Object Detection in Natural Images Using the Brain Programming Paradigm with a Multi-objective Approach

Eddie Clemente, Gustavo Olague(✉), Daniel Hernández, José L. Briseño, and José Mercado

Proyecto EvoVisión, 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, 22860 Ensenada, BC, Mexico
{eclemen,dahernan,jmercado}@cicese.edu.mx,
{olague,briseno}@cicese.mx
http://cienciascomp.cicese.mx/evovision/

Abstract. In the last few decades the human vision system has been the focus of several researches, using it as a model for solving the object detection problem in digital images. In this work this approach is taken to define the algorithm called Artificial Visual Cortex (AVC) which is inspired in the information flow in the human visual cortex. Additionally, a new methodology for image description is proposed, which allows the detection and description of an object in the scene. Furthermore, this paper describes a new multi-objective learning technique called brain programming. This paradigm is implemented for the training stage of the proposed model in order to classify the *persons* set of the GRAZ-02 image database. The solutions found in this research outperform other techniques in the state-of-the-art.

Keywords: Artificial Visual Cortex · Object recognition · Object detection · Brain programming

1 Introduction

Sight is one of the most important senses for human beings, approximately 70 % of the information received by the brain comes from visual perception; this information helps in the process of making decisions and interacting with the environment. Hence, several scientific communities, such as computer vision, have focus their research in understanding the human vision system in order to emulate it. In this sense, there are several computational models, [1–10], inspired in the hierarchical structure of the human visual system, its neuropsychological theories and neurophysiological characteristics; some examples are: the feature integration theory [11], biased competition theory [12], Recognition-By-Components [13], simple and complex cells model [14], the two path cortical

© Springer International Publishing Switzerland 2015
A.M. Mora and G. Squillero (Eds.): EvoApplications 2015, LNCS 9028, pp. 201–213, 2015.
DOI: 10.1007/978-3-319-16549-3_17

model [15], to mention but a few. There is a model of particular interest for this paper, because it is inspired in the human visual cortex and it is implemented for object recognition; this model, proposed by Olague *et al.*, it is called Artificial Visual Cortex (AVC) [16], it shows great performance in establishing the absence or presence of an object in an image. In this way, we enhanced this model for the object detection task, in order to recognize and locate an object within a digital image.

The AVC model is based mainly on two models: a psychological model called feature integration theory and a neurophysiological model called the two pathway cortical model. The feature integration theory explains that visual attention in human beings is performed in two stages. The first one is called the pre-attentive stage, where the visual information is processed in parallel over the feature dimensions that compose a scene, which are: shape, color, orientation, spacial frequency, brightness and motion direction. The second stage is called focal attention, it integrates the features that were process independently on the previous stage, and focuses the attention on a region of the scene. Hence, visual attention is the capability of a creature, living or artificial, of focusing an object of interest on a visual environment [17]. Visual attention can be formally defined as “the process that establishes a relationship between the different properties in the scene, perceived through the visual system, and the objective of finding the best aspect for solving the task at hand” [18]. The second theory used for this work is the two pathway cortical model; this neurophysiological model states that the are two information pathways within the visual cortex, the dorsal and ventral streams; both subsystems receive the same visual information as input, but the difference between them is related to the information transformations performed in each of them [15]. The dorsal stream is mainly related to the spacial detection of objects and visual attention [19]. Thus, it is also known as the “where?” or “how?” pathway; the regions of the brain related to this task are: V1, V2, V3, V5, MST and PP, each region has a specific functionality and they are hierarchically organized. On the other hand, the ventral stream is linked to object recognition and shape representation; hence, it is also called the “what?” stream. The brain areas involved in this functionality are: V1, V2, V4, TEO and TE. Nevertheless, both information streams are interrelated in order to achieve their tasks [12,19,20].

The integration of these theories within a computational model as the AVC is based on the idea of defining an image as the graph of a function, which is then transformed by a series of operators within a hierarchical process; where each computational stage emulates the transformations that the visual information undergoes in the brain. In this way, the AVC model was designed for categorizing images regardless of the color, orientation, illumination conditions, scale or position of the object in the image, and one of its innovations is the way it selects prominent image features in order to build an abstract representation of the object. Hence, in images where the object of interest occupies a big portion of the image the classification rate achieved by the AVC model is 98 % or higher. Nevertheless, when the system is applied on natural images, like those present in the database proposed in [21], where the objects are smaller and immersed in a high content environment, the performances of the AVC is lower. This might occur because most of the features selected to build the image descriptor are selected from the environment, instead of the object of interest; on the other hand, it might occur due to the fact that the descriptor is built using scattered points from the last feature map, called mental map (MM), where only the most prominent region is selected for the description of the image, but this might not correspond to the object of interest. Therefore, in this work we propose a new methodology for building the image descriptor, where the description is performed using an image region instead of sparse points; rendering the AVC model capable of selecting a region of the object of interest using class specific object attributes. Then, implicitly finding the object's location. Also, we propose a feedback operation using the first stages of the model for building the descriptor, since the first maps contain more information about the object. This new paradigm is called AVCMO due to the multi-objective approach taken for the training stage of the model in order to detect and describe the object of interest.

The remainder of this paper is organized as follows, Sect. 2 details the different stages of our approach, as well as the implementation of a multi-objective evolutionary system as the training paradigm for the AVCMO model. After, Sect. 3 describes the performance achieved by the AVCMO model for classifying the *persons* class of the GRAZ-02 database, and finally, the conclusion from this work are explained in Sect. 4.

2 Methodology

This section describes the AVCMO model, focusing on the new methodoly for building the image descriptor; also, the brain programming algorithm with a multi-objective approach is detailed here.

2.1 Description of the AVCMO

The AVCMO is divided in two main stages. In the first stage the system acquires and transforms the attributes that characterize the object; and in the second stage, the system locates the most prominent image region and extracts a description vector, which is later applied for classification purposes. These two stages are detailed next.

Feature Acquisition and Transformation. The input for the system is a digital color image in the RGB color model (red, green, blue); which is then transformed to the CMYK (cyan, magenta, yellow, black) and HSV (hue, saturation, value) color models, in order to build the set $I_{color} = \{I_r, I_g, I_b, I_c, I_m, I_y, I_k, I_h, I_s, I_v\}$, where each element corresponds to a component of the color models. The color bands in I_{color} are then transformed by four evolved visual operators (EVO) defined as $EVO_d : I_{color} \rightarrow VM_d$; where each operator is applied

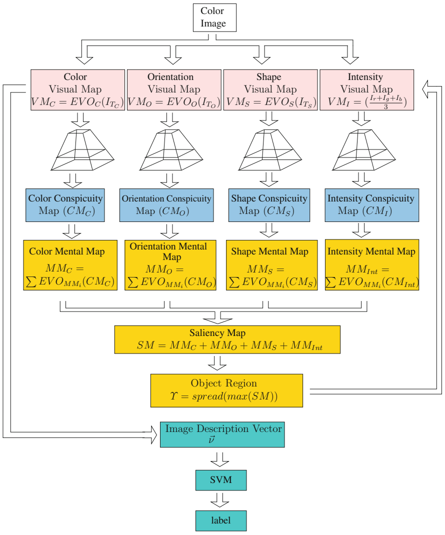

Color Image
Color Visual Map $VM_C = EVO_C(I_{T_C})$
Orientation Visual Map $VM_O = EVO_O(I_{T_O})$
Shape Visual Map $VM_S = EVO_S(I_{T_S})$
Intensity Visual Map $VM_I = (\frac{I_r+I_g+I_b}{3})$
Color Conspicuity Map (CM_C)
Orientation Conspicuity Map (CM_O)
Shape Conspicuity Map (CM_S)
Intensity Conspicuity Map (CM_I)
Color Mental Map $MM_C = \sum EVO_{MM_i}(CM_C)$
Orientation Mental Map $MM_O = \sum EVO_{MM_i}(CM_O)$
Shape Mental Map $MM_S = \sum EVO_{MM_i}(CM_S)$
Intensity Mental Map $MM_{Int} = \sum EVO_{MM_i}(CM_{Int})$
Saliency Map $SM = MM_C + MM_O + MM_S + MM_{Int}$
Object Region $\Upsilon = spread(max(SM))$
Image Description Vector $\vec{v}$
SVM
label

Fig. 1. Visual information flow.

independently aiming to highlight specific features of the object of interest, such as color (C), orientation (O), shape (S) and intensity (Int); these features are called dimension (d) and follow an independent information flow, see Fig. 1. In this manner, d is an element in the dimension set $d \in \{C, S, O, Int\}$; hence, each visual map (VM_d) highlights promientent information from the object in the different features.

Then, once the visual maps are calculated they go through a center-surround process, this process is based on the functionality of the ganglion cells, where the activation of the cells corresponds to the difference between the stimulus on the central receptive field and the border one. From a computational point of view, the objective of this process is to generate a conspicuous map (CM) per dimension, in accordance with the model in [3]. This subroutine is defined by two steps; first, starting from the VM_d an eight level Gaussian Pyramid is created $P_d^\sigma = \{P_d^{\sigma=0}, P_d^{\sigma=1}, P_d^{\sigma=2}, ..., P_d^{\sigma=8}\}$, where σ denotes the Gaussian blurring at each level and its size. The second step of this process uses this pyramid as input in order to generate six new maps as follows:

$$Q_d^j = P_d^{\sigma=\lfloor\frac{j+9}{2}\rfloor+1} - P_d^{\sigma=\lfloor\frac{j+2}{2}\rfloor+1},$$

where $j = 1, 2, ..., 6$. Since the levels in P_d^σ have different size all the levels are scaled to the smaller size for calculating the differences. Then, each of these six maps is normalized and integrated through a summation operation, the resulting map is normalized and scaled to the size of the input VM_d; hence, this resulting map defines the CM_d.

Object Detection and Description. After building the conspicous maps, the next stage of the AVCMO aims to establish the image region with the most prominent information about the object of interest and create a description vector from it. This stage is analogous to the functionality of the V4 brain area, as well as the inferior temporal cortex (IT), since these two areas are related to the object classification task. Computationally speaking, in this stage of the process a set of visual operators are applied in order to create a mental map (MM) per dimension, see Fig. 1. In this way, a set of operator EVO_{MM} is applied over each CM_d seeking the most prominent features per dimension, this is: $MM_d = \sum_{i=1}^{k}(EVO_{MM_i}(CM_d))$, where d is the feature dimension and k representes the cardinality of the set EVO_{MM}. Thus, the sumatory integrates the output of all the operators in EVO_{MM}, creating the MM_d.

Once the mental maps are created, they are normalized between 0 and 1 using a lineal interpolation, see Eq. 1; then, they are integrated into a single saliency map SM, as shown in Eq. 2.

$$MM_d = \frac{MM_d - min(MM_d)}{max(MM_d) - min(MM_d)} \ . \tag{1}$$

$$SM = MM_C + MM_O + MM_F + MM_{Int} \ . \tag{2}$$

When the SM is obtained, the coordinates of the highest value in the map are stored in a location vector $\boldsymbol{p}$. Then, a propagation operation is performed around this position, this requires a process of n iterations, where we seek to add to the locations vector $\boldsymbol{p}(i)$ the position of the highest value located in the neighborhood around the points stored in $\boldsymbol{p}$. In this way, the n elements of $\boldsymbol{p}(i)$ define a region Υ on the saliency map, which establishes the area where the object is located. Even though Υ defines the object location, the values used to describe the object will be extracted from previous stages of the process. This is, the region Υ will be projected over the visual maps with the aim of obtaining the best features in each dimensions; then, the pixels with the highest values within each region in the VM_d are selected. Again, a propagation operation is performed in order to obtain the m highest values in each visual map. Finally, the m values from each dimension are concatenated creating a description vector $\boldsymbol{\nu}$ of size n, which is then input into a support vector machine (SVM) for classification purposes. The construction of the description vector is detailed in the Algorithm 1 and it is depicted in Fig. 1.

2.2 Multi-objective Brain Programming

In the brain programming paradigm each solution has the same hierarchical structure defined by the AVCMO and what differentiates the solutions are the set of operators within them. This idea comes from analogy to the natural system, where evolution could modify the functionality of each brain area without altering the order in which they work. Brain programming follows the evolution cycle of genetic programming, but it proposes a new heterogeneous multi-tree representation for the individuals, as well as new crossover and mutation operator for this new representation.

```
Input: $SM, VM_d$ /*Saliency map and the array of visual maps */
Output: $\boldsymbol{p}, \boldsymbol{\nu}$ /*Region coordinates for each visual map and the description vector */
$\boldsymbol{p}[1] \leftarrow$ coordMaxVal($SM$)
$SM[\boldsymbol{p}[1].x, \boldsymbol{p}[1].y] \leftarrow 0$
for $i \leftarrow 2$ to $n$ increase 1 do
    $\boldsymbol{p}[i] \leftarrow$ coordMaxValNeighbor($SM, \boldsymbol{p}$)
    $SM[\boldsymbol{p}[i].x, \boldsymbol{p}[i].y] \leftarrow 0$
end
foreach $VM_d$ do
    $\boldsymbol{p}_d[1] \leftarrow$ coordMaxValReg($VM_d, \boldsymbol{p}$)
    $v_d[1] \leftarrow$ getVal($VM_d, \boldsymbol{p}_d[1]$)
    $VM_d[\boldsymbol{p}_d[i].x, \boldsymbol{p}_d[i].y] \leftarrow 0$
    for $i \leftarrow 2$ to $n/4$ increase 1 do
        $\boldsymbol{p}_d[i] \leftarrow$ coordMaxValNeighbor($VM_d, \boldsymbol{p}_d$)
        $\boldsymbol{\nu}_d[i] \leftarrow$ getVal($VM_d, \boldsymbol{p}_d[i]$)
        $VM_d[\boldsymbol{p}_d[i].x, \boldsymbol{p}_d[i].y] \leftarrow 0$
        $\boldsymbol{\nu} \leftarrow$ concat($\boldsymbol{\nu}, \boldsymbol{\nu}_d$)
    end
end
return $\boldsymbol{p}, \boldsymbol{\nu}$
```

Algorithm 1. DESCRIPTOR

Genotype. One important aspect of the *EVO*s is their independence, this facilitates their computational representation as an array. In this way, we can consider the next analogy with the biological system. The array of *EVO*s is similar to a chromosome, and each operator can be considered as a gene, where each function or terminal used to build the *EVO* as analogous to the nucleotides that form a gene. This means that the representation or genotype has three levels, the first one considers the whole chromosome as a unit, the second level are the genes and the third level are the functions that define the operators, see Fig. 2. Thus, the phenotype, defined as the physical manifestation of the genes, is the result of applying the within the structure of the Artificial Visual Cortex.

Genetic Operators. There are two types of crossover operations, one for chromosome level and the other for gene level operations, these are detailed next:

- *Chromosome level crossover:* the objective of this operator is the genetic combination and information exchange between chromosomes, this process is performed by exchanging array segments that constitute each of the parents. The method used is known as cut and splice. First, a crossover point is randomly selected on from the parent with the shortest string, then the same point is selected for the other parent; after, the new individual, offspring 1, is generated by joining the left size of the string from parent 1 and the right size from parent 2. In a similar way, the offspring 2 is built by using the left side of the string in parent 2 and the right size in parent 1. This process is depicted in Fig. 2a.
- *Gene level crossover:* this operator focuses on the operators that compose the gene. A crossover point is selected for each three using the smallest one. Then, parent 1 is selected to create offspring 1, where the sub-threes below the cross point are replaces by the sub-threes from below the cross point. Similarly, offspring 2 is created by taking parent 2 and replacing the sub-three from

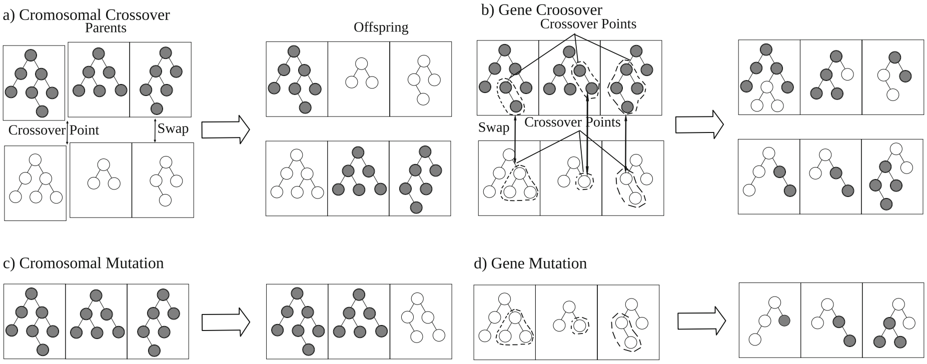


Fig. 2. The genetic operators are perform at two levels; Figure (a) shows the crossover operation at chromosome level, and (b) at gene level, while (c) and (d) depict the mutation operation at chromosome and gene level respectively.

parent 1. In this way, two new individuals are created. This process is shown in Fig. 2b.

Mutation Operators. Once the new individuals are created, they might be modified by one or two kinds of mutation operators: chromosome level mutation and gene level mutation. These operators work as follows:

– *Chromosome level mutation:* it consists on exchanging each of the operators that constitute the chromosome with a randomly generated operator, completely changing the genotype. This procedure can be seen in Fig. 2c.
– *Gene level mutation:* for each syntactic three a random mutation point is selected, then the sub-three below the mutation point is replaced by a new random sub-three. This kind of mutation only changes a portion of the each operator. This mutation operator is depicted in Fig. 2d.

Functions and Terminals. In the proposed model each EVO is independent and is built using its own set of functions and terminals, see Table 1. Hence, we specially selected functions for each dimension, aiming to find the best features to characterize the object. Therefore, for the operator EVO_O we use Gaussian smoothing filters with $\sigma = 1$ and $\sigma = 2$, as well as first and second order derivatives on the x and y directions. Meanwhile, for the color dimension, we selected functions like: color opponencies ($Op_{r-g}(I)$, $Op_{b-y}(I)$), complement function ($(A)^c$); for building the EVO_C operator. In a similar way, aiming to find prominent shape features we propose to implement mathematical morphology functions such as: dilation ($A \oplus SE_x$), erosion ($A \ominus SE_x$), opening ($A \circledcirc SE_s$), closing ($A \odot SE_s$), as well as other operations that result from combining these four; this set of functions is applied to construct the EVO_S operator. In the case of the terminals, these are defined by the I_{color} set, as well as the output from some functions applied over elements of the same set.

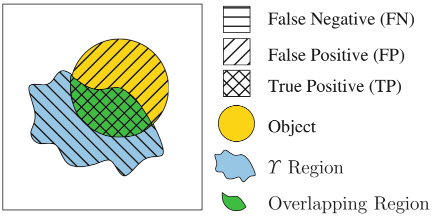
Table 1. Functions and terminals for the *EVO*.

Functions for EVO_O	Terminals for EVO_O
$A+B$, $A-B$, $A \times B$, A/B, $\|A\|$, $\|A+B\|$, $\|A-B\|$, $log(A)$, $(A)^2$, $\sqrt{A}$, $k+A$, $k-A$, $k \times A$, A/k, $round(A)$, $\lfloor A \rfloor$, $\lceil A \rceil$, $inf(A,B)$, $sup(A,B)$, $G_{\sigma=1}(A)$, $G_{\sigma=2}(A)$, $D_x(A)$, $D_y(A)$, $thr(A)$	I_r, I_g, I_b, I_c, I_m, I_y, I_k, I_h, I_s, I_v, $D_x(I_x)$, $D_{xx}(I_x)$, $D_y(I_x)$, $D_{yy}(I_x)$, $D_{xy}(I_x)$
Functions for EVO_C	Terminals for EVO_C
$A+B$, $A-B$, $A \times B$, A/B, $log(A)$, $exp(A)$, $(A)^2$, $\sqrt{A}$, $(A)^c$, $thr(A)$	I_r, I_g, I_b, I_c, I_m, I_y, I_k, I_h, I_s, I_v, $Op_{r-g}(I)$,$Op_{b-y}(I)$
Functions for EVO_S	Terminals for EVO_F
$A+B$, $A-B$, $A \times B$, A/B, $k+A$, $k-A$, $k \times A$, A/k, $round(A)$, $\lfloor A \rfloor$, $\lceil A \rceil$, $A \oplus SE_d$, $A \oplus SE_s$, $A \oplus SE_{dm}$, $A \ominus SE_d$, $A \ominus SE_s$, $A \ominus SE_{dm}$, $Sk(A)$, $Perim(A)$, $A \circledast SE_d$, $A \circledast SE_s$, $A \circledast SE_{dm}$, $T_{hat}(A)$, $B_{hat}(A)$, $A \circledcirc SE_s$, $A \odot SE_s$, $thr(A)$	I_r, I_g, I_b, I_c, I_m, I_y, I_k, I_h, I_s, I_v
Functions for EVO_{MM}	Terminals for EVO_{MM}
$A+B$, $A-B$, $A \times B$, A/B, $\|A+B\|$, $\|A-B\|$, $log(A)$, $(A)^2$, $\sqrt{A}$, $G_{\sigma=1}(A)$, $G_{\sigma=2}(A)$, $D_x(A)$, $D_y(A)$	MC_d, $D_x(MC_d)$, $D_{xx}(MC_d)$, $D_y(MC_d)$, $D_{yy}(MC_d)$, $D_{xy}(MC_d)$

Fitness Function. The fitness function measures the performances of the solutions, which is related to the task at hand. In this case, we focus on the classification and localization of an object in an image. Thus, based on the characteristics of the model we propose two functions, one for measuring the classification performance and one for determining the quality of the solutions for localizing the object in the image. The first objective is the called Equal Error Rate (EER). This metric defines the probability of an algorithm for deciding if two instances correspond to the same class [22]. The EER is defined as the value that satisfies $fpr = fnr$; where fnr is the false negative rate and fpr is the false positive rate, fulfilling the following restriction: $fnr = 1 - tpr$; where, tpr is the true positive rate. From a ROC (Receiver Operating Characteristic) curve, the EER can be calculated by extending a line from (0,1) to (1,0), the point where this line crosses the curve corresponds to the EER. In this way the first objective is defined as follows:

$$Objective_1 = EER. \tag{3}$$

The second objective is based on calculating the correspondance between a groundtruth of the object location in the image and the region Υ selected as the posible position. In this case, we use the F-measure defined by: $F_\alpha(\rho, \vartheta) = \dfrac{(1+\alpha) \cdot (\rho \cdot \vartheta)}{(\alpha \cdot \rho) + \vartheta}$, where α controls the balance between precision ρ and recall ϑ, with $0 \leq \alpha \leq \infty$. If $\alpha < 1$ then ρ is greater than ϑ; on the contrary, if $\alpha > 1$ then ϑ is greater. Finally, when $\alpha = 1$, we say that the precision and coverage



$\rho = \frac{TP}{TP+FP}$, $\vartheta = \frac{TP}{TP+FN}$

Fig. 3. Correspondence between the attended image region and the object region occupied by the object of interest applied for evaluating the precision ρ and recall ϑ values.

are balanced. In this work we consider that $\alpha = 1$. The true positive elements correspond to the pixels that belong to the region Υ and the object region, the false positive are the pixels in Υ that are not part of the object of interest, while the false negative are the points in the object that are not included in the region Υ, as seen in Fig. 3. In this way, after processing n images that contain the object, the fitness function is defined by the average of the F-measure over the ω images, this is:

$$Objective_2 = \frac{1}{\omega}\sum_{i=1}^{\omega}\left(\frac{2\cdot(\rho\cdot\vartheta)}{\rho+\vartheta}\right). \tag{4}$$

3 Experiments and Results

In this work, we approach the classification problem from a presence/absence perspective. We established a protocol composed of three steps; the first two define the training stage of the model, while the third one corresponds to the testing phase. Therefore we need three sets of images for the experiments, one per step. This protocol is described next:

1. *Training:* this step starts by evaluating each solution with an image set called training; one image descriptor is created per image. Then, these descriptors are used to generate a SVM model which labels of each descriptor linking the image to a class. In order to avoid over training we perform the second step.
2. *Validation:* in this step, we evaluate all the solutions using another image set, called validation; then, the descriptors found for this set are classified using the SVM model created in step 1. The classification results from this step are used as a fitness function and we continue the evolutionary process.
3. *Test:* once the brain programming optimization is finished, we take the solutions from the last generation along their corresponding SVM model in order to evaluate their classification performance on another image set, called test; this evaluation provides the performances of the solutions in order to compare them with other methods.

3.1 Image Data Base

The image data based used for this work is called GRAZ-02, it was proposed in [21]. This data base was constructed by using similar environments for all the classes. GRAZ-02 is composed of three classes, where 311 images belong to the persons class, 365 to the bikes class, 420 to the cars class and 380 images conform the background set. This last set does not contain persons, bikes or cars. For this work only the persons class was selected, using the same number of training and testing images as the experiments presented in [21]. Then, 150 images were selected for the training set, 75 images for validation and 75 images for the testing set. One of the advantages of this data base is that it provides segmented images for each of the classes, which was taken as reference for the F-measure evaluation. Figure 4 shows some image samples from the persons class in the GRAZ-02 data base.

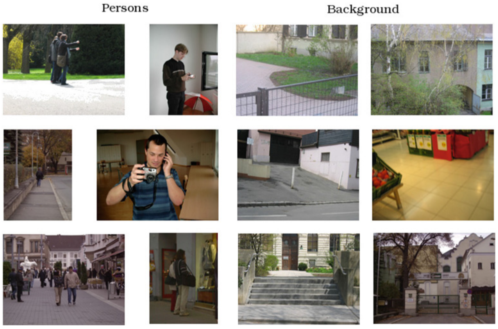
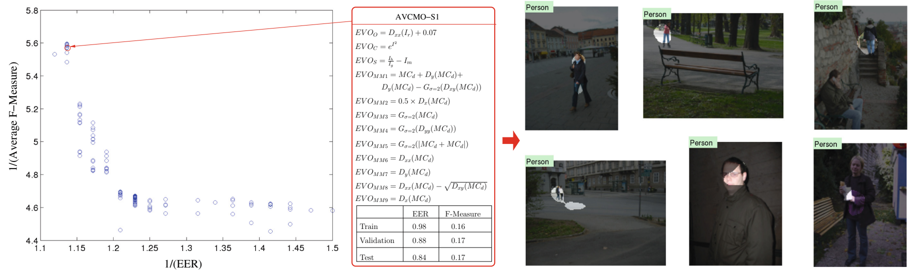


Fig. 4. Sample images from the *persons* class.

3.2 Comparison with Other Methods

The evolutionary parameters for the experiments are: 30 generations, 400 individuals per generation, the initialization of the syntactic trees as done using the *half-and-half* method, using 9 levels as maximum tree depth, with a maximum of 15 genes per chromosome. These parameters were proposed after a tuning procedure. For the parent selection process we used the SPEA2 algorithm [23].

Figure 5 presents the solutions after evaluating the validation set. Note that the graphs show the inverse of the EER versus the average of the F-measure, since the optimization process is performed as a minimization task. One of the issues of an multi-objective approach is finding the best solution, in this case, it was selected according to its performance in classifying the test image set, in order to compare our solution with other methodologies. Nevertheless, due the multiobjective approach there are some solutions that achieve better results in locating the person within the image. Thus, the selected solution $AVCMO-S1$ is detailed in Fig. 5. The $AVCMO-S1$ model outperforms the methods in the



AVCMO–S1

$EVO_O = D_{xx}(I_r) + 0.07$

$EVO_C = e^{I^2}$

$EVO_S = \frac{I_h}{I_g} - I_m$

$EVO_{MM1} = MC_d + D_y(MC_d) + D_y(MC_d) - G_{\sigma=2}(D_{xy}(MC_d))$

$EVO_{MM2} = 0.5 \times D_x(MC_d)$

$EVO_{MM3} = G_{\sigma=2}(MC_d)$

$EVO_{MM4} = G_{\sigma=2}(D_{yy}(MC_d))$

$EVO_{MM5} = G_{\sigma=2}(|MC_d + MC_d|)$

$EVO_{MM6} = D_{xx}(MC_d)$

$EVO_{MM7} = D_y(MC_d)$

$EVO_{MM8} = D_{xx}(MC_d) - \sqrt{D_{xy}(MC_d)}$

$EVO_{MM9} = D_x(MC_d)$

	EER	F-Measure
Train	0.98	0.16
Validation	0.88	0.17
Test	0.84	0.17

Fig. 5. The chart on the left side of the Figure shows the 400 solutions for the validation set. Note that after 30 generations there are some identical solutions due to the diversity loss, also there are several solutions with the same performances score. The $AVCMO-S1$ was selected since it achieves the best score in the testing process. Some test images where the solution is applied can be seen on the right side of the Figure.

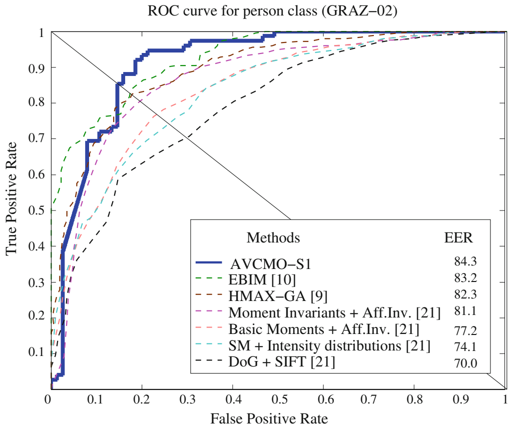


Fig. 6. Performance comparison of classification with other methods.

state-of-the-art, see Fig. 6, and it is also capable of finding the object location in the image; some examples of this process are shown in Fig. 5.

4 Conclusions and Future Work

This work shows a new methodology for building the description vector in the AVC model. This new strategy seeks to create the descriptor using the information of the image region where the object is located, implicitly finding its location. The training process for this model was performed using the evolutionary technique called brain programming, implemented from a multi-objective perspective. This new approach was applied for classifying the persons class of the GRAZ-02 data base, achieving better results than those in the state-of-the-art. Some future work would be to extend the experimentation to other classes in GRAZ-02 and GRAZ-01.

References

1. Fukushima, K.: Neural network model for selective attention in visual pattern recognition and associative recall. Appl. Opt. **26**(23), 4985–4992 (1987)
2. Olshausen, B., Anderson, C., Van Essen, D.: A neurobiological model of visual attention and invariant pattern recognition based on dynamic routing of information. J. Neurosci. **13**(11), 4700–4719 (1993)
3. Walther, D., Itti, L., Riesenhuber, M., Poggio, T.A., Koch, C.: Attentional selection for object recognition - a gentle way. In: Bülthoff, H.H., Lee, S.-W., Poggio, T.A., Wallraven, C. (eds.) BMCV 2002. LNCS, vol. 2525, pp. 472–479. Springer, Heidelberg (2002)
4. Itti, L., Koch, C., Niebur, E.: A model of saliency-based visual attention for rapid scene analysis. IEEE Trans. Pattern Anal. Mach. Intell. **20**(11), 1254–1259 (1998)
5. Riesenhuber, M., Poggio, T.: Hierarchical models of object recognition in cortex. Nat. Neurosci. **11**(2), 1019–1025 (1999)
6. Serre, T., Kouh, C., Cadieu, M., Knoblich, G., Kreiman, U., Poggio, T.: Theory of object recognition: computations and circuits in the feedforward path of the ventral stream in primate visual cortex. Technical report. MIT-CSAIL (2005)
7. Mutch, J., Lowe, D.: Object class recognition and localization using sparse features with limited receptive fields. Int. J. Comput. Vis. **80**(1), 45–57 (2008)
8. Wersing, H., Körner, E.: Learning optimized features for hierarchical models of invariant object recognition. Neural Comput. **15**(7), 1559–1588 (2003)
9. Ghodrati, M., Khaligh-Razavi, S., Ebrahimpour, R., Rajaei, K., Pooyan, M.: How can selection of biologically inspired features improve the performance of a robust object recognition model? Plos One **7**(2), 1–15 (2012)
10. Huang, Y., Huang, K., Tao, D., Tan, T., Li, X.: Enhanced biologically inspired model for object recognition. IEEE Trans. Syst. Man Cybern. B Cybern. **41**(6), 1668–1680 (2011)
11. Treisman, A.M., Gelade, G.: A feature-integration theory of attention. Cogn. psychol. **12**(1), 97–136 (1980)
12. Desimone, R., Duncan, J.: Neural mechanisms of selective visual attention. Ann. Rev. **18**, 193–222 (1995)
13. Biederman, I.: Recognition-by-components a theory of human image understanding. Psychol. Rev. **94**(2), 115–147 (1987)
14. Hubel, D.: Exploration of the primary visual cortex, 1955–78. Nature **299**(5883), 515–524 (1982)
15. Mishkin, M., Ungerleider, L., Macko, K.: Object vision and spatial vision: two cortical pathways. TINS **6**, 414–417 (1983)
16. Olague, G., Clemente, E., Dozal, L., Hernández, D.E.: Evolving an artificial visual cortex for object recognition with brain programming. In: Schuetze, O., Coello Coello, C.A., Tantar, A.-A., Tantar, E., Bouvry, P., Del Moral, P., Legrand, P. (eds.) EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation III, pp. 97–115. Springer, Heidelberg (2014)
17. Koch, C., Ullman, S.: Shifts in selective visual attention: towards the underlying neural circuitry. Hum. Neurobiol. **4**, 219–227 (1985)

18. Olague, G., Dozal, L., Clemente, E., Ocampo, A.: Optimizing an artificial dorsal stream on purpose for visual attention. In: Schuetze, O., Coello Coello, C.A., Tantar, A.-A., Tantar, E., Bouvry, P., Del Moral, P., Legrand, P. (eds.) EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation III, pp. 141–166. Springer, Heidelberg (2014)
19. Milner, A., Goodale, M. (eds.): The Visual Brain in Action, 2nd edn. Oxford University Press, Oxford (2006)
20. Farivar, R.: Dorsal-ventral integration in object recognition. Brain Res. Rev. **61**(2), 144–153 (2009)
21. Opelt, A., Pinz, A., Fussenegger, M., Auer, P.: Generic object recognition with boosting. IEEE Trans. Pattern Anal. Mach. Intell. **28**(3), 416–431 (2006)
22. Reid, P.: Biometrics for Network Security. Prentice Hall Professional, Upper Saddle River (2004)
23. Bleuler, S., Laumanns, M., Thiele, L., Zitzler, E.: PISA – a platform and programming language independent interface for search algorithms. In: Fonseca, C.M., Fleming, P.J., Zitzler, E., Deb, K., Thiele, L. (eds.) EMO 2003. LNCS, vol. 2632, pp. 494–508. Springer, Heidelberg (2003)