classify a test sample, it first selects the best-matched metaclass, then it classifies the test sample using only discriminators belonging to that metaclass. The result of this new two-level classification approach is that the confusion matrix is now almost diagonalized (see italic values in Table 1b), and the *F-measure* reaches the value of 94.1, that is, the system improved its classification power by 4.4 %.

5 Conclusions

The DRASiW model makes available the learned knowledge in form of an internal data structure called “Mental” Image. This information, which is the synthesis of the learning process, is explicitly available at the programming level and it can be used in several application domains. In this paper we showed how exploitation of MIs, in the context of a DRASiW computational process, allows to pursue different goals/tasks: 1) using global metrics and/or invariants of MIs as additional information (feedback) the system can take advantage of in order to control its functioning (self-healing); 2) verifying the correctness of a training procedure; 3) tuning/adapting classification process by detecting and exploiting more discriminating regions in MIs; 4) facilitating user-system interface and communication.

³<https://archive.ics.uci.edu/ml/machine-learning-databases/00240/>.

Table 1 Confusion matrices on HAR data set: (a) DRASIW; (b) DRASIW with MI processing

(a)							(b)								
Class	1	2	3	4	5	6	Recall	Class	1	2	3	4	5	6	Recall
1 – Walking	477	2	17	0	0	0	96.1	1 – Walking	482	1	13	0	0	0	97.2
2 – Upstairs	29	431	9	0	2	0	91.5	2 – Upstairs	12	457	2	0	0	0	97.0
3 – Downstairs	47	60	310	0	2	1	73.8	3 – Downstairs	18	47	355	0	0	0	84.5
4 – Standing	0	2	0	400	88	1	81.4	4 – Standing	0	0	0	416	75	0	84.7
5 – Sitting	0	0	1	33	498	0	93.6	5 – Sitting	0	0	0	7	525	0	98.7
6 – Laying	0	0	1	0	2	534	99.4	6 – Laying	0	0	0	0	0	537	100
Precision	86.2	87.1	91.7	91.9	84.4	99.6	89.7	Precision	94.1	90.5	95.9	98.4	87.5	100	94.1

We are aware that the natural unfolding of this work is looking for new ways of using MIs in the context of neurosymbolic systems. Indeed, this is the main investigation direction we will pursue in the next future on this topics. Although it would be nice to have a general formalism and/or (rule-based) high-order language to express the information contained in MIs, we are afraid that any choice would be inevitably effective only in a specific (or class of) problem domains.

References

1. Aleksander, I., De Gregorio, M., França, F.M.G., Lima, P.M.V., Morton, H.: A brief introduction to weightless neural systems. In: Proceedings of the 17th European Symposium on Artificial Neural Networks, pp. 299–305 (2009)
2. Aleksander, I., Morton, H.: An Introduction to Neural Computing. Chapman & Hall, London (1990)
3. Aleksander, I., Thomas, W.V., Bowden, P.A.: WISARD a radical step forward in image recognition. *Sensor Rev.* **4**, 120–124 (1984)
4. Anguita, D., Ghio, A., Oneto, L., Parra, X., Reyes–Ortiz, J.L.: A public domain dataset for human activity recognition using smartphones. In: Proceedings of the 21st European Symposium on Artificial Neural Networks, pp. 437–442 (2013)
5. Bouwmans, T.: Recent advanced statistical background modeling for foreground detection: a systematic survey. *Recent Patents Comput. Sci.* **4**(3), 147–176 (2011)
6. Burattini, E., De Gregorio, M., Tamburrini, G.: Generation and classification of recall images by neurosymbolic computation. In: Proceedings of the 2nd European Conference on Cognitive Modelling, pp. 127–134 (1998)
7. Davies, N., Siewiorek, D.P., Sukthankar, R.: Activity-based computing. *IEEE Pervasive Comput.* **7**(2), 20–21 (2008)
8. De Gregorio, M.: On the reversibility of multi-discriminator systems. Technical Report 125/97, Istituto di Cibernetica, CNR (1997)
9. De Gregorio, M., Giordano, M.: Change Detection with Weightless Neural Networks, *IEEE Change Detection Workshop – CVPR 2014*, pp. 403–407 (2014)
10. Feng, T.J., Houkes, Z., Korsten, M., Spreeuwiers, L.: Internal measuring models in trained neural networks for parameter estimation from images. In: IPA, pp. 230–233 (1992)
11. Grieco, B.P., Lima, P.M.V., De Gregorio, M., França, F.M.G.: Producing pattern examples from “mental” images. *Neurocomputing* **73**(79), 1057–1064 (2010)
12. Ludermir, T.B., Carvalho, A.C., Braga, A.P., Souto, M.C.P.: Weightless neural models: a review of current and past works. *Neural Comput. Surv.* **2**, 41–61 (1999)
13. Mantas, C., Puche, J., Mantas, J.: Extraction of similarity based fuzzy rules from artificial neural networks. *Int. J. Approx. Reason.* **43**(2), 202–221 (2006)
14. Mc Ivor, A.: Background subtraction techniques. In: International Conference on Image and Vision Computing New Zealand, IVCNZ (2000)
15. Neagu, C.D., Palade, V.: Neural explicit and implicit knowledge representation. In: Proceedings of the 4th International Conference on Knowledge-Based Intelligent Engineering Systems and Allied Technologies, vol. 1, pp. 213–216 (2000)
16. Panahi, S., Sheikhi, S., Hadadan, S., Gheissari, N.: Evaluation of background subtraction methods. In: *Digital Image Computing: Techniques and Applications*, pp. 357–364 (2008)
17. Piccardi, M.: Background subtraction techniques: a review. In: *IEEE International Conference on Systems, Man and Cybernetics*, pp. 3099–3104 (2004)
18. Sato, M., Tsukimoto, H.: Rule extraction from neural networks via decision tree induction. In: *International Joint Conference on Neural Networks*, vol. 3, pp. 1870–1875 (2001)

19. Soares, C.M., da Silva, C.L.F., De Gregorio, M., França, F.M.G.: Uma implementação em software do classificador WiSARD. In: 5th SBRN, pp. 225–229 (1998)
20. Staffa, M., Rossi, S., Giordano, M., De Gregorio, M., Siciliano, B.: Segmentation performance in tracking deformable objects via WNNs. In: Robotics and Automation (ICRA) (2015)
21. Towell, G., Shavlik, J.: Extracting refined rules from knowledge-based neural networks. *Mach. Learn.* **13**(1), pp. 71–101 (1993)
22. Yilmaz, A., Javed, O., Shah, M.: Object tracking: a survey. *ACM Comput. Surv.* **38**(4), pp. 1–45 (2006)