Object Recognition with an Optimized Ventral Stream Model Using Genetic Programming

Eddie Clemente[1,2], Gustavo Olague[1], León Dozal[1], and Martín Mancilla[1]

[1] Proyecto EvoVision,
Departamento de Ciencias de la Computación, División de Física Aplicada,
Centro de Investigación Científica y de Estudios Superiores de Ensenada,
Carretera Ensenada-Tijuana No. 3918, Zona Playitas, Ensenada, 22860, B.C., México
{eclemen,ldozal,olague}@cicese.edu.mx
http://cienciascomp.cicese.mx/evovision/

[2] Tecnológico de Estudios Superiores de Ecatepec. Avenida Tecnológico S/N,
Esq. Av. Carlos Hank González, Valle de Anáhuac, Ecatepec de Morelos

Abstract. Computational neuroscience is a discipline devoted to the study of brain function from an information processing standpoint. The ventral stream, also known as the "what" pathway, is widely accepted as the model for processing the visual information related to object identification. This paper proposes to evolve a mathematical description of the ventral stream where key features are identified in order to simplify the whole information processing. The idea is to create an artificial ventral stream by evolving the structure through an evolutionary computing approach. In previous research, the "what" pathway is described as being composed of two main stages: the interest region detection and feature description. For both stages a set of operations were identified with the aim of simplifying the total computational cost by avoiding a number of costly operations that are normally executed in the template matching and bag of feature approaches. Therefore, instead of applying a set of previously learned patches, product of an off-line training process, the idea is to enforce a functional approach. Experiments were carried out with a standard database and the results show that instead of 1200 operations, the new model needs about 200 operations.

Keywords: Evolutionary Artificial Ventral Stream, Complex Designing System, Heterogeneous and Hierarchical Genetic Programming.

1 Introduction

The human brain is the best example of a purposeful system that transforms numerous complex signals into a set of complex actions. Today, the exact way in which the brain organizes and controls its actions remains a mystery. The endeavour of understanding the inner working of the brain is challenged by several communities grouped into the field of neuroscience that includes the following disciplines: psychology, neurology, psychiatry, cognitive science, cybernetics, computer science, and philosophy, to mention but a few. The advent of

C. Di Chio et al. (Eds.): EvoApplications 2012, LNCS 7248, pp. 315–325, 2012.

computer technology starts a new age in which the brain is modeled as an information processing system giving raise to a field known as computational neuroscience. Although the complexity of the brain is recognized within the domain of evolutionary computation; there is no meaningful work on the development of algorithms modeling the ventral stream and their application to the solution of complex problems [5]. The research described in the present paper aims to develop a new research area in which evolutionary computing is combined with previous proposals from computational neuroscience to assess the validity of algorithmic representations for problem solving. In particular, the goal of the present paper is to illustrate the necessary steps for evolving an artificial ventral stream applied to the problem of object recognition.

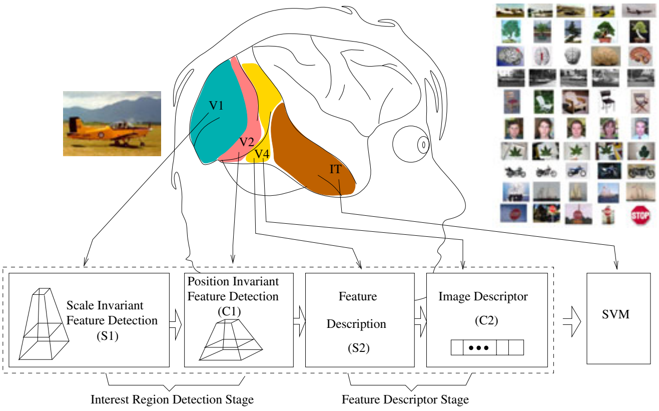
V1
V2
V4
IT
Scale Invariant Feature Detection (S1)
Position Invariant Feature Detection (C1)
Feature Description (S2)
Image Descriptor (C2)
SVM
Interest Region Detection Stage
Feature Descriptor Stage

Fig. 1. Analogy between the ventral stream and the proposed computational model

Figure 1 depicts the classical model of the ventral stream that is known as the standard model for object recognition. In this model, the ventral stream begins with the primary visual cortex, $V1$, goes through visual area, $V2$, then through visual area, $V4$, and finally to the inferior temporal cortex. Therefore, the ventral stream is said to be organized as a hierarchical and functionality specialized processing pathway [6]. The idea exposed in this paper is to evolve an artificial occipitoparietal pathway in such a way of defining an interest region stage; as well as, a feature description stage. The proposed model starts with a color image that is decomposed into a set of alternating "S" and "C" layers, which are named after the discovery of Hubel and Wiesel of the simple and complex cells [7]. This idea was originally implemented by Fukushima in the neocognitron system [4]. This system was further enhanced by other authors including the convolutional networks [8] and the HMAX model by Riesenhuber and Poggio [9]. In all these models the simple layers apply local filters in order to compute higher-order features and the complex layers increase invariance by combining units of the same type.

1.1 Problem Statement

Despite powerful paradigms for object recognition developed in the last decades; it is acknowledged that a solution remains elusive. The problem studied in this paper is the recognition of object categories and this is solved with a biological inspired model that is optimized through an evolutionary algorithm. The goal is to search for the best expressions that are grouped into a hierarchical structure by emulating the functionality of the ventral stream using genetic programming. The major result that is presented in this work is the simplification of the computational process that brings a significant economy in the final algorithm.

The remainder of the paper is organised as follows: section 2 describes the artificial ventral stream divided in two parts known as detection and description; section 3 presents the hierarchical and heterogeneous genetic programming that is used as a way of solving the computational problem; section 4 provides experimental results, and section 5 draws the conclusions and discusses possible future work.

2 An Artificial Ventral Stream

The aim of this section is to describe an artificial ventral stream (AVS) with the goal of solving the problem of object class recognition. In this way, an artificial ventral stream is defined as a computational system, which mimics the functionality of the visual ventral pathway by replicating the main characteristics that are normally found in the natural system. In this sense, previous research developed by computer scientists and neuroscientists such as: [4,1,12,11,10] follow a line of research where the natural and artificial systems are explained through a data-driven scenario. Thus, the idea is to extract a set of features from an image database using a hierarchical representation. The ventral stream is modeled as a process that replicates the functionality of the primary visual cortex $V1$, the over extrastriate visual areas $V2$ and $V4$, and the inferotemporal cortex IT. Thus, the image is transformed into a new representation where: bars, edges, and gratings, are outlined and the whole information is combined into an output vector that represents the original image. This process is characterized by the application of *a priori* information in the form of image patches, which are normally used during the training of the proposed model. In this way, the artificial ventral stream is evaluated by a classification system that is implemented with a support vector machine.

Contrary to previous research, the idea developed in this paper is to propose a function driven scenario based on the genetic programming paradigm. This section proposes a way in which an artificial ventral stream could be evolved in order to emulate key functions that are used to describe the human ventral stream; specifically the standard model. These functions, called Evolutionary Visual Operators (EVO), are optimized in order to render an improved design of the whole visual stream. In particular, the aim is to recognize the building blocks that are used in the solution of a multi-class object recognition problem. Thus, the search functions highlight the set of suitable features that point out the properties of the object such as: color, form, and orientation. The main advantage of the proposed system is reflected on the lower amount of computations that provided a significant economy without sacrificing the overall quality. Next, we describe both detection and description stages following the hierarchical model of the ventral stream.

2.1 Interest Region Detection

The interest region detection stage is depicted in Figure 2. The input color image $I(R, G, B, x, y)$ is decomposed into a pyramid of 12 scales $(I_{\sigma_1}, ..., I_{\sigma_{12}})$, each being smaller than the last one within a factor of $2^{\frac{1}{4}}$; while maintaining its aspect ratio and the information of the corresponding color bands (R, G, B). In this way, the pyramid could be seen as a multidimensional structure that is used to introduce scale invariance information along the multiple bands and at the same time integrating the multiple color bands R, G, B. In this stage, the idea is to apply an EVO to the image pyramid in order to simplify the amount of information. The EVO should be understood as a general concept that is applied to the artificial ventral stream; in such a way, that for each step of the information processing there are specialized programs that fit the problem in an optimal way.

For example, during the stage devoted to the scale invariant feature detection, an interest region operator (IRO) is evolved in order to be adapted to this specific function, see Figure 2. Hence, the IRO can be seen as an specialized operator designed with a GP process that extract special or unique features such as: borders, lines at different orientations, corners; and finally, others that do not need to be human readable. Moreover, one property of genetic programming is the characteristic of being a white box, which is something of great value in the approach that is presented here. In this work, the IRO's domain is defined by the color and orientation at 12 scales $I_{\sigma_1}, ..., I_{\sigma_{12}}$ and its codomain is the

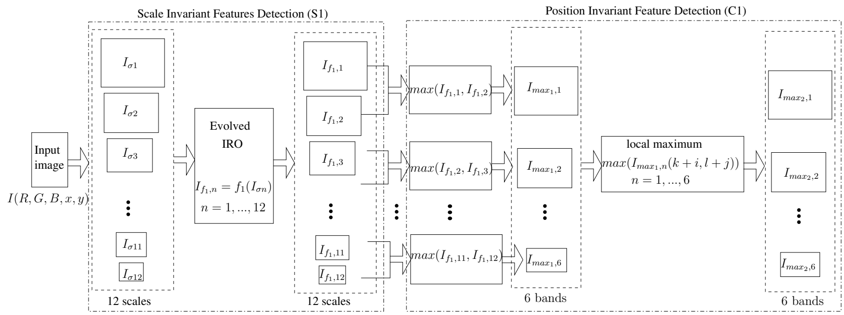

Fig. 2. Flowchart of the interest region detection stage

resulting pyramid of images $I_{f_1,1}, ..., I_{f_1,12}$ that are obtained after applying a suitable IRO. These steps have the functionality of replicating the layer V1 that consists of a set of simple cells. Note, that the structure of the IRO is built from the terminals and functions provided in Table 1. Here, the terminals not only include the RGB color space; but also, the C, M, Y, H, S, I, K that are obtained from the corresponding transformation between color spaces.

Next, in order to enhance the data a *maximum* operation is applied over the local regions, $max(I_{(f_1,2n-1)}, I_{(f_1,2n)})$ with $n = 1, .., 6$, between each pair of consecutive images of the 12 scale pyramid. Then, another maximum filter is applied in order to substitute each sample of the 6 bands $(I_{max_1,1}, ..., I_{max_1,6})$ by the maximum within an interval around a region ϵ of size $i \times j$ around the sample's position (k, l):

$$I_{max_2,n} = \max[I_{max_1,n}(k + i, l + j)] \tag{1}$$

In this particular case, $i = j = 9$ and k and l move at steps of 5 elements for each band. These two operations improve the position and scale invariance within a larger region and it also reduces the information into fewer bands: $(I_{(max_2,1)}, ..., I_{(max_2,6)})$.

This process emulates the first stage of a simple feedforward hierarchical architecture that is composed of a set of simple cells, modeled here with the IRO, and the cortical complex cells which bring some tolerance respect to small changes in position and scale. Therefore, in the proposed model, the layers $S1$ and $C1$ have the purpose of detecting features that are invariant to scale and position. The next section explains how to describe image regions containing the detected features.

Table 1. Set of terminals and functions

Terminals IRO:	$R, G, B, C, M, Y, H, S, I, K, D_x(R), D_x(G), D_x(B), D_x(C), D_x(M), D_x(Y), D_x(H), D_x(S), D_x(I), D_x(K), D_y(R), D_y(G), D_y(B), D_y(C), D_y(M), D_y(Y), D_y(H), D_y(S), D_y(I), D_y(K), D_{xx}(R), D_{xx}(G), D_{xx}(B), D_{xx}(C), D_{xx}(M), D_{xx}(Y), D_{xx}(H), D_{xx}(S), D_{xx}(I), D_{xx}(K), D_{yy}(R), D_{yy}(G), D_{yy}(B), D_{yy}(C), D_{yy}(M), D_{yy}(Y), D_{yy}(H), D_{yy}(S), D_{yy}(I), D_{yy}(K)$
Functions IRO:	$+, -, /, *, \mid - \mid, \mid + \mid, (\cdot)^2, log(\cdot), D_x(\cdot), D_y(\cdot), D_{xx}(\cdot), D_{xy}(\cdot)$ $D_{yy}(\cdot), Gauss_{\sigma=1}(\cdot), Gauss_{\sigma=2}(\cdot), 0.05(\cdot)$
Terminals IDO:	$C1, D_x(C1), D_{xx}(C1), D_y(C1), D_{yy}(C1), D_{xy}(C1)$
Functions IDO:	$+, -, /, *, \mid - \mid, \mid + \mid, \sqrt{\cdot}, (\cdot)^2, log(\cdot), D_x(\cdot), D_y(\cdot), D_{xx}(\cdot)$ $D_{xy}(\cdot), D_{yy}(\cdot), Gauss_{\sigma_1}(\cdot), Gauss_{\sigma_2}(\cdot), 0.05(\cdot)$

2.2 Feature Description

Once that all regions have been highlighted the next step is to describe such important regions. The typical approach is based on template matching between the information obtained in the previous section and a number of prototype patches. The goal is to learn a set of prototypes that are known as a universal dictionary of features and which are used by all object categories. Hopefully, the SVM can recognize the prototypes that correspond to a specific image of a single category. On the other hand, in this paper the functionality of layer $S2$ is evolved in order to enhance the set of prominent features that was highlighted by the interest region detector. It should be noted that each evolved function is a composite function that is capable of substituting several prototype features; thus, reducing significantly the total number of operations needed to define all object features that are used to describe and classify the input images. According to Figure 3, the information provided by $C1$ is feedforward to $k-1$ operators that emulate a set of lower order hypercomplex cells. In other words, for each input image $I_{max_2,n}$ with $n = 1, ..., 6$, a set of functions $f_i(I_{max_2,n})$ with $i = 2, ..., k$, are applied in order to highlight the necessary characteristics that recognize each object class. Note, that each of these functions is an EVO built by the GP from the terminals and functions shown in Table 1, and which performs the patch information descriptive operation (IDO) during this second stage. Hence, this set of functions replaces the universal dictionary proposed by [12,10,11] and it could be said that it corresponds to a function driven approach.

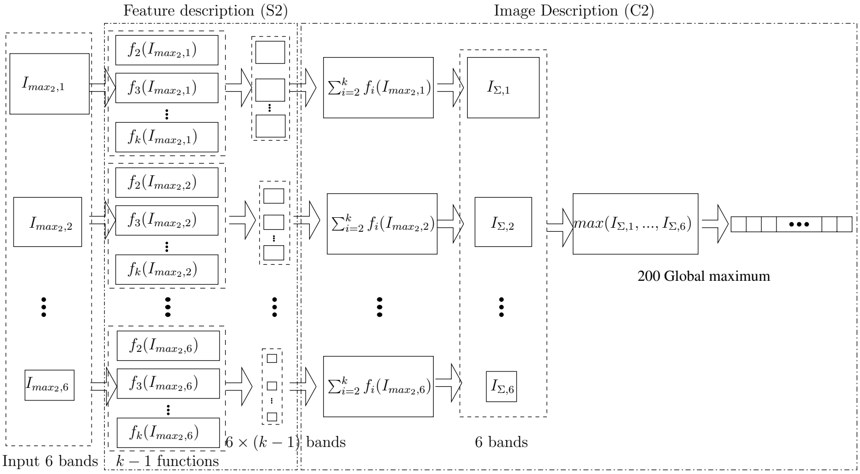

Fig. 3. Flowchart of the feature description stage

Finally, the methodology includes a layer $C2$ for which the outputs of the $k-1$ EVOs are combined and feedforwarded into $I_{\Sigma,n} = \sum_{i=2}^{k} f_i(I_{max_2,n})$ with $n = 1, ..., 6$, resulting into a new pyramid of 6 bands. This step is different to the traditional layer $C2$ where an Euclidean norm is applied to identify the best patches. In this way, the approach proposed here requires only to add the $k-1$ functions' responses. Thus, the image description vector is built by selecting the 200th highest values from the image pyramid that is sort out of the $C2$ layer.

3 Heterogeneous and Hierarchical GP

This section describes the heterogeneous and hierarchical genetic programming (HHGP) that was implemented to optimize the artificial ventral stream. Figure 4 depicts the main steps in the search of an optimal AVS using a mixture of tree-based representations organized similarly to a linear genetic programming structure. The representation that is proposed ensures the development of complex functions, while freely increasing the number of programs. In this way, the structure can growth in number and size of its elements. It is important to remark that each individual should be understood as the whole AVS and it is therefore not only a list of tree-based programs but the whole processing depicted in Figures 2 and 3. Thus, the algorithm executes the following steps. First, it randomly creates an initial population of 30 AVS, where each one is represented as a string of heterogeneous and hierarchical functions called chromosome. In this way, each function corresponds to a gene and is represented as a tree with a maximum depth of 7 levels. Also, each string has a maximum length of 10 genes or functions. Thus, the initial population is initialized with a ramped half and half technique for each gene and the size of the whole chromosome is randomly created. The variation is performed with four different operators that work at the chromosome and gene levels and all operations are selected based on fitness following the scheme proposed by Koza in which the probability of selecting all genetic operations sum to one. Hence, the probability of crossover at chromosome and gene levels is 0.4 respectively and the mutation at chromosome and gene levels is 0.1 for each operation. In this way, the evolutionary loop start the execution of each AVS by computing its fitness using a SVM that is

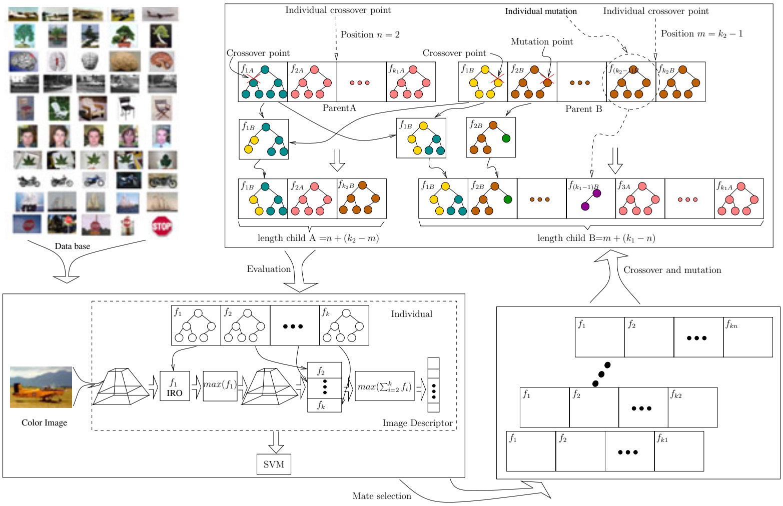

Fig. 4. General flowchart of the methodology to synthesize an artificial ventral stream

used to calculate the accuracy for solving a multiclass recognition problem. The algorithm considers 10 classes and 30 images per class and it uses the Caltech 101 database with the following objects: cars, brains, airplanes, bonsai, chairs, faces, leaves, schooner, motorcycles, and stop signals. Next, an AVS is selected from the population with a probability based on fitness using a roulette-wheel selection to participate in the genetic recombination; while, the best AVS is retained for further processing. In this way, a new population is created from the selected AVS by applying only one genetic operator; for example, the crossover or mutation operation at chromosome or gene levels. As in genetic algorithms our HHGP program executes the crossover between two selected AVS at the chromosome level by applying a "cut and splice" crossover. Each parent string has a separate choice of crossover point; for example, in Figure 4 the position $n = 2$ of Parent A with length k_1 and the position $m = k_2 - 1$ from Parent B with length k_2. Thus, all data beyond the selected crossover point in either AVS string is swapped between both parents A and B. Hence, the resulting child A has a length of $n + (k_2 - m)$ genes that in this case is 3; and the child B has a length of $m + (k_1 - n)$ genes. Moreover, The result of applying a crossover at the gene level is performed by randomly selecting two parent AVS based on fitness in order to execute a subtree crossover between both selected genes. Note, that the IRO can only be selected for a subtree crossover between two parents f_{1A} and f_{1B}. Thus, the f_{1A}'s subtree is replaced with the f_{1B}'s subtree and vice versa to create two new AVS genes. On the other hand, the mutation at the chromosome level leads the selection of a random gene of a given parent to replace such function by a new randomly mutated gene; for example, the position $k_2 - 1$ at parent B; see Figure 4. Moreover, the mutation at the gene level is calculated over an AVS by applying a subtree mutation to a probabilistically selected gene; in other words, a mutation point is chosen at a selected gene and the subtree up to that point is removed and replaced with a new subtree as is illustrated in Figure 4, where the tree f_{2B} is mutated. Finally, the evolutionary loop is terminated until an acceptable classification is reached; i.e., the accuracy is equal to 100% or the total number of generations $N = 50$ is reached.

4 Experimental Results

This section describes the results of evolving the AVS with the multi-class problem in a Dell Precision T7500 Workstation, Intel Xeon 8 Core, NVIDIA Quadro FX 3800 and running Linux OpenSUSE 11.1. The SVM implementation was developed with the libSVM [2]. The best result of the HHGP gives a classification accuracy of 78% in training. In order to compare and validate the performance of the evolved AVS a test against the original HMAX implementation written in Matlab [9]; as well as, the CUDA version of the HMAX model is provided here. The test consists on the comparison of the number of convolutions, speed and performance. Tables 2 and 3 provide the classification results for testing the HMAX and the evolved AVS using 15 images per class. Note, that the HMAX implementations use gray scale images, while the evolved AVS was programmed

Table 2. This table shows the confusion matrix obtained during the testing of the HMAX and its classification accuracy = 71.33% (107/150 images)

	Airplanes	Bonsai	Brains	Cars	Chairs	Faces	Leaves	Motorcycle	Schooner	Stop Signal
Airplanes	11	1	0	1	0	0	0	2	0	0
Bonsai	0	10	3	1	0	1	0	0	0	0
Brains	0	1	10	0	1	0	1	1	0	1
Cars	1	1	0	11	0	1	0	0	1	0
Chairs	0	0	1	1	11	0	0	1	1	0
Faces	0	2	1	0	0	10	1	0	0	1
Leaves	0	0	1	0	0	1	12	0	0	0
Motorcycle	2	0	0	2	0	0	0	11	1	0
Schooner	0	0	1	0	0	2	0	2	10	0
Stop Signal	0	0	1	0	0	2	1	0	0	11

Table 3. This table shows the confusion matrix obtained during the testing of the AVS and its classification accuracy = 80% (120/150 images)

	Airplanes	Bonsai	Brains	Cars	Chairs	Faces	Leaves	Motorcycle	Schooner	Stop Signal
Airplanes	3	1	0	6	0	1	0	3	1	0
Bonsai	0	13	0	0	0	0	0	0	2	0
Brains	1	0	13	0	0	0	0	1	0	0
Cars	0	0	0	14	0	0	0	1	0	0
Chairs	0	2	0	0	10	3	0	0	0	0
Faces	0	0	0	0	0	15	0	0	0	0
Leaves	0	0	0	0	0	0	15	0	0	0
Motorcycle	0	0	0	3	0	0	0	12	0	0
Schooner	0	1	0	0	0	1	0	1	12	0
Stop Signal	0	0	0	0	0	1	0	0	1	13

using the color space. Table 5 shows the number of convolutions per function to illustrate that a significant number of computations was reduced with our proposal, even considering the application of color space. Hence, the AVS applies 216 convolutions while the HMAX model uses 1248 convolutions using a universal dictionary of 200 patches. Therefore, the performance of the AVS process was improved significantly since the total number of convolutions is reflected on a lower computational time, see Table 4.

Table 4. This table shows the total running time

Image size	HMAX MATLAB	HMAX CUDA	Artificial V. S.	
896 × 592	34s	3.5s	2.6s	
601 × 401	24s	2.7s	1.25s	
180 × 113	9s	1s	0.23s	

Table 5. Number of convolutions (NC) for each function of the best individual

$f_1 = 0.05D_x(D_y(I))$	$f_2 = \frac{log(D_{xx}(C1))}{log(D_x(D_x(C1)-C1))}$	$f_3 = (\lvert D_{xx}(C1) - D_x(D_{yy}(C1))\rvert)(\lvert D_y(C1)\rvert)$	$f_4 = log(D_{xx}(C1))$	$f_5 = D_{yy}(C1) + D_x(D_y(C1))$
$NC = 24$	$NC = 24$	$NC = 36$	$NC = 12$	$NC = 24$
$f_6 = D_{yyy}(C1)$	$f_7 = (log(D_x(D_y(C1))))(C1 \cdot Gauss_{\sigma=2}(C1))$	$f_8 = \lvert D_y(C1) - D_y(C1)\rvert$	$f_9 = D_y(D_x(D_y(C1))) - log(D_y(C1))$	$f_{10} = 0.05(D_{yy}(C1) - D_x(D_y(C1)))$
$NC = 18$	$NC = 18$	$NC = 12$	$NC = 24$	$NC = 24$

5 Conclusions

The goal of this paper was to develop an approach based on GP to solve an object recognition problem using as model the ventral stream. The proposal follows a functional approach with several genetic programs being evolved in a hierarchical structure. All programs use different elements within the terminal and function sets according to the particular stage of the artificial ventral stream. The main result is a simplification of the overall structure that provides a lower computational cost. In future work we would like to explore other models.

Acknowledgments. This research was founded by CONACyT, Project 155045 - "Evolución de Cerebros Artificiales en Visión por Computadora".

References

1. Bartlet, W.: SEEMORE: Combining Color, Shape, and Texture Histogramming in a Neurally Inspired Approach to Visual Object Recognition. Neural Computation 9, 777–804 (1997)
2. Chih-Chung, C., Chih-Jen, L.: LIBSVM: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology 2(27), 1–27 (2011) Software available at, `http://www.csie.ntu.edu.tw/~cjlin/libsvm`
3. Fei-Fei, L., Fergus, R., Perona, P.: Learning generative visual models from few training examples: an incremental Bayesian approach tested on 101 object categories. In: IEEE Workshop on Generative-Model Based Vision, CVPR 2004 (2004)
4. Fukushima, K.: Necognitron: A Self-Organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position. Biological Cybernetics 36, 193–202 (1980)
5. Holland, J.H.: Complex Adaptive Systems. A New Era in Computation 121(1), 17–30 (1993)
6. Hubel, D., Wiesel, T.: Receptive Fields of Single Neurones in the Cat Striate Cortex. J. Physiol. 148, 574–591 (1959)
7. Hubel, D.: Exploration of the Primary Visual Cortex. Nature 299, 515–524 (1982)
8. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based Learning applied to Document Recognition. Proceedings of the IEEE (1998)
9. Riesenhuber, M., Poggio, T.: Hierarchical Models of Object Recognition in Cortex. Nature Neuroscience 2(11), 1019–1025 (1999)

10. Mutch, J., Lowe, D.: Object Class Recognition and Localization Using Sparse Features with Limited Receptive Fields. International Journal of Computer Vision, IJCV (2008)
11. Serre, T., Wolf, L., Bilechi, S., Riesenhuber, M., Poggio, T.: Robust Object Recognition with Cortex-Like Mechanisms. IEEE Transactions on Pattern Analysis and Machine Intelligence 29(3), 411–426 (2007)
12. Ullman, S., Vidal-Naquet, M., Sali, E.: Visual features of intermediate complexity and their use in classification. Nature Neurosciencie 5(7), 682–687 (2002)
13. Ungerleider, L., Haxby, J.: "'What' and 'where' in the Human Brain". Current Opinion in Neurobiology 4, 157–165 (1994)