**Proceedings of the 2007 IEEE Symposium on
Foundations of Computational Intelligence (FOCI 2007)**

Artificial immune systems based novelty detection with CNN-UM

György Cserey and Tamás Roska
Faculty of Information Technology, Péter Pázmány University,
and Computer and Automation Research Institute of
the Hungarian Academy of Sciences, Budapest, Hungary
Email: cserey, roska@itk.ppke.hu

***Abstract*— In this paper, we show that our earlier presented immune response inspired algorithmic framework [1], [3] for spatial-temporal target detection applications using CNN technology [9], [17], [10] can be implemented on the latest CNN-UM chip (Ace16k) [16] and Bi-i system [15]. The implementation of the algorithm is real-time and able to detect novelty events in image flows reliably, running 10000 templates/s with video-frame (25 frame/s) speed and on image size of 128x128. Besides that some results of the implementation of this AIS model and its application for natural image flows are shown, the realized adaptation and mutation methods are also introduced.**

I. Introduction

Every day it is worth wondering over the beauties of Nature, life and its biological processes, came to that the complexity of only one cell. We can wonder and admire, however we will never be able to understand it fully. We often take courage, with our humbleness is on the small side, to copy or horn in its processes. The alibi of Medicine is simple: the protection of human life. An engineer can offer his knowledge to the doctor, and give better instruments to him or find ideas from studying medicine and apply them to provide more effective solutions of engineering problems. Artificial Immune Systems is a growing research area where engineers would like to get ideas while studying the processes of our immune system.

Artificial Immune Systems (AIS) mimic the human immune system that has refined capabilities and methodologies, to build efficient algorithms that solve engineering problems. Moreover, our immune system possesses important properties, such as diversity, noise and fault tolerance, learning and memory and self-organization, which give it an advantage compared to other standard methods [11], [8], [12].

The cell-level interaction of immune system is based on identification and recognition of 3D molecule patterns. During our research the object we proposed is a creation of a model, which, similarly to the 3D spatial pattern detection of the immune system, is able to detect and recognize dynamic objects in 2D image flows. We intended to design topographical algorithms and their experimental realization where huge number of target objects are monitored in real time to detect previously unknown events. So our goal was spatial-temporal novelty detection.

Novelty detection is the identification of new or unknown data or signal that a machine learning system is not aware of during training. Novelty detection is one of the fundamental requirements of a good classification or identification system since sometimes the test data contains information about objects that were not known at the time of training the model.

Novelty detection can be a challenge in several areas. Nowadays, one of these areas is robotics, where novelty detection - the differentiation of the general sensor input and the sensory pattern not yet experienced - provides useful knowledge to mobile robots in a dynamically changing environment [18].

Basically, there are two approaches of novelty detection: statistical based and neural network based approaches [13], [14].

Statistical approaches are mostly based on modeling data based on its statistical properties and using this information to estimate whether a test samples comes from the same distribution or not [13].

Neural networks have been widely used for novelty detection. Compared to statistical methods, some issues for novelty detection are more critical to neural networks such as their ability to generalize, computational expense while training and further expense when they need to be retrained [14].

To compare our work to the research areas of novelty detection, we can observe that it is rather statistical approach, because it has similar properties as statistical approaches have: easily re-trainable and the evaluation of the algorithm is based on the sub-patterns of the images.

Sensor-close computation can be crucial from the point of view of utilization efficiency, since it could help solving some of the general problems of traditional systems, namely the reduction of the bandwidth of image transfer from the sensor to the computational unit and the time needed to process images in real-time.

Computer architectures based on the *cellular nonlinear/neural network* (CNN: Cellular nonlinear/neural networks are regular, single or multi-layer, parallel processing structures with analog nonlinear dynamic units (cells). The state value of the individual processors is continuous in time and their connectivity is local in space.) paradigm and its 128×128 sized VLSI implementations offer adequate solution for high-speed pattern matching. In our algorithms designed for CNN Universal Machine (*CNN-UM*: A cellular wave computer architecture that includes CNN dynamics as its main instruction. The CNN-UM makes it possible to efficiently combine analog array operations with local logic.) wave computers we used

already published template classes (Template: The program of CNNs is completely determined by the pattern of the local interactions, the so-called template). It was an important aspect of the chosen templates being able to be executed reliably on available CNN-UM hardware systems. In the course of the implementation of our analogical CNN algorithms we intended to raise efficiency, using both CNN and conventional digital solutions and implemented the most suitable algorithmic steps on the appropriate machine.

The organization of this paper is the following. First we present the algorithm based on our earlier work to make this paper more understandable. Section 3 introduces the adaptation and mutation methods. Section 4 descusses the main foundational aspects of the presented system. In Section 5, we show some Ace16k experiments. Conclusions can be found in Section 6.

II. Algorithm framework

Although the details of our algorithm and its subroutines can be found in [1], [4], [5], we shortly summarize our framework to the reader following the layered approach which is often used for designing artificial immune systems [6], [8]. The basis of every system is the application domain. For this domain, the representation of the components of the system has to be considered. Having a suitable representation, the interactions of the elements of the system have to be quantified by one or more affinity measures. Usually Hamming or Euclidean distances is chosen, but it is dependent upon the representation. The next layer represents the processes or algorithms and subroutines to govern the dynamics of the system.

A. Representation

Our model defines the antigens and T-lymphocytes as two data items with different characteristics and goals. The antigens can be represented by $n \times n$ sized binary (black and white) matrixes. Colors can be coded with *1* (black) and *-1* (white) numbers. Each antigen is usually a 3×3 or 5×5 subpattern of a binary picture which is extracted from the input image flow by a special feature extraction method [2]. These patterns can be recognized by our T-lymphocytes, called match-templates [19]. They are usually 3×3 or 5×5 matrixes (On Ace16k chip we could use only 3×3 sized templates.) and contain *1*, *-1* and *0* numbers.

B. Affinity measurement

In our model the *S* shape-space has 9 or 25 dimensions, because the sub-pattern matrixes can be represented by 25 or 9 long binary vectors and the templates correspond to 25 or 9 long vectors (coordinates can be -1, 1, or 0). The distance measure between an antigen ($Ag = \langle Ag_1, Ag_2, ...Ag_L \rangle$) and a template $T = \langle T_1, T_2, ...T_L \rangle$ is

$$D = \sum_{i=1}^{L} \delta_i, \text{ where } \delta_i \begin{cases} 0, & \text{if } T_i = Ag_i \text{ or } T_i = 0, \\ 1, & \text{if } T_i \neq Ag_i \end{cases} \quad (1)$$

Our match template class does the recognition if and only if the *D* distance is *0*. Contrary to common AIS, where the molecules usually are represented by similar vectors, the sub-patterns and template vectors generally differ in our model. There are "don't care" elements in the templates, whose positions are fixed within their vectors. Therefore, we could not give a definition of affinity as other AISs have. If a match-template has d "don't care" elements, it can detect 2^d different sub-patterns. The more "don't care" elements it has, the more different sub-patterns are, which are detected. Therefore, the affinity of a template can be characterized by the number of the "don't care" elements. This affinity is called template affinity α. This affinity has a similar effect to the usual affinity or cross-reactivity threshold in AIS. A sub-pattern can be matched successfully by $2^L = \sum_{\alpha=0}^{L} \binom{L}{\alpha}$ different match-templates, where α is the affinity, defined formerly. The maximum number of sub-patterns that can be recognized by a template set is $\sum 2^{\alpha_i}$, where the α_i is the template affinity of ith template of the template set.

C. Algorithm and its subroutines

The immune system endeavors to solve a target detection problem where the objects to be detected are not predetermined. These objects can be numerous. The system has to react as quickly as possible to distinguish between non-pathogen and pathogen objects to protect the body from the pathogens.

The algorithm has two parts – initialization and recognition which sequentially follow each other. In the course of initialization, a "non-dangerous" template set (T cells) is created. This template set contains templates which are not able to recognize the initial objects – the non-dangerous objects. This process is called negative selection. The output of the negative selection – templates – performs the recognition in the second part. The randomly chosen templates (lymphocytes in the bone marrow) are tested (in the thymus) against a pattern flow which is extracted (by the antigen presenting cells) from the initial 2D image flow. The feature extractor module converts the gray-scale input flow to binary. Those templates, which do not match any of the patterns, are selected as the "non-dangerous" ones.

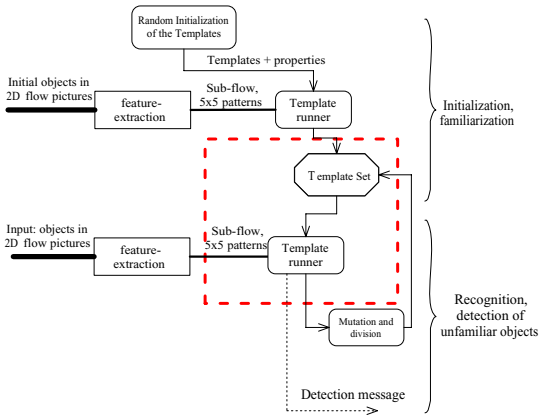

Fig. 1. Target detection CNN algorithm [1]

In the recognition phase, every member of the selected

template-set tests the actual pattern extracted from the input image flow of the recognition part, and if it recognizes the unfamiliar pattern then a detection message is generated. These templates are called prosperous and get higher priority, and thus have more opportunities to recognize patterns. The mutation and division module operates on the templates and its sequence to improve the effectiveness of the algorithm. More details can be found in the next section.

III. Mutation subroutines

Emphasizing the significance of the mutation of the immune system [7] we would like to mention three motivations why it was applied in our implementation.

Increased affinity: Increased affinity will increase the chance of binding with the antigen, and therefore increase the efficiency of the attack by the immune cells. If a pathogen is associated with multiple patterns it is very well possible that a T-Cell that recognizes one of these shades does not recognize the other one.

Memory: Some of the most successful immune cells will mutate into long lived memory cells. This effect serves mainly to increase the speed of the immune response for latter invasions of the same pathogen. For this reason, in our model the T-Cells have to be equipped with some sort of activity flag showing how effective they have been in recognizing pathogenic patterns.

Set Completion: Due to the great number of different immune cells, it is hardly ever possible to cover the whole space with the actual immune cell set. Having some form of rotation in non-recognizing T-Cells by mimicking cell death and the spawning of new randomly created T-Cells will assure set completion over a larger span of time.

A. Implementation

In the next part of this section two methods of the implementation mutation are described.

Affinity Maturation: When a template is successful in the recognition of a pattern, there is a certain chance that it will generate a mutation. Mutation candidates are generated as follows: If a template is general (i.e., has multiple zeros) one of the zeros of the template will be mutated into either a +1 or a -1, making the mutation more specific. If it is already specific ($d < 2$), the mutation will just be random. The generated mutation candidate is tested on the input antigen to see whether it is more successful in recognition than the original version. The success of the template is determined by the number of matches found in the antigen. If the mutation candidate is found to be more successful than the original, it is tested to see whether it is triggering a response to the self set. In case the mutation is accepted, the original will be replaced. In case a test on the mutation candidate fails, the mutation will be rejected, and the candidate is removed (cell death). Since we want the system to be running in real time it is not feasible to run the whole self set each time we want to perform a mutation. Therefore this test is spread out over time. After every input image the mutation candidates will be tested on one of the images from the self set. Affinity maturation realizes increased detection efficiency, and makes the system more suitable for implementing pathogen recognition. The whole process of mutation for successful templates is shown as a flow chart in Fig. 2 and Fig. 3.

Set Completion: The total number of patterns recognized by the entire template set is kept in a variable. When this value drops below a threshold related to the total number of possible patterns, new random templates will be spawned. Of course, these templates are tested against the self set before being allowed into the pool.

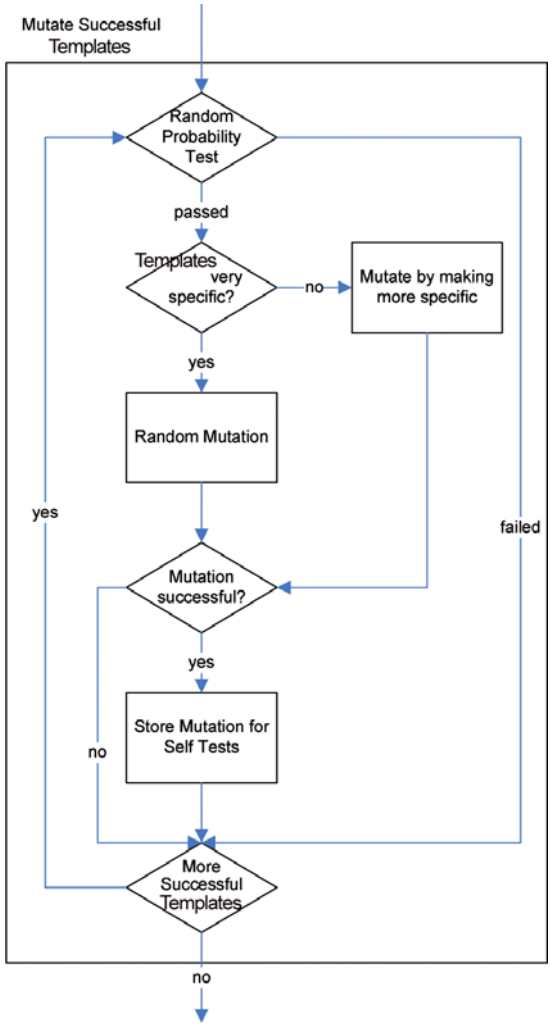

Fig. 2. Flow chart of the realized mutation procedure for affinity maturation. It shows the actual mutation procedure block.

The need for having mutation in the natural immune system different from the need for it in our CNN model. The possible goals for mutation are concluded to be increasing efficiency, maintaining set completion, realizing memory, and creating specific pathogen recognition.

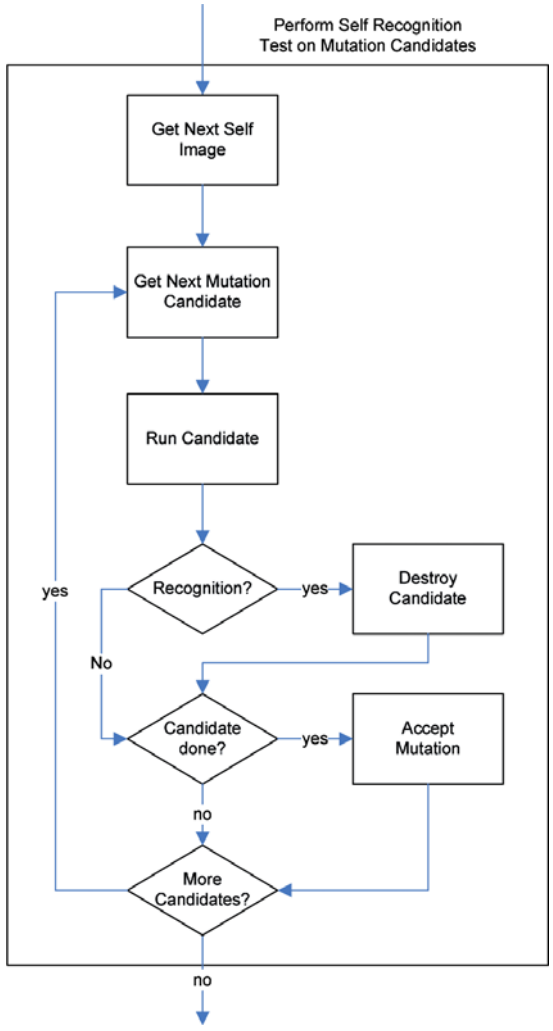

Fig. 3. Flow chart of the realized mutation procedure for affinity maturation. It shows the details of the self recognition test for mutation candidates.

IV. Foundational aspects

Our system was designed to detect and recognize dynamic objects in 2D image flows in real time. Analyzing this solution basically there are two main aspects of this system. The difference is not only hardware and software, but they are theoretically different areas. As it is described in section 2 and 3, the algorithm basically follows the design steps of artificial immune systems, which provide several useful features. The algorithm was implemented on CNN-UM architecture, where well designed image processing algorithms can be executed with high speed. Let us itemize the issues and properties of our framework from both AIS and CNN aspects.

A. AIS issues

- The modules of the algorithm are inspired by the processes of the immune system. E.g. production of lymphocytes in the bone marrow, affinity maturation during mutation or negative selection in thymus.
- Significances of the mutation are increased affinity, memory and set completion as described above.
- The representation can be given by using adequate mathematical description.
- Its affinity measurement can be described in general AIS form, but some properties are different as it is discussed in section 2.

B. CNN issues

- The algorithm is well suited to run on CNN-UM architecture. The kernel of the algorithm - which has the biggest computational cost - is designed for effective evaluation. The combination of the parallel processing and fast execution result in real time running.
- The chosen match template class can be run reliably on Ace16k type chips which are available on the market.
- The implementation of the algorithm exploits the architecture advantages of Bi-i systems.

V. Ace16k experiments

A. Bi-i (Bio-inspired) Architecture

The Bi-i architecture [15] is built on Ace16k CNN type and Texas DSP type microprocessors. This system is standalone and has a communication processor which is able to exchange information over a 100 Mbit/sec network using TCP/IP protocol. In between different processing stages the algorithmic framework of the Bi-i contains several automatic control and feedback mechanisms. There is a proper selection of the visual input of a low resolution CNN (Ace16k sensor-processor) and a high resolution CMOS sensor. The algorithm can acquire 128×128 CNN size images from the high resolution CMOS sensor cut by given positions. Besides using the own optical sensor of the CNN chip, an algorithm can navigate in a 1280×1024 high resolution projection of the enviroment at a processing rate of a few thousand frames/sec. The most important attributes of this system are the following: Bio-inspired sensor-computer, fault tolerant, high speed, standalone system [15]. The key component reaching its high performance - the ability of capturing and processing 10000 frames of 128×128 sized images in a second - is the CNN-UM type array processor chip (Ace16k).

B. Experiments

In this section, we describe two experiments, which were presented on Bi-i system and implemented on ACE16k chip. The inputs were real images transferred from the optical interface of the chip. The feature extraction module and the algorithmic template core were implemented on the chip. The evaluation of results, template storage, mutation and administration of their different properties were performed on a Texas digital signal processor. This combination gave better performance with real-time processing.

The first experiment shows dynamic adaptation of the system to the environment. In the rows, from left to right, the order of the images are the following. The first image is the gray-scale input. Its binary converted version is the second

image. You can see the detection points of all the templates in the third image. The last image is the combination of the input and the detection points, where the detection points belong to a template whose number of detection points is higher than a value. From up to down the following can be observed: at the beginning, the system has several detection points, but during learning, the number of detection points decrease. Once objective is covered, the template set was enriched with new templates by the mutation routines. New detection points show that the input has changed again while the covering was removed (Fig. 4).

Fig. 4. The system's dynamic adaptation to the environment. In the rows, from left to right, the order of the images are the following. The first image is the gray-scale input. Its binary converted version is the second image. You can see the detection points of all the templates in the third image. The last image is the combination of the input and detection points, where the detection points belong to a template whose number of detection points is higher than a given number. From up to down the number of frame indexes are 1, 23, 28, 64 in time.

In the second experiment, we show that the implemented system with particular parameters can be appropriate for object recognition and border estimation. The relation between the images is similar as in Fig. 4. From up to down the following can be observed: the first image shows that the system is already adapted to the environment, there is no detection. In the second and third row, two results of an image sequence can be seen, where a palm in front of the camera shows the detection points, mainly on the fingers and upper part of the palm are detected (Fig. 5).

The CNN-UM chip (Ace16k) implementation of our algorithm is able to detect novelty events in image flows reliably, running 10000 templates/s with video-frame (25 frame/s) speed and on image size of 128×128.

VI. Conclusion

Nature has developed a powerful 3D pattern recognizer defense system. Our model was inspired by this, and the results

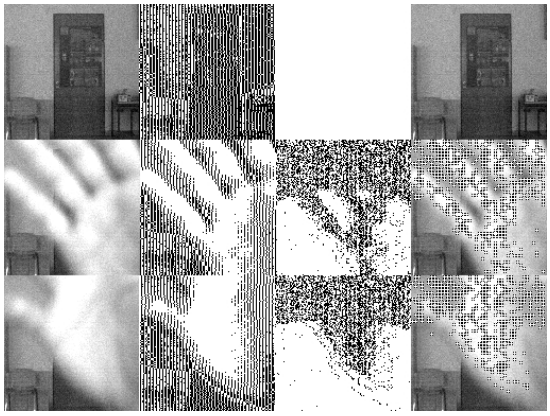

Fig. 5. This figure shows that the implemented system with particular parameters can be appropriate for object recognition and contour estimation. The relation between the images is similar as in Fig. 4. From up to down the number of the frames are 18, 21, 30 in time.

show that it is efficiently usable on 2D patterns (pictures). CNN's spatio-temporal dynamics with fast template processing is an effective tool for modelling the spatio-temporal dynamics of the immune system. The immune system provides a special class of algorithms, covers the target space, is able to learn and has memory. The proposed strategy has been successfully applied in a sample texture analyzer application and gave promising results.

Acknowledgment

The Operational Program for Economic Competitiveness (GVOP KMA) which supports the multidisciplinary doctoral school at the Faculty of Information Technology of the Péter Pázmány Catholic University, and the support of the Computer and Automation Institute of the Hungarian Academy of Sciences are gratefully acknowledged. The authors are also grateful to Professor András Falus and Jasper Stolte for the discussions and their suggestions.

References

[1] Gy. Cserey, A. Falus, and T. Roska, "Immune response inspired spatial-temporal target detection algorithms with CNN-UM," *International Journal of Circuit Theory and Applications*, vol. 34, pp. 21–47, 2006.

[2] Gy. Cserey, A. Falus, W. Porod, and T. Roska, "Feature extraction CNN algorithms for artificial immune systems," in *Proceedings of International Joint Conference on Neural Networks, IJCNN 2004*, vol. 1, (Budapest, Hungary), pp. 147–152, July 2004.

[3] Gy. Cserey, W. Porod, and T. Roska, "An artificial immune system based visual analysis model and its real-time terrain surveillance application," in *Proceedings of International Conference on Artificial Immune Systems, ICARIS 2004*, (Catania, Italy), pp. 250–262, Springer-Verlag, Inc, Sept. 2004. Lecture Notes in Computer Science 3239.

[4] J. Stolte and Gy. Cserey, "Artificial immune systems based sound event detection with CNN-UM," in *Proceedings of European Conference on Circuit Theory and Design, ECCTD '05*, vol. 3, (Cork, Ireland), pp. 11–14, Sept. 2005.

[5] Gy. Cserey, A. Falus, W. Porod, and T. Roska, "An artificial immune system for visual applications with CNN-UM," in *Proceedings of the IEEE International Symposium on Circuits and Systems, ISCAS 2004*, vol. 3, (Vancouver, Canada), pp. 17–20, May 2004.

[6] G. Nicosia, V. Cutello, P. J. Bentley, and J. Timmis, "Artificial immune systems," in *Third International Conference, ICARIS 2004*, vol. 3239, (Catania, Italy), Lecture Notes in Computer Science, Springer-Verlag, Sept. 2004.

[7] G. Nicosia, V. Cutello, G. Narzisi, and M. Pavone, "Real coded clonal selection algorithm for global numerical optimization using a new inversely proportional hypermutation operator," in *The 21st Annual ACM Symposium on Applied Computing, SAC 2006*, vol. 2, (Dijon, France), pp. 950–954, Apr. 2006.

[8] L. N. de Castro and J. Timmis, *Artificial Immune Systems: A New Computational Intelligence Approach*. Springer, 2002.

[9] T. Roska and L. O. Chua, "The CNN Universal Machine," *IEEE Transactions on Circuits and Systems*, vol. 40, pp. 163–173, 1993.

[10] T. Roska, "Computational and computer complexity of analogic cellular wave computers," in *Proceedings of the 7th IEEE International Workshop on Cellular Neural Networks and their Applications, CNNA 2002*, (Frankfurt, Germany), pp. 323–335, July 2002.

[11] D. Dasgupta, *Artificial Immune Systems and Their Applications*. Springer-Verlag, 1999.

[12] S. Stepney, R. E. Smith, J. Timmis, M. J. Tyrell, A. M. Neal, and A. N. W. None, "Conceptual frameworks for artificial immune systems," *International Journal of Unconventional Computing*, vol. 1, no. 3, pp. 315–338, 2005.

[13] M. Markos and S. Sameer, "Novelty detection: a review part 1: statistical approaches," *Signal Processing*, vol. 83, pp. 2481–2497, 2004.

[14] M. Markos and S. Sameer, "Novelty detection: a review part 2: neural network based approaches," *Signal Processing*, vol. 83, pp. 2499–2521, 2004.

[15] A. Zarándy and C. Rekeczky, "Bi-i: a standalone ultra high speed cellular vision system," *IEEE Circuits and Systems Magazine*, vol. 5, no. 2, pp. 36–45, 2005.

[16] A. Rodriguez-Vazquez, G. Linan-Cembrano, L. Carranza, E. Roca-Moreno, R. Carmona-Galan, F. Jimenez-Garrido, R. Dominguez-Castro, and S. Meana, "ACE16k: the third generation of mixed-signal SIMD-CNN ACE chips toward VSoCs," *IEEE Transactions on Circuits and Systems Part I: Regular Papers*, vol. 51, no. 5, pp. 851–863, 2004.

[17] L. O. Chua and T. Roska, *Cellular neural networks and visual computing, Foundations and applications*. Cambridge University Press, 2002.

[18] S. Marsland, U. Nehmzow, and J. Shapiro, "On-line novelty detection for autonomous mobile robots," *Journal of Robotics and Autonomous Systems*, vol. 51, pp. 191–206, 2005.

[19] T. Roska, L. Kk, L. Nemes, A. Zarndy, M. Brendel, and P. Szolgay, "CNN software library (templates and algorithms), version 7.2.," Tech. Rep. DNS-CADET-15, Analogical and Neural Computing Laboratory, Computer and Automation Research Institute, Hungarian Academy of Sciences (MTA SzTAKI), 1998.