

# Brain Programming for the Evolution of an Artificial Dorsal Stream

León Dozal · Gustavo Olague · Eddie Clemente · Daniel E. Hernández

Received: 2 April 2013 / Accepted: 14 February 2014 / Published online: 8 March 2014  
© Springer Science+Business Media New York 2014

**Abstract** This work describes the application of a new strategy called brain programming for automating the design of visual attention (VA) models. Nowadays, a term known as cognitive vision coined within the computer vision and cognitive research communities has been introduced to delimitate the kind of computer vision systems that are robust, resilient and adaptable to the task at hand through the incorporation of cognitive abilities. In particular, visual attention is considered as a critical factor whose main goal is to establish a relationship between the different properties or features of the scene with the aim of selecting the most suitable aspects for the task at hand. This paper follows a main trend in cognitive computation where the visual pathway is modeled through a succession of levels or layers. Here, the VA task is defined with the idea that several areas of the brain are in charge of its functionality in a hierarchical way. To achieve such functionality, we propose that an artificial process, mimicking the natural counterpart, would be charged of looking for a set of complex operations using an optimization/search process. The idea is to include such

operations within a VA model that will be evolved according to a specific task. The aim of the whole process is to provide with the best solutions among the space of possible visual attention programs (VAPs) for a given problem. In this way, the article presents a methodology for automating the design of VAPs. Therefore, the final design can be seen as a cognitive vision system that is engaged in a purposive goal-directed behavior. The results obtained on a well-known testbed confirm that the proposal is able to automatically design VAPs that outperform previous man-made systems developed by VA experts, while providing readable results through a set of mathematical and computational structures.

**Keywords** Visual attention · Brain programming · Cognitive vision

## Introduction

The goal of gazing toward the objects of interest with a camera attached to a computer has contributed to the creation of an important research area, devoted to the development of computational models for visual attention (VA). Nowadays, within the neuroscience community, VA is understood as a natural process performed by the brain, whose functionality is to perceive salient visual features. This cognitive task is necessary since it is impossible to focus the sight at two different objects during a single unit of time and space. Moreover, VA is a skill which allows a creature, living or artificial, to direct their gaze rapidly toward the objects of interest in the visual environment [1]. Thus, the objects of interest refer to those locations or regions in the image that are projected from the environment, and which contain important information at a given

---

L. Dozal · G. Olague (✉) · E. Clemente · D. E. Hernández  
CICESE, Carretera Ensenada-Tijuana No. 3918, Zona Playitas,  
Ensenada, B.C., Mexico  
e-mail: gustavo.olague@gmail.com

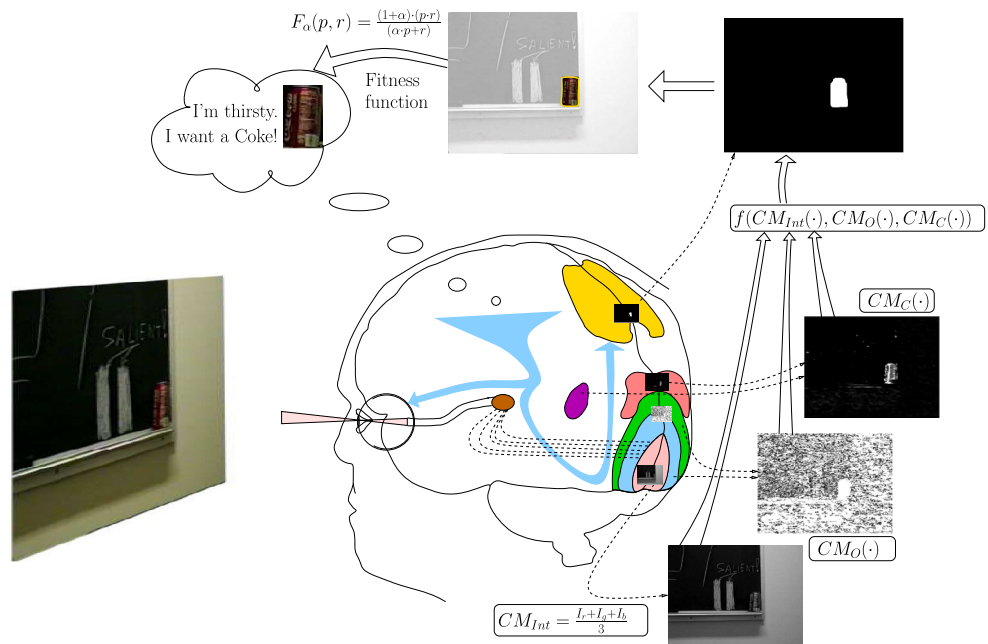
L. Dozal  
e-mail: leon.dozal@gmail.com

E. Clemente  
e-mail: eddie.clemente@gmail.com

D. E. Hernández  
e-mail: danielhdz.morales@gmail.com

E. Clemente  
Tecnológico de Estudios Superiores de Ecatepec, Avenida  
Tecnológico S/N, Esq. Av. Carlos Hank González, Valle de  
Anáhuac, Ecatepec de Morelos, Mexico

**Fig. 1** The correspondence between the dorsal stream areas and the stages of the artificial model. The idea is to emulate the transformations that the input image undergoes along the visual attention pathway of the natural system



time. Hence, visual attention is considered as one of the most important mechanisms in the visual system, whose functionality in the brain or visual cortical areas is to provide with the selectivity process that filters the visual information coming from the retina through the visual cortex. In the past, it was widely believed that human observers constructed a complete representation of everything in their visual field [2, 3]. Today, within the computational neuroscience and computer vision communities, it is well known that in order to process the high amount of visual information, it is necessary to break down the problem of image understanding into a series of simple, local and serial image operations that can achieve near real-time performance with typical computer systems.

In nature, the VA functionality, regarding the localization of objects, is related to the brain areas around the dorsal stream, see Fig. 1. Thus, the dorsal pathway has been defined as projecting from V1 through V2, V3, the middle temporal area (MT), the medial superior temporal area (MST) and finally to the posterior parietal cortex [4]. Nevertheless, there is a lack of consensus about the specific brain areas, structure and functionality that conform the dorsal stream. For example, in the work of Milner and Goodale [5], the dorsal stream is also known as the “how” stream; while in the work developed by Baluch and Itti [6], the dorsal stream process differs significantly with respect to previous works.

Nowadays, classical explanations of VA are in accordance with the idea that the dorsal stream is influenced by bottom-up (BU) and top-down (TD) factors. Thus, some works explain the existence of two interacting neural systems that are

involved in the control of both factors to approach the problem of visual attention [7–12]. In general, TD or cognitive factors are voluntary; i.e., knowledge, expectations, desires and current goals. While, BU or reactive factors depend entirely on stimulus; i.e., novelty and unexpectedness. As a result, VA is usually studied as two separate proposals [8] focusing on BU or TD processes. Next, we formulate the main problem to be studied in this paper.

### Problem Statement

This paper describes a system that automatically designs novel computational models of an artificial dorsal stream (ADS). The problem is that of solving the automatic design of visual attention programs (VAPs) without manually reengineering the best solutions according to the task at hand; i.e., the application to a specific image database. Therefore, VA can be understood as a process that seeks to emulate its natural counterpart by mimicking two main tasks [26].

- *Acquisition*, in this stage, features are captured early, automatically and in parallel across the visual field using visual receptors. Normally, the receptors are specialized to react to specific scene properties such as orientation, color, brightness and movement. Afterward, the information of receptors is transmitted and processed along the different areas of the brain.
- *Integration*, in this stage, the separate visual features present in the environment are combined in order to perceive the most prominent region while seeking to attend the objects of interest contained within the scene.

In this work, VA is seen as the product of an optimization process, where BU and TD factors are integrated to produce this single basic phenomenon. Thus, as in biological VA, the algorithm is composed of two tasks that are used in the perception and action cycle. The first task, related to perception, is ascribable to the limited processing capacity of the visual system; while, the second function related to action is a consequence of the selectivity or ability to filter unwanted visual information. In this way, it is said that VA selectivity is controlled by both cognitive or TD factors such as knowledge, expectation and current goals; as well as, involuntary or BU factors that refer to reactive and sensory stimulation [9, 11]. Nonetheless, an emphasis is given here to the ability of solving VA problems under a unique optimization framework [13]. It is for this reason that in this paper VA is studied as a unique process resulting from a single mechanism designed to obey a given general purpose, which could be made of different particular goals aiming to design a relationship between the observer and the scene. In particular, this paper details the application of the proposed methodology for the solution of challenging TD problems that have been previously employed in the literature and which are used here as a benchmark to compare our proposal with four state-of-the-art algorithms [14]. We considered this framework as an *unified approach for VA* [15].

As a result, we identify some problems related to both stages of VA. Firstly, the selectivity process related to the acquisition stage is understood as the ability to filter out unnecessary visual information and whose main aim is to render visible the objects of interest. In other words, selectivity responds the questions: “What features should be utilized?” and “When to use a given feature?” However, the answers are not evident and should be searched among the space of possible VAPs. Secondly, the integration of the best possible visual features into a single map of salience, which is seen as a difficult task since different visual dimensions produce a combinatorial problem. Finally, we have the problem of formulating VA in terms of an optimization/search process that looks for an ADS capable of attending a given target. In this paper, we will illustrate how an evolutionary approach allows us to address these issues by implementing a brain programming (BP) strategy.

### Research Contributions

This paper outlines the following research contributions.

- First, the automated design of an ADS is explained, where the problem is formulated in terms of an optimization/search-based approach. The explanation presented extends the previous one of Dozal et al. [13, 15] and Olague et al. [16].
- Second, a new strategy called brain programming mainly based on genetic programming (GP) is introduced. The approach follows the hierarchical paradigm developed within the computational neuroscience community in combination with the idea that special functions, in the form of mathematical and computational programs, can be evolved to solve the problem of VA. Indeed, the proposed approach is able to design competitive programs that challenge the designs of human experts according to the achieved results using a standard testbed.
- Finally, the performance of each experiment is reported in Table 5, followed by a detailed analysis of each successful experiment. In this way, the final structures of the best VAPs are described together with some details about the inner workings of the system. Moreover, the article presents results that give statistical insights about the dynamics and overall performance of the evolutionary system.

This paper is organized as follows. A first section devoted to the review of previous works in the area of VA is presented, while giving a brief introduction to GP. Next, the methods of ADS and brain programming are explained from a computational standpoint. Finally, experimental results are provided followed by the conclusions.

### Related Work

Since the late nineteenth century, the VA mechanism has been studied by researchers from different scientific disciplines, such as neurologists, physiologists, psychologists, and in the last three-decades by people working at the intersection of neuroscience, cognitive science and artificial intelligence, which is commonly known as cognitive computation. This last community has studied the problem of VA as a feasible way to reduce the complexity of visual information processing [17–21]. Recently, a discipline known as cognitive vision [22, 23] was created from the combination of the computer vision and cognitive research areas. Thus, it is said that cognitive vision systems should be able to engage in purposive goal-directed behavior, while adapting robustly to unexpected changes of the visual field, and it should have the ability to anticipate the occurrence of objects or events [22]. Moreover, VA is considered a critical issue in cognitive vision [24].

Owing to the importance of the VA mechanism, many different definitions have been formulated. Some of those ideas are summarized next. At the beginning of the 1980s, Posner et al. [25] proposed that VA could be understood in terms of a “spotlight” that enhances the efficiency of detection. Later, Treisman and Gelade [26] compared VA with the concept of “glue,” which integrates the initially

separable features into unitary objects. Then, in the same decade Koch and Ullman [1] described VA as a skill which allows the creature to direct their gaze rapidly toward the objects of interest. Later, two different proposals were developed. In the middle of the 1990s, Desimone and Duncan [9] introduced VA as an emergent property of visual processing that is seen as the product of many neural mechanisms working to resolve the problems of resource competition and control of behavior. Finally, Wolfe [27] explains VA as an “enabler” rather than an actor. In other words, instead of saying that attention somehow identifies an object it is said that attention enables object recognition processes to work on a single item at a given time. Hence, after considering the previous works, we introduce a new VA definition consisting on a functional relationship that filters the visual information according to the task at hand.

**Definition 1** (*Visual attention*) The concept of VA is defined as a process that establishes a relationship between the different properties or features of the scene, which are perceived through the visual system, with the aim of selecting the most suitable aspect for the task at hand.

In this way, we consider that VA performs both the BU and TD processes through a single computational structure with the goal of achieving the task at hand while considering a *purposeful framework* [28]. Therefore, according to *the feature-integration theory* of Treisman and Gelade [26], the VA process is divided in two stages: acquisition and integration. Firstly, during the acquisition, it is said that the receptors are adapted to perceive special features such as orientation, color, brightness and movement. Secondly, the information of receptors is transmitted and processed along different areas of the brain with the goal of creating a suitable description of the objects of interest. Therefore, during the integration stage, all separate features used by the focus of attention task are merged into a single computational structure describing the salient region. In this way, from a computational standpoint, the whole program structure can be seen as a way of describing a region of interest within an image; where the final result is useful for the visual perception task of attending the objects of interest.

#### *Brief History of VA from a Computational Standpoint*

As we have reviewed *the feature-integration theory* of visual attention was proposed by Treisman and Gelade [26]. This theory is the most widely accepted paradigm for VA within the cognitive science community, since it is considered as a fundamental step for VA. In this way, the first neurological and plausible computational model for VA was outlined by Koch and Ullman [1] where it is introduced a model composed of three stages. First, a set

of elementary features are computed across the visual field in parallel. Second, the resulting maps are combined into a single saliency map that encodes the prominence of the scene. Third, a winner-take-all (WTA) network is used to iteratively select the most prominent regions. As a result, the WTA network shifts to the next most prominent region after inhibiting the most active location. A year later, Fukushima [29] proposed a model of selective attention useful for visual pattern recognition, which mimics the functionality of the human brain. This model is a hierarchical multilayered neural network capable of paying attention sequentially to two or more patterns. Afterward, Burt [30] described three mechanisms for VA and their computational implementation: foveation, tracking and high-level interpretation. Then, Sandon [31] in 1990 made the first implementation of a VA model based on the work of Koch and Ullman. Later, Olshausen et al. [32] proposed a neurobiological solution to the problem of object recognition using a model of VA that is capable of creating objects representations that are invariant to position and scale. In this case, their work was motivated by the theory of Palmer [33], who suggested that the result of attention should be focused on the object, considering a canonical or object-based reference frame. At the same time, Milanese [34] proposed a VA system that uses mechanisms inspired from biological processes, which were further adopted by the computer vision community and are now commonly utilized until these days [14]. Moreover, Tsotsos et al. [35] developed a biologically plausible model of VA making special emphasis to the computational utility through the application of special VA routines. In addition to the computational explanations that had been proposed in the literature, their idea includes a new winner-take-all algorithm that better matches the functionality of the primate visual system by providing a method for controlling a robot vision system. Nevertheless, one of the most well-known computational models of VA is probably that of Itti et al. [36], which provided a software that popularized the theoretical processes. Later, Itti and Koch [14] compared four different strategies for feature combination to achieve object detection. In their work, they built a simple feed-forward architecture that consists of two main stages: extraction of early visual features and saliency map computation. The first stage could be implemented in parallel, rendering it suitable for real-time applications. Later, Torralba [37] in 2003 created a VA model that incorporates contextual statistical information. The quality of the results showed that contextual information improves object detection by directing the visual search toward the prominent features of the scene. Another important work is that of Walther and Koch [38] where a biologically plausible model is proposed for

forming and attending proto-objects aiming to enhance the task of object recognition. Afterward, Cutsuridis [39] discusses a cognitive model of saliency based on the interaction between BU and TD stimulus, which drives VA between saccades. In his work, he offered a hypothesis of how the participating brain areas work together to attend a specific region from a visual scene. Another study developed by Kootstra et al. [40] propose a BU model that predicts eye fixations based on symmetry instead of the contrast feature. Their main result shows that symmetry provides a better prediction in the case of human eye fixations. Finally, Marat et al. [41] considered the problem of face recognition; they proposed the incorporation of a “face” saliency map within existing VA models. The idea is that this map emphasizes areas where a face should be detected in order to improve the prediction of a set of eye movements during video playing.

### *Brief Introduction to GP*

Evolutionary computation incorporates a wide range of techniques to solve problems based on Darwin’s theory of evolution; and the well-known principle of natural selection, commonly known as “survival of the fittest.” Today, the most popular of evolutionary algorithms is GP which was formalized by Koza [42]; and it is understood as a machine learning approach useful in the design of computer programs. GP is an offshoot of genetic algorithms that automatically solves problems without requiring the designer to know or specify in advance the final form or structure of the problem solution. Thus, GP is said to be a systematic, domain independent method; nevertheless, the user usually needs to define the way in which GP evolves programs often in a domain-specific language, through a user-defined task, using a set of functions and terminals together with the given criteria.

In artificial evolution, such as in nature, there is a population of individuals. In GP, the individuals in the population are computer programs, which are usually written as syntax trees or as corresponding expressions in prefix notation. Each individual in the population is evaluated to establish its quality at solving a given problem. Note that such measure is called the fitness function, which is widely known as the objective function. In general, the individuals in the initial iteration, or generation, of the process will have poor fitness. Nonetheless, some individuals in the population will prove to be slightly fitter than others. The fitter individuals survive and produce offspring that replace older and relatively unfit individuals from the population. In Koza’s style of GP, the population is usually of fixed size. The paradigm incorporates the idea of genetic inheritance through some specialized computational operations that are used to

produce new programs by either mutation or crossover. The Darwinian principle of reproduction and survival of the fittest, as well as the genetic operations of crossover and mutation, are both inspired by the biological exchange of genetic material and are used to create a new set of individuals to improve the current population. Mutation operators make a random change to a copy of a selected program. In other words, a selected point mutation replaces a node within the tree by another randomly generated tree without imposing an arity constraint; i.e., number of arguments. Crossover works with two selected programs, called parents, from the current population; the idea is to exchange code between parents by randomly selecting subtrees. In Koza’s style of programming subtree, crossover removes one branch from one tree and inserts it into the other. These simple processes, mutation and crossover, should ensure that the new programs are syntactically valid. Also, a problem known as bloat, related to the uncontrollable growth of the programs, is usually approached with a user-defined bound.

Thus, GP can be seen as a machine learning approach that looks into the search space of all possible computer programs that are defined appropriately to the problem domain. In the same way, as any other stochastic process, the GP algorithm can never guarantee an optimal solution; however, this kind of heuristic methods can help in the solution of problems where deterministic methods are unable to offer a suitable solution. In our GP-based approach, that we are calling brain programming, a set of programs are searched with the goal of synthesizing key mathematical operations applied within the hierarchical computational structure of the ADS.

### **Methods**

This section provides a general overview about the proposed optimization methodology useful in the design of an ADS. First, we introduce the biologically inspired model for VA that serves as the general framework; it is on this model where key functions will be optimized through the paradigm of artificial evolution. Next, we provide some details about the implementation of the proposed BP algorithm, which has been applied to generate optimal ADSs to solve a challenging VA-database problem.

#### The ADS Algorithm

The ADS algorithm is as a computational model that mimics the biological process of focus of attention [1, 29, 30]. The ADS attempts to emulate the natural process that takes place along the visual cortex, specifically through the

dorsal stream. Today, the ADS is said to perform its task according to the neurological *theory of feature-integration* [26]. This theory states that the operation of an ADS is divided into a two main-stage process that works sequentially to fulfill the VA task. The stages are 1) visual feature acquisition and 2) feature-integration. From a computational perspective, the ADS is comprised of several functions arranged in a forward manner. Figure 2 shows the flow diagram of the ADS algorithm, and in the following subsections, we explain both stages of the ADS.

### Visual Feature Extraction and Visual Maps

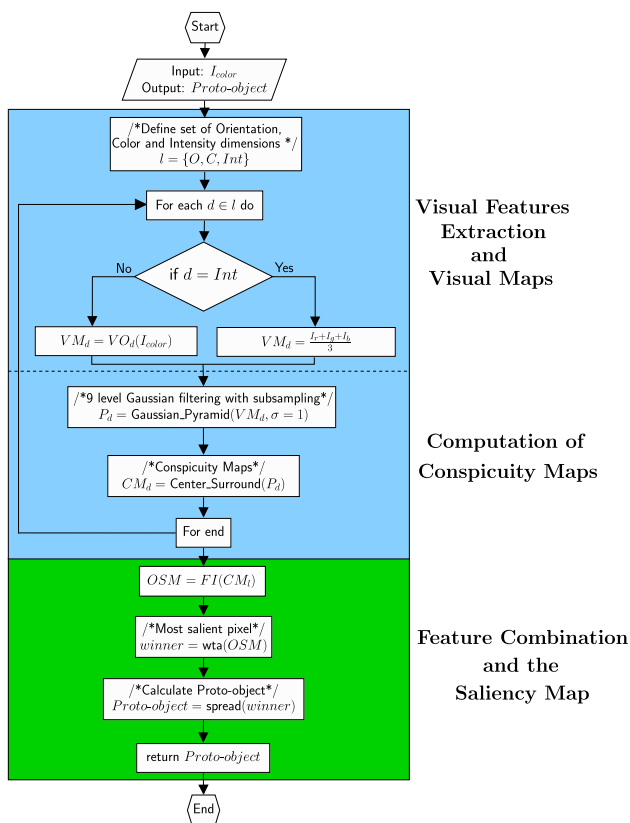
In the visual system of primates, visual features follow a pathway across the retina, superior colliculus, lateral geniculate nucleus and early visual cortical areas [43]. Along the pathway, the low level mechanisms applied to the task of feature extraction act in parallel over the entire visual field to provide the stimulus for the subsequent stages of the ADS, which highlight the prominent regions of the image [1, 26]. In fact, it has been shown by Julesz [44] that a set of texture discriminators, called textons, provide a specific set of features that are detected in parallel. These features are specialized in the detection of

color, line orientation and certain shape parameters such as curvature and convexity. Moreover, Treisman and Gelade [26] also described such features as acting in parallel with each feature defined to conform a particular dimension. Next, some basic definitions are introduced to understand the overall strategy.

**Definition 2 (Image as the graph of a function)** Let  $f$  be a function  $f : U \subset \mathbb{R}^2 \rightarrow \mathbb{R}$ . The graph or image  $I$  of  $f$  is the subset of  $\mathbb{R}^3$  that consist of the points  $(x, y, f(x, y))$ , in which the ordered pair  $(x, y)$  is a point in  $U$  and  $f(x, y)$  is the value of  $f$  at that point. Symbolically, the image  $I = \{(x, y, f(x, y)) \in \mathbb{R}^3 | (x, y) \in U\}$ .

This definition recalls that the scene is perceived through a camera composed of a two-dimensional array of sensors that measures the amount of light that reaches them. Note that the definition can refer to an image taken with a camera or to any of the false images issued by an image operation. In this way, the images are the result of the impression of light across the two-dimensional sensor plane, as well as the output issued by the image processing. Moreover, the image  $I$  obtained by the camera is defined as the graph of a function, since this mathematical concept is used at the moment of describing the transformations that will be used in our functional approach. Thus, this definition helps us to understand the image as the initial input to a function-driven paradigm of the visual cortex model. Previous works [4, 5, 9] described the brain as divided into several specialized areas, each of them performing a specific task. In our work, we claim that such functionality can be imitated by a set of mathematical functions and operators together with some computational structures.

In general, digital color images are composed of three bands at different wavelengths of light: red, green and blue. In this way, it is possible to transform an image that is represented in the RGB color model into another, such as the CMYK and HSV models. In this paper, the input set of images is defined as follows:  $I_{\text{color}} = \{I_r, I_g, I_b, I_c, I_m, I_y, I_k, I_h, I_s, I_v\}$  that provides the initial representation of the scene. Next, the input image in  $I_{\text{color}}$  is transformed by a visual operator (VO) to recreate the feature extraction process of the brain, resulting into a visual map (VM). In our work, each VO is defined as a mapping  $\text{VO}_d : I_{\text{color}} \rightarrow \text{VM}_d$  where the domain of the function is  $I_{\text{color}}$  and the codomain of this operation is a visual map  $\text{VM}_d$  along a particular dimension. Each kind of feature is classified or grouped into several dimensions, such as color, orientation, spatial frequency, brightness and direction of movement; this is according to the theory of Treisman and Gelade [26]. The current work follows the implementation of Itti et al. [36] where the transformed pixel value is represented by



**Fig. 2** Flowchart of the artificial dorsal stream

the feature prominence along each dimension considering orientation, color and intensity; hence,  $d \in \{O, C, \text{Int}\}$ . Figure 2 shows that features are extracted sequentially one at a time by applying the corresponding operator  $\text{VO}_d$ . These features are detailed next.

**Orientation** The extraction of orientation information on the natural visual system occurs by the action of simple and complex cells at the primary visual cortex (V1), as well as in the V2 layer. These cells are in charge of estimating the sensitivity to the orientation stimulus by decomposing the image into a set of small linear segments at different orientations and scales [45]. From a computational perspective, the operator  $\text{VO}_O$  extracts prominent edges that are present in the image and is defined as follows:

$$\text{VO}_O: I_{\text{color}} \rightarrow \text{VM}_O. \quad (1)$$

The result of this operation is a visual map  $\text{VM}_O$  for which the pixel values represent the prominence of the feature considering the orientation dimension. Typical systems considered a first processing step that produces simple features modeled as a neural network or through convolutions of the input images with Gabor filters of various orientations, spatial frequencies and scales [29, 34, 36, 37]. The goal is to compute a variety of structures, such as corners, line intersections and parts of objects.

**Color** In the natural system, the color is encoded through a set of photoreceptor cells known as cones, which are located at the retina. The human color vision depends on three types of photosensitive molecules: short wavelength or blue sensitive (SWS), middle wavelength or green sensitive (MWS) and long wavelength or red sensitive (LWS) [46]. In this way, the visual pigments consist of the molecules that transform light energy into an electrical impulse. In nature, the yellow color is a special case since it cannot be perceived by the cones but rather at the retinal ganglion cells. Moreover, it is well known that in several modules of the visual cortex, there are cells that respond to color stimulus as in the V1, V2, and V4 layers. On the other hand, on the computational framework these ideas are achieved with the following visual operator:

$$\text{VO}_C: I_{\text{color}} \rightarrow \text{VM}_C. \quad (2)$$

The result of this operation is an image or visual map  $\text{VM}_C$  containing the prominence in color. In previous work, these pre-attentive mechanisms, sensitive to color, are computed through spatial differences on the image channels using a center-surround contrast; i.e., red–green and blue–yellow color opponencies [34, 36, 38]. The goal is to find high-contrast regions within the image.

**Intensity** The intensity is a magnitude that indicates the amount of light striking a photosensitive receptor. Physiologically, human beings have specialized ganglion cells to record this feature. In general, each ganglion cell has a circular receptive field that responds to the intensity of light reaching it [9, 26]. In order to obtain the intensity a simple formula is applied:

$$\text{VM}_{\text{Int}} = \frac{I_r + I_g + I_b}{3}, \quad (3)$$

where  $I_r$ ,  $I_g$  and  $I_b$  are the red, green and blue bands of the image in the RGB color space [34, 36, 38]. The result of this operation is a visual map  $\text{VM}_{\text{Int}}$  that is directly computed with the pixel values, obtained by the camera, and which represents the intensity defined with the above formula.

Although the acquisition of visual features can be performed in a parallel manner in the implementation of the ADS algorithm, depicted in Fig. 2, a for loop is used to iterate across the dimensions. During one loop of the algorithm, the operator  $\text{VO}_d$  is applied over the input  $I_{\text{color}}$ . After the construction of each visual map, the system must calculate an additional map known as the conspicuity map for each dimension. The following section is dedicated to the description of such process.

#### Computation of the Conspicuity Maps

Once the VMs are obtained, the next step is to compute the conspicuity maps (CMs). These maps are obtained by means of a function that is applied to simulate the center-surround (CS) receptive fields. This part of the computational system is inspired from the natural structure that measures the differences between firing rates at the center ( $c$ ) and surroundings ( $s$ ) areas of the ganglion cells. In our work, the CMs are obtained following the Walther and Koch model [38]. The computational center-surround mechanism compares the average value of a center region to the average value of a surrounding region in the visual receptive fields [18]. In particular, the center-surround is implemented in the model as the difference between fine and coarse scales using a pyramid of nine scales [36]. This function is defined as follows:

$$\text{CS}: \text{VM}_d^A \rightarrow \text{CM}_d \quad \forall d \in \{O, C, \text{Int}\},$$

where  $A = \{1, 2, \dots, 9\}$  is the set of scales  $\alpha \in A$ , and  $\text{VM}_d^A$  is a pyramid for each dimension  $d$  that is obtained by employing a Gaussian filter applied to each of the corresponding VMs using  $\sigma = \alpha$ , which can be written as  $\text{VM}_d^\alpha = G_{\sigma=\alpha} \star \text{VM}_d$  with  $\star$  representing the convolution operation. In this way, the CS mapping is composed of two steps that are defined next.

- First, an across-scale subtraction  $\ominus$  is performed resulting in a center-surround pyramid  $VM_d^o$  of six levels; in such a way that the value of the pixel corresponds to a difference of Gaussians from the nine level pyramid  $VM_d^A$ . Thus, the across-scale subtraction is defined as follows:

$$VM_d^o = N(|VM_d^c \ominus VM_d^s|) \quad \forall d \in \{O, C, Int\},$$

where  $N(\cdot)$  is a normalization operator,  $c = \{3, 4, 5\}$  and  $s = \{c + 3, c + 4\}$ . Note that for each scale value of  $c$  exists two values of  $s$ ; where  $s, c \subset A$  and the final result is a pyramid with six levels  $\omega = \{1, 2, \dots, 6\}$  calculated through six across-scale subtractions.

- Second, the  $VM_d^o$  pyramids for each dimension perform an across-scale summation  $\oplus$  in order to obtain all conspicuity maps  $CM_d$ . This can be written as follows:

$$CM_d = N(\oplus_{\omega=1}^6 VM_d^o) \quad \forall d \in \{O, C, Int\}.$$

At this point, we have a normalized CM for each dimension. Once that the feature extraction and center-surround processes are finished the “for” loop ends, see Fig. 2. The next stage on the ADS algorithm deals with the combination of the CMs into a single saliency map.

#### Feature Combination and the Saliency Map

The aim of the feature combination process is to fuse the CMs into a single map of salience information. This process highlights a set of locations through a saliency map (SM), resulting in a set of prominent regions within the image. Anatomically, the location of the SM within the brain is unknown. Moreover, within the neuroscience community, a clear description of how the brain makes such an integration is still a subject of discussion. In this way, Koch and Ullman [1] proposed as hypothesis that the SM resides either at the lateral geniculate nucleus (LGN), or at the striate cortex (V1). Along this line of research, several authors have proposed other explanations for the location of the SM; for example the thalamic nucleus and the pulvinar regions proposed in [47], the V1 as described in [48], the V4 region explained in [49], even at the posterior parietal cortex detailed in [50] and finally at the lateral intraparietal area as illustrated in [51].

The feature-integration process is a difficult task since the CMs are part of different and unrelated visual modalities working at the sensory system level. Indeed, we claim that a goal-driven paradigm could serve the purpose of integrating such problematic and apparently different characteristics of the feature-integration process. In this way, most models involving BU and TD factors can be seen as highly relevant regarding the achievement and

pursuance of the VA process. Thus, the criteria that guide the search toward the most suitable combination of characteristics should be defined in accordance with the task at hand. Therefore, the integration of CMs is accomplished by a feature-integration operator FI as follows:

$$FI: CM_d \rightarrow SM \quad \forall d \in \{O, C, Int\}. \quad (4)$$

Commonly the SM is the result of the average of all CMs. Once the integration of features is achieved, a WTA network is applied to compute the most salient pixel from the resulting SM. Then, a region around the winner, known as proto-object, is delimited using a spread function that is controlled by a threshold. Thus, the proto-object indicates the location of the most prominent region within the original image [52, 53]. Note that an image can contain more than one proto-object; hence, the WTA process should be capable of distinguishing several regions. The feature-integration stage is depicted in Fig. 2.

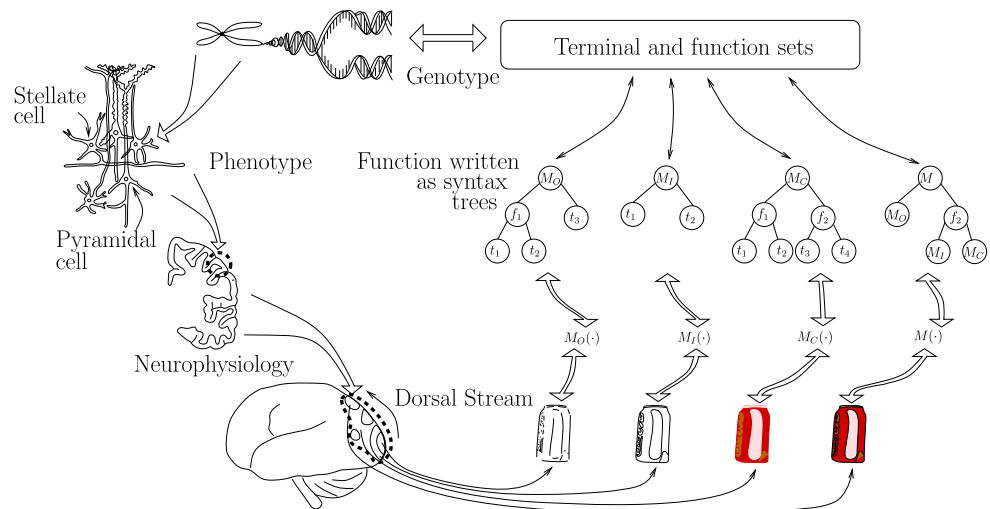
#### Brain Programming

The present work follows a function-driven methodology that approaches the biological visual process from the standpoint of its functionality, while paying special attention to the goals. Our method differs to the data-driven approach that is based on a large amount of knowledge, defined by the experts, encoded as a set of patches; where a patch is a small contrasting part of an image. Those patches come from the hard-coded operations that are applied at the different stages of the ADS. The proposed methodology, brain programming, describes a manner in which an ADS can be optimized to imitate the functionality of specialized areas of the brain through a set of operators. This paper presents a function-driven approach to extract and combine the relevant information that solves a specific visual task; in this case the VA problem. In particular, we claim that brain programming, a GP-based strategy embedded into a well-known hierarchical process and combined with key elements such as center-surround mechanisms, including normalization and pyramid scale processes, are able to provide with the suitable tools that are necessary to implement this methodology in highly creative ways.

Clearly, the classical approach of single or multiple tree GP process is insufficient to evolve the whole ADS owing to the high level of complexity. Nevertheless, the approach described in this paper provides a strategy that is able to automatically design such programs or solutions. In this section, we describe the main aspects to implement the BP, a new evolutionary technique, with the aim of designing ADSs for VA problems, which will be further tested against four state-of-the-art TD strategies using a well-known database. Thus, BP is a bioinspired strategy where the major changes



**Fig. 3** The analogy between the natural and artificial systems. The idea is based on the evolution of several mathematical operators, immersed or encapsulated within the artificial dorsal stream, that emulate the functionality of the corresponding natural areas



with respect to the GP came from the fact that the possible solutions or individuals cannot be defined as an array of trees, but rather as a complex structure composed of several trees and other predefined processes. The motivation for these changes is the idea of emulating the functionality of an organ, such as the brain, and its complexity, see Fig. 3. Our approach is inspired by the idea that the brain is subject to the process of evolution [54]. We introduced these changes in order to deal with the evolution of complex structures to be tested in a challenging classical problem of VA. As a result, BP exhibits the ability to design novel VA programs that achieve a lower false detection rate than those designed by human experts.

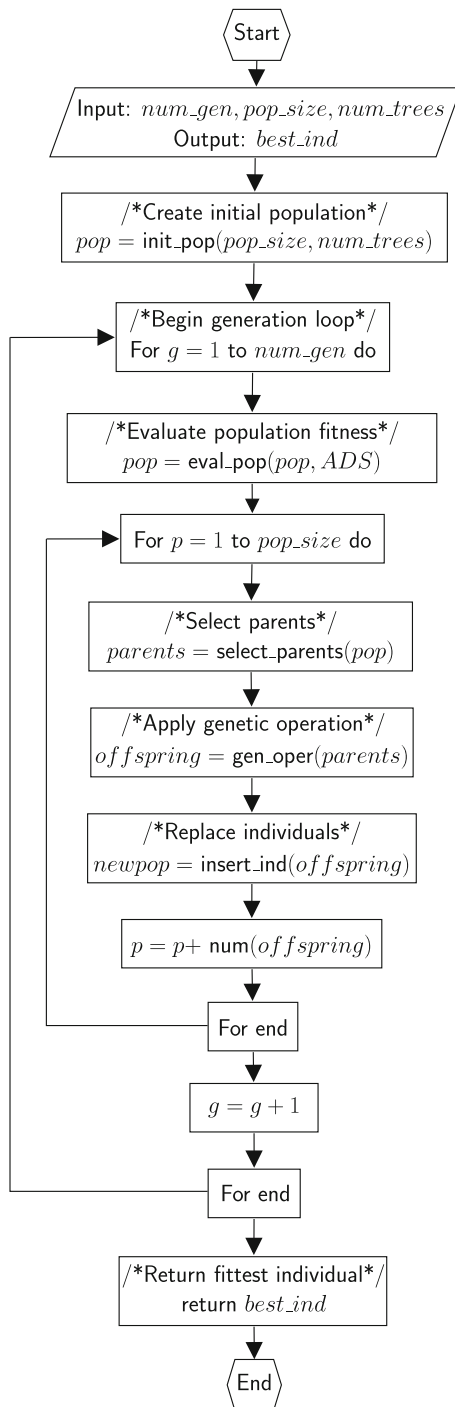
### Genetic Representation of the ADS

In our work, the genetic representation of the ADS, the genotype, consists of a triplet of trees embedded within an ADS. In GP, programs are usually encoded as syntax trees made up of internal and leaf nodes, which are defined by a set of primitive elements also called function set, and terminal set. Thus, the sets of functions and terminals represent the problem search space. Note that we will refer to the genotype only by the triplet of trees although the final result, phenotype, is achieved by a more complex system; in this paper the ADS. Indeed, the genotype looks for a set of key operations that in combination with the whole structure define the desired design by mimicking the operations at different brain areas. Nevertheless, our implementation of brain programming follows a general GP-based approach since the genetic representation is the link to the real-world problem together with the definition of the fitness function. Figure 4 provides the classical diagram of the sequence of operations in an evolutionary algorithm where the evaluation of the fitness function is calculated through a measure applied to the ADS. In this

section, we will explain the necessary steps that were taken to define the genetic representation of the ADS. The idea is to evolve the different VOs, as well as the operation in charge of the feature-integration stage. In the rest of the paper, we will refer to such operations as the evolved visual operators (EVOs) and the evolved feature-integration (EFI), resulting into an optimized saliency map (OSM). Note also that this last operation could be related to the concept of automatically defined functions [42] despite the fact that the EVOs working as terminals are not directly feeded into the EFI tree and that the EVOs are evolving within a fix structure.

In this paper, each tree has its own sets of functions and terminals that were carefully chosen according to the desired functionality that we attempt to emulate. The first tree mimics the orientation, or the functionality of the orientation-sensitive cells in V1. Thus, we propose to evolve the operator for orientation ( $EVO_O$ ) through a set of elements specially selected to highlight edges, corners and other orientation-related features using the set of terminals and functions provided in Table 1. The notation is as follows: the input for the functions in  $F_O$  can be any of the terminals in the  $T_O$  set, as well as the composition among them;  $G_\sigma$  are Gaussian smoothing filters with  $\sigma = 1, 2$ ; and  $D_u$  represents the image derivatives along the direction  $u \in \{x, y, xx, yy, xy\}$ .

The second operator encodes the color dimension or the operation of photoreceptor cells and color-sensitive cells present in the V1 and V4 layers of the visual cortex. Again, we propose to evolve the color visual operator ( $EVO_C$ ) to reproduce the color perception process. In fact, there is a theory that explains how color perception has evolved in nature [55]; thus, it makes sense to evolve such process. In the case of artificial evolution, the evolutionary process uses the set of functions and terminals provided in Table 2 to create the  $EVO_C$ . The notation provided in Table 2 is as follows: the input for the functions in  $F_C$  can be any of the



**Fig. 4** Flowchart of the brain programming strategy

**Table 1** Functions  $F_O$  and terminals  $T_O$  applied for building the operator  $EVO_O$

$F_O = \{+, -, \times, \div,   +  ,   -  , \sqrt{\quad}, ()^2, \log_2(\quad), G_{\sigma=1}, G_{\sigma=2},   , 0.5 * (\quad), D_x(\quad), D_y(\quad)\}$
$T_O = \{I_{\beta} \in I_{color}, G_{\sigma=1}(I_{\beta}), G_{\sigma=2}(I_{\beta}), D_x(I_{\beta}), D_y(I_{\beta}), D_{xx}(I_{\beta}), D_{yy}(I_{\beta}), D_{xy}(I_{\beta})\}$

**Table 2** Functions  $F_C$  and terminals  $T_C$  used by the operator  $EVO_C$

$F_C = \{+, -, \times, \div,   +  ,   -  , \sqrt{\quad}, ()^2, \log_2(\quad), \exp(\quad), \text{complement}(\quad)\}$
$T_C = \{I_{color}\}$

**Table 3** Functions and terminals applied within the EFI stage

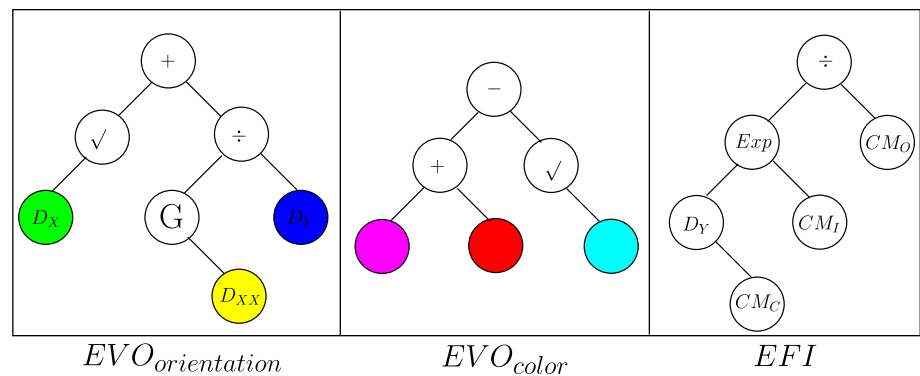
$F_{FI} = \{+, -, \times, \div,   +  ,   -  , \sqrt{\quad}, ()^2, \exp(\quad), G_{\sigma},   , D_x(\quad), D_y(\quad), EQ(\quad)\}$
$T_{FI} = \{CM_{int}, CM_O, CM_C, D_u(CM_{int}), D_u(CM_O), D_u(CM_C)\}$

terminals in  $T_C$ , as well as the composition among them. Note that some functions of  $EVO_C$  are the same of those in  $EVO_O$ . In addition, the function `complement()` provides a negative image that is the complement of an intensity or RGB image; in other words, each pixel value is subtracted from the maximum pixel value and the difference is used as the pixel value in the output image. Thus, in the output image, dark areas become lighter and light areas become darker.

Finally, the third tree encodes the way in which the features are combined to obtain the OSM. This is accomplished with the idea of creating a fusion operator that highlights the features of the objects of interest. Here, the evolutionary method uses the set of functions and terminals listed in Table 3. Note that the set of terminals is completely different to the previous ones, and it incorporates the output of the previous stages of the ADS, not to be confused with the output of the EVOs. The functions are similar to some elements of the function set used in  $EVO_O$  with the addition of `exp()` and the equalization of the histogram `EQ()`.

Figure 5 shows an example of the proposed genotype that we are applying during the evolution. These three functions are embedded into the flow diagram of Fig. 2 and evolved with the evolutionary program outlined in Fig. 4. At the beginning of the BP algorithm, it receives as input the number of generations, the population size and the number of trees to be evolved per individual. All individuals of the initial population are a triplet of random trees indicated by the *num\_trees* parameter. In our experiments, the number of generations, *num\_gen*, is 30 and the number of individuals per population *size\_pop* is also 30. The initialization method that is applied to the population is known as the *ramped half-and-half* method proposed by Koza [42]. This approach creates half of the initial population with the *grow* method and the other half with the *full* method. The *grow* method produces unbalanced trees allowing branches of different lengths while the *full* method makes balanced trees, with all branches of the

**Fig. 5** Example of the ADS's genotype



**Table 4** Main parameters settings of the BP algorithm

Parameters	Description
Generations	30
Population size	30 individuals
Initialization	Ramped half-and-half
Crossover at chromosome level	0.4
Crossover at gene level	0.4
Mutation at chromosome level	0.1
Mutation at gene level	0.1
Tree depth	Dynamic depth selection
Dynamic max depth	7 levels
Real max depth	9 levels
Selection	Roulette-wheel
Elitism	Keep the best individual

same length. The length of the tree must not exceed the specified maximum to prevent that trees grow up without control during evolution. In our work, the size of the individuals should not exceed the user-defined maximum depth in order to avoid uncontrolled growing of the trees to prevent bloat. The depth of a tree is defined as the length of the longest non-backtracking path from the root to an endpoint. Tree depth limits the size of any given individual within the population and it is dynamically controlled by two maximum tree depths parameters. The first is the *dynamic max depth* which is a maximum tree depth that may not be surpassed by any individuals unless its fitness is better than the best solution found so far. If this happens, the dynamic max depth is augmented to the tree depth of the new fittest individual. Conversely, it is reduced if the new best individual has a lower tree depth. The second one is the *real max depth* that is hard limited and which no individual may surpass under any circumstances. Finally, the termination criterion was defined as the maximum number of generations; thus, the evolutionary process reaches at least a suitable top-down model during each single run. Table 4 summarizes the parameter values applied by BP during the experimental runs. These parameters have standard values [42].

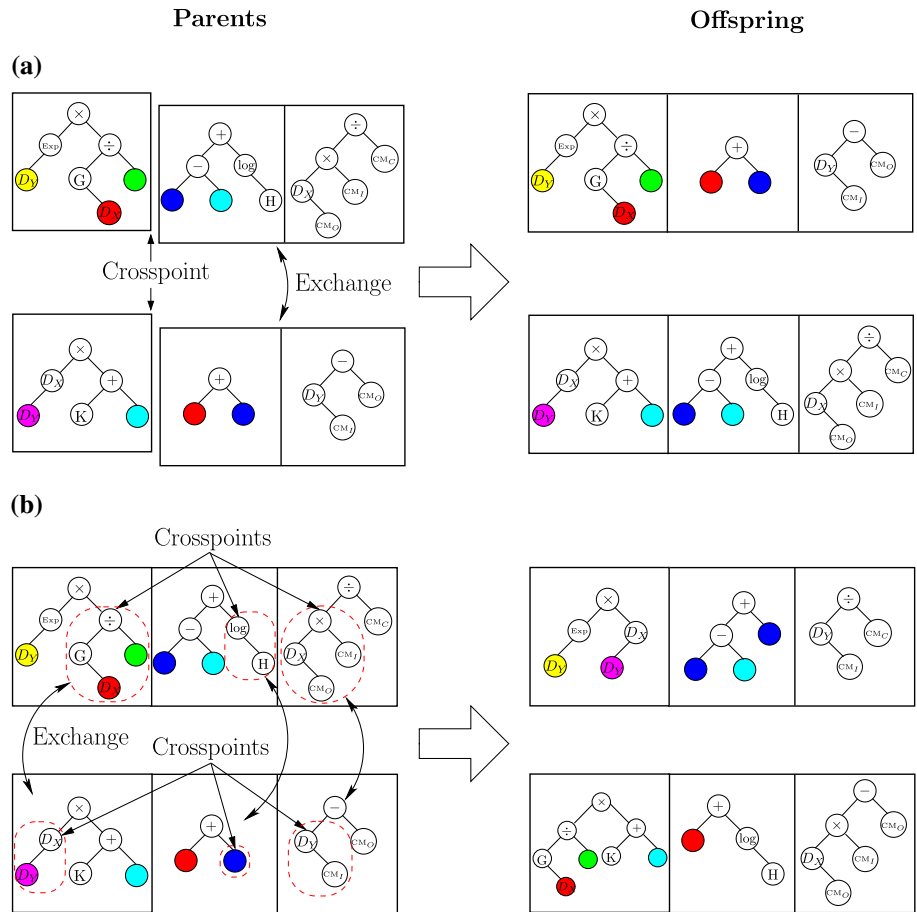
In the following sections, the explanation of the GP-based evolutionary process is further detailed through the definition of the applied genetic operations and the criterion for evaluation. The idea is to outline our approach and illustrate a new way of obtaining VAPs.

### Genetic Operations

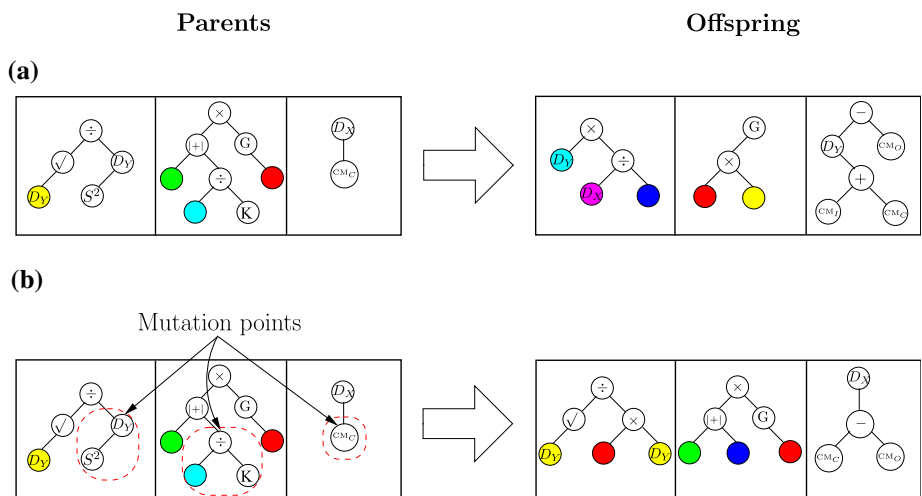
After the initial population is created, and evaluated with the fitness function, the next step of the BP algorithm is the selection and genetic variation of individuals from the population in order to breed the next generation. This is accomplished through the genetic operations of selection, crossover and mutation, see Fig. 4. The parents are obtained using a fitness-proportionate selection method implemented with the *roulette-wheel* strategy, which consists in assigning to each individual a probability of selection proportional to their fitness value. Thus, the fittest ADSs are more likely to be selected, and the genetic crossover and mutation are sequentially applied until a new population is created. Note that the difference between GP and BP arises from the genotype representation, the hierarchical representation and special processes such as the Gaussian pyramid and center-surround process. In this way, the existence of a higher level of complexity within the genotype structure motivates the development of new genetic operators acting on different stratum that we call gene and chromosomal levels.

Both methods of crossover need a pair of parents to perform the corresponding operation. In the case of chromosomal crossover, the crosspoint is located at chromosome level and the corresponding parts of the genotype are swapped to produce the new offspring. For the gene crossover, multiple crosspoints are chosen randomly at each parent in the respective trees; then, the selected subtrees are exchanged between the corresponding parents to create the new offspring; see Fig. 6. On the other hand, the chromosomal mutation is applied to the triplet of trees in such a way of randomly substituting all genes of the compound chromosome with new operators. Finally, the

**Fig. 6** Schematics that illustrate the crossover operations at the chromosomal and gene levels. **a** Chromosomal Crossover, **b** Gene Crossover



**Fig. 7** Schematics that illustrate the mutation operations at the chromosomal and gene levels. **a** Chromosomal Mutation, **b** Gene Mutation



mutation at the gene level consists of the random selection of a node within a single tree, called point of mutation. This selected node is deleted and replaced by an entirely new sub-tree, which is randomly generated to create the new tree. This is carried out for each selected tree of the compound genotype, and the strategy is repeated to produce a new individual and eventually a whole new population; see Fig. 7. Note that if during the gene mutation, the

selected node is the root of the tree, the whole tree will be substituted by a completely new random tree.

In our work, during the production of the new population, the genetic operators are applied to find new solutions based on a probability that is assigned for each operation with the scheme proposed by Koza [42]. In this case, the operations are computed independently and their total probability adds to one. Hence, the crossover probability at gene and

chromosomal levels are equal to 0.4 each; while the mutation probability for both levels is equal to 0.2, see Table 4. These genetic operators allow the variation of the genetic material, while promoting genetic innovation of individuals through all levels, and maintaining the diversity of the population.

### Fitness Function or Objective Function

In brain programming like in any evolutionary method, after the population is created, all individuals are evaluated, see Fig. 4. For this reason, it is necessary to formulate a well-posed fitness function in accordance with the goal that the ADS is attempting to reach. In other words, the fitness function of BP represents the characterization of the purpose, the answer to the question “What are the individuals for?” [28]. Thus, like in all optimization approaches, this step defines the way of implementing the goal within a computer program. In this work, we propose to use the F-measure within the BP process, as the fitness function for evaluating the ADS performance for each generated structure. Next, we explain how the fitness function was implemented to approach a VA problem.

This section introduces the F-measure as the criterion for evaluating the ADS within the proposed optimization framework. The F-measure was created in the information retrieval (IR) community. Van Rijsbergen [56] originally proposed this measure through the Effectiveness function (E). The definition is considered as the original formulation of the F-measure, see Equation (5).

$$E = 1 - \frac{1}{\alpha \frac{1}{p} + (1 - \alpha) \frac{1}{r}}, \quad (5)$$

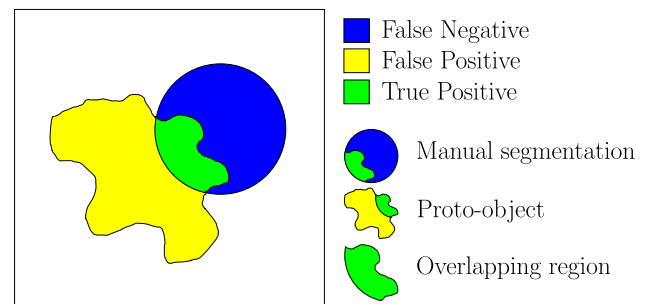
where  $\alpha = \frac{1}{\beta^2 + 1}$  and  $0 \leq \beta \leq \infty$ . In the context of IR, the computation of recall and precision is defined in terms of a set of retrieved documents  $F$  and a set of relevant documents  $R$ :

$$r = \frac{|F \cap R|}{|R|} \quad \text{and} \quad p = \frac{|F \cap R|}{|F|}, \quad (6)$$

where  $r$  is recall  $\{r: 0 \leq r \leq 1\}$  and  $p$  is precision  $\{p: 0 \leq p \leq 1\}$ . Note that the F-measure is the weighted harmonic mean of precision and recall. Moreover, the weights give the advantage of balancing the values of recall and precision. Thus, the general formula of the F-measure is defined as follows:

$$F_\alpha(p, r) = \frac{(1 + \alpha) \cdot (p \cdot r)}{(\alpha \cdot p + r)}, \quad (7)$$

where  $\alpha$  is the parameter that controls the balance between  $p$  and  $r$  with  $0 \leq \alpha \leq \infty$ . Note that if  $\alpha < 1$  the weight variable  $p$  is higher; while if  $\alpha > 1$ , then the weight



**Fig. 8** The diagram illustrates areas; manual segmentation, proto-object, overlapping region; that are used to compute precision and recall variables

variable  $r$  is higher, and when  $\alpha = 1$ , the precision and recall are said to be well-balanced.

In previous works, the F-measure has been used as an evaluation criterion for applications related to computer vision. For example, Pérez and Olague [57] proposed the F-measure as a quantitative measure for evaluating evolved SIFT descriptor operators for the classical problem of image matching. In the same year, Gimenez and Evans [58] proposed it as a method of comparison of different segmentation algorithms. Later, Atmosukarto et al. [59] used the F-measure as fitness function for a GP algorithm to learn descriptors of shape variations that are used to rate different facial abnormalities. These examples show that the F-measure has been successfully applied to image-related problems and we decide to test it within a well-known VA problem.

Figure 8 provides a sketch describing the main areas that were applied during the computation of the F-measure. This is applied to Itti’s database that was used as benchmark [14]. Note that a proto-object refers to the selected region issued by the VAP, while manual segmentation denotes the area in the image that has been manually segmented for training and testing. Thus, the selected manual segmentation of an object is considered as an ideal that the evolved models replicating the VA process should seek to attend. For the computation of the fitness function, we consider that the evolved ADS has only one chance for correctly identifying the manually segmented region that serves as ground truth. In order to estimate the VA performance with the F-measure, we compare the evolved proto-object with the manually segmented region. In this way, the relationship between the proto-object with the values of recall and precision must be defined. Thus, the overlapping region between the proto-object and the manually segmented one defines the area where the true positive pixels are located. The remaining areas outside the overlapping region refer to the false-negative and false-positive pixels. These concepts are formally defined next.

Let  $M$  represent the set of pixels of the manually segmented region, and let  $A$  be the set of pixels attended by the ADS. Thus, the set of pixels  $O$  containing the overlapping region is described through the intersection of both sets  $O = A \cap M$ . Once that this relationship is stressed, the next step is to substitute the corresponding variables in Equation (6) to derive the definitions for recall ( $r$ ) and precision ( $p$ ) as follows:

$$r = \frac{|O|}{|M|} \quad \text{and} \quad p = \frac{|O|}{|A|}, \quad (8)$$

where  $|X|$  denotes the number of pixels in  $X \in \{M, A, O\}$ . Hence, the following equation represents the criterion  $Q$  that is applied as the fitness function:

$$Q = \operatorname{argmax} \left\{ F_{\alpha}(P_{k,i}^x, R_{k,i}^x) = \sum_{k=1}^m \sum_{i=1}^n \frac{(1 + \alpha) \cdot (p_i \cdot pr_i)}{(\alpha \cdot p_i) + r_i} \right\},$$

where  $Q: F_{\alpha}(P^s, R^s) \geq F_{\alpha}(P^t, R^t)$

(9)

with  $n$  representing the number of thresholds of the spread function, and  $m$  defines the number of training images. Precision data of an image are denoted with  $P_k^x = (p_1, p_2, \dots, p_n)$  and recall data by  $R_k^x = (r_1, r_2, \dots, r_n)$ ; where  $x$  represents the set of all possible solutions and  $s, t \subseteq x$ . Therefore,  $Q$  represents the ascending ranking with the highest value corresponding to the focus of attention model  $x$  that best attends the particular object over all training images. Therefore, once again, the F-measure could be seen as a suitable approach whose results demonstrated the easiness and usefulness at the moment of specifying the goal of artificial evolution, which are both reflected on the quality achieved during the final results. In particular, the performance of the solutions is related to the focused regions containing the relevant objects that aim a specific purpose.

## Experimental Results

This section presents an analysis and discussion about the experimental results. The experiments were designed considering a state-of-the-art image database composed of three different objects: red can, emergency warning triangle and traffic signals [14]. The databases used for training and testing can be obtained from the USC ilab<sup>1</sup>, as well as the source code of the saliency toolbox<sup>2</sup>. In this paper, each test consists of 30 independent runs of the proposed BP algorithm: the set of images for training and testing, as well

as the parameters that were fixed during the whole run. The experiments were performed on a Dell Precision T7500 Workstation, Intel Xeon 8 Core, CPU E5506 at 2.13Ghz, NVIDIA Quadro FX 3800, running Linux OpenSUSE 11.1 operating system.

### Image Database

In this paper, the experimental tests consist of three databases of natural color images that are used to compare the BP algorithm against four state-of-the-art methodologies [14]. The first database consists of 104 images containing a red can as target. This object in the first database is considered as the simplest target of the three due to its simplicity since it can be characterized by only one color feature. In fact, the results confirm this hypothesis. For the experiments, we follow the proposed protocol described in [14]. The first experiment uses 45 images for training and 59 images for testing. The second database is composed of 64 photographs containing a vehicle emergency triangle, which can be seen as a more complicated target. The database includes the warning triangle under different orientations, scale factors and illumination. The 64 images are divided in two sets of equal size for training and testing. In the case of the third database, the collection is composed of 90 images containing one or more traffic signals. This database contains a broad variety of targets with diverse colors, shapes, sizes and textures. For this reason, it is considered as a more complicated database. Here, the database is divided in two sets of 45 images. Finally, we would like to mention that the targets in the three databases could be partially occluded, opaque, with multiple backgrounds or suffering from reflection of light when viewed from different perspectives.

### Analysis of Results

Next, we present the results of this work accompanied by an interpretation of the obtained data. The results of the BP algorithm show evidence of the relatively easiness at solving all databases, that were used as benchmark, while finding the best solutions in comparison with the four state-of-the-art algorithms [14].

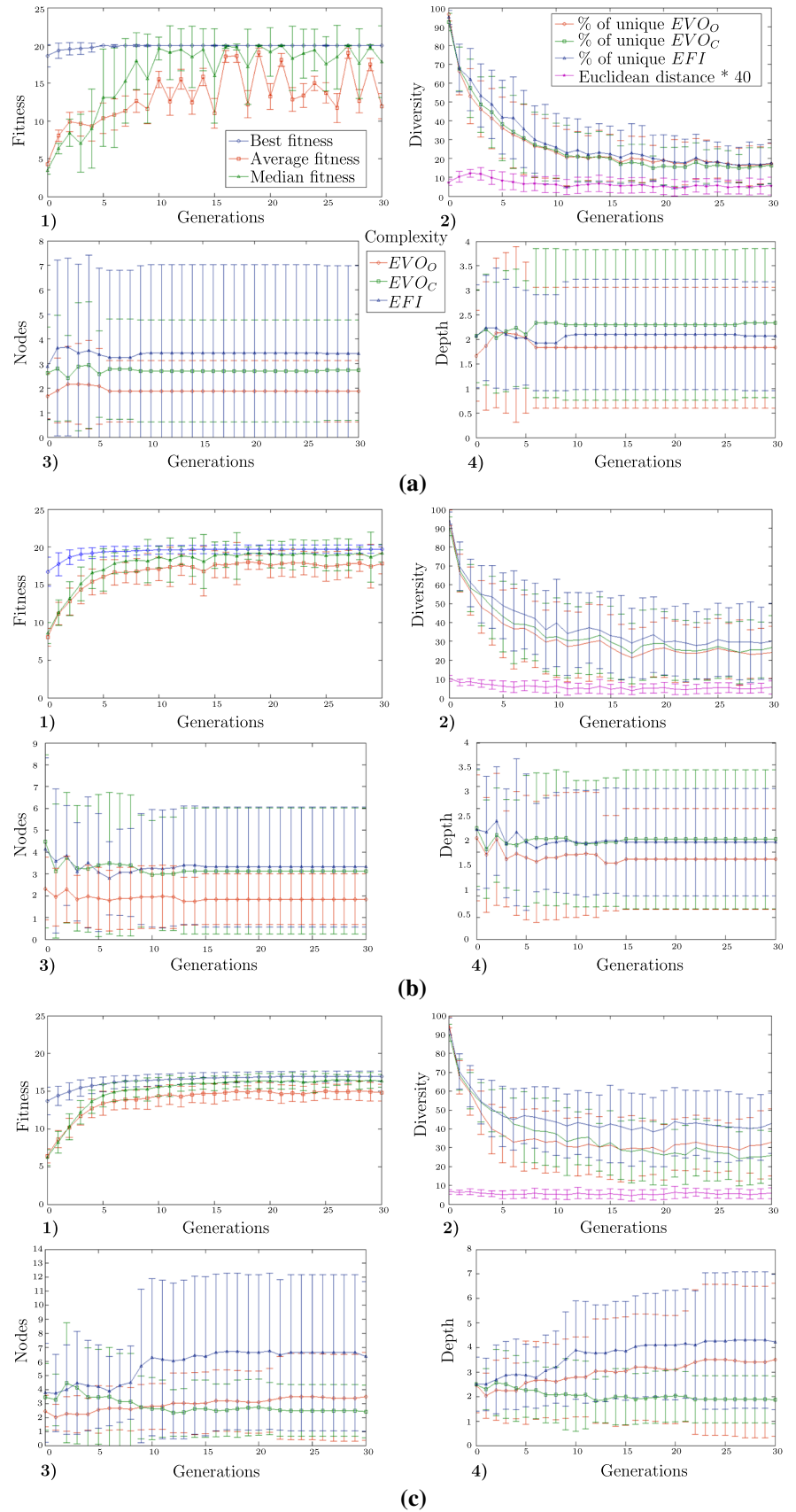
#### BP Analysis

In this section, some typical statistics about GP are provided: fitness, diversity of population, number of nodes and depth of trees, since those aspects are highly informative and commonly used to describe the whole GP process. Figure 9 depicts the statistics considering 30 runs of the BP algorithm executed during 30 generations. All graphs give an account about the performance and complexity of the population during the training stage.

<sup>1</sup> <http://ilab.usc.edu/imgdbs/>

<sup>2</sup> <http://www.saliencytoolbox.net/>

**Fig. 9** Brain programming statistics along the execution of 30 experiments grouped by the target's database: **a** red can, **b** warning triangle and **c** traffic signal. In 1, the fitness chart shows the average, median and best fitness. While 2 plots the diversity of population through the percentage of uniqueness in the EVOs and EFI as well as the Euclidean distance among the population individuals. Finally, 3 and 4 depict the complexity of the structure of the EVOs and EFI based on the amount of nodes and depth, respectively



The first example is known as the red can database problem. Figure 9a.1 presents the graph of best fitness where we can observe the simplicity at solving the red can attention problem. Indeed, we can say that the problem is almost trivial for the ADS, which cannot be considered as a real test from the GP standpoint. Nevertheless, this example serves the purpose of illustrating the functionality of the BP algorithm. Note that from the first generation, the ADS scores a fitness value of 17, which is quite close to the best fitness value of 20. Note that BP converges to the optimal ADS by the fifth generation. Moreover, the average and median population fitness roughly improve through the whole run despite the early convergence around the global optima where multiple local maximum exist from the genotype standpoint. Thus, the optimization provided by the BP algorithm achieves better ADS designs as evolution proceed.

The graph provided in Fig. 9a.2 shows the loss of population diversity along the evolution according to the percentage of uniqueness. In fact, such behavior found on the BP strategy is characteristic of the evolutionary computation methodologies. Thus, controlling this aspect is of major importance since population's diversity promotes a broad exploration of the search space while preventing the population from prematurely converging to a local optima. In this work, we measure the population diversity among the triplet of trees in the BP population; this is computed through the percentage of unique EVOs and EFI. The diversity graph depicts a decreasing trend on the level of uniqueness to 15 % probably due to early convergence. Moreover, diversity was also measured in the fitness space through the Euclidean distance among individuals. In this case, between the fitness values of the ADSs whose measure takes into account the fact that behavior for different programs can be very similar or even identical. As a result, the plot illustrates that while the uniqueness of EVOs decreases significantly the total difference about the ADSs performance remains constant.

One severe problem in GP-based strategies is related to the tree representation where programs transform into larger and slower structures without any improvement in their performance, while producing also a problem on their generalization ability; this is known as the bloat problem. Figure 9a.3 and a.4 plot the scored complexity measured by the number of nodes and depth according to the tree structures. Here, we can observe in the BP experiment that the complexity remains constant during the whole evolutionary run, while keeping the average value around the 2.0 level per tree; far below the dynamic maximum depth value of 7. Hence, the overall structure allows an improvement on the measured fitness function. We consider that the problem of bloat is minimized since our proposed design includes the hierarchical structure and

several well-known principles of cognitive neuroscience. In this way, the methodology does not need to find the whole algorithmic process for solving a given task.

After examining the results achieved by BP on the emergency warning triangle database at Fig. 9b, it is possible to observe that the behavior regarding the fitness plot is similar to that achieved on the red can database. Nevertheless, in this case, the convergence toward the maximum fitness value for all runs is achieved by the 13th generation. Furthermore, the average and median fitness improve smoothly along the run, which is the expected result for BP, see Fig. 9b.1. In the same way as in previous experiments, diversity is calculated through the uniqueness measure of EVOs that decreases rapidly toward a value around the 25 %, where the Euclidean distance remains almost the same during the whole evolutionary run. Moreover, in terms of complexity, Fig. 9b.3, b.4 shows that it is possible to observe a slight decrease in the number of nodes and depth level for all operators along the 30 generations which is an unusual behavior for a GP-based strategy, while reaching an average value between 1.83 and 3.32 for the nodes, and of 1.8 and 2.29 for the depth levels.

The results achieved by the BP strategy on a third database including traffic signals are shown in Fig. 9c. Contrary to previous experiments, and even when the average best fitness maintains an increasing rate the evolution does not converge toward the maximum fitness. In this occasion, the fitness scores an average maximum value of 16.98. The last fitness improvement was obtained during the 27th generation. Moreover, Fig. 9c.2 presents several levels of diversity that are clearly separated among them, with the lowest percentage of uniqueness scoring 28.77 % for EVO<sub>O</sub>, 25.11 % for EVO<sub>C</sub> and 38.33 % for EFI. Meanwhile, the Euclidean distance achieves a small gain. Thus, regarding the level of complexity, see Fig. 9c.3, c.4: the EVO<sub>C</sub> shows a diminution in average toward 2.43 for the number of nodes and 1.86 for the depth level; the EVO<sub>O</sub> shows an increment in average toward 3.5 for the number of nodes and 3.5 for the depth level; and finally, the EFI of 6.63 on the number of nodes and 4.3 for the depth level.

These results show that the behavior of the BP algorithm is characterized by the image database. As a result, the expected behavior of the BP strategy follows the regular trend toward the best solution of optimization approaches, with the exception of complexity that seems to avoid the typical intricate designs of standard GP approaches. In fact, these results point to the enforcement of the parsimonious principle where the most effective solutions are also the simplest ones. Next, the final solutions are analyzed from the standpoint of their effectiveness. We then evaluated the performance of the proposed model in well-known visual attention experiments using a standard testbed. Finally, we



**Table 5** The false detection rate achieved after finding the target (mean  $\pm$  standard deviation)

	Itti and Koch results, see [14]				
	Naive	N(.)	Iterative	Trained	
Red can	2.90 $\pm$ 2.50	1.67 $\pm$ 2.01	1.24 $\pm$ 1.42	0.35 $\pm$ 1.03	
Triangle	2.44 $\pm$ 2.20	1.69 $\pm$ 2.28	1.42 $\pm$ 1.67	0.87 $\pm$ 1.29	
Traffic	1.84 $\pm$ 2.13	0.49 $\pm$ 1.06	0.52 $\pm$ 1.05	0.24 $\pm$ 0.77	
Brain programming results					
	BP <sub>0</sub>	BP <sub>1</sub>	BP <sub>2</sub>	BP <sub>3</sub>	BP <sub>4</sub>
Red can	<b>0.31 <math>\pm</math> 0.88</b>	<b>0.25 <math>\pm</math> 0.73</b>	<b>0.29 <math>\pm</math> 0.77</b>	<b>0.29 <math>\pm</math> 0.77</b>	<b>0.29 <math>\pm</math> 0.77</b>
Triangle	<b>0.16 <math>\pm</math> 0.45</b>	<b>0.28 <math>\pm</math> 0.77</b>	2.37 $\pm$ 5.18 <sup>1</sup>	<b>0.37 <math>\pm</math> 0.66</b>	0.06 $\pm$ 0.24 <sup>2</sup>
Traffic	0.80 $\pm$ 7.76	0.31 $\pm$ 0.87	0.38 $\pm$ 1.19	0.16 $\pm$ 0.52 <sup>1</sup>	0.62 $\pm$ 1.99 <sup>1</sup>
	BP <sub>5</sub>	BP <sub>6</sub>	BP <sub>7</sub>	BP <sub>8</sub>	BP <sub>9</sub>
Red can	<b>0.29 <math>\pm</math> 1.13</b>	<b>0.29 <math>\pm</math> 0.77</b>	<b>0.22 <math>\pm</math> 0.77</b>	<b>0.29 <math>\pm</math> 0.77</b>	<b>0.12 <math>\pm</math> 0.46</b>
Triangle	<b>0.22 <math>\pm</math> 0.49</b>	<b>0.56 <math>\pm</math> 1.6</b>	<b>0.19 <math>\pm</math> 0.4</b>	1.28 $\pm$ 2.25	<b>0.12 <math>\pm</math> 0.42</b>
Traffic	0.14 $\pm$ 0.42 <sup>3</sup>	<b>0.20 <math>\pm</math> 0.46</b>	0.25 $\pm$ 0.49 <sup>1</sup>	0.27 $\pm$ 0.81	0.56 $\pm$ 1.52 <sup>2</sup>
	BP <sub>10</sub>	BP <sub>11</sub>	BP <sub>12</sub>	BP <sub>13</sub>	BP <sub>14</sub>
Red can	<b>0.27 <math>\pm</math> 0.74</b>	0.19 $\pm$ 0.61 <sup>1</sup>	<b>0.27 <math>\pm</math> 0.74</b>	<b>0.24 <math>\pm</math> 0.9</b>	<b>0.08 <math>\pm</math> 0.34</b> *
Triangle	<b>0.13 <math>\pm</math> 0.71</b>	<b>0.09 <math>\pm</math> 0.3</b>	<b>0.44 <math>\pm</math> 0.88</b>	0.19 $\pm$ 0.40 <sup>6</sup>	<b>0.37 <math>\pm</math> 0.75</b>
Traffic	0.55 $\pm$ 1.31	0.55 $\pm$ 1.11	<b>0.18 <math>\pm</math> 0.49</b>	0.60 $\pm$ 1.27	0.06 $\pm$ 0.35 <sup>12</sup>
	BP <sub>15</sub>	BP <sub>16</sub>	BP <sub>17</sub>	BP <sub>18</sub>	BP <sub>19</sub>
Red can	0.17 $\pm$ 0.50 <sup>1</sup>	<b>0.31 <math>\pm</math> 0.81</b>	<b>0.27 <math>\pm</math> 0.96</b>	0.07 $\pm$ 0.26 <sup>3</sup>	<b>0.29 <math>\pm</math> 0.77</b>
Triangle	0.07 $\pm$ 0.26 <sup>3</sup>	<b>0.25 <math>\pm</math> 0.44</b>	<b>0.09 <math>\pm</math> 0.39</b>	0.4 $\pm$ 1.16 <sup>2</sup>	<b>0.13 <math>\pm</math> 0.71</b>
Traffic	0.46 $\pm$ 0.87 <sup>4</sup>	0.20 $\pm$ 0.56 <sup>4</sup>	0.48 $\pm$ 1.02 <sup>3</sup>	0.36 $\pm$ 1.0	0.56 $\pm$ 1.06
	BP <sub>20</sub>	BP <sub>21</sub>	BP <sub>22</sub>	BP <sub>23</sub>	BP <sub>24</sub>
Red can	<b>0.29 <math>\pm</math> 0.77</b>	0.22 $\pm$ 0.70 <sup>1</sup>	0.22 $\pm$ 0.70 <sup>1</sup>	0.27 $\pm$ 0.80 <sup>3</sup>	0.29 $\pm$ 0.75 <sup>1</sup>
Triangle	0.06 $\pm$ 0.25 <sup>1</sup>	<b>0.09 <math>\pm</math> 0.3</b>	0.39 $\pm$ 0.56 <sup>1</sup>	0.27 $\pm$ 0.52 <sup>2</sup>	<b>0.34 <math>\pm</math> 0.83</b>
Traffic	0.62 $\pm$ 0.95 <sup>5</sup>	0.33 $\pm$ 1.24	<b>0.13 <math>\pm</math> 0.4</b> *	0.33 $\pm$ 1.09	0.16 $\pm$ 0.48 <sup>1</sup>
	BP <sub>25</sub>	BP <sub>26</sub>	BP <sub>27</sub>	BP <sub>28</sub>	BP <sub>29</sub>
Red can	<b>0.29 <math>\pm</math> 0.77</b>	0.02 $\pm$ 0.13 <sup>2</sup>	0.21 $\pm$ 0.69 <sup>1</sup>	0.18 $\pm$ 0.54 <sup>2</sup>	<b>0.19 <math>\pm</math> 0.51</b>
Triangle	<b>0.22 <math>\pm</math> 0.49</b>	<b>0.06 <math>\pm</math> 0.35</b> *	<b>0.09 <math>\pm</math> 0.3</b>	<b>0.09 <math>\pm</math> 0.3</b>	<b>0.09 <math>\pm</math> 0.3</b>
Traffic	0.14 $\pm$ 0.35 <sup>1</sup>	0.09 $\pm$ 0.29 <sup>1</sup>	<b>0.16 <math>\pm</math> 0.37</b>	0.43 $\pm$ 0.82 <sup>1</sup>	0.64 $\pm$ 1.28

The numbers in superscripts define in how many images the test fail, while the asterisk identifies the best solutions for the three databases

repeat the above experiments with reduced sets of functions and terminals.

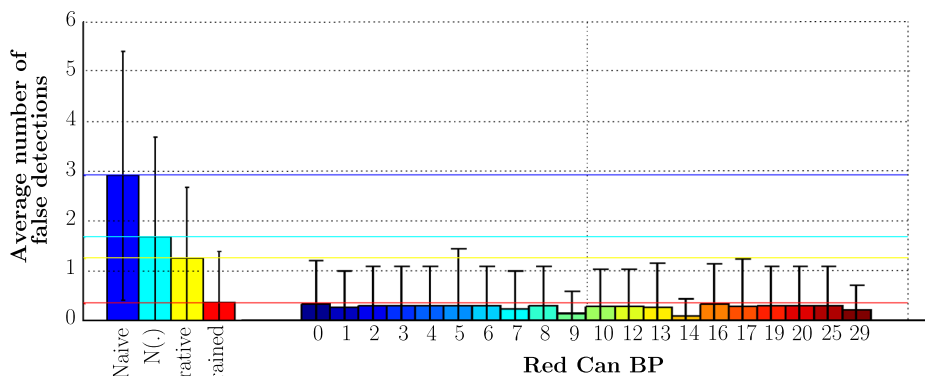
### Effectiveness of Solutions

The best individuals obtained through the BP runs were tested with different sets of images; this process is known as the testing stage. In these experiments, each data were randomly divided into two subsets with a balance number of images, the first subset called the training example was used to learn the EVOs and EFI, and the second subset was

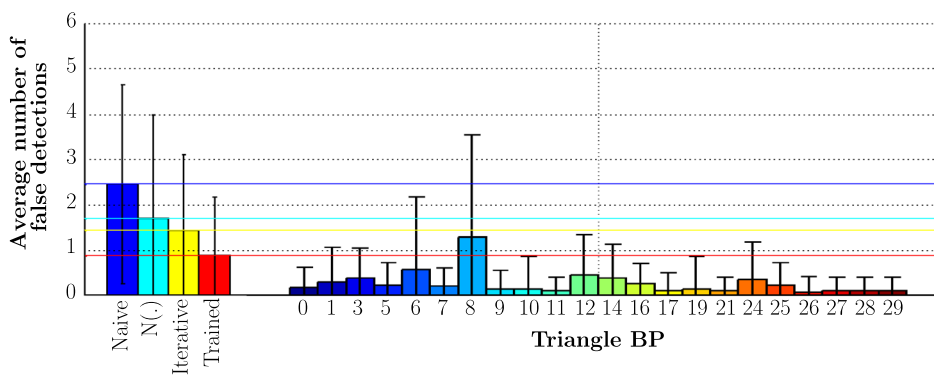
used for testing the proposed model. All images are the originals of the standard test, and the performance was measured as the number of false detections computed after the failure in the identification of the most salient target [14]. All experiments were tested on 30 independent runs, and the average and standard deviation are reported.

In this section, the results of the best individuals for each BP run are listed and compared against the solutions obtained with four previous approaches developed by Itti and Koch [14], in order to study the problem of integrating the optimal features, produced through different visual

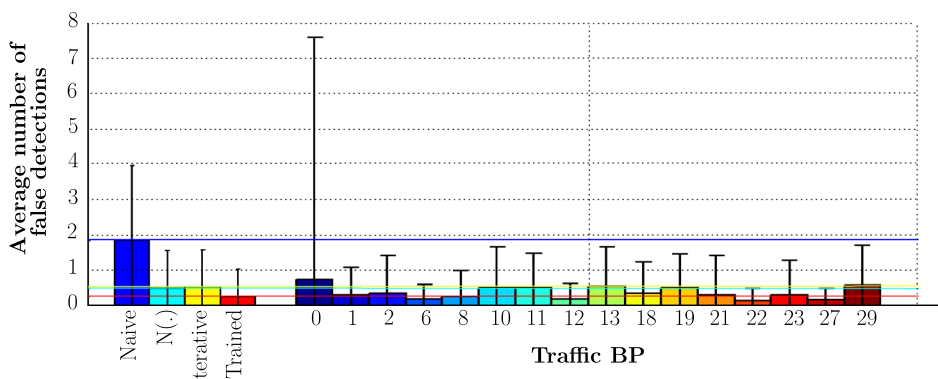
**Fig. 10** The charts provide the false detection rate for the best individuals along 30 executions



(a) Prev. Results *Mean and Std. Deviation for Red Can Images*



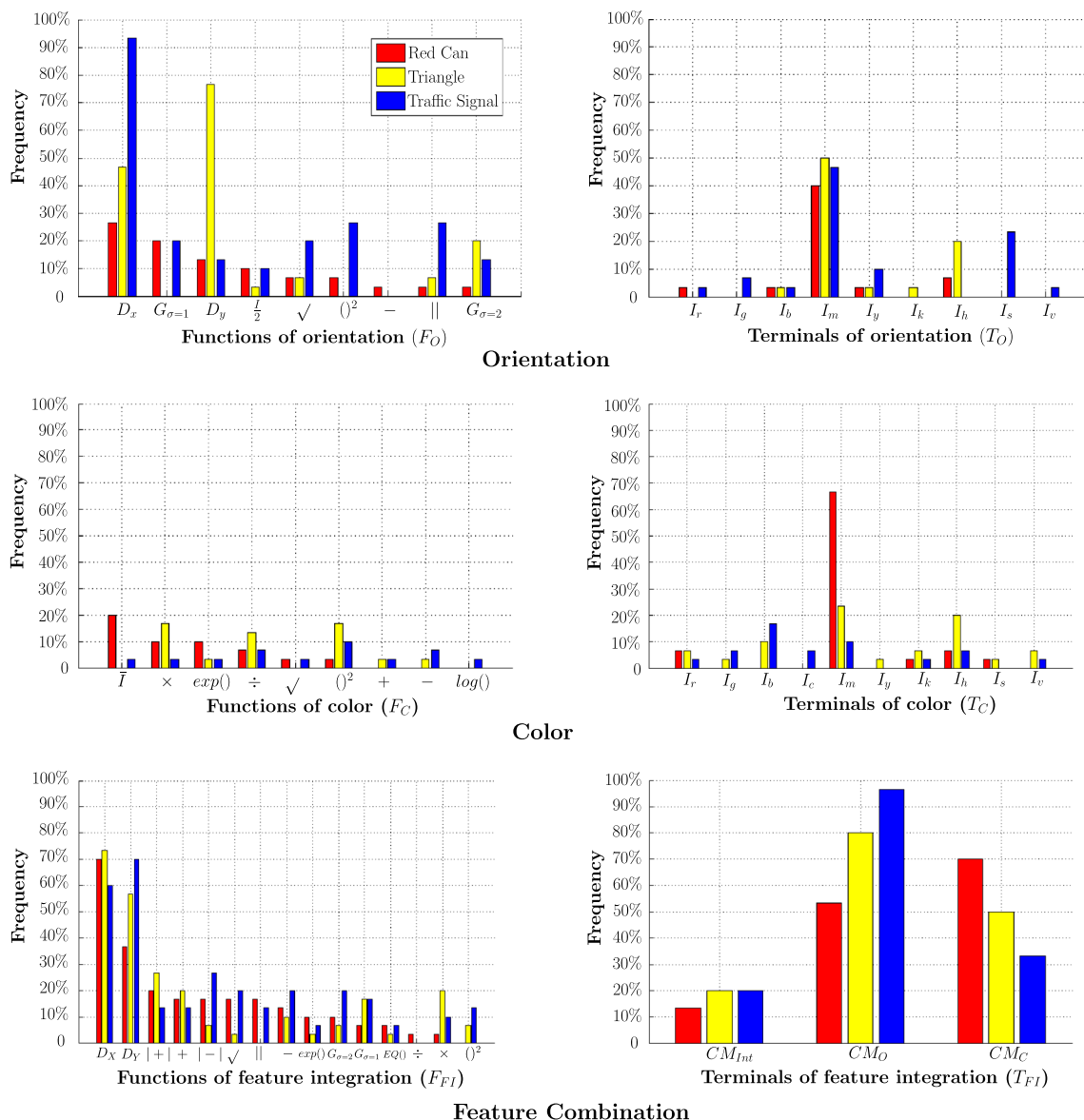
(b) Prev. Results *Mean and Std. Deviation for Triangle Images*



(c) Prev. Results *Mean and Std. Deviation for Traffic Signal Images*

modalities, into a single SM. Such work studied four strategies for feature combination: naive model, model with non-iterative normalization  $N(\cdot)$ , model with iterative normalization and trained model, see Table 5. Moreover, in order to develop a comparison of the results, we adopt the testing sets and measurement criterion about the effectiveness of solutions as reported in [14]. In this way, the false detection rate is computed for the corresponding target as well as the standard deviation. Note that a target is considered as detected if the subset of pixels, being different from the empty set, falls within the attended region also known as proto-object.

The results at the top of Table 5 provide the outcome of the four strategies tested with three different databases. According to these results, the best strategy is the *trained model*. Now, the results achieved by the BP algorithm are also sorted, and the best, that actually outperform the *trained model*, are highlighted in boldface. Moreover, the best solutions for each database are highlighted in italic-boldface and an asterisk. Finally, there were some problems in which the system was unable to find a solution for some specific images during the testing stage. Those results have a superscript that provides the precise number of images where the testing failed.



**Fig. 11** The charts show the statistics about the frequency of use of functions and terminals computed from the best individuals along 30 executions. Note that the terminal graphs,  $T_O$  and  $T_{FI}$ , do not include the derivatives since all of them were counted as functions

According to Table 5, the following data could be inferred. Firstly, regarding the red can target, the BP process finds a better solution in 66.66 % of the runs in comparison with the *trained model*, see Fig. 10a. Moreover, the best ADS achieved its highest score during the 14th test-run, being discovered during the third generation. In fact, this solution scores a false detection rate of 0.08 that corresponds to 5 out of 59 images. Furthermore, the  $ADS_{C14}$  does not record a single false detection during the training stage. Now, after studying the warning triangle results, the BP obtains a better solution in 70 % of the times in comparison with the *trained model*, see Fig. 10b. Here, the best  $ADS_{T26}$  was acquired at the 26th trial and eighth generation. The result achieves a performance with

a rate of 0.06 false detections, in other words, just 2 attempts out of 32 testing images, without a single false detection during the training stage. Finally, we observe the difficulty of detecting the traffic signals since the images are more complicated. Indeed, the best discovered solutions were better than the *trained model* in 13.33 % of the runs, see Fig. 10c. In this way, the best solution was discovered in the 22nd run at the 14th generation. As a result, the BP algorithm found the best  $ADS_{TS22}$  with a false detection rate of 0.13 that is equal to six fails out of the 45 testing images. Contrary to previous results, the  $ADS_{TS22}$  achieves a higher score of 17.33 during the training stage in comparison with the perfect fitness score of 20.

**Table 6** The selection of 6 out of the 30 best ADS<sub>C</sub> classified according to the structural information

Ind	Formula	Fit <sub>train</sub>	Fit <sub>test</sub>	Average
Sample of ADS <sub>C</sub> based on the orientation dimension				
ADS <sub>C7</sub>	EVO <sub>O</sub> : $D_x(G_{\sigma=1}(I_m))$ EVO <sub>O</sub> : n/a EFI: $CM_O$	20	17.6271	18.8135
ADS <sub>C13</sub>	EVO <sub>O</sub> : $G_{\sigma=2}(0.5 * (G_{\sigma=2}(D_x(I_m))))$ EVO <sub>C</sub> : n/a EFI: $ D_x(CM_O) + \sqrt{D_x(CM_O)} $	20	17.8305	18.9152
Sample of ADS <sub>C</sub> based on the color dimension				
ADS <sub>C11</sub>	EVO <sub>O</sub> : n/a EVO <sub>C</sub> : $\exp(I_m)$ EFI: $D_x(CM_C)$	20	16.9492	18.4746
ADS <sub>C22</sub>	EVO <sub>O</sub> : n/a EVO <sub>C</sub> : $I_m$ EFI: $D_x(CM_C)$	20	16.9492	18.4746
Sample of ADS <sub>C</sub> based on the orientation and color dimensions				
ADS <sub>C9</sub>	EVO <sub>O</sub> : $G_{\sigma=1}(I_m)$ EVO <sub>C</sub> : $(\exp(I_r) \times (I_m \div \text{complement}(I_h))) \times \exp(I_h)$ EFI: $((D_y(D_y(CM_C)) + D_x(D_x(CM_O))) - \exp(D_x(D_x(D_x(CM_{Int})))))) - G_{\sigma=1}(G_{\sigma=2}(G_{\sigma=1}(\sqrt{\frac{D_x(CM_O)}{D_x(CM_C)}})))$	20	18.3051	19.1525
ADS <sub>C14</sub>	EVO <sub>O</sub> : $D_x(I_m)$ EVO <sub>C</sub> : $I_m$ EFI: $ D_x(CM_C) - D_y(CM_O) $	20	18.6441	19.3220

**Table 7** The selection of 7 out of the 30 best ADS<sub>T</sub> classified according to the structural information

Ind	Formula	Fit <sub>train</sub>	Fit <sub>test</sub>	Average
Sample of ADS <sub>T</sub> based on the orientation dimension				
ADS <sub>T10</sub>	EVO <sub>O</sub> : $D_x(D_y(I_m))$	20	19.3750	19.6875
ADS <sub>T19</sub>	EVO <sub>C</sub> : n/a EFI: $D_y(CM_O)$			
ADS <sub>T26</sub>	EVO <sub>O</sub> : $D_x(D_y(I_m))$ EVO <sub>C</sub> : n/a EFI: $D_x(CM_O) + D_y(CM_O)$	20	19.3750	19.6875
Sample of ADS <sub>T</sub> based on the color dimension				
AVS <sub>T3</sub>	EVO <sub>O</sub> : n/a EVO <sub>C</sub> : $((I_m \times I_m) \times I_m)^2 \times I_m^2$ EFI: $CM_C$	20	14.3750	17.1875
AVS <sub>T7</sub>	EVO <sub>O</sub> : n/a EVO <sub>C</sub> : $\exp(\exp(I_b \times I_m))$ EFI: $ D_y(CM_{Int}) + CM_C  + D_y(D_y(CM_C))$	18.7500	16.2500	17.5000
Sample of ADS <sub>T</sub> based on both orientation and color dimensions				
AVS <sub>T0</sub>	EVO <sub>O</sub> : $D_y(G_{\sigma=2}(D_x(I_m)))$ EVO <sub>C</sub> : $I_k$ EFI: $G_{\sigma=2}(CM_O + D_x(D_x(CM_C)))$	19.3750	18.8125	19.0937
AVS <sub>T5</sub>	EVO <sub>O</sub> : $D_x(D_y(I_m))$ EVO <sub>C</sub> : $I_r \div I_h$ EFI: $ D_x(CM_O) + (D_y(CM_{Int}) + D_x(CM_C)) $	20	16.2500	18.1250

**Table 8** The selection of 4 out of the 30 best ADS<sub>TS</sub> classified according to the structural information

Ind	Formula	Fit <sub>train</sub>	Fit <sub>test</sub>	Average
Sample of ADS <sub>TS</sub> based on the orientation dimension				
AVS <sub>TS12</sub>	EVO <sub>O</sub> : $G_{\sigma=1}((G_{\sigma=1}(D_x(I_m)))^2)$ EVO <sub>C</sub> : n/a EFI: $G_{\sigma=1}(D_x(D_x(CM_O)) - D_x(D_x(CM_O))) - D_y(D_y(CM_O))$	17.3333	17.3333	17.3333
AVS <sub>TS22</sub>	EVO <sub>O</sub> : $D_x(G_{\sigma=2}(\sqrt{D_x(I_m)}))$ EVO <sub>C</sub> : n/a EFI: $  CM_O - \sqrt{CM_O}  - \sqrt{CM_O} $	17.3333	16.8889	17.1111
Sample of ADS <sub>TS</sub> based on both orientation and color dimensions				
AVS <sub>TS6</sub>	EVO <sub>O</sub> : $0.5*(D_x(G_{\sigma=2}(G_{\sigma=2}(D_x(G_{\sigma=2}(D_{xy}(I_m)))))))$ EVO <sub>C</sub> : $I_b$ EFI: $G_{\sigma=1}((CM_O - D_y(CM_C)) - D_y(CM_C))$	18.2222	16.4444	17.3333
AVS <sub>TS23</sub>	EVO <sub>O</sub> : $  D_x(I_s)  $ EVO <sub>C</sub> : $I_b$ EFI: $ (D_x(D_x(CM_O)) - G_{\sigma=1}(D_x(D_x(CM_C))))^2 - \sqrt{D_y(CM_O)}  +  (D_y(CM_O))^2 + D_y(CM_C) $	17.7778	16.7778	17.2778

As a conclusion, BP was able to find, throughout 30 runs, better solutions than those obtained in previous research. Next, we describe the structural analysis of the solutions.

*Structural Analysis of Solutions*

In this section, the solutions are analyzed from the structural standpoint by describing the distribution of functions and terminals through the histograms of frequency of use, see Fig. 11. Our goal was to identify the best components of the trees for three visual attention tasks. An important question is: which functions and terminals have more influence on the VA task? We study the final best individuals along the 30 runs. Thus, the functions and terminals are counted only if their respective EVO is combined by the EFI. Hence, the best solutions are listed in Tables 6, 7 and 8, according to the visual dimensions that were selected when finding a solution to the problem.

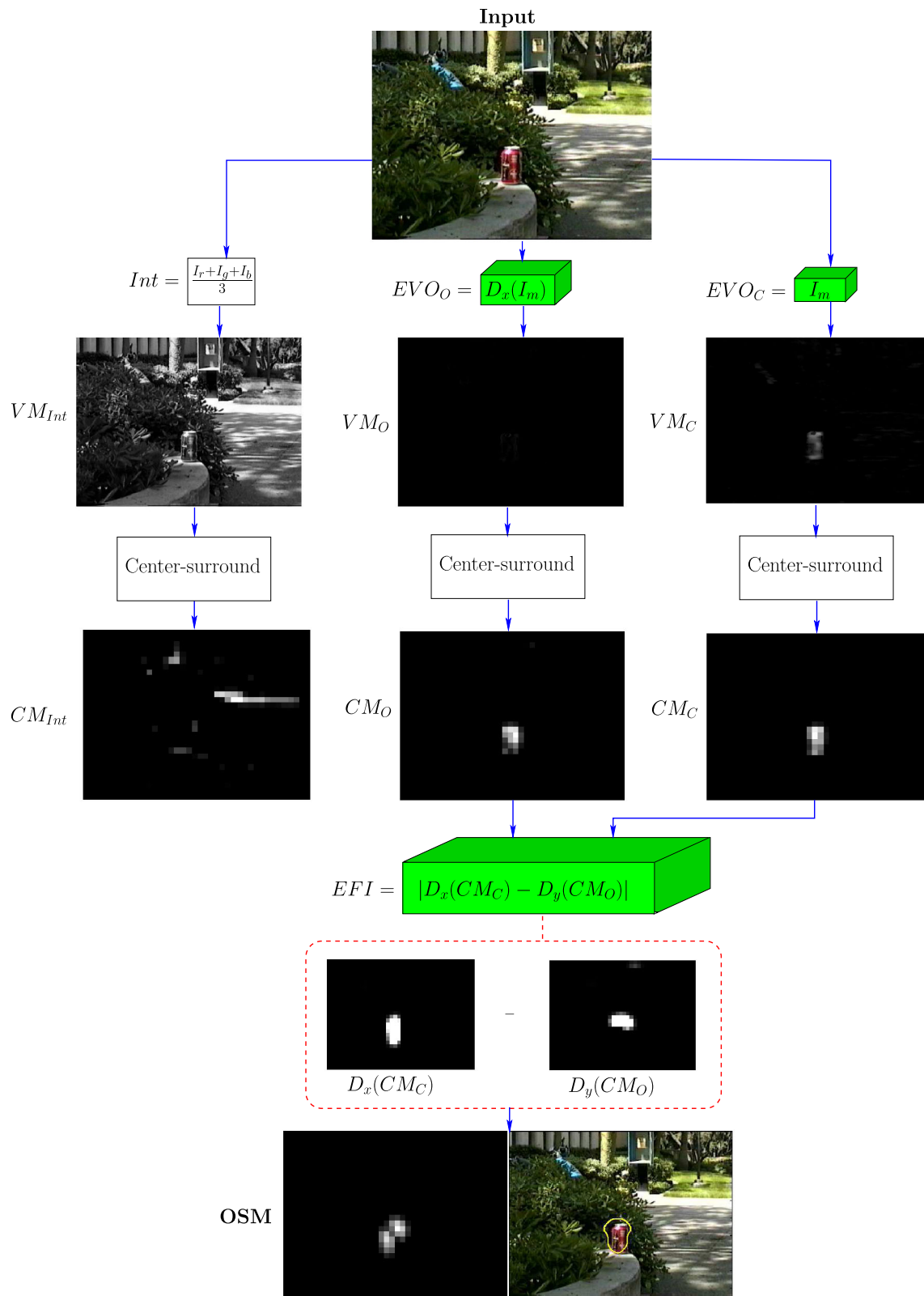
In the experimental results about the red can, highlighted in red color, the best ADS<sub>C</sub> for VA utilize the color feature 70 % of the time; see Fig. 11. This result was expected since color is considered as the main characteristic of the red can; followed by the orientation at the rate of 52 %. Furthermore, according to the color feature, the most useful terminal was  $I_m$  at a rate of 68 % of the time, while the complement() function was applied in only 20 % of the time. On the other hand, in terms of orientation, the  $I_m$  was again the most useful terminal appearing in 40 % of the ADSs, with similar results for the functions  $D_x$ ,  $D_y$  and  $G_{\sigma=1}$ .

Final results about the warning triangle solutions ADS<sub>T</sub>, highlighted in yellow, converged to the functions and

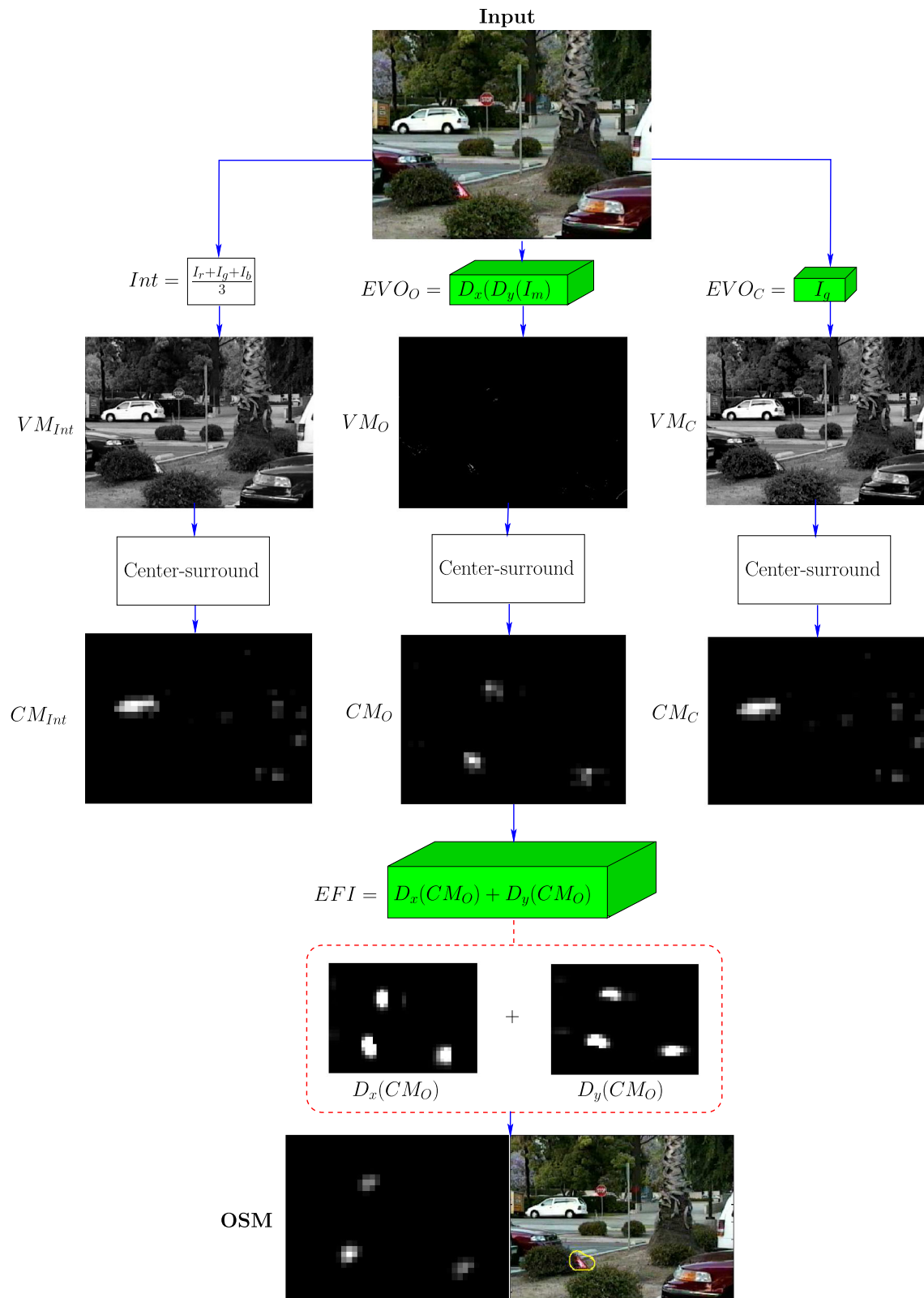
terminals along the orientation and color dimensions. Note that 80 % of the ADS<sub>T</sub> utilize orientation while 50 % color information combined with the functions  $D_x$ ,  $D_y$ ,  $| + |$ ,  $+$  and  $\times$ . As a result, the most prominent dimension was orientation. Indeed, the EVO<sub>O</sub> was basically a composition of  $D_x$ ,  $D_y$ ,  $G_{\sigma=2}$ ,  $\times$ ,  $\div$  and  $()^2$  functions; together with the terminals  $I_m$  and  $I_h$ , see Fig. 11. We can observe that the results about the traffic signals confirm that 97 % of the ADS<sub>TS</sub> are based on orientation and that the EVO<sub>O</sub> apply the terminals  $I_m$  and  $I_s$ , as well as the functions  $D_x$ ,  $()^2$ ,  $| |$ ,  $\sqrt{\quad}$  and  $G_{\sigma=1}$ . The blue bars in Fig. 11 illustrate these results. Hence, we can say that the proposed optimization process is able to find good visual operators that extract the main characteristics according to the task at hand.

In this work, the best ADSs can be organized in terms of the most useful characteristics that were applied during the EFI stage. Table 6 presents three pairs of ADS<sub>C</sub> that exhibit the highest performance in the “can” experiments. Note that all operators base their process on the orientation and color dimensions. Note also that intensity was never selected among the optimal solutions for the red can problem. Thus, the first two individuals use the *magenta* band since it matches better the color of the red can, while  $D_x$  was applied probably due to the vertical position of the can across all images in the testbed. On the other hand, the ADS<sub>C11</sub> and ADS<sub>C22</sub> provide local minima that are very similar in structure, since they are both based on  $I_m$  and  $D_x$  terminals. Note that the best ADS<sub>C14</sub> applies a subtraction between  $D_x$  and  $D_y$  computed with their corresponding CMs.

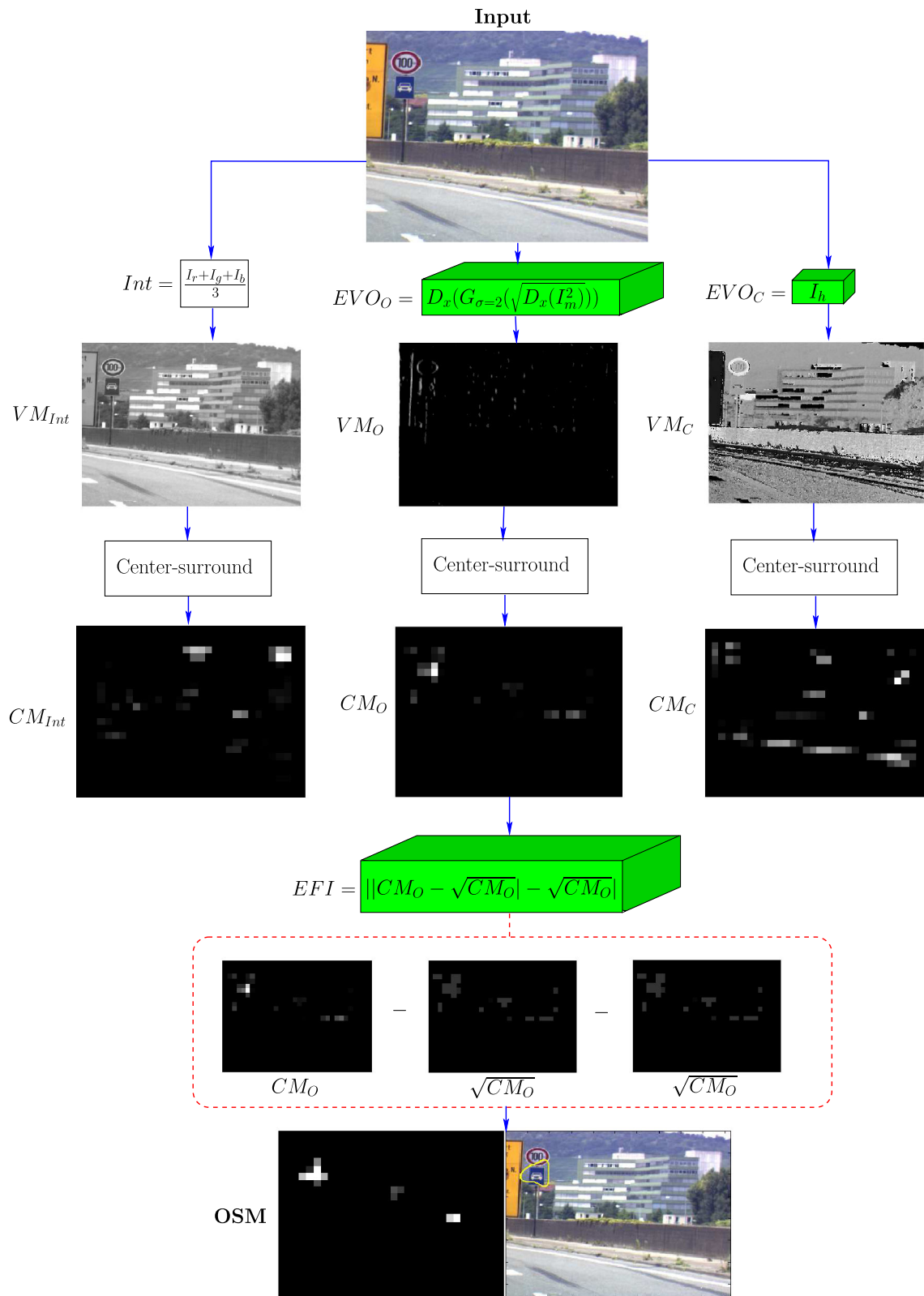
In this way, we can observe in Table 7 the selection of seven ADS<sub>T</sub> that were able to focus reliably on the



**Fig. 12** Some details about the inner workings of the best individual ADS<sub>C14</sub>, which rightly focus the coke in the image database



**Fig. 13** Some details about the inner workings of the best individual  $ADS_{T26}$ , which rightly focus the emergency warning triangle in the image database



**Fig. 14** Some details about the inner workings of the best individual ADS<sub>TS22</sub>, which rightly focus the traffic signals in the image database



**Table 9** Sets of selected and discarded functions and terminals for the red can database, where  $I_{x_1} \in \{I_r, I_b, I_m, I_y, I_h\}$  and  $I_{x_2} \in \{I_g, I_c, I_k, I_s, I_v\}$ EVO<sub>O</sub>

$$F_O = \{ -, \sqrt{\quad}, ()^2, G_{\sigma=1}, G_{\sigma=2}, \parallel, 0.5*(), D_x, D_y \}$$

$$F_O \text{ discarded} = \{ +, \times, \div, | + |, | - |, \log_2() \}$$

$$T_O = \{ I_{x_1}, G_{\sigma=1}(I_{x_1}), G_{\sigma=2}(I_{x_1}), D_x(I_{x_1}), D_y(I_{x_1}), D_{xx}(I_{x_1}), D_{yy}(I_{x_1}), D_{xy}(I_{x_1}) \}$$

$$T_O \text{ discarded} = \{ I_{x_2}, G_{\sigma=1}(I_{x_2}), G_{\sigma=2}(I_{x_2}), D_x(I_{x_2}), D_y(I_{x_2}), D_{xx}(I_{x_2}), D_{yy}(I_{x_2}), D_{xy}(I_{x_2}) \}$$

EVO<sub>C</sub>

$$F_C = \{ \times, \div, \sqrt{\quad}, \exp(), \text{complement}() \}$$

$$F_C \text{ discarded} = \{ +, -, | + |, | - |, \log_2() \}$$

$$T_C = \{ I_r, I_m, I_k, I_h, I_s \}$$

$$T_C \text{ discarded} = \{ I_g, I_b, I_c, I_y, I_v \}$$

EFI

$$F_{FI} = \{ +, -, \times, \div, | + |, | - |, \sqrt{\quad}, \exp(), G_{\sigma=1}, G_{\sigma=2}, \parallel, D_x(), D_y(), EQ() \}$$

$$F_{FI} \text{ discarded} = \{ ()^2 \}$$

$$T_{FI} = \{ \text{CM}_{\text{Int}}, \text{CM}_O, \text{CM}_C, D_u(\text{CM}_{\text{Int}}), D_u(\text{CM}_O), D_u(\text{CM}_C) \}$$

$$T_{FI} \text{ discarded} = \emptyset$$

**Table 10** Sets of selected and discarded functions and terminals for the warning triangle database, where  $I_{x_1} \in \{I_b, I_m, I_y, I_k, I_h\}$  and  $I_{x_2} \in \{I_r, I_g, I_c, I_s, I_v\}$ EVO<sub>O</sub>

$$F_O = \{ \sqrt{\quad}, G_{\sigma=2}, \parallel, 0.5*(), D_x, D_y \}$$

$$F_O \text{ discarded} = \{ -, +, ()^2, G_{\sigma=1}, \times, \div, | + |, | - |, \log_2() \}$$

$$T_O = \{ I_{x_1}, G_{\sigma=1}(I_{x_1}), G_{\sigma=2}(I_{x_1}), D_x(I_{x_1}), D_y(I_{x_1}), D_{xx}(I_{x_1}), D_{yy}(I_{x_1}), D_{xy}(I_{x_1}) \}$$

$$T_O \text{ discarded} = \{ I_{x_2}, G_{\sigma=1}(I_{x_2}), G_{\sigma=2}(I_{x_2}), D_x(I_{x_2}), D_y(I_{x_2}), D_{xx}(I_{x_2}), D_{yy}(I_{x_2}), D_{xy}(I_{x_2}) \}$$

EVO<sub>C</sub>

$$F_C = \{ \times, \div, ()^2, +, -, \exp() \}$$

$$F_C \text{ discarded} = \{ \sqrt{\quad}, | + |, | - |, \log_2(), \text{complement}() \}$$

$$T_C = \{ I_r, I_g, I_b, I_m, I_y, I_k, I_h, I_s, I_v \}$$

$$T_C \text{ discarded} = \{ I_c \}$$

EFI

$$F_{FI} = \{ +, -, \times, | + |, | - |, \sqrt{\quad}, \exp(), G_{\sigma=1}, G_{\sigma=2}, D_x(), D_y(), ()^2, EQ() \}$$

$$F_{FI} \text{ discarded} = \{ ||, \div \}$$

$$T_{FI} = \{ \text{CM}_{\text{Int}}, \text{CM}_O, \text{CM}_C, D_u(\text{CM}_{\text{Int}}), D_u(\text{CM}_O), D_u(\text{CM}_C) \}$$

$$T_{FI} \text{ discarded} = \emptyset$$

emergency warning triangle. Nevertheless, we have only six ADSs since the genotype or structural composition of ADS<sub>T10</sub> and ADS<sub>T19</sub> is identical. Note that both avoid the color dimension. Thus, we can state that the algorithm discovered a local minimum whose overall performance scores identical results. Furthermore, the best solution ADS<sub>T26</sub> is also based on orientation, and its similarity to the previous ones is striking. Note that ADS<sub>T26</sub> has the same fitness of ADS<sub>T10</sub> and ADS<sub>T19</sub>, but it is slightly

different in terms of the structural composition since at the moment of performing feature-integration the individual applies a new derivative to the conspicuous map. These results consider that the fitness function is evaluated in one attempt of the focus of attention, see “[Fitness function or objective function](#)” section. Nevertheless, we can say that ADS<sub>T26</sub> is better since it has a lower false detection according to Table 5. Hence, the best individual ADS<sub>T26</sub> is a combination of  $D_x$  and  $D_y$  over the *magenta* band; while

**Table 11** Sets of selected and discarded functions and terminals for the traffic signals database, where  $I_{x_1} \in \{I_r, I_g, I_b, I_m, I_y, I_s, I_v\}$  and  $I_{x_2} \in \{I_c, I_k, I_h\}$

EVO <sub>O</sub>
$F_O = \{ \sqrt{\cdot}, G_{\sigma=1}, G_{\sigma=2}, \parallel, 0.5^*(\cdot), (\cdot)^2, D_x, D_y \}$ $F_O \text{ discarded} = \{ -, +, \times, \div,   +  ,   -  , \log_2(\cdot) \}$ $T_O = \{ I_{x_1}, G_{\sigma=1}(I_{x_1}), G_{\sigma=2}(I_{x_1}), D_x(I_{x_1}), D_y(I_{x_1}), D_{xx}(I_{x_1}), D_{yy}(I_{x_1}), D_{xy}(I_{x_1}) \}$ $T_O \text{ discarded} = \{ I_{x_2}, G_{\sigma=1}(I_{x_2}), G_{\sigma=2}(I_{x_2}), D_x(I_{x_2}), D_y(I_{x_2}), D_{xx}(I_{x_2}), D_{yy}(I_{x_2}), D_{xy}(I_{x_2}) \}$
EVO <sub>C</sub>
$F_C = \{ \times, \div, \sqrt{\cdot}, (\cdot)^2, +, -, \exp(\cdot), \log_2(\cdot), \text{complement}(\cdot) \}$ $F_C \text{ discarded} = \{   +  ,   -   \}$ $T_C = \{ I_r, I_g, I_b, I_c, I_m, I_k, I_h, I_v \}$ $T_C \text{ discarded} = \{ I_y, I_s \}$
EFI
$F_{FI} = \{ +, -, \times,   +  ,   -  , \sqrt{\cdot}, \exp(\cdot), G_{\sigma=1}, G_{\sigma=2}, D_x(\cdot), D_y(\cdot), (\cdot)^2, \parallel, EQ(\cdot) \}$ $F_{FI} \text{ discarded} = \{ \div \}$ $T_{FI} = \{ CM_{int}, CM_O, CM_C, D_u(CM_{int}), D_u(CM_O), D_u(CM_C) \}$ $T_{FI} \text{ discarded} = \emptyset$

the ADS<sub>T3</sub> and ADS<sub>T7</sub> also apply a nonlinear combination of functions working with the *magenta* color. The application of  $I_m$ , and the  $D_x$  and  $D_y$  functions is also used on individuals ADS<sub>T0</sub> and ADS<sub>T5</sub> in combination with *red* and *hue* bands. Finally, the overall best individual in this test was the ADS<sub>T26</sub>.

In the case of the traffic signals, the results are more complex than the two previous ones. Note that the database for the traffic signals includes a wider variety of images; hence, the final solutions are more complex. In the final results, we can appreciate that some particular functions and terminals have been picked up more frequently. Also, among the 30 best ADS<sub>TS</sub> no one is exclusively based on the color feature. Therefore, we select only four ADSs that are written on Table 8. The first two, based on orientation, the ADS<sub>TS12</sub> and ADS<sub>TS22</sub>, are basically created through a combination of  $I_m$ ,  $D_x$  and  $D_y$  terminals, as well as the *Gaussian* function. Note that ADS<sub>TS22</sub> is the best discovered solution as well as the simplest one. Moreover, the next two ADS<sub>TS</sub> have a higher complexity and both use color and orientation data, despite the fact that color information consists only of the *blue* band. These ADSs are few examples of solutions that the BP process is able to find. Note that the best solutions do not necessarily include all VOs because the EFI could exclude them. In the following section, we provide as examples three ADSs to visualize the information flow as a way of illustrating the inner workings of the best attention process for each image database, see Figs. 12, 13 and 14.

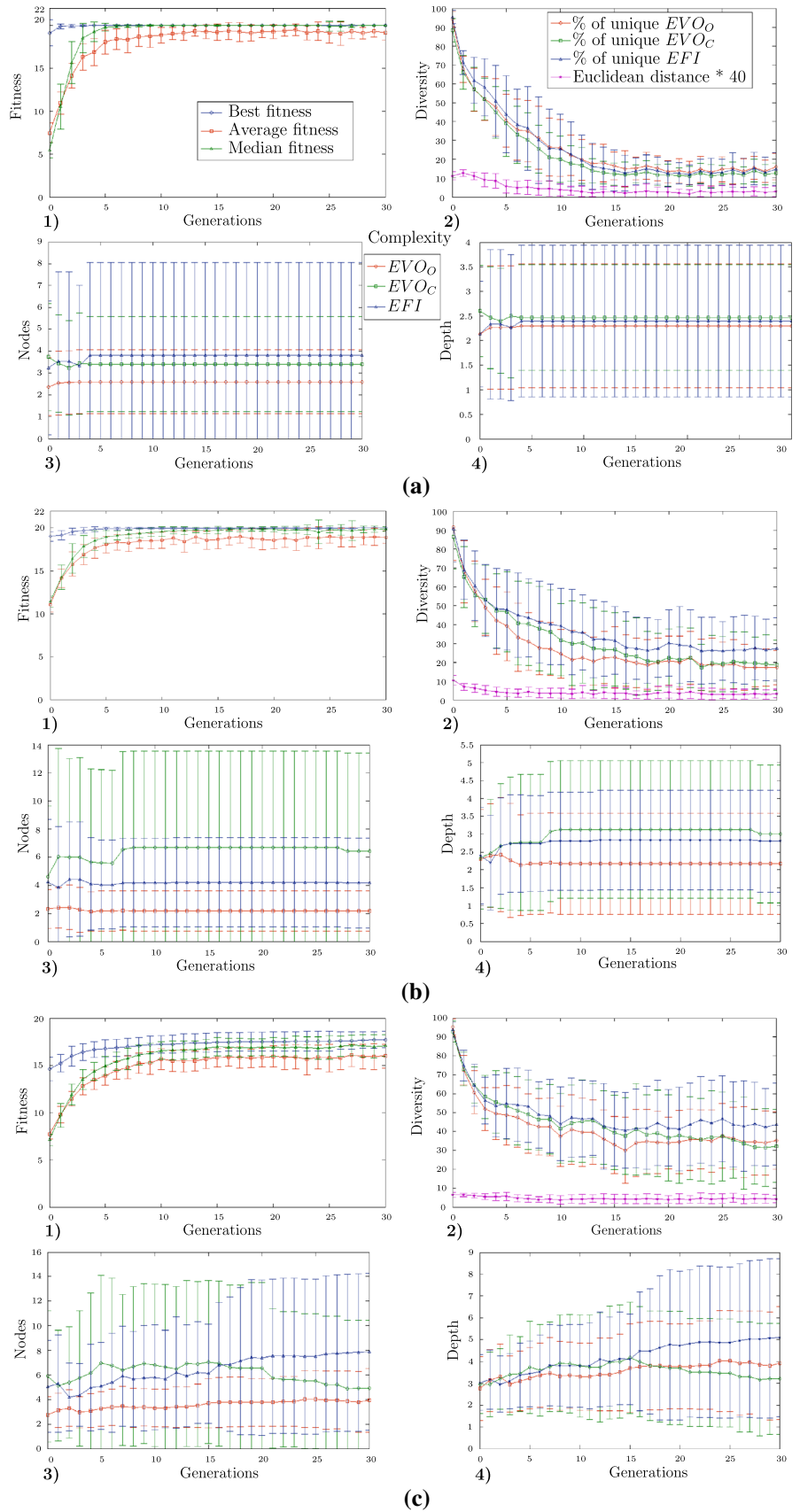
*Final Designs for the Three Databases*

This section contains three examples discovered with our optimization methodology that illustrate the functionality of the best ADS designs using a diagram where some key aspects about the structure are illustrated.

Figure 12 depicts the ADS<sub>C14</sub> that is used as a first example; where it is possible to observe the following noteworthy results. The *magenta* color band is used in EVO<sub>O</sub> and EVO<sub>C</sub> to filter a big amount of unnecessary information known in the literature as distraction elements or the image regions that can mislead the location of the proto-object. This process allows the system to highlight more accurately the red can. Moreover, the EVO<sub>O</sub> applies a  $D_x$  as a way of suppressing the horizontal data while keeping the red can information. Next, the center-surround function used within the algorithm task eliminates the foreground information along the regions with high contrast. As a result, we obtain two conspicuity maps, CM<sub>O</sub> and CM<sub>C</sub>, that are able to highlight and discriminate the red can region as the most prominent of the image. Afterward, the feature-integration stage consists of the absolute value computed from the difference across the main directions  $D_x$  and  $D_y$ . Finally, the result leads us to the OSM that indicates the desired location achieved by the focus of attention system.

The next example corresponds to the warning triangle problem. Figure 13 shows the information flow process for the ADS<sub>T26</sub>. Again, the solution works through the

**Fig. 15** Brain programming statistics along the execution of 30 experiments with the most utilized functions and terminals, grouped by the target's database: **a** red can, **b** warning triangle and **c** traffic signal. **1** shows the graph of best, average and median fitness. **2** plots the diversity of population through the percentage of uniqueness in the EVOs and EFI; as well as the Euclidean distance among the population individuals. Finally, **3** and **4** depict the complexity of the structure of the EVOs and EFI based on the amount of nodes and depth, respectively



**Table 12** The best triplet of trees achieved with the reduced set of functions and terminals for the traffic signals database

Ind	Formula
ADS <sub>TS18</sub>	Ori: $D_x(I_s)$ Col: $\sqrt{(I_g \times I_v) + \exp(\sqrt{\exp(I_m)})} + \sqrt{(I_g \times I_v) + (\sqrt{I_m} \times \sqrt{\exp(I_m)} + \sqrt{(I_g \times I_v) + \exp(\sqrt{\exp(I_m)}))})}$ FI: $G_{\sigma=2}( G_{\sigma=2}(G_{\sigma=2}( G_{\sigma=2}( D_y(CM_O)  - D_{xx}(CM_C)  - D_x( CM_O ))  - D_x(G_{\sigma=1}( CM_O ))) $

*magenta* color band. Note that this is mainly due to the color of the emergency warning triangle. In this way, distractions like the white car in the background are eliminated. On the other hand, since there are multiple red color objects, the partial results obtained after the visual maps show several regions that could become more prominent after the center-surround function. Nevertheless, besides the color information, the EVO<sub>O</sub> extracts the orientation information through the application of two derivatives  $D_y$  and  $D_x$  accordingly. In this way, the information about the color and orientation is applied to produce results that highlight the main characteristics of the triangle. As a result, the EFI operator combines the orientation along the main directions of the image resulting in an OSM where the most salient region corresponds precisely to the exact place of the emergency triangle. Indeed, ADS<sub>C14</sub> and ADS<sub>T26</sub> have a very similar way of working.

Finally, the last example illustrates the best individual discovered for the traffic signal problem. Figure 14 shows the ADS<sub>TS22</sub> working on a typical example. In this regard, the ADS<sub>TS22</sub> utilizes exclusively the orientation operator EVO<sub>O</sub>. Once again, the *magenta* was the color band that the evolutionary algorithm selected for the solution of the VA regarding the traffic signals. Moreover, EVO<sub>O</sub> applies two  $D_x$  to remove horizontal features and preserve the vertical position of the traffic signals. After the application of the center-surround function, resulting in the CM<sub>O</sub>, we can distinguish several conspicuous regions. However, the EFI operator subtracts twice the CM<sub>O</sub> with the  $\sqrt{CM_O}$ , which can be interpreted as a filter that helps to decrease the noise level of the CM<sub>O</sub>. Hence, the OSM presents a lower number of salient regions, and more importantly, it focuses on the traffic signal region.

#### Final Experiments with Reduced Sets of Functions and Terminals

In this section, we decide to proceed further with the analysis of the results achieved during the previous experiments related to the structural composition and the effectiveness of solutions. As we already mentioned, the best individuals obtained through the proposed strategy do not use all the functions and terminals that were proposed empirically. Therefore, we tested the BP algorithm with reduced sets of

functions and terminals based on the previous results. Tables 9, 10 and 11 provide the sets of applied and discarded functions and terminals for each experiment. Note that the discarded sets represent the elements that were never used by any of the 30 best solutions of the corresponding test. However, there are patterns that emerge from these results. For example, it is noteworthy to observe that the three experiments arose different results; with the exception on the selection of the EFI stage, since all terminals were used and the functions at least once. Note also that independently of the three tests the following sets of functions  $\{+, \times, \div, | + |, | - |, \log_2()\}$  and terminals  $\{I_c\}$  were never used by the EVO<sub>O</sub>. Moreover, the set of functions  $\{| + |, | - | \}$  was never used by the EVO<sub>C</sub>.

Figure 15 shows the results of applying a new series of experiments to test the learning process of BP with the described reduced sets. Note that for the three experiments, all the statistics were significantly improved, specially those related to the fitness function. Thus, the average and median fitness present a lower standard deviation; while for the case of the traffic signals, the results show an improvement on the best fitness.

Finally, from the structural standpoint, we can say that in this last series of experiments, the evolutionary process converges toward the local minima reported in “[Structural analysis of solutions](#)” section. For example, in the red can experiment, a similar solution to ADS<sub>C11</sub> appears once and two similar solutions to ADS<sub>C22</sub> were found along with one copy of it. This phenomena also occurs for the warning triangle, where the exact same individual ADS<sub>T10</sub> was rediscovered in three occasions. Contrary to these results, in the traffic signals test, neither identical solutions nor similar individuals were found. Table 12 shows the structure of the best individual obtained during the 18th run at the 27th generation of this last experiment. This solution scores a false detection rate of 0.11 that correspond to five fails out of the 45 testing images. Moreover, this individual achieves the highest score of 19.11 during training.

#### Conclusion

This study presented a novel GP-based algorithm that we called brain programming and a new VA model based on

the artificial dorsal stream, which through the BP algorithm is able to design a suitable program for the problem of attending a specific target object. The BP algorithm is a new optimization/search approach that looks for the best way of identifying the visual features and the right way of combining them to highlight the necessary visual information for the task at hand. We conducted a visual search experiment using three different image databases in order to test our approach. The analysis of the results shows that BP is an efficient tool to optimize ADSs whose results outperform previous man-made VA models on a standard testbed.

Based on our experimental results along 30 evolutionary runs, we observe that BP is able to provide suitable and coherent designs, while maintaining the simplicity in the solutions according to the task at hand. In this way, regarding the EVOs that are embodied in the ADS structure, we found that they fulfilled the parsimony principle. In other words, the simplest solution is always the one offered by the system since the system is able to find the local minima.

In addition, we found interesting properties in the EVOs. For instance, we observe that some EVOs are highly specialized toward the extraction and combination of the main characteristics of the object of interest. Thus, for the red can attention problem, the EFIs focused on the application of orientation and color dimensions, being color the most prominent feature for the red can problem. Meanwhile, the EVO<sub>O</sub> utilizes the *magenta* color band and the  $D_x$  function. This is understandable since *magenta* is closer in the color space in comparison with the “red” color of the can, while  $D_x$  enhance the appearance of its vertical position. On the other hand, the principal dimensions of the emergency warning triangle are also color and orientation. Here, most of the EFIs selected the orientation dimension: the EVO<sub>O</sub> utilizes  $D_x$  and  $D_y$  to highlight both vertical and horizontal edges while *magenta* band is applied to highlight the color of the can. Finally, regarding the traffic signals, the vast majority of EFIs are based once again on the orientation dimension EVO<sub>O</sub>, which is mainly conformed of the  $I_m$  terminal and the  $D_x$  function. Moreover, the traffic signals have several colors and shapes, and for this reason, it is more difficult for BP to select suitable features. In summary, our results suggest that through the application of the BP strategy, it is possible to automatically design VA systems for focusing a specific target.

In future work, this research offers several new avenues. Firstly, we plan to test the BP strategy for the ADS problem with a bigger database of natural images in combination with pop-out search images. Secondly, the proposed strategy allows us to extend the ADS model by the aggregation of new feature dimensions such as movement, disparity or shape. Finally, from the evolutionary computing standpoint

we would like to incorporate other evolutionary techniques, such as niching, in order to study the problem of diversity.

**Acknowledgements** This research was founded by CONACyT through the Project 155045 - “Evolución de Cerebros Artificiales en Visión por Computadora”. First author supported by scholarship 164678 from CONACyT. Dr. Gustavo Olague graciously acknowledges the support of the Seventh Framework Program of the European Union through the Marie Curie International Research Staff Scheme, FP7-PEOPLE-2013-IRSES, Grant 612689 ACoBSEC and the project Analysis and Classification of Mental States of Vigilance with Evolutionary Computation. This research was also supported by TESE through the project DIMI-MCIM-004/08. In addition, the authors acknowledge the valuable comments of the anonymous reviewers, the Editor of Cognitive Computation, and the International Editorial Board whose enthusiasm is gladly appreciated.

## References

- Koch C, Ullman S. Shifts in selective visual attention: towards the underlying neural circuitry. *Hum Neurobiol.* 1985;4:219–27.
- Feldman JA. Four frames suffice: a provisional model of vision and space. *J Behav Brain Sci.* 1985;8:265–89.
- Treuhub A. *The cognitive brain.* Cambridge, MA: MIT Press; 1991. p. 384.
- Ungerleider LG, Mishkin M. Two cortical visual systems. In: *Analysis of visual behavior.* Cambridge, MA: MIT Press; 1982. p. 549–85.
- Milner D, Goodale MA. *The visual brain in action.* 2nd ed. Oxford: Oxford University Press; 1995. p. 320.
- Baluch F, Itti L. Mechanisms of top-down attention. *Trends Neurosci.* 2011;34(4):210–24.
- Yarbus A. *Eye movements and vision.* 1st ed. New York: Plenum Press; 1967. p. 222.
- James W. *The principles of psychology, vol 1.* Mineola: Dover; 1950. p. 696.
- Desimone R, Duncan J. Neural mechanisms of selective visual attention. *Annu Rev Neurosci.* 1995;18(1):193–222.
- Egeth H, Yantis S. Visual attention: control, representation, and time course. *Annu Rev Psychol.* 1997;48(1):269–97.
- Corbetta M, Shulman GL. Control of goal-directed and stimulus-driven attention in the brain. *Nat Rev Neurosci.* 2002;3(3):201–15.
- Underwood G. Cognitive processes in eye guidance: algorithms for attention in image processing. *Cogn Comput.* 2009;1:69–76.
- Dozal L, Olague G, Clemente E, Sánchez M. Evolving visual attention programs through evo features. In: *Proceedings of the 2012 European conference on applications of evolutionary computation.* Berlin, Heidelberg: Springer, *EvoApplications'12*; 2012. p. 326–35.
- Itti L, Koch C. Feature combination strategies for saliency-based visual attention systems. *J Electron Imaging.* 2001;10:161–9.
- Dozal L, Olague G, Clemente E. Evolving an artificial dorsal stream on purpose for visual attention. *EVOLVE—a bridge between probability set oriented numerics and evolutionary computation II.* *Adv Intell Syst Comput.* 2013;175:371–85.
- Olague G, Dozal L, Clemente E, Ocampo A. Optimizing an artificial dorsal stream on purpose for visual attention. In: Schütze O et al, editors. *EVOLVE—a bridge between probability set oriented numerics and evolutionary computation III.* *Studies in Computational Intelligence,* Springer 2014;500:141–66.
- Tsotsos JK. Analyzing vision at the complexity level. *Behav Brain Sci.* 1990;13:423–69.

18. Frintrop S, Rome E, Christensen HI. Computational visual attention systems and their cognitive foundation: a survey. *ACM Trans Appl Percept.* 2010;7(1):1–39.
19. Wischnewski M, Belardinelli A, Schneider WX, Steil JJ. Where to look next? Combining static and dynamic proto-objects in a TVA-based model of visual attention. *Cogn Comput.* 2010;2(4):326–43.
20. Taylor JG, Cutsuridis V. Saliency, attention, active visual search, and picture scanning. *Editor Spec Issue Cogn Comput.* 2011;3(1):1–3.
21. Neokleous KC, Avraamides MN, Neocleous CK, Schizas CN. Selective attention and consciousness: investigating their relation through computational modelling. *Cogn Comput.* 2011;3(1):321–31.
22. Vernon D. Cognitive vision: the case for embodied perception. *Image Vis Comput.* 2008;26:127–40.
23. Nagel HH. Reflections on cognitive vision systems. In: Crowley JL, Piater JH, Vincze M, Paletta L, editors. *Proceedings of the third international conference on computer vision systems.* Heidelberg: Springer; 2003. p. 34–43.
24. Tsotsos JK. Cognitive vision needs attention to link sensing with recognition. In: Christensen HI, Nagel HH, editors. *Cognitive vision systems: sampling the spectrum of approaches, LNCS.* Heidelberg: Springer; 2006. p. 25–35.
25. Posner MI, Snyder CR, Davidson BJ. Attention and the detection of signals. *J Exp Psychol.* 1980;109(2):160–74.
26. Treisman AM, Gelade G. A feature-integration theory of attention. *Cogn Psychol.* 1980;12(1):97–136.
27. Wolfe JM. Visual attention. In: De Valois KK, editor. *Seeing.* 2nd ed. San Diego, CA: Academic Press; 2000. p. 335–86.
28. Olague G. Evolutionary computer vision—the first footprints. *Springer (accepted for publication, to appear).*
29. Fukushima K. A neural network model for selective attention in visual pattern recognition. *Biol Cybern.* 1986;55(1):5–15.
30. Burt P. Attention mechanisms for vision in a dynamic world. In: *9th international conference on pattern recognition;* 1988. p. 977–87.
31. Sandon PA. Simulating visual attention. *J Cogn Neurosci.* 1990;2(3):213–31.
32. Olshausen BA, Anderson CH, Van Essen DC. A neurobiological model of visual attention and invariant pattern recognition based on dynamic routing of information. *J Neurosci.* 1993;13(11):4700–19.
33. Palmer SE. The psychology of perceptual organization: a transformational approach. In: Beck J, Hope B, Rosenfeld A, editors. *Human and machine vision.* New York: Academic Press; 1983. p. 269–339.
34. Milanese R. Detecting salient regions in an image: from biological evidence to computer implementation. Ph.D. thesis, Department of Computer Science, University of Genova, Switzerland; 1993.
35. Tsotsos JK, Culhane SM, Wai WYK, Lai Y, Davis N, Nuffo F. Modeling visual attention via selective tuning. *Artif Intell.* 1995;78(1–2):507–45.
36. Itti L, Koch C, Niebur E. A model of saliency-based visual attention for rapid scene analysis. *IEEE Trans Pattern Anal Mach Intell.* 1998;20(11):1254–59.
37. Torralba A. Modeling global scene factors in attention. *J Opt Soc Am A Opt Image Sci Vis.* 2003;20(7):1407–18.
38. Walther D, Koch C. Modeling attention to salient proto-objects. *Neural Netw.* 2006;19(9):1395–407.
39. Cutsuridis V. A cognitive model of saliency, attention, and picture scanning. *Cogn Comput.* 2009;1(4):292–9.
40. Kootstra G, Boer B, Schomaker LRB. Predicting eye fixations on complex visual stimuli using local symmetry. *Cogn Comput.* 2011;3(1):223–40.
41. Marat S, Rahman A, Pellerin D, Guyader N, Houzet D. Improving visual saliency by adding ‘face feature map’ and ‘center bias’. *Cogn Comput.* 2013;5(1):63–75.
42. Koza JR. Genetic programming: on the programming of computers by means of natural selection. 1st ed. Cambridge: A Bradford Book; 1992. p. 840.
43. Suder K, Wörgötter F. The control of low level information flow in the visual system. *Rev Neurosci.* 2000;11(2–3):127–46.
44. Julesz B. A brief outline of the texton theory of human vision. *Trends Neurosci.* 1984;7(2):41–45.
45. Hubel DH, Wiesel TN. Receptive fields of single neurons in the cat’s striate cortex. *J Physiol.* 1959; 148(3):574–91.
46. Boynton RM. Human color vision. 2nd ed. New York: Holt, Rinehart and Winston; 1979. p. 438.
47. Robinson DL, Petersen SE. The pulvinar and visual saliency. *Trends Neurosci.* 1992;15(4):127–32.
48. Zhaoping L. A saliency map in primary visual cortex. *Trends Cogn Sci.* 2002;6(1):9–16.
49. Mazer JA, Gallant JL. Goal-related activity in V4 during free viewing visual search. Evidence for a ventral stream visual saliency map. *Neuron.* 2003;40(6):1241–50.
50. Gottlieb J. From thought to action: the parietal cortex as a bridge between perception, action, and cognition. *Neuron.* 2007; 53(1):9–16.
51. Taylor K, Stein J. Attention, intention and saliency in the posterior parietal cortex. *Neurocomputing.* 1999;26–27:901–10.
52. Rensink RA. Seeing, sensing and scrutinizing. *Vis Res.* 2000;40(10–12):1469–87.
53. Rensink RA. The dynamic representation of scenes. *Vis cogn.* 2000;7(1–3):17–42.
54. Barton RA. Visual specialization and brain evolution in primates. *Proc Biol Sci.* 1998; 265(1409):1933–7.
55. Yokoyama S. Molecular evolution of color vision in vertebrates. *Gene.* 2002;300(1–2):69–78.
56. Rijsbergen CJV. Information retrieval. 2nd ed. London: Butterworths; 1979. p. 208.
57. Pérez CB, Olague G. Learning invariant region descriptor operators with genetic programming and the f-measure. In: *ICPR 19th international conference on pattern recognition;* 2008. p. 1–4.
58. Gimenez D, Evans AN. An evaluation of area morphology scale-spaces for colour images. *Comput Vis Image Underst.* 2008;110(1):32–42.
59. Atmosukarto I, Shapiro LG, Heike C. The use of genetic programming for learning 3D craniofacial shape quantifications. In: *ICPR 20th international conference on pattern recognition;* 2010. p. 2444–7.