Automatic CNN Multi-Template Tree Generation.

V. M. Preciado, D. Guinea, J. Vicente, M. C. Garcia-Alegre and A. Ribeiro.

Instituto de Automatica Industrial- Spanish Council for Scientific Research
La Poveda (Arganda del Rey) - 28500 Madrid, Spain. preciado@iai.csic.es

ABSTRACT: *A fruitful field into the CNN research domain is being the development of analogic algorithms that combine single templates to perform complex image processing. The results would be extremely useful for pattern recognition in industrial and robotic applications. This work presents a general methodology for the automatic generation of analogic algorithms by means of a genetic search. A genetic algorithm for generating multi-template trees, concept derived from the AI field, is applied to the automatic generation of analogic algorithms, based in both genetic-evolutionary search and heuristic approaches.*

1. Introduction

The traditional image processing techniques require a lot of computational effort as pixel information is acquire and processed sequentially and data flow trough an A/D conversion stage. This generates a time delay that is unacceptable for real time image processing in visual tasks requiring the process of several millions of pixels per second (e.g. automatic industrial inspection, visual based navigation in robotics).

A massively parallel architecture that works with analog signals could offer a solution for these applications. This is just the basis idea of Cellular Neural Network (CNN's): an array of analogic dynamic processors whose cells interact directly within a finite local neighborhood [1]. The local CNN connectivity allows its implementation as VLSI chips that can operate at a very high speed, with a high complexity level [2]. Nowadays CNN architectures implemented as VLSI chips shows the aptitude of extremely high speed compared with traditional digital image processing tools. The proliferation of ever more sophisticated CNN architectures, and the great effort observed during last years to implant practical system based on CNN chips, drives the development of analog algorithms able to perform complex image processing tasks required in industrial applications [3], robotic systems, classification [4], and compression [5].

The objective of this work is the generation of a learning machine within the CNN paradigm, capable of finding solutions for complex image processing tasks. First a general machine for automatic analog algorithm design independent of the problem to solve is proposed. It uses an evolutionary strategy that is an extension of the genetic programming [6]. Second, this work introduces a set of sub-mechanisms to increase the power of the genetic programming and to reduce the enormous search space to optimise time. Some concepts are related with AI theory, in such a way that the performed work is at the intersection of AI, Image Processing, and CNNs fields.

Previous works present some CNN simulators with the feature of the automation of single template generation [7][8]. In this line our former work [9] describes an example-based learning method using a Genetic Algorithm for automatic generation of a single template. Current work gives flexibility to former approach proposing a method for the automatic generation of a template sequence instead of a single template GA based.

2. Multi-Template Tree Representation

The main objective of this work is the automatic generation of an adequate template sequence to transform an input image into an output image, using an initial state. This template sequence can be represented by means of a tree formed by templates, namely multi-template tree, Figure 1.

The notation utilized to codify the multitemplate tree , in a way useful for our automatic GA searching, is the *prefix* or Polish notation due to Lukasiewicz. According to this notation an operator *O* which performs an action on two objects *x* and *y* is represented *O x y*. In our case, O consist of a single template, x is an input image and y is the initial state for the CNN differential equation to be solved. No parenthesis are needed and the principle operator for any term appear at the head of that term.

Assuming the general case of binary templates as unary templates are a particular case of binary template having one of its inputs " a priori" set to black, gray or white intensity level. In this way we can express a multi-template tree, as that of Figure 1 by the following expression: *Tem.4(Tem.2(Tem.1(X, B), A), Tem.3(X, A)).*

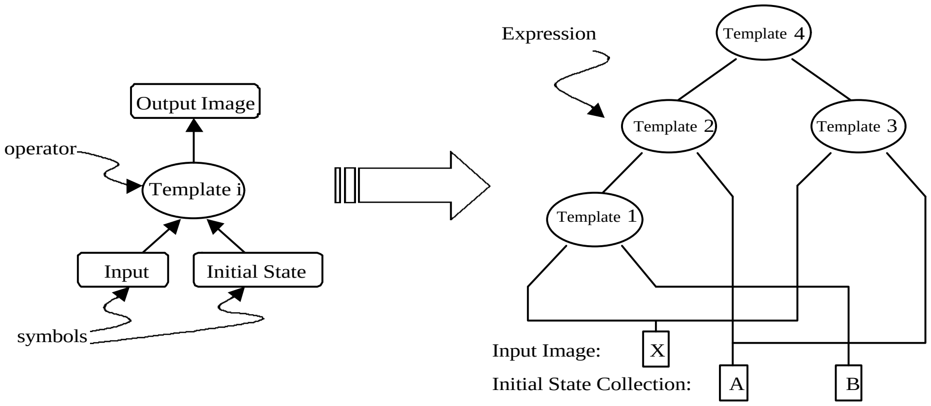

Fig. 1. Tree and node representation

So an expression is a set of images and templates coherently related to perform a complex image processing. Within the same framework, a well-formed expression can itself be regarded as an object, only if it verifies the Rosenbloom theorem:

A sequence of symbols S in prefix notation is a well-formed expression if and only if:
1. rank (S)=-1;
2. rank (sub-expression on the left of S)³0;
where rank is defined by:
rank (binary operator)=1;
rank (unary operator)=0;
rank (constant)=-1;
rank (S1 concatenated with S2)=rank (S1)+rank (S2).

3. The Genetic Algorithm

To search the most adequate template sequence or multi-template tree, a class of learning system called genetic algorithms is used [10]. These GA algorithms are probabilistic search algorithms which simulates natural evolution and are very useful in combinatorial optimization.

In these algorithms the explored search space at each iteration, is called population. The population is formed by a collection of individuals that are represented by a string, which are often referred to as chromosomes. The purpose of using a GA is to find from the search space, the individual with the best “genetic material”. Thus it is necessary to quantify the individual quality and this is performed through an evaluation function, namely fitness function.

In summary, the algorithm firstly chooses the initial population of potential solution and defines the fitness function. Then, in each iteration, the individuals, parents, are selected to produce new individuals, children, of the next generation (which is a new algorithm iteration) by means of the combination of the their genetic material. This genetic material combination is denoted as crossover operation. Then for each new individual, there is a probability close to zero that the individual can “mutate”, resulting in small modifications of their genetic material, called mutation operation.

It is important to remark that the mutation operation is needed to explore new states and prevents the algorithm from local minimum. Crossover tend to increase the average quality of a population, so the selection of an adequate crossover and mutation operators increases the GA probability to reach a near-optimal solution in a reasonable number of iterations.

3.1. Individual Representation and Initial Population

In the proposed method, each individual codifies a multi-templates tree as a string in Polish notation. The initial random population and the offspring produced by each genetic operation must be a "well-formed expression" , verifying the Rosenbloom theorem. Although the individuals are strings in Polish notation, to simplify the following dissertation, a tree representation for each individual is assumed.

Therefore, the way to generate an individual is accomplished by fixing an upper and lower number of possible operations, to be carried on. The probability for a node to be either an operator or an image is given by a probability value Po for a binary operation and $Pi=1-Po$ for an image. Following this pattern, we add new nodes taking into account the former probabilities and testing whether or not the upper and lower operation number are exceeded. The individual lenght is variable, consequently the initial population generation is performed through a list that links new term based upon the probability previously set by the user.

3.2 The Fitness Function

The fitness function is an energy function proportional to the to the difference between pixels from the current output image ($I_{Current}$) and the desired output image ($I_{Desired}$). For each individual t, the fitness function is expressed as,

$$f(t) = \sum_{pixels} \left| I_{Desired} - I_{Current}(t) \right|$$

which has to be minimized in the GA search process.

3.3. Genetic operators

The GA operators, crossover and mutation can be defined as follows, *Figure 2*:

Crossover operator. A crossover point is randomly chosen in the first and second parent Then the subtree rooted at the crossover point of the first parent is eliminated and replaced by the subtree coming from the second parent. Crossover is the predominant operation and in our proposal it acts with a high probability about 0.85-0.90.

Mutation: The mutation operation used is the one defined by Koza [6]. The individual is probabilistically selected from the population and a point is randomly chosen, then the subtree rooted at that point is erased, and a new subtree is generated using the same random growing process used to originate the initial population. This mutation operation is performed sparingly. The probability of the mutation is 0.01, at each iteration.

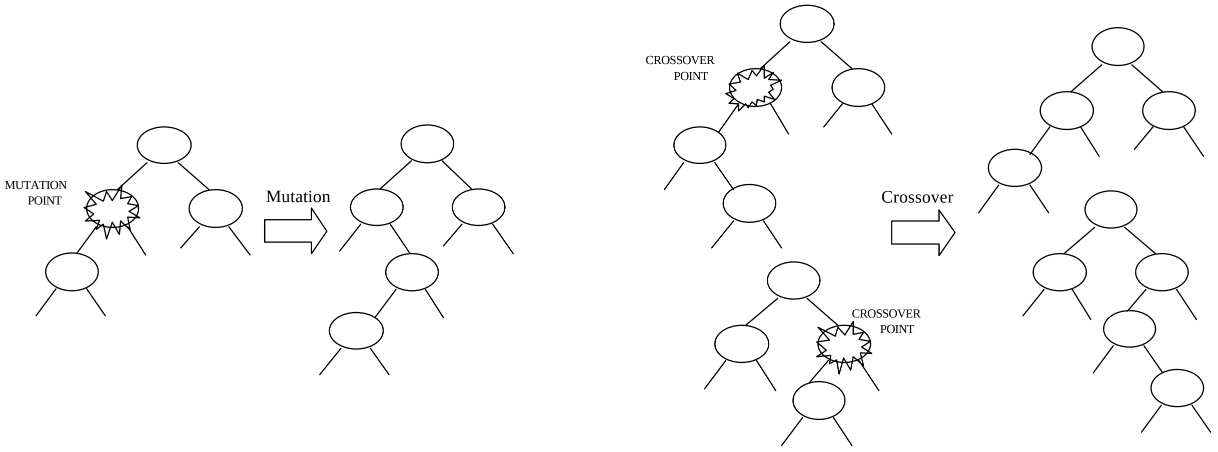

Fig.2. Mutation and Crossover operators.

4. Search Space Size and its Prune by means of Heuristic Methods

Under the hypothesis of binary templates we can affirm that in a tree formed by S templates there are S+1 free branches that can be filled by other terms (in general other trees). It can be demonstrated, as far as the initial root of a tree is an binary template, Figure 3a, and the growth process is performed by adding new templates located in each of these branches, Figure 3b and 3c. Each new template erases one of these free branches and produce two new ones. Accordingly, for each new template, a new branch appears and the initial difference between templates and branches remains the same. Beyond that, any tree of S templates can produce S+1 trees by adding one new operator, so the number of possible tree with S operators is a factorial function of S, S!.

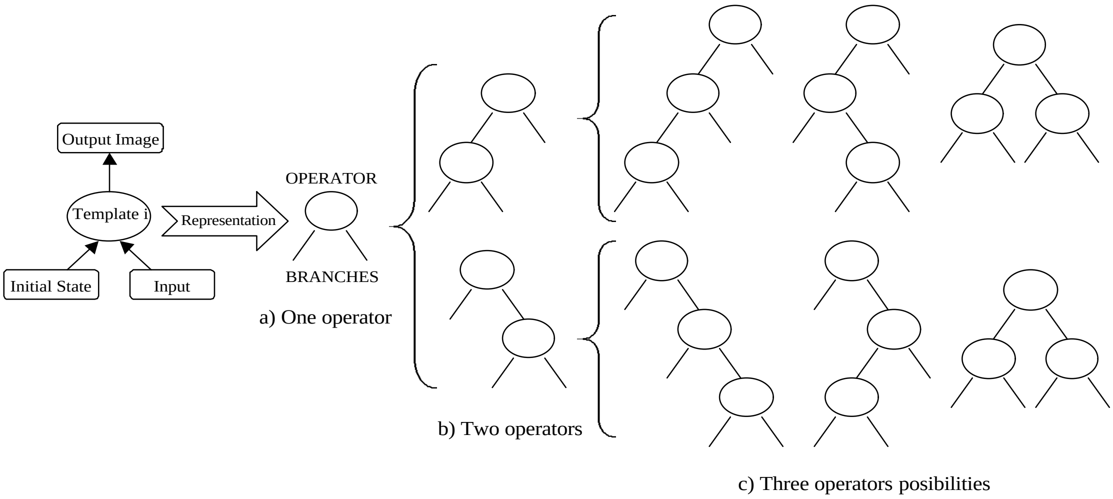

Fig.3 The tree growing process

Assuming that the number of possible templates extracted from a template library, is N then the search space dimension is equal to the number of possible variations of S elements taken N by N. So, for a tree formed by S nodes the search space has $\mathbf{N^S}$ elements. Therefore, the search space size obtained by the set of all possible trees built with S operators, will be $N^S \cdot S!$. Finally, if the set of all possible trees are confined between a lower and an upper number of possible operations, S_L and S_U, then the number of elements in the search space would be:

$$\sum_{S=S_L}^{S_U} SIZE_{N\_templates}^{S\_Nodes} = \sum_{S=S_L}^{S_U} N^S \cdot S!$$

As previously stated, a new tree is created by adding elements in a string which represents the tree in Polish notation, taking into account the Rosenbloom's theorem to validate the tree syntax. According to former premises, a tree formed by S templates is represented by a string that includes S templates and S+1 input images. The probability of a tree of S binary operators is equal to the probability to obtain the before mentionned, that is $Po^S \cdot (1-Po)^{S+1}$.

The influence that Po has in the probability that the tree size would be S, is displayed in Figure 4, being S the number of operations in the tree, and *P(po,S)* the probability of generating a tree of S operations conditioned to a Po value equal to po. The graphic shows that as far as Po grows, the function *P(po,S)* decreases for lower values of S and increases for the higher values of S. This implies higher probability values for big trees as po increases.. This bias becomes critic when po tends to 1, then P(1,S) is equal to one for S=∞ and zero otherwise.

A straightforward heuristic to shrink the space search size deals with the reduction of the number N of elements in the template library, thus the 88 templates have been grouped in 19 sets of elements having similar behavior. The space search size is reduced if we only take into account a resentative from each set. Then an initial search is accomplished in the reduced search space to perform later on a refinement process wherein the quality of each one of the components of the selected sets can be tested to find the most adequate multi-template tree.

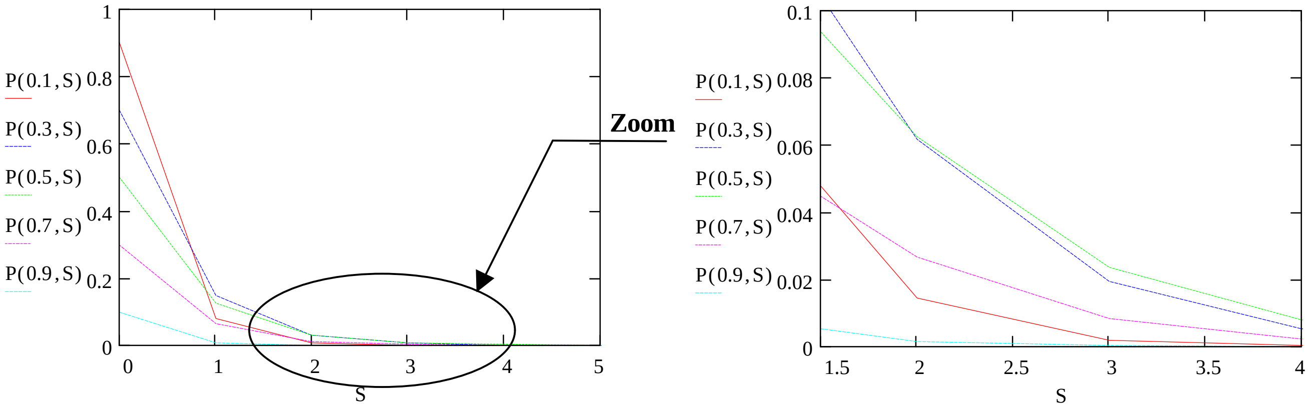

Fig. 4. Influence of Po in the tree size.

A second heuristic, also to reduce the search space size includes an operation hierarchy in the random generation and mutation of the trees. It performs a template classification in three categories: a) Grey to Grey, b) Grey to Binary and c) Binary to Binary. Hence, all trees start from a Grey to Grey root, and when a Grey to Binary or Binary to Binary template appears, the subtree growing from this node is exclusively fashioned by Binary to Binary templates. So, the grey scale operations are isolated from binary ones being the grey to binary templates the interface between them.

5. Conclusions and Further Research

Up to now, the template design has been usually focused as an extrapolation of traditional image processing methods, and only recently complex mathematical techniques as morphology and PDE related methods, have been proposed. But in each case the success of the algorithm strongly depends on the designer expertise.

Thus, the automatic algorithm generation by GAs, here proposed, offers a new methodology to obtain an adequate sequence of operations, independently of the human subjectivy, for the development of analogic algorithm to solve general vision tasks. One of the aims of current work is to introduce this methodology for the design of CNNs algorithms in general industrial applications.

Futher research will be devoted to the automatic generation of analogic programs, taking as a model computer programs with all the diversity and complexity that they convey. In other words, programs usually contain subroutines (also called automatically defined functions, ADFs, or function-defining branches), iterations (automatically defined iterations or ADIs), loops (automatically defined loops or ADLs), recursions (automatically defined recursions or ADRs), and memory of different dimensionality and size (automatically defined stores or ADSs).

6. Acknowledgements

Present work was fully supported by Research Grants. CAM-06G-038-96: “Automation based on an artificial system at the poultry industry”, CICYT 96-1392-C02-01: “Integrated active vision system”, E.U. Thematic Network: “Networked industrial design and control applications using genetic algorithms and evolution strategies: INGENET", BRRT-CT97-5034.

REFERENCES

[1] L. O. Chua, L. Yang. "Cellular Neural Networks: Theory". IEEE Trans. on Circuits and Systems. Vol. 35, No. 10. pp. 1257-1272. October 1988.

[2] S. Espejo, R. Carmona, R. Domínguez-Castro, A. Rodriguez-Vazquez, “A VLSI oriented continuous-time CNN model”, Int. Journ. Of Circuit Theory and Applications, vol. 24, pp. 341-356, 1996.

[3]P.Szolgay, I.Kispal and T.Kozek, “An Experimental System for Optical Detection of Layout Errors of Printed Circuits Boards Using Learned CNN Templates”, Proceedings of the International Workshop on Cellular Neural Networks and their Applications (CNNA-92), pp. 203-209, Munich, 1992.

[4] T.Sziranyi and M.Csapodi, “Texture classification and Segmentation by Cellular Neural Network using Genetic Learning”, Computer Vision and Image Understanding, Vol. 71, No. 3, pp. 255-270, September 1998.

[5]P.L.Venetianer, F.Werblin, T.Roska and L.O.Chua, “Analogic CNN Algorithms for some Image Compression and Restoration Tasks”, IEEE Transactions on Circuits and Systems, Vol. 42, No. 5, 1995.

[6] J. R. Koza: “Detailed Description of Genetic Programming”. Genetic Programming: On the Programming of Computers by Means of Natural Selection, MIT Press, 1992.

[7] R. Caponetto, M. Lavorgna, A. Martinez and L. Occhipinti, “Cellular Neural Network Simulator for Image Processing Applications”, 1998 Fifth IEEE International Workshop on Cellular Neural Network and Their Applications Proceedings, pp.360-365. April 1998.

[8] Martin Hanggi and George S. Mostchytz, “Stochastic and Hybrid Approaches Toward Robust Templates”, 1998 Fifth IEEE International Workshop on Cellular Neural Network and Their Applications Proceedings, pp.366-371. April 1998.

[9] V. Preciado, D. Guinea, J. Vicente, M.C.Garcia-Alegre and A.Ribeiro, ”CNN´s design by GA´s”, European Congress on Computational Methods in Applied Sciences and Engineering Proceedings, ECCOMAS2000, Sept. 2000. Barcelona.