ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 23-25 April 2014, i6doc.com publ., ISBN 978-287419095-7. Available from http://www.i6doc.com/fr/livre/?GCOI=28001100432440.

Online tracking of multiple objects using WiSARD

Rafael Lima de Carvalho[1,3], Danilo S. C. Carvalho[1] , Félix Mora-Camino[4], Priscila V. M. Lima[2], Felipe M. G. França[1] *

1 – COPPE, 2 – iNCE, Universidade Federal do Rio de Janeiro, BRAZIL

3 - Universidade Federal do Tocantins, UFT, BRAZIL

4 - Ecole Nationale de l’Aviation Civile - Laboratoire d’Automatique, FRANCE

Abstract. This paper evaluates the WiSARD weightless model as a classification system on the problem of tracking multiple objects in real-time. Exploring the structure of this model, the proposed solution applies a re-learning stage in order to avoid interferences caused by background noise or variations in the target shape. Once the tracker finds a target at the first time, it applies only local searches around the neighbourhood in order to have fast response. This approach is evaluated through some experiments on real-world video data.

1 Introduction

Tracking objects in real-time is an important and challenging task, useful for many applications. Among its challenges, the real-time requirement is an obstacle for many off-the-shelf tracker solutions due the high cost of processing. Therefore, a fast tracker solution with exploration of the total observed area is decisive for such type of application. The exploration of the whole image area has a high cost, so using parallel approaches sounds an interesting way to achieve it.

SanMiguel et al. [1] proposed a framework for video tracking algorithms quality estimation, which features the capability of evaluating video trackers with multiple failures and recoveries over long sequences. Percini and Del Bimbo [2] presented a tracking method that uses multiple instances of scale invariant local features, and a non parametric learning algorithm based on the transitive matching property, showing state of the art tracking performance on public available benchmark datasets. In [3], the WiSARD model has been successfully used by an artificial vision system in order to follow the cadence of ships, implying in a model of the movement of an observed vessel.

In this work, a part of [1] is used as the means of evaluating tracker accuracy. Besides, the adopted methodology takes the opposite approach of [2] by adopting a minimum number of features, focusing on portability and speed of the tracker. Moreover, the objective of this paper is evaluate the use of WiSARD neural model, taking advantages of its structure in order to overcome the real-time requirement of an on-line tracker for general objects.

*The authors acknowledge the Brazilian Agencies CAPES and CNPq for funding parts of this research.

This paper is organised as follows. The WiSARD models is presented in section 2. In following, the section 3 describes how we used the WiSARD as the classification system of the proposed tracker application. Then, the experimental setup followed by some results are presented in section 4. Finally, in section 5 some conclusions and future improvements are pointed.

2 WiSARD

The WISARD is a weightless neural network model conceived, initially, for bill recognition and to be implemented in hardware. WiSARD stands for Wilkie, Stonham and Aleksander's Recognition Device [4]. The model has its neuron unit based in the RAM memory. In training mode, this RAM memory stores "1" on its memory position addressed by the binary input pattern (the non-addressed entries remain "0"). While in classifying mode, the RAM outputs the value addressed by its input. Thus, a RAM fires when the input pattern addresses a value equals to "1".

As an advantage over the McCulloch and Pitts neuron model, the RAM-neuron is enabled to learn any Boolean function. However, the neuron itself has no generalisation capabilities. Thus, the simplest weightless neural network with such ability is known as *discriminator*. A discriminator is a single layer network with K RAM neurons capable of handling KN inputs. Therefore, a WiSARD network is composed of a set of these discriminators. Each one is responsible for classifying a different pattern. This way, the WiSARD network has the same number of inputs as its discriminators. Figure 1 illustrates the architecture of the WiSARD's discriminator.

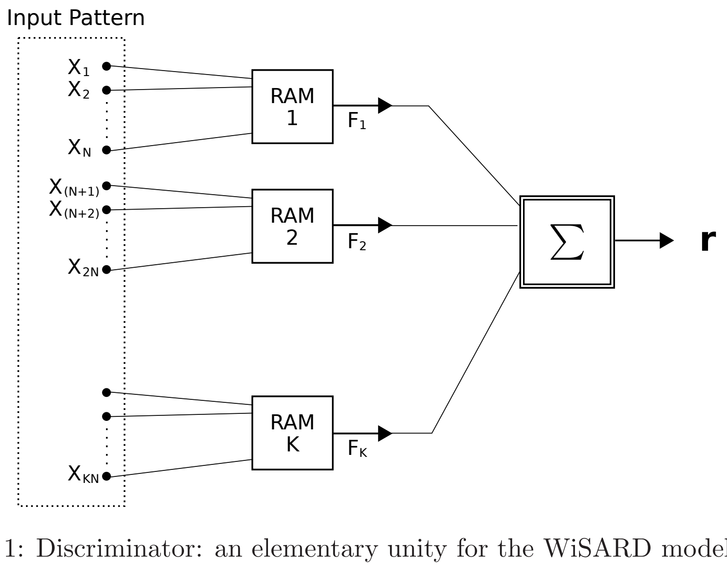

Fig. 1: Discriminator: an elementary unity for the WiSARD model [4].

3 A Weightless Tracker

The proposed WiSARD tracker is composed of components called unit trackers (UT), which holds information about the bounding box of the object to be tracked and the WiSARD instance. In this tracker, the quantity of UTs is defined *a priori*. Each UT has also two search algorithms: a global one, which is responsible for searching over the entire image; and a local one, which searches around a local neighbourhood.

In the beginning of the tracking task, the proposed solution requests all of its tracker units to perform a global search. After that, a local search is invoked, in order to improve the accuracy of the global search answer. If the resulting unit answer is greater than a minimal threshold, the location is stored in a history list.. When the second input image is presented, the tracker retrieves the last history entry, updating the bounding box position through local searches only.

As presented in section 2, the original RAM-based neuron of the WiSARD model stores a binary information about the presence or absence of a determined address (pattern). The RAM-based neural networks are subject to the overtraining problem. If the training set has many different patterns, most of RAMs composing the network may fill all (or almost all) available addresses. This event is called *saturation* and makes the network loose its classifying capabilities.

In order to overcome the overtraining problem, [5] proposed a WiSARD extension called DRASiW, which stores the RAM addressing frequency. This approach allows one to know which parts of the pattern (sub-patterns) happens more frequently. Furthermore, the remaining task is to isolate the relevant sub-patterns from the others [6]. A *bleaching* process is shown by [7], which proposes to accomplish this task by using the frequency information as a filter for the RAM-neuron fire mechanism. When using this filter, a RAM is able to fire only when the frequency of the input address is greater or equal to a threshold, known as *bleaching threshold*.

During the tracking process, in addition the background and luminance disturbances, the moving objects tend to change their shape over time. In order to address this problem, two re-learning algorithms are proposed: *byMean*, which triggers the re-train procedure when the mean of the answers is less than a threshold, and *byDiff*, which accounts for the historical differences between answers in a buffer. The algorithm *byDiff* calls the re-learn procedure whenever the buffer reaches a given size and the sum of differences is greater than a threshold. Both algorithms increase the bleaching threshold before re-training.

Two local search algorithms have been developed as well: *linearLS* and *probLS*. The former explores the whole local neighbourhood delimited by a number n of pixels around a given window position. The latter randomly chooses p different points (rounds) around the neighbourhood also delimited by n pixels. Two global search algorithms have also been proposed. The first one, identified as *stepGS*, slides the unit window over the whole image, moving by n pixels at each step. Finally, the other one, identified as *threadedGS*, uses a grid of $n \times m$

instances of the *probLS* algorithm, associating each instance with a different thread.

4 Experimental setup and results

The proposed tracker has been evaluated using a part of the CAVIAR dataset[1] (the result of only one dataset is shown in this paper). Each image of this dataset has 384x288 pixels and before processing a frame, the tracker binarizes the input image using a luminance threshold. The tracker quality evaluation is done by calculating the amount of intersection between the bounding boxes given by the ground-truth data and the ones given by the tracker.

In order to evaluate the tracker, we assembled sixteen different combinations of the algorithms described in section 3, identified by *config0* to *config15*. The combinations of searching algorithms are divided into two pairs, as follows. The *searchPair1* has the global search *threadedGS* (16×16 *probLS*s, neighbourhood of 10 pixels, and 25 rounds), and the *linearLS* (neighbourhood of 10 pixels) as the local search algorithm. The *searchPair2* has the global search *stepGS* (step of 5 pixels), and the *probLS* (neighbourhood of 20 pixels and 40 rounds) as the local search algorithm.

Configs from 0 to 3 use the *searchPair1* and the *byDiff* (threshold equals to 0.08 and history size $n = 3$) re-learning strategy. Configs from 4 to 7 use the *searchPair2* and the *byDiff* (threshold equals to 0.08 and history size $n = 1$) re-learning strategy. Configs from 8 to 11 use *searchPair1* and the *byMean* (threshold equals to 0.08) re-learning algorithm, while configs from 9 to 15 use the *searchPair2* with the same re-relearning algorithm. The parameters used by the algorithms composing the configurations were found empirically.

We ran the tracker application, using the 16 aforementioned configurations, on an Intel(R) Core(TM) i7-3770 CPU 3.40GHz processor. We have used the UNIX time application to measure the processing time. Table 1 summarises the results by showing the mean and standard deviation of each tracked target, as well as the quantity of Frames per Second for each configuration. Figure 2 shows an output image produced by the tracker.

[1]EC Funded CAVIAR project/IST 2001 37540, found at `http://homepages.inf.ed.ac.uk/rbf/CAVIAR/`

ESANN 2014 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 23-25 April 2014, i6doc.com publ., ISBN 978-287419095-7. Available from http://www.i6doc.com/fr/livre/?GCOI=28001100432440.

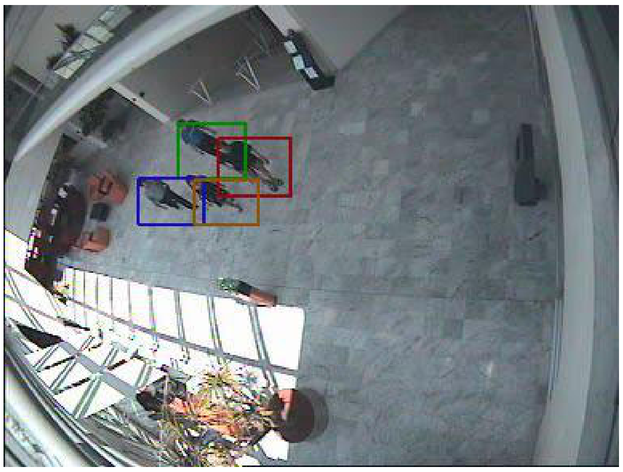

Fig. 2: An image from CAVIAR dataset and the bounding boxes drawn by the tracker.

Setup	Obj1		Obj2		Obj3		Obj4		FPS
	μ	σ	μ	σ	μ	σ	μ	σ	
config0	0.74	0.30	1.00	0.00	1.00	0.07	0.86	0.29	48
config1	0.70	0.21	0.98	0.10	0.76	0.17	1.00	0.04	56
config2	0.58	0.26	0.94	0.22	0.87	0.32	0.97	0.07	55
config3	0.87	0.20	0.95	0.20	0.69	0.25	0.98	0.05	52
config4	0.86	0.27	0.92	0.22	0.71	0.40	0.99	0.04	162
config5	0.68	0.19	0.94	0.18	1.00	0.01	0.94	0.14	166
config6	0.65	0.31	0.92	0.22	0.70	0.38	0.57	0.34	165
config7	0.74	0.29	0.88	0.23	0.82	0.36	0.79	0.26	158
config8	0.90	0.10	0.91	0.23	0.87	0.29	0.99	0.04	52
config9	0.64	0.24	1.00	0.00	0.52	0.31	0.97	0.10	56
config10	0.70	0.25	0.97	0.11	0.89	0.28	0.98	0.05	56
config11	0.62	0.26	0.94	0.20	0.88	0.28	0.94	0.14	51
config12	0.63	0.31	0.99	0.03	0.92	0.31	0.95	0.11	164
config13	0.68	0.25	0.93	0.19	0.70	0.42	0.97	0.08	170
config14	0.80	0.28	0.88	0.22	0.68	0.41	0.86	0.22	161
config15	0.86	0.20	0.58	0.22	0.80	0.33	0.82	0.26	153

Table 1: Summary of results.

The optimal values for the tracking quality measure are the mean equals to 1 and the standard deviation tending to 0, meaning the window is over the object all the time. Values lesser than 1 indicate the tracker has lost its target (or part of it) during the tracking time.

5 Final Remarks

In this paper, the WiSARD model was evaluated as a solution in an on-line multi-object tracking application. Despite the use of a very limited input information (binarized image), the WiSARD model has shown promising results towards an adequate classifying algorithm for this type of application. Nevertheless, more data sets should be tested in order to validate the proposed system. In the configurations shown in section 4, the configuration12 presented good results in both track quality and quantity of frames per second. Despite fulfilling the realtime requirement, the tracker has still lost the tracked objects for some frames.

A positive factor of this model is the bleaching process who plays an important role in the re-learning stage. However, the re-learning rules currently present in this tracker are too simple. One of the observed problems of this model is loosing targets, in the presence of occlusions. In this case, target loss occur due to two reasons: the neural network answer be slightly better in a position other than the occluded object; and for delays in the re-learning trigger. As future improvements, smarter rules should be investigated in order to get to the right moment of invoking the re-learning stage. Future research directions include the development of algorithms for auto adjustment of the tracker parameters and the use of grayscale images such as the ranking algorithm proposed in [8].

References

[1] J. SanMiguel, A. Cavallaro, and J. Martínez. Adaptive online performance evaluation of video trackers. *IEEE Transactions on Image Processing*, 21(5):2812 – 2823, 2012.

[2] Federico Pernici and Alberto Del Bimbo. Object tracking by oversampling local features. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, in press, 2014.

[3] H.L. França, J.C.P. da Silva, M. De Gregorio, O. Lengerke, M.S. Dutra, and F.M.G. França. Movement persuit control of an offshore automated platform via a ram-based neural network. In *Control Automation Robotics Vision (ICARCV), 2010 11th International Conference on*, pages 2437 –2441, dec. 2010.

[4] Igor Aleksander and Helen Morton. *An introduction to Neural Computing*. Thomson Computer Press, Berkshire House, London, UK, second edition edition, 1995.

[5] Massimo de Gregorio. On the reversibility of multi-discriminator systems. Technical Report Technical Report 125/97, Istituto di Cibernetica–CNR, 1997.

[6] D. S. Carvalho, H. C. C. Carneiro, F. M. G. França, and P. V. Lima. B-bleaching: Agile overtraining avoidance in the wisard weightless neural classifier. *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, pages 515–520, April 2013.

[7] Bruno P.A. Grieco, Priscila M.V. Lima, Massimo De Gregorio, and Felipe M.G. França. Producing pattern examples from "mental" images. *Neurocomputing*, 73(7–9):1057 – 1064, 2010.

[8] Kazimali M. Khaki. *Weightless Neural Networks for Face and Pattern Recognition: an evaluation using open-source databases.* PhD thesis, Brunel University, London, 2013.