

Adversarial Attacks Assessment of Salient Object Detection via Symbolic Learning

Gustavo Olague, *Senior Member, IEEE*, Roberto Pineda, Gerardo Ibarra-Vazquez, Matthieu Olague, Axel Martinez, Sambit Bakshi, *Senior Member, IEEE*, Jonathan Vargas, and Isnardo Reducindo

Abstract—Machine learning is at the center of mainstream technology and outperforms classical approaches to handcrafted feature design. Aside from its learning process for artificial feature extraction, it has an end-to-end paradigm from input to output, reaching outstandingly accurate results. However, security concerns about its robustness to malicious and imperceptible perturbations have drawn attention since its prediction can be changed entirely. Salient object detection is a research area where deep convolutional neural networks have proven effective but whose trustworthiness represents a significant issue requiring analysis and solutions to hackers' attacks. Brain programming is a kind of symbolic learning in the vein of good old-fashioned artificial intelligence. This work provides evidence that symbolic learning robustness is crucial in designing reliable visual attention systems since it can withstand even the most intense perturbations. We test this evolutionary computation methodology against several adversarial attacks and noise perturbations using standard databases and a real-world problem of a shorebird called the Snowy Plover portraying a visual attention task. We compare our methodology with five different deep learning approaches, proving that they do not match the symbolic paradigm regarding robustness. All neural networks suffer significant performance losses, while brain programming stands its ground and remains unaffected. Also, by studying the Snowy Plover, we remark on the importance of security in surveillance activities regarding wildlife protection and conservation.

Index Terms—Adversarial examples, Visual attention, Brain programming, Adversarial patch, Wildlife conservation.

1 INTRODUCTION

ADVERSARIAL robustness is a critical feature that recent deep learning (DL) architectures are searching for due to their inability to defend themselves from adversarial attacks (AAs), which are strategies that attempt to fool them. Therefore, despite the enormous advantages DL has brought to several research areas, including visual attention (VA), they are untrustworthy. Adversarial examples (AEs) are maliciously-constructed inputs that fool machine learning models by usually degrading the output performance of image classification.

Robustness is a required property when security, safety, and certainty are mandatory in surveillance, life conservation, and defense applications. Robustness is part of a new wave of computer science that is not alien to evolutionary

computation [1]. Robustness captures whether the system can deliver correct service conditions beyond the typical domain of operation and without fundamental changes to the original system, i.e., the ability to resist shocks without adapting or changing its behavior.

1.1 Related Work

Recent work has shown that these attacks are generalized to salient object detection (SOD) [2]. Li et al. have made the problem of AAs evident since VA is usually an input for more complex visual tasks. The performance of highlighting conspicuous objects is seriously affected by corrupt input images (AEs) in such a way that succeeding stages produce poor results, compromising the entire system. This first work describing the lack of robustness shows how DL compromises the application of such technologies in situations where security and reliability are critical features to avoid failure when detecting the object of interest. Nevertheless, this first work considers only non-target attacks leaving aside other methodologies regarding AAs [3]. In Section 1.2, we review a scheme to classify AAs and introduce later the experimental design strategy that we conduct in this work.

In the most recent survey of SOD, the authors included a small section about the lack of robustness against AEs [4]. Wang et al. provided some experiments showing that VA based on deep SOD models is brittle since their performance degrades when contaminated with corrupted inputs. In the first set of experiments, the authors contrasted the behavior of three deep models against heuristic methods. According to their results, heuristic methods do not match deep models in terms of performance; hence, comparing both approaches from a robustness standpoint against AEs is pointless.

- G. Olague is with the Department of Computer Science, CICESE Research Center, Carretera Ensenada-Tijuana 3918, 22860, México. Email: olague@cicese.mx
- R. Pineda is with the Department of Computer Science, CICESE Research Center, Carretera Ensenada-Tijuana 3918, 22860, México.
- G. Ibarra-Vazquez is with the Institute for Future of Education. Monterrey Institute of Technology and Higher Education, Monterrey, N.L., 64849, México.
- M. Olague is with Anáhuac University, Querétaro, El Márquez, 72246, México. email: matthieu.olague03@anahuac.mx
- A. Martinez is with the Department of Computer Science, CICESE Research Center, Carretera Ensenada-Tijuana 3918, 22860, México.
- S. Bakshi is with Department of Computer Science and Engineering, National Institute of Technology Rourkela, Odisha 769008, India. email: bakshisambit@ieee.org
- J. Vargas is with Terra-Peninsular and Cornell Lab of Ornithology, México.
- I. Reducindo is with the Information Science Faculty of Universidad Autónoma de San Luis Potosí, México.

Manuscript received April 19, 2005; revised August 26, 2015.

Among the three deep models, the authors highlight the excellent robustness of Pixel-wise Contextual Attention for Saliency Detection (PICANet) against a wide range of input perturbations, including Gaussian blur, Gaussian noise, and rotation [5]. Authors attribute this result to its effective non-local operation, revealing that effective network designs improve robustness to random perturbations. Later, in the experimental section, we challenge this idea by showing that this network exhibits poor robustness to Gaussian, Salt & Pepper, and Speckle noise. In [4], authors presented a second set of experiments by adopting and modifying an AA algorithm for semantic segmentation to evaluate robustness. They kept the same three deep SOD models for the analysis while excluding the heuristic methods. The results confirm the poor robustness against AEs since attacks designed with the knowledge of deep SOD models fail to highlight the complete salient objects while wrongly recognizing salient objects as part of the background areas. Such attacks prevent SOD models from producing reliable salient object candidates for other safety-critical applications. However, the proposed attack presents weak transferability between the three different network structures. Again, our results contradict this idea, as reported in the experimental section, by showing that many designed attacks affect networks with different architectures.

As we have observed from the reviewed literature, AAs in SOD are rarely studied. Nevertheless, this topic has the potential to become mainstream in computer science as one of the current studies on the susceptibility of deep neural networks for classification tasks [6]. In fact, in this study, we incorporated the most recent work of DL for SOD considering AAs [7], which we introduce in Section 2.1. Currently, SOD plays a significant role in several security applications for detecting candidate interest targets, and the fact that a technique from evolutionary computation is reliable and secure against inconspicuous attacks crafted to affect the functioning of a vast range of government and commercial applications is, without a doubt, a research area to be explored in the future. The present study represents the first attempt to show the effectiveness of artificial evolution as a symbolic learning paradigm against malicious threats in the case of SOD.

1.2 Problem Statement

AA is a relevant topic in the current study of DL models, which it started to attract attention in image classification models where the phenomenon was first discovered. Soon, the subject spread to diverse areas where it has been demonstrated to appear as a vulnerability in DL modeling. In this section, we summarize the explanation of how AAs affect DL models while focusing on VA modeling. Therefore, given a set of images $\mathcal{I} = \{\mathbf{I}_1, \mathbf{I}_2, \dots, \mathbf{I}_n\}$ and their corresponding proto-objects $\mathcal{P} = \{\mathcal{P}_1, \mathcal{P}_2, \dots, \mathcal{P}_n\}$, the deep SOD model establishes a relationship between an image (\mathbf{I}_i) and the proto-object (\mathcal{P}_i) through the following equation:

$$\mathcal{P} = f(\mathbf{I}) = A(\mathbf{W}^T \mathbf{I}) \quad , \quad (1)$$

where function $f(\cdot)$ is the deep SOD model, \mathbf{W} are the associated weight parameters, and $A(\cdot)$ is an activation function.

AAs denote the erroneous behavior of such models when the input image suffers a small change in its pixels by adding a perturbation ρ to create the AE $\mathbf{I}_\rho = \mathbf{I} + \rho$ such that:

$$f(\mathbf{I}) \neq f(\mathbf{I}_\rho) \text{ s.t. } \|\mathbf{I} - \mathbf{I}_\rho\|_p < \alpha \quad (2)$$

where $p \in \mathbb{N} \mid p \geq 1$ and $\alpha \in \mathbb{R} \mid \alpha \geq 0$. The AE is considered the intentionally modified input image that is recognized differently than the \mathbf{I} . The level of change in the pixels is defined by $\|\mathbf{I} - \mathbf{I}_\rho\|_p < \alpha$ that constrains the AE to be too small so it may be imperceptible to the human eye, although this constraint can be overlooked. A simple explanation of how AAs break DL models, rendering them vulnerable, is that if we compute the dot product with the associated weight matrix from the model. For the AE, we obtained $\mathbf{W}^T \mathbf{I}_\rho = \mathbf{W}^T \mathbf{I} + \mathbf{W}^T \rho$. This is caused by the linearity of neural networks, which are intentionally designed with activation functions that perform linearly so that they are easier to optimize. Therefore, the AE will grow the activation function by $\mathbf{W}^T \rho$.

Additionally, the linearity of DL models reveals another consequence of AAs. Every AE that is easy or difficult to compute in a specific model affects another model, even with different architectures. They only have to be trained for the same task. However, we extend this effect between different tasks, as explained in Section 2.4.3. This phenomenon is defined as the transferability effect that establishes the spread of this erroneous behavior between DL models with such small perturbations. AAs are usually classified depending on their strategy, which defines how the perturbation is created. White box attacks assume the complete knowledge of the architecture model, its parameter values, and the trained task. On the contrary, a black box attack optimizes the perturbation during a testing procedure to change the original prediction without knowing the model. These strategies are also considered to fool a model into believing a specific label (targeted attack) or that the predicted label is irrelevant (untargeted attack) while it is not the original label.

1.3 Contributions

This research is part of discovering new properties that evolutionary algorithms possess against DL. We extend the first results reported at the 25th International Conference on the Applications of Evolutionary Computation, where we reported the first discoveries about the robustness against AAs of brain programming (BP) for the problem of SOD [1]. We note the following contributions:

- 1) This study shows the robustness of a symbolic SOD algorithm against AAs by conducting a deeper analysis and contrasting it with five DL models, three AAs, three different noises, and five different datasets.
- 2) This study considers not only standard databases, but a real-world problem focused on wildlife conservation that proves a real challenge for deep-learning SOD methods.

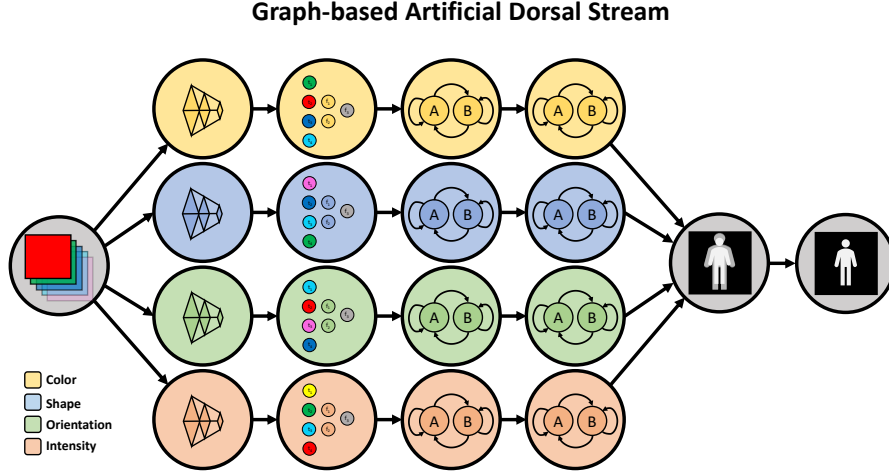


Fig. 1: The proposed method encodes an individual as a template, and we apply artificial evolution to discover a set of mathematical and computational functions optimized for the task of salient object detection. The evolved solutions are robust against different AEs, as the experiments exemplify.

2 METHODOLOGY

This section contains four subsections. First, we describe six learning-based methods commonly applied to SOD problems. Then, we provide a mathematical formulation for learning-based SOD modeling. Later, we introduce the idea of robustness through continuity criteria. Finally, we list our choice of AAs.

2.1 Symbolic and Subsymbolic Approaches for SOD

Nowadays, SOD researchers predominantly prefer solutions based on DL in contrast to traditional solutions that lack the idea of learning, so the latter's influence has been reduced recently. However, current research continues the tradition by using the same databases and assessment schemes outlined before the flourishing of DL in VA. BP is a design scheme where we incorporate symbolic learning to enhance traditional methodologies. All research follows the path of empirical investigation while modeling hand-designed solutions regardless of the approach. In conventional methods, features were designed by hand, and in DL methods, architectures are still designed by hand. Here, we propose to compare five different networks with a recently designed algorithm under the BP scheme, the results of which we will describe later. The five selected networks have similar characteristics since all architectures are fully convolutional networks, adopt a fully supervised learning, bottom-up or top-down network scheme, and follow single-task learning.

- *BP* is an evolutionary computation paradigm loosely based on the inner workings of the visual cortex. In this paper, we focus on the analogy to the dorsal stream to create computational models of SOD from an input image (Figure 1). The goal is to show the robustness of the methodology against AEs. The strategy follows

a goal-oriented framework where we study learning as a symbolic optimization process where an individual consists of a template describing the VA model while discovering critical parts of the algorithm through artificial evolution. This study focuses on the mathematical notation to formulate the SOD problem from the viewpoint of robustness and leaves the full explanation of the algorithm to consult in a previous document [8].

- Adversarial Robust SOD Networks with Learnable Noise (LeNo) proposes the usage of simple shallow noise and noise estimation embedded in the encoder and decoder of arbitrary SOD networks to defend against AAs [7]. Introduced in 2022, LeNo initializes the shallow noise with a cross-shaped Gaussian distribution inspired by the center prior of the human VA mechanism. Instead of adding additional network components for post-processing, the proposed noise estimation modifies only one decoder channel. This method outperforms previous works on adversarial and clean images, contributing to the stronger robustness of SOD.
- SOD via Extremely-Downsampled Network (EDN) focuses on enhancing high-level features through extreme downsampling to effectively learn a global view of the whole image, leading to accurate salient object localization [9]. Proposed in 2022, improved multi-level feature fusion is accomplished through the construction of the scale-correlated pyramid convolution to build a decoder for recovering object details from the above extreme downsampling. Not only does this work achieve state-of-the-art performance, but it also focuses on working at real-time speeds.
- Boundary-Aware SOD (BASNet) is a deep convolutional neural network (CNN) focusing on the boundary quality that appears in 2019 [10]. The proposed

BASNet consists of two modules, the first inspired by U-Net and a semantic segmentation model (SegNet), which implement a salient object prediction module as an Encoder Decoder network, capturing high-level global contexts and low-level details at the same time. The input convolutional layer and the first four stages are adopted from [11], the transfer Residual Neural Network (ResNet-34), while the residual refine module consists of filters followed by a batch normalization and a Rectified Linear Unit (ReLU) activation function [12].

- Learning PICA Net aims to generate an attention map at each pixel over its context region and construct an attended contextual feature to enhance the feature representability of convolutional networks [5]. The authors proposed the network in 2018 based on two pixel-wise attention modes: global attention and local attention. For each location, the former generates attention based on the Recurrent Neural Network over the whole feature map [13], while the latter works on a local region using several convolutional layers. The whole network is based on the VGG 16 layer [14] as the backbone while applying U-Net, which is a CNN developed for biomedical image segmentation.
- The Deep Hierarchical Saliency Network (DHSNet) is an end-to-end CNN detecting salient objects that appeared in 2016 [15]. It consists of a global view step followed by a Hierarchical Recurrent CNN. Based on the Very Deep Convolutional Networks (VGG net), the first network creates a coarse global saliency map to roughly detect and localize salient objects. The second step refines saliency maps in detail by incorporating local contexts.

We have appreciated all five networks based their architectures on previous development of CNNs. The idea of incorporating symbolic learning based on previous neuroscientific research is equivalent; therefore, we aim to show the robustness of symbolic modeling.

2.2 SOD Evaluation

The image-based SOD problem can be formulated as follows. Given an input image $\mathbf{I} \in \mathbb{R}^{u \times v \times 3}$ of size $u \times v$ for each color channel, an SOD model f maps the input image \mathbf{I} to a proto-object $\mathcal{P} = f(\mathbf{I}, F, T, a) | \mathcal{P} \in \{0, 1\}^{u \times v}$, where F and T represent the function and terminal sets, respectively, from the feature extraction, and a is the set of parameters controlling the evolutionary process.

BP is the algorithm in charge of tuning (F, T) , looking for optimal feature extraction from the input images using the visual operators embedded into the artificial dorsal stream. For learning-based SOD, the meta-model $f(\cdot)$ is learned through a set of training samples. Given a set of images $\mathcal{I} = \{\mathbf{I}_1, \mathbf{I}_2, \dots, \mathbf{I}_n\}$, with n defining the number of training images, and corresponding binary SOD ground-truth masks

$$\mathcal{G} = \{G_1, \dots, G_n | G \in \{0, 1\}^{u \times v}\},$$

the goal of learning is to find $f \in \mathcal{F}$ that minimizes the prediction error, i.e., $\sum_n Q(\mathcal{P}_n, G_n)$, where Q as a quality criterion in this case is a certain distance measure (e.g., F_β

measure), and \mathcal{F} is the set of potential mapping functions. Deep SOD methods typically model $f(\cdot)$ through modern DL techniques, as we reviewed in Section 2.1.

In the case of BP, we have $f(\mathbf{I}, F, T, a)$, such that

$$\mathcal{P}^* \in \mathcal{S} : \mathcal{Z}(\mathcal{P}^*, G) \geq \mathcal{Z}(\mathcal{P}, G)$$

which maximizes the overlap between \mathcal{P} and G from the solution space \mathcal{S} , such that

$$Z = \sum_{i=1}^n \mathcal{Z}(\mathcal{P}_i, G_i)$$

where \mathcal{Z} is a statistical measure of accuracy, $\mathcal{Z} \triangleq F_\beta(\cdot)$, that is computed as follows:

$$F_\beta(P_i, R_i) = \sum_{i=1}^n \sum_{j=1}^m \frac{(1 + \beta^2)p_j \cdot r_j}{\beta^2 p_j + r_j} \quad (3)$$

with m representing the number of thresholds to build the proto-object, and n defines the number of training images. Precision data of an image are denoted with $P_j = (p_1, p_2, \dots, p_n)$ and recall data by $R_j = (r_1, r_2, \dots, r_n)$ with

$$p = \frac{t_p}{t_p + f_p},$$

$$r = \frac{t_p}{t_p + f_n},$$

where t_p means true positive, f_p is false positive, and f_n denotes false negative. F_β measures the effectiveness of retrieval with respect to a user who attaches β times as much importance to recall as precision. β^2 is set to 0.3 to emphasize the precision following the standard protocol [16].

We empirically calculate precision and recall based on a number n of thresholds for each image in the dataset while considering each algorithm run. The F_β measure of each of these results is calculated based on the generated saliency map and splicing it with the ground-truth image. After this, we take the maximum value to obtain each image's best F_β score. Then, we calculate the average of all the images' F_β scores to obtain the final F_β value according to Equation (3), which corresponds to Z .

2.3 Model Robustness through Continuity Criteria

Let $\mathcal{P} = f(\mathbf{I})$, where $f(\cdot)$ is the model (NN, BP, etc.). For example, $\mathcal{P} = f(\mathbf{I}, F, T, a) \sim f(\mathbf{I}) \therefore f(\mathbf{I}) \neq f(\mathbf{I}_\rho) : \|\mathbf{I} - \mathbf{I}_\rho\| < \alpha$.

AEs find input \mathbf{I}_ρ in the subspace \mathcal{I}' , such that $\mathbf{I}_\rho \in \mathcal{I}'$ and $f(\mathbf{I}) \neq f(\mathbf{I}_\rho)$. Nevertheless we denote robustness in terms of function continuity. Given a model's function $f(\cdot)$ in an input subspace. \mathcal{I} is said to be robust at $\mathbf{I} \in \mathcal{I}$; if $\mathbf{I}_k \mapsto \mathbf{I}$, then $f(\mathbf{I}_k) \mapsto f(\mathbf{I})$. Equivalently, $f(\mathbf{I})$ is robust at \mathbf{I} , for all $\mathbf{I}' \in \mathcal{I}$, if given a $\epsilon > 0$, there is a $\delta > 0$, such that $\|\mathbf{I} - \mathbf{I}'\| < \delta$ implies $\|f(\mathbf{I}) - f(\mathbf{I}')\| < \epsilon$. Hence, if $f(\mathbf{I})$ is robust for every \mathbf{I} , then $f(\mathbf{I})$ is said to be robust on \mathcal{I} .

2.4 AAs

2.4.1 Fast Gradient Sign Method (FGSM)

FGSM was first implemented by Goodfellow et al. by noting that linear behaviors of high-dimensional spaces easily generate AEs [17]. FGSM proposes to increase the loss of the detector by solving the following equation: $\rho = \epsilon \text{sign}(\Delta J(\theta, \mathbf{I}, \mathbf{G}))$, where $\Delta J()$ computes the gradient of the cost function around the current value of the model parameters θ with respect to the image \mathbf{I} , and the ground truth \mathbf{G} . $\text{sign}()$ denotes the sign function, which maximizes the magnitude of the loss and ϵ is a small scalar value that restricts the norm L_∞ of the perturbation.

2.4.2 Multipixel Attack

The multipixel adversarial perturbation is a black box untar-getted attack since it does not require network information. The one-pixel attack is the basis for this algorithm [18]. However, the original algorithm is not suitable for real-world problems since it can only work for icon images. In summary, let the vector $\mathbf{I} = (\mathbf{I}_1, \dots, \mathbf{I}_n)$ be an n -dimensional image, which is the input of the salient object detector $f()$ that correctly predicts the object t from the image. The statistics of \mathbf{I} associated with the object t is $\mathcal{Z}(f(\mathbf{I}), f(\mathbf{I} + e(\mathbf{I})))$. It builds an additive adversarial perturbation vector $e(\mathbf{I}) = (e_1, \dots, e_n)$ according to \mathbf{I} , the salient object detector, and the limitation of maximum modifications d , a small number that expresses the dimensions that are modified, while other dimensions of $e(\mathbf{I})$ that are left are zeros. The main purpose is to find the optimal solution $e(\mathbf{I})^*$ that solves the following equation:

$$\begin{aligned} \min_{e(\mathbf{I})^*} \mathcal{Z}(f(\mathbf{I}), f(\mathbf{I} + e(\mathbf{I}))) \\ \text{s.t. } \|e(\mathbf{I})\|_0 \leq d \end{aligned}$$

2.4.3 Adversarial Patch (AP)

The AP is a white box attack that creates a particular form of perturbation, which builds a universal perturbation with a patch shape [19]. Even though this attack is performed to be used on image classification tasks, its universality makes it perfect to be analyzed with the transferability effect that this perturbation provokes. The AP builds the perturbation, maximizing $f_{\text{target}}(\mathbf{I} + \hat{\mathbf{p}})$ to a specific class where it finds the optimal patch $\hat{\mathbf{p}}$. Universal perturbations such as the AP pose a powerful foolproof that relies on the wide variety of transformations that can be applied to the patch to fool the system, or it can even be printed to work in real-world conditions.

The algorithm trains the AP $\hat{\mathbf{p}}$ using a set of images \mathcal{I} , and a variant of the expectation over transformation framework to optimize the following equation:

$$\hat{\mathbf{p}} = \underset{\mathbf{p}}{\text{argmax}} \mathbb{E}_{\mathbf{I} \in \mathcal{I}, t \in T, l \in L} [\log f(y_{\text{target}}, A(\mathbf{p}, \mathbf{I}, l, t))] \quad (4)$$

where y_{target} represents the target class in the image classification model and $A(\mathbf{p}, \mathbf{I}, l, t)$ is a function that first applies the transformation t from a distribution over transformations T to the patch \mathbf{p} . Next, it puts the transformed patch \mathbf{p} at the location l from a distribution over locations L to the image \mathbf{I} . The effectiveness of these patches resides

in the expectation over the training images that increases success, regardless of the background.

2.4.4 Noise-based Attacks

AEs are usually interpreted as strategically perturbed images that fool DL models. However, randomly perturbed images are also studied to verify the robustness of SOD models. We employ three types of noise (Gaussian, Salt & Pepper, and Speckle noise) to study the behavior of such algorithms with randomly perturbed inputs. Gaussian noise adds a white noise with a probability density function of a normally distributed random variable with an expected value μ and variance σ . The general probability density function is as follows:

$$g(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (5)$$

Salt & Pepper noise with density (d) assigns each pixel (p), from the total pixels (tp) in an image, a random probability value (pv) from a standard uniform distribution between $(0, 1)$. Therefore, it applies to the following cases

- $p = 0$, for $pv \in (0, d/2)$ limited to $d \times tp/2$ pixels.
- $p = \text{max value}$, for $pv \in [d/2, d]$ limited to $d \times tp/2$ pixels.
- $p = p$, for $pv \in [d, 1)$.

Speckle noise add a multiplicative perturbation using the following equation $J = \mathbf{I} + m \times \mathbf{I}$, where m is a uniformly distributed random noise with $\mu = 0$ and $\sigma = 0.05$.

3 EXPERIMENTAL RESULTS

We organize this section into three subsections describing the databases, the generation of AEs, and validation results.

3.1 Databases

In this work, we use five different image databases as a way to evaluate adversarial robustness. Four of these databases show common everyday objects. However, we experimented with a new image database not used in previous SOD research that holds real-world objects. By “*real-world*,” we refer to objects in a scene positioned in their natural and unperturbed environment and occurring naturally without any external factors. A good example is nature photography, where photographers capture an animal without manipulating the scene. Data collection normally includes respecting the many properties that might be present when taking a photograph, i.e., light and weather conditions, debris, low contrast, obstructions, and object position, among others. Experimenting with a real-world database is a big step toward SOD, given that many research papers focus on using images specifically designed to test a novel algorithm and where the properties for detection are mostly advantageous [4]. These databases mostly show large objects with a high contrast between the foreground and background. The databases used in this work that refer to the point just mentioned are FT [20], ImgSal [21], PASCAL-S [16], and DUTS [22].

FT contains 800 images for training and 200 for testing, and Figure 2 provides some images of the dataset. The dataset includes diverse portraits of animate and inanimate



Fig. 2: Example images of the FT database with the corresponding ground-truth.



Fig. 5: Example images of the SNPL database with the corresponding ground-truth.

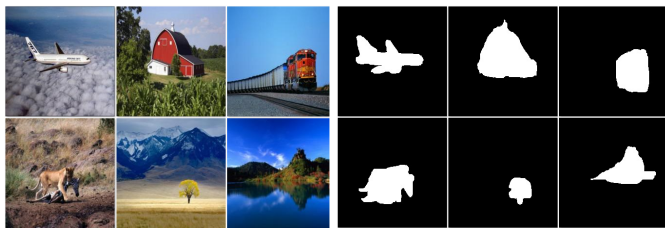


Fig. 3: Example images of the ImgSal database with the corresponding ground-truth.



Fig. 6: Example images of the DUTS database with the corresponding ground-truth.

objects with image sizes ranging from 324×216 up to 400×300 pixels. PASCAL-S consists of 680 images for training and 170 images for testing (Figure 3). PASCAL-S contains scenes of domestic animals, persons, and means of air and sea transport, with images ranging from 200×300 to 375×500 pixels. IMGSA provides 235 images, split into 188 training images and 47 images for testing (Figure 4). This dataset includes wild animals, flora, and different objects and people with image sizes of 480×640 pixels. Finally, regarding standard datasets, DUTS is comprised of 10,553 training images and 5,019 test images (Figure 6). DUTS portrays people, animals, insects, and objects in a wide variety of scenarios with image sizes ranging from 266×400 to 400×192 pixels.

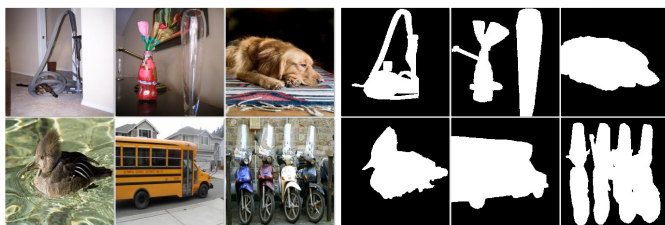


Fig. 4: Example images of the PASCAL-S database with the corresponding ground-truth.

This work has another purpose: for experimenting with AAs using a live bird database. The Snowy Plover (*Charadrius nivosus*) was listed by the U.S. Fish & Wildlife Service (USFWS) as a threatened species under the federal Endangered Species Act. The Snowy Plover is a small shorebird that can reach a length of 6.7 in (17 cm) with

a wingspan of 13.4 in (34 cm). The bird has a short, thin, black bill and gray legs. The upper body varies from grayish to light-brown, with a white belly and black on the forehead and ears. The Snowy Plover inhabits sandy beaches. Many organizations have been working for the protection of the Snowy Plover, incorporating mainstream technology based on DL. However, AAs expose the vulnerabilities of many neural networks used for wildlife conservation. These neural networks are part of real-life monitoring systems such as camera traps for animal identification, and counting [23], drones for scanning a habitat [24], and even acoustic recording devices [25]. Many of these applications involve defending against the attacks of individuals performing illegal activities that put wildlife in danger, such as poaching, which has become an increasing problem in today's world and a subject that conservationists desperately need a solution to, not only regarding accuracy but reliability and security. Other activities threatening species' well-being are the constant increase in human population and their disturbance of the animal's natural habitat, such as beaches where the Snowy Plover and many other birds reside. By exposing the vulnerabilities of these neural networks, the scientific community can focus on improving the resilience of their algorithms to prevent attacks by malicious individuals. This work shows that an evolutionary computation technique is appropriate to approach SOD, with the advantage of not being susceptible to AAs while performing high detection.

The fourth real-world database used in this work comprises 250 images of the Snowy Plover, named the SNPL database (Figure 5). We took images using a Nikon DSLR camera and a 200-500mm f/5.6 Supertelefoto lens, considering different weather, exposure, and range conditions. We took each of the 250 photographs in the bird's natural habitat, consisting of sandy beaches and mudflats on the

Pacific coast of Baja California, Mexico. Each one of these images shows one or more Snowy Plovers, whose size on the scene varies according to the distance at which we took a photograph while adjusting the focal length. Another essential feature involves the background, where a variety of distractors, such as plants, water bodies, and small objects (shells and debris), are present. Also, many of these images contain occlusions that partially cover the (ground truth) object of interest. Note also that the animal's sandy beach habitat and the primary color of its feathers have similar color tones. Many of these images lack a high contrast between the background and the Snowy Plover. Due to these characteristics, the SNPL database is a great basis to study such a real-world problem. This dataset represents a challenging scenario to test different SOD models. Also, we can compare it with the previous datasets to understand the limitations of academic datasets whose optimal size and exposure conditions prevent us from understanding the effect of AEs on VA.

3.2 Generation of AEs

In order to test the robustness, or resistance of the detection algorithms to AAs, including BP, we generated AEs for the validation set of each database used in this work. We used the Python programming language, version 3.8.3 and the machine learning framework PyTorch, version 1.6.0 [26] and explained the process for generating each AE.

We created six sets of AEs for the validation set of each database using FGSM coupled with the training stage of PICANet. We generated each of the six sets with a value of $\epsilon = \{2, 4, 8, 16, 32, 64\}$, so the images of each set have disturbances with different intensities, with 2 being the lowest perturbation and 64 being the highest variation. The second row of Figure 7 shows examples of this perturbation applied to an image of the Snowy Plover.

Regarding the AP, we created two sets using ResNet-50. The first set contains images where the patch covers an area of 70×70 pixels, named Patch, corresponding to 31% of the original image. The second set contains images whose patch is 50×50 pixels, named Patch(S), covering 22% of the original image (Figure 7). The reason for generating two sets with patches of different sizes was to assess whether the performance of the detection algorithms depends on the patch size, how much the results vary, or whether it ultimately depends on the pure disturbance regardless of size. We placed the patch randomly, without discriminating its location even when the patch was wholly or partially covering the object of interest.

We validated each dataset using the multipixel method, whose disturbance value $d = 10,000$ specifies the number of modified pixels. This result reflects a significant number of pixels that are not inconspicuous while allowing protruding objects to remain noticeable. This attack did not require knowledge of the internal parameters of any machine learning approach.

Next, we created for each database a set of AEs using Gaussian noise. Regarding all generated examples, we apply a $\sigma = 30$, which implements a grain of magnitude sufficient to distinguish the objects in the scene without a problem. Generating these examples did not require knowing the

internal parameters of any learning approach. Also, we validated each dataset with Salt & Pepper noise. Instead of applying it to grayscale images, we use color images, so the noise does not appear as black and white points but as color dots. This effect is because the Salt & Pepper noise takes the values of the maximum and minimum pixels of each color band and distributes them throughout the image. The generation of Salt & Pepper noise did not require knowledge of the internal parameters of any neural network. Finally, we create AEs for each dataset using Speckle noise. Since Speckle noise originates from physical conditions, it was necessary to implement a method to simulate it. Thus, random values were taken from a Gaussian distribution along image dimensions and multiplied by a variance of 0.3. Similar to the other noises, the Speckle noise does not require knowing the parameters of any neural network to implement it.

3.3 Validation Results

In this section, we explain how the experiment demonstrates the threats that recent models could face. We provide four tables divided by each studied database: FT, ImgSal, PASCAL-S, and SNPL. We organize each table by the kind of attack: a white box attack (FGSM), the transferability effect (AP), and black box attacks (multipixel and noise-based attacks). Values in bold correspond to the lowest score an algorithm obtained among all types of AAs. Finally, we present the results of applying the symbolic and subsymbolic methods learned with the FT dataset to the AAs made on the DUTS database. The goal is to validate the robustness of the proposed models using a different dataset from that used during learning.

3.3.1 FT database

Table 1 shows the results of applying an adversarial FGSM attack to the five neural networks and the BP algorithm using the FT database. In tests with the original images (no AA), neural networks outperformed the BP algorithm. However, as the attack is applied and the epsilon value increases, the networks' performance drops. PICANet's performance suffers a significant loss, starting with a score of 81.45% and steadily falling to 40.52% at $\epsilon = 64$, thus losing around 50% of its performance. Although the BASNet network has the best score in the test, with the original images achieving 92.67%, it also loses its performance and drops to 58.65%. The second best model with the original images is EDN, which achieved 90.93% before dropping to 53.43% with $\epsilon = 64$. LeNo achieves the fourth-best score at 87.49% while declining to 65.54%, which is lower than those of DHSNet and BP. Although the DHSNet network does not lower its score so drastically, it is also affected. On the other hand, BP is virtually unaffected after applying all epsilon values, indicating that it possesses significant robustness and is quite resistant to this type of attack, being able to detect protruding objects in the scene even with the most intense disturbance, which is unlike neural networks. It is worth mentioning that the FT database consists of images whose objects of interest cover a large part of the image, so these algorithms have an excellent facility for achieving detection, which is reflected in the high scores.

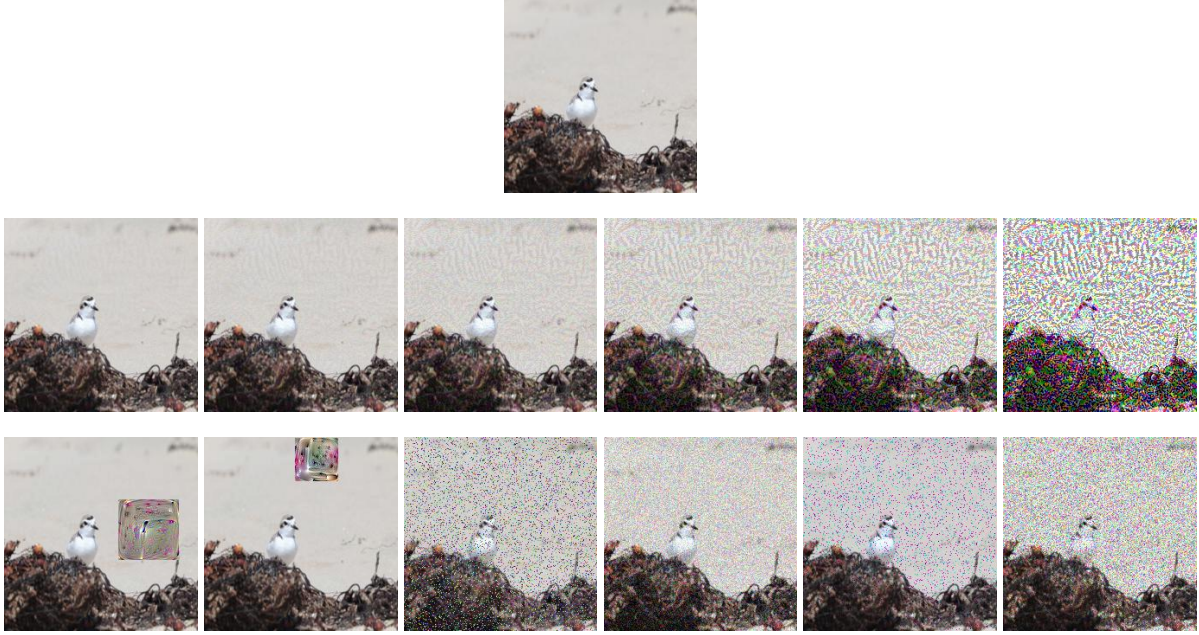


Fig. 7: Perturbations over an image of the SNPL database. On top is the original image, while in the second row we observe the image with different levels of perturbation using FGSM; the third row provides the resulting AEs using two versions of the adversarial patch, the multipixel attack, and the three different noises.

Table 1 also shows the results when applying the multipixel, AP, and noise attacks to the five neural networks and the BP algorithm using the FT database. PICA Net is primarily affected by the multipixel attack, reaching the lowest score of 66.86%, while EDN and LeNo reached the second and third lowest scores, respectively. The remaining methods (BP, BASNet, and DHSNet) keep their scores after the multipixel attack. We can observe that the AP causes a score decrease in all detection algorithms, including BP. However, this decrease is more severe in the BASNet and PICA Net networks, which had shown a high score with the undisturbed images. The DHSNet network, which had been shown to be unaffected by the FGSM attack, finally loses its score with the AP, dropping to 74.48%. On the other hand, the BP algorithm shows for the first time a decrease in its detection capacity, down to 67.96%, which suggests that the AP is a fairly powerful attack since the patches inserted in the images are, in many cases, classified as outstanding objects. EDN and LeNo scores also decreased after AP was applied. Lastly, although the smallest AP manages to lower the score of most algorithms, it fails to compare with the score of the largest patch, which was expected but may not be the case in the following experiments.

Regarding noise, the detection algorithms BP, BASNET, and DHSNet are practically unaffected, but not so with LeNo, EDN, and PICA Net. These last three networks lose detection ability against all three types of noise, although the score loss is not as drastic, suggesting that they are still resistant to these attacks. It is important to note that one of the possible reasons for the high robustness to noise attacks is because this database, as already mentioned, contains objects of interest of considerable size and high contrast with the background, making them easy to spot. This feature contrasts with the results of the following databases, where

noise attacks affect the score of the neural networks while BP maintains its high robustness.

3.3.2 *ImgSal database*

Table 2 shows the results of applying FGSM to the five neural networks and the BP algorithm using the ImgSal database. It is important to emphasize that the ImgSal database does not have correctly segmented ground truths, so the results in the table should be taken as an approximation of the actual score that the algorithms would have returned if the segmentation had been correct. In this case, the five neural networks, even BASNet and PICA Net, are severely affected by the FGSM attack, losing great detection capacity from the value $\epsilon = 8$. Their performance has already dropped considerably for the value $\epsilon = 64$, in the following order: EDN, BASNet, PICA Net, LeNo, and DHSNet, down to 25% for EDN. BP remains at an acceptable score even after the intense disturbance $\epsilon = 64$, dropping only by 8%. This result again underscores the robustness of BP against this type of AA.

Table 2 also displays the results when applying the multipixel, AP, and noise attacks to the five neural networks and the BP algorithm using the ImgSal database. In this case, unlike the FT databases, the multipixel attack significantly affects the five neural networks' performance. This result may be because the objects in the ImgSal database images are not as large as those in FT, so detection is more challenging. The BP algorithm is not affected by this attack. Regarding the AP, this seriously affects the detection capacity of all algorithms, with BASNet and PICA Net losing around 50% of their performance. Also, the small patch causes singular behavior in the LeNo, BP, and DHSNet algorithms; the score obtained is lower than in the regular-sized patch. This result indicates that a larger patch will not necessarily affect the performance of an algorithm to a greater extent.

TABLE 1: Summary of the results obtained with adversarial attacks (FGSM, multipixel, patch, and noises) applied to the FT database.

Algorithm	Measure	Original	FT						Multipixel	Patch	Patch(S)	Gaussian	S&P	Speckle
			$\epsilon = 2$	$\epsilon = 4$	$\epsilon = 8$	$\epsilon = 16$	$\epsilon = 32$	$\epsilon = 64$						
Brain [8]	Avg.	73.84	73.79	73.63	73.69	73.44	72.97	71.64	74.31	67.96	69.63	73.84	73.90	73.96
	σ	12.48	12.53	12.52	12.53	12.56	12.70	13.68	12.23	14.02	14.06	12.28	12.60	12.35
LeNo [7]	Avg.	87.49	87.40	87.17	87.16	86.74	83.65	65.54	72.99	80.17	80.61	86.59	83.56	82.37
	σ	14.38	14.43	14.85	14.58	15.18	17.57	25.22	24.17	16.31	16.34	15.55	18.64	19.35
EDN [9]	Avg.	90.93	90.87	90.77	90.42	88.25	82.20	53.43	69.10	81.48	83.28	86.08	86.01	82.88
	σ	12.82	12.81	12.89	13.12	16.12	21.04	26.38	26.95	17.06	16.12	18.33	18.77	21.95
BASNet [10]	Avg.	92.67	90.92	89.74	88.92	87.30	82.60	58.65	90.79	67.01	73.00	92.27	92.41	90.88
	σ	12.51	15.38	16.94	16.78	18.02	23.28	37.96	15.54	20.47	23.16	12.72	12.69	14.75
PICANet [5]	Avg.	81.45	74.26	67.84	61.01	57.04	51.21	40.52	66.86	61.04	69.78	78.39	78.95	75.89
	σ	15.27	18.74	20.83	22.58	22.91	22.76	22.19	22.14	19.79	17.06	17.42	19.40	20.97
DHSNet [15]	Avg.	88.63	87.92	87.46	86.68	85.62	83.85	78.92	85.67	74.48	79.04	87.84	86.98	86.89
	σ	12.43	12.86	13.30	13.74	14.48	15.95	19.90	14.50	17.67	16.38	13.31	14.12	14.73

In the case of noise, BP is not affected by any method, keeping its score very close to the original. Instead, the five neural networks decreased their performance, especially in Salt & Pepper and Speckle noises. Unlike FT, where no algorithm was affected by noise, the case of ImgSal does affect them due to the nature of its images, which pose an even more significant challenge at the time of detection.

3.3.3 PASCAL-S database

Table 3 shows the results of applying an adversarial FGSM attack to the five neural networks and the BP algorithm using the PASCAL database. At first glance, we can appreciate that the BP algorithm is not affected by this attack, keeping its score very close to the original and consistently above 60%. This result demonstrates once again the excellent robustness of BP. On the other hand, as we already observed in previous databases, neural networks do not keep up with this type of attack, and this case is no exception. Even though BASNet had the highest score in the original images (80.20%), its score dropped to 14.96% in the highest epsilon. The BP algorithm may not have had the highest score using the original images, but it outperforms four out of five neural networks, with the exception of LeNo, when applying the FGSM with the most severe perturbation.

Table 3 also shows the results when applying the multipixel, AP, and noise attacks to the five neural networks and the BP algorithm using the PASCAL-S database. In the case of the multipixel attack, the neural networks are considerably affected, especially BASNet, which drops from 80.20% to 57.26%. For the AP, the BASNet score drops even further, while PICANet mysteriously recovers at 4.63% compared with its multipixel score. This result indicates that the algorithm's performance highly correlates with the database, and in the AP, the PICANet algorithm tolerated the patch slightly better than EDN. Note that the LeNo score is slightly better than the PICANet score regarding the AP. However, robustness is more evident in the BP algorithm, which can acceptably withstand these two types of attacks, starting at 62.94% in the original images and up to 57.5% in Patch(S). Again, attacking BP with a normal-sized patch did not result in a lower score than attacking it with a small patch.

In the case of noise attacks, the BP and LeNo algorithms are largely unaffected, but we cannot say the same regarding the other neural networks. Even though it seems that the noise attack does not manage to affect the performance of these four neural networks so drastically, there is no doubt that there is a decrease in the detection capacity. EDN, BASNet, and PICANet are down an average of 8%–15%, while

DHSNet, which already proves robust to noise attacks, is down only 3% this time.

3.3.4 SNPL database

Unlike previous databases, the Snowy Plover (SNPL) images contain characteristics that come quite close to a real-world problem. In other words, we planned the photographs of these images in natural environments, with cameras for bird photography, and with different weather, lighting, and zoom conditions. This dataset allows it to test detection algorithms that are mostly only tested against databases with larger objects of interest that are optimally illuminated and have no real-world purpose.

Table 4 shows the results of applying an adversarial FGSM attack to the five neural networks and the BP algorithm using the SNPL database. The first observation in this experiment is the decrease in the scores of most of the algorithms compared with the previous databases. Except for PICANet in the test with the original images, the neural networks and BP lose most of their performance when trying to achieve detection with the SNPL database. This result is due to the different zoom and obstruction characteristics in this set of images. Even though the neural networks scored reasonably well against the other databases, where the objects of interest have a relatively high salience, a more significant challenge begins to break them. This outcome becomes much more evident when attacking them with AEs. For the case of FGSM, all neural networks show a gradual decrease starting at $\epsilon = 2$ and ending with a practically nonexistent score when attacking with a value of $\epsilon = 64$. LeNo loses more than half of the score reaching 39.79%, while BASNet completely loses the score with 0.99%. In the case of BP, when testing the BP algorithm with the original images, it starts with a score of 52.6%, below those of the PICANet, LeNo, EDN, and BASNet scores, in order of performance, but still better than that of DHSNet. However, by attacking with $\epsilon = 32$, BP already beat all neural networks, finishing with a score well above theirs. In the end, when attacking with $\epsilon = 64$, the BASNet network shows a total decrease of 98.3%, 89.6% for PICANet, 87.8% for EDN, 59.4% for DHSNet, and 55.86% for LeNo, while BP only drops to 19.7%.

Table 4 also shows the results when applying the multipixel, AP, and noise attacks to the five neural networks and the BP algorithm using the SNPL database. In the case of the multipixel attack, the massive decrease in the score of PICANet and BASNet is remarkable, to the point that the PICANet network dropped from 86.04% to 13.29% only with this attack, and the BASNet network suffered

TABLE 2: Summary of the results obtained with adversarial attacks (FGSM, multipixel, patch, and noises) applied to the ImgSal database.

ImgSal														
Algorithm	Measure	Original	$\epsilon = 2$	$\epsilon = 4$	$\epsilon = 8$	$\epsilon = 16$	$\epsilon = 32$	$\epsilon = 64$	Multipixel	Patch	Patch(S)	Gaussian	S&P	Speckle
Brain [8]	Avg.	62.66	62.56	62.52	61.96	60.30	58.87	54.20	60.37	46.25	46.12	62.27	62.18	61.28
	σ	23.96	23.83	24.09	24.11	24.43	24.72	24.73	23.54	25.13	24.61	23.91	24.21	23.73
LeNo [7]	Avg.	72.80	72.54	72.60	72.74	70.75	57.62	28.92	60.79	59.51	57.42	69.73	67.42	63.80
	σ	24.52	25.10	24.70	24.48	25.80	29.76	24.15	29.48	26.80	27.38	25.88	24.69	27.37
EDN [9]	Avg.	69.85	69.30	69.00	69.09	63.63	42.62	17.41	41.27	52.17	54.55	55.28	53.95	44.66
	σ	24.73	25.22	24.98	24.64	27.07	30.24	16.72	30.25	28.22	27.92	30.22	29.76	29.98
BASNet [10]	Avg.	63.75	59.84	55.96	53.78	54.76	44.34	20.69	54.19	27.92	31.49	60.01	57.27	55.06
	σ	31.73	33.32	34.43	35.62	34.19	35.76	30.10	35.81	26.28	27.64	34.68	34.57	34.18
PiCANet [5]	Avg.	71.60	63.73	56.14	50.71	49.52	44.85	25.78	50.33	35.6	38.49	65.12	67.23	64.29
	σ	22.16	28.51	29.29	28.36	28.58	28.75	22.96	30.62	23.57	22.48	29.82	30.56	29.41
DHSNet [15]	Avg.	53.63	53.25	52.51	50.86	49.22	45.96	39.89	49.12	41.94	41.60	51.19	48.83	49.04
	σ	24.85	24.62	24.73	24.97	24.81	24.98	24.34	25.66	26.80	25.84	24.83	25.37	25.77

TABLE 3: Summary of the results obtained with adversarial attacks (FGSM, multipixel, patch, and noises) applied to the PASCAL-S database.

PASCAL-S														
Algorithm	Measure	Original	$\epsilon = 2$	$\epsilon = 4$	$\epsilon = 8$	$\epsilon = 16$	$\epsilon = 32$	$\epsilon = 64$	Multipixel	Patch	Patch(S)	Gaussian	S&P	Speckle
Brain [8]	Avg.	62.94	62.79	62.34	62.53	62.38	61.53	60.53	61.74	58.05	57.50	62.55	61.51	61.55
	σ	20.89	20.65	20.80	21.13	21.60	21.28	21.54	18.28	18.84	17.96	18.12	18.41	18.58
LeNo [7]	Avg.	73.59	73.11	73.48	72.97	72.01	70.01	62.29	69.65	69.74	70.37	72.58	72.51	72.72
	σ	17.15	17.16	17.19	17.27	17.62	18.56	21.33	19.51	18.49	17.54	17.94	17.45	17.25
EDN [9]	Avg.	72.68	72.49	72.24	71.82	69.07	58.11	47.96	57.45	66.52	68.89	65.26	67.26	63.93
	σ	18.67	18.78	18.99	18.83	19.45	22.13	21.99	21.85	20.50	18.84	20.91	19.71	21.29
BASNet [10]	Avg.	80.20	77.24	74.64	71.13	65.96	50.82	14.96	57.26	53.12	54.67	67.66	74.66	66.49
	σ	21.15	23.16	23.53	26.03	27.74	33.78	29.49	32.54	26.26	29.63	28.86	25.69	29.32
PiCANet [5]	Avg.	75.81	68.16	59.38	51.12	52.00	53.61	50.08	63.31	67.94	69.34	70.16	72.06	68.65
	σ	19.80	20.74	20.17	21.48	21.54	20.64	20.33	19.69	17.24	17.53	17.29	17.25	19.19
DHSNet [15]	Avg.	68.14	66.57	65.41	63.44	61.65	60.03	57.07	64.54	62.03	64.41	65.69	66.32	65.19
	σ	21.03	21.36	21.07	20.96	21.17	21.37	21.38	20.92	21.67	20.83	20.46	20.97	21.42

the same consequences. Also, EDN and LeNo achieved poor performances, decreasing their scores by 46.44% and 38.06%, respectively. Again, due to the nature of the SNPL database images, these networks fail to maintain the robustness necessary to achieve detection. The BP algorithm only shows a 9.3% decrease in its score, demonstrating its high robustness. For the case of the AP, PiCANet manages to be surprisingly robust, going down from 86.04% to 72.06% in the regular-size patch and 82.66% in the smallest size patch, achieving a high level of detection even when we designed the patch to be another type of protruding object in the scene. On the other hand, BP manages to hold close to 40% of the patch attacks of both sizes, outperforming the BASNet and DHSNet networks. Note that BASNet scored the worst results, with 20.47% for the regular-size patch and 20.53% for the small patch. LeNo decreases its score reaching to 53.17% for the regular patch while improving up to 61.08% for the small patch, losing 24% and 12.7%, respectively. EDN maintains a similar score for both patches at around 49%, but its performance still decreased by 25%.

Noise attacks continue to show that neural networks are not immune to them. In all types of noise attacks, the BP algorithm outperforms the five neural networks, with only Speckle noise managing to lower its score to 44.98% while remaining above all networks. PiCANet and BASNet suffer huge performance drops, with Speckle noise destroying their scores almost entirely (PiCANet down to 91% and BASNet down to a total of 80%). LeNo, DHSNet, and EDN demonstrate some robustness, but not enough to outperform BP.

3.3.5 DUTS database

As a way to validate our results, we selected 300 random images from the 5019 testing pictures available in the DUTS dataset [22]. In this manner, we were able to generate a validation set that was comparable in size to prior databases. Additionally, we used the models that resulted from training on the FT database so that we could observe

how the different algorithms would behave on an unrelated collection of photos.

Table 5 shows the results of applying an adversarial FGSM attack to the five neural networks and the BP algorithm using the DUTS database. The consistency in the results obtained by the BP algorithm is worth mentioning, hovering around 50%. Similar to the results obtained with previous databases, the LeNo and EDN neural networks show resiliency for smaller gradient attacks, and then considerably deteriorate after $\epsilon = 32$ attacks. Scoring 57.1% in the original images, LeNo drops its score to 37.0% with $\epsilon = 64$, while EDN has a more prominent drop in score, starting at 58.63%, and falling down to 26.2%. BASNet has the highest score in this experiment, with 82.43%, but also starts deteriorating sooner with $\epsilon = 4$ attacks. PiCANet scores slightly better (62.25%) than LeNo and EDN, but quickly deteriorates after $\epsilon = 2$ attacks, reaching a score of 19.37% on the strongest epsilon attack. Finally, while DHSNet does not deteriorate as prominently as other neural networks, it does have the lowest score out of all algorithms, starting at 25.77%, and reaching a score of 16.42% with $\epsilon = 64$.

Table 5 also shows the results when applying the multipixel, AP, and noise attacks to the five neural networks and the BP algorithm using the DUTS database. Again, BP shows the highest robustness, with the AP causing its most significant drop in performance, going from 50.22% down to 47.66%. PiCANet, EDN, LeNo, and DHSNet show a high sensitivity against multipixel, and speckle noise attacks, while algorithms such as BASNet seem to be most affected by AP attacks.

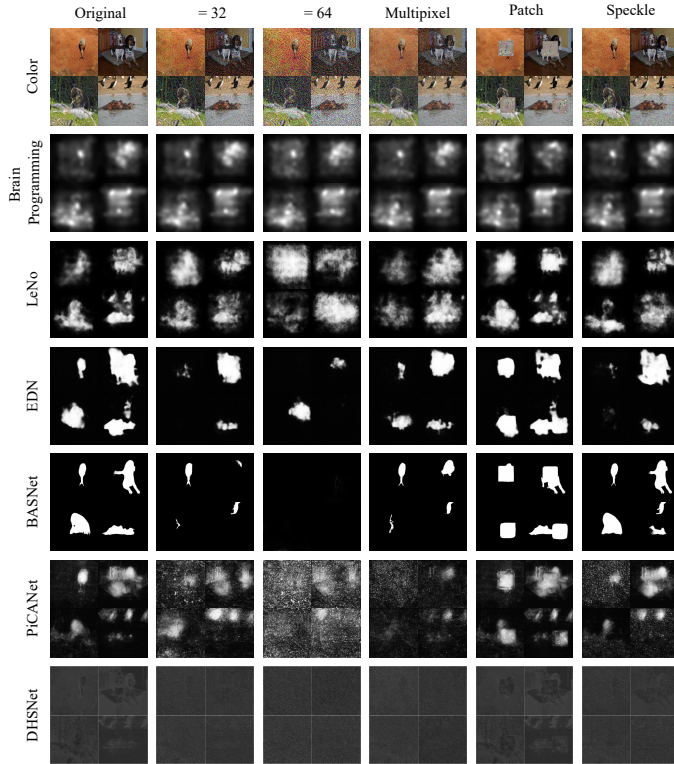
Figure 8 illustrates example saliency maps generated on the validation DUTS dataset and provides a visual explanation of the behavior observed in Table 5. We included AAs that produced the most variance across all algorithms in comparison with the original saliency maps. For reference, we also included the colored images that were passed as inputs to the various algorithms to achieve a better un-

TABLE 4: Summary of the results obtained with adversarial attacks (FGSM, multipixel, patch, and noises) applied to the SNPL database.

Algorithm	Measure	SNPL										Gaussian	S&P	Speckle
		Original	$\epsilon = 2$	$\epsilon = 4$	$\epsilon = 8$	$\epsilon = 16$	$\epsilon = 32$	$\epsilon = 64$	Multipixel	Patch	Patch(S)			
Brain [8]	Avg.	52.60	53.31	52.13	50.31	49.09	45.01	42.24	47.70	41.26	39.32	48.40	45.74	44.98
Programming	σ	19.72	19.23	19.64	20.73	21.21	20.27	20.28	21.12	21.65	19.89	20.98	21.91	21.26
LeNo [7]	Avg.	69.97	69.72	70.51	69.42	63.80	39.79	30.88	43.34	53.17	61.08	51.39	46.58	39.15
	σ	17.67	17.49	16.90	17.66	19.21	21.98	20.23	23.98	19.95	20.41	22.03	25.01	21.92
EDN [9]	Avg.	66.32	66.10	65.44	62.78	52.22	32.69	8.10	35.52	49.43	49.58	42.46	42.75	30.95
	σ	18.23	18.64	19.09	20.95	23.87	23.99	6.54	26.22	21.34	24.74	24.73	27.08	23.83
BASNet [10]	Avg.	60.27	57.36	53.88	50.66	35.28	13.63	0.99	18.09	20.47	20.53	31.27	30.63	10.01
	σ	35.12	35.40	34.43	31.85	31.80	24.72	2.05	28.11	21.30	26.71	34.50	34.73	19.84
PiCANet [5]	Avg.	86.04	72.91	63.7	57.64	44.85	9.26	8.90	13.29	72.61	82.66	32.79	26.93	7.37
	σ	14.64	23.56	25.12	25.32	22.41	8.16	5.65	12.09	25.58	21.02	22.22	23.42	6.63
DHSNet [15]	Avg.	43.98	43.23	43.04	42.12	38.63	31.67	17.84	36.13	34.76	30.01	38.92	38.19	33.60
	σ	19.88	19.46	19.37	19.88	19.57	19.48	15.77	21.02	17.76	15.59	19.42	20.86	20.01

TABLE 5: Summary of the results obtained with adversarial attacks (FGSM, multipixel, patch, and noises) applied to the DUTS database, using models trained on the FT database.

Algorithm	Measure	DUTS										Gaussian	S&P	Speckle
		Original	$\epsilon = 2$	$\epsilon = 4$	$\epsilon = 8$	$\epsilon = 16$	$\epsilon = 32$	$\epsilon = 64$	Multipixel	Patch	Patch(S)			
Brain [8]	Avg.	50.22	50.06	50.08	49.96	49.90	50.4	49.39	49.82	47.66	49.02	50.07	50.13	50.47
Programming	σ	22.23	22.32	22.33	22.24	22.42	22.63	22.40	22.47	22.25	22.39	22.27	22.36	22.37
LeNo [7]	Avg.	57.10	57.59	57.70	56.77	55.13	49.13	37.00	47.21	51.96	51.03	54.10	50.57	48.16
	σ	25.04	24.79	24.77	24.64	24.92	25.19	23.29	26.28	24.04	24.69	25.02	25.34	25.23
EDN [9]	Avg.	58.63	58.53	58.32	57.71	55.40	44.02	26.20	44.70	50.73	52.28	52.25	51.09	46.51
	σ	25.61	26.64	25.68	26.11	26.46	26.89	21.84	27.11	25.85	25.28	26.67	26.59	27.44
BASNet [10]	Avg.	82.43	81.20	76.67	67.08	53.96	42.14	29.11	63.34	45.97	55.97	71.18	75.16	67.64
	σ	24.13	24.87	28.10	31.97	33.47	30.92	23.10	30.47	26.73	29.69	28.81	27.14	30.33
PiCANet [5]	Avg.	62.25	55.86	42.06	31.59	27.48	24.17	19.37	40.62	47.71	48.29	50.61	51.06	47.34
	σ	23.64	24.81	24.55	22.15	20.75	19.33	16.64	23.87	24.34	23.63	24.94	26.30	25.79
DHSNet [15]	Avg.	25.77	25.71	25.54	24.68	21.65	17.32	16.42	17.25	23.90	24.50	18.48	19.90	18.16
	σ	17.81	17.79	17.71	17.36	16.17	14.64	14.59	14.70	16.76	17.16	14.84	15.43	14.93

Fig. 8: Saliency maps are generated on the DUTS database using algorithms trained with the FT database. Columns group results by the original image, FGSM attack with $\epsilon = 32$ and $\epsilon = 64$, multipixel, patch, and speckle noise attacks. The first row shows the original colored image with its corresponding AAs, followed by the results generated by the various algorithms.

first thing that comes to our attention is how constant the resulting saliency maps are. The cases where we can see the most distortion are in the AP, just as the table results hint. If we look carefully at the colored images, we will notice that the cases showing the biggest distortion are those where the patch fully covers the salient object. Now, the LeNo algorithm produces results that are reminiscent of smoke; it is worth noting how every single one of the attacks portrayed in the figure manages to completely change the computed output. While LeNo showed the highest resiliency out of the five networks according to the table results, these examples show that even the most robust state-of-the-art neural network does not comply with the proposed definition of robustness. In the cases of EDN and BASNet, we can observe how FGSM attacks tend to obscure the resulting saliency maps, leading to the highest loss in performance. PiCANet seems to be very susceptible to various attacks since the algorithm appears to prefer the attacked pixels of the image over the salient object. Lastly, we can observe how DHSNet is also affected by AAs, failing to produce consistent saliency maps.

3.3.6 Summary of Results

Table 6 shows the standard deviations of each of the detection models for each database. Here, it is clearly shown that the dispersion of the results obtained by the BP algorithm is low compared with those of all other neural networks. This outcome is due to the high robustness of BP, which keeps the experiments' values with AAs close to the original images. In the case of neural networks, the results deviated quite a bit, especially in the case of PiCANet for the SNPL database. Even though it scored highly in the original images, it fell apart in the attacks. DHSNet also demonstrated low dispersion in the data; however, as seen in the tables above, it is susceptible to attacks such as FGSM, so it cannot compete with BP.

derstanding of the produced outputs. Starting with BP, the

TABLE 6: Standard deviation of each detection model with respect to adversary attack experiments for each database.

Algorithm	Standard Deviation σ				
	FT	ImgSal	PASCAL-S	SNPL	DUTS
BP [8]	1.93	5.96	1.73	4.47	0.75
LeNo [7]	6.56	12.02	3.03	13.81	5.77
EDN [9]	10.63	15.15	7.32	17.10	8.93
BASNet [10]	11.09	13.74	17.14	19.39	16.20
PICANet [5]	12.04	13.82	8.90	29.76	12.94
DHSNet [15]	4.37	4.56	3.02	7.15	3.67

4 CONCLUSION

In this work, we studied SOD with an emphasis on measuring the robustness of various algorithms concerning AAs. The BP methodology was defined to detect conspicuous objects by involving a process similar to that of the artificial visual cortex in humans. After the experiments, BP proved highly robust, supporting even the highest-disturbance images. AEs were implemented, such as the FGSM, AP, Multipixel Attack, and three types of noise: Gaussian, Salt & Pepper, and Speckle. By attacking the neural networks with each of these AEs, we confirmed that they are not reliable at all. Even the slightest disturbance in the input images lowers their performance significantly, not to mention that feeding them a strong disturbance renders them virtually unusable. The advantage of BP is that it is highly robust and reliable during detection, and although a neural network achieves higher performances than the evolutionary algorithm, it can collapse with virtually any disturbance to its input data. These experiments were carried out in five databases: FT, ImgSal, PASCAL, DUTS, and SNPL. The first four contain highly detectable and unnatural objects since they cover a large part of the scene and have high contrast. Nevertheless, the SNPL database is different, as it contains objects with characteristics that resemble the real world more closely. By using this database, we can get closer to a detection problem with a practical purpose that directly helps solve a real-world problem, which is, in this case, wildlife conservation. In future research, we will focus on improving the performance of BP.

ACKNOWLEDGMENTS

The following projects support this research: 1) Project titled “Estudio de la programación cerebral en problemas de reconocimiento a gran escala y sus aplicaciones en el mundo real” CICESE-634135. 2) Project titled “Deep learning applications for computer vision task” funded by NITROAA with support of Lenovo P920 and Dell Inception 7820 workstation and NVIDIA Corporation with support from the NVIDIA Titan V and Quadro RTX 8000 GPU. 3) Project titled “Applications of Drone Vision using Deep Learning” funded by Technical Education Quality Improvement Programme (referred to as TEQIP-III), National Project Implementation Unit, Government of India. Jonathan Vargas graciously acknowledges the economic support granted by Cornell University through the program of Coastal Solutions Fellows Program. Roberto Pineda would like to thank CONACyT Mexico for the scholarship awarded to conduct studies at CICESE. Matthieu Olague would like to thank the Anáhuac

University–Querétaro for the scholarship granted to carry out his studies in Mechatronics Engineering.

REFERENCES

- [1] R. Pineda, G. Olague, G. Ibarra-Vazquez, A. Martinez, J. Vargas, and I. Reducindo, “Brain programming and its resilience using a real-world database of a snowy plover shorebird,” in *Applications of evolutionary computation*, J. L. Jiménez Laredo, J. I. Hidalgo, and K. O. Babaagba, Eds. Berlin: Springer International Publishing, 2022, pp. 603–618.
- [2] H. Li, G. Li, and Y. Yu, “Rosa: Robust salient object detection against adversarial attacks,” *IEEE Transactions on Cybernetics*, vol. 50, no. 11, pp. 4835–4847, 2020.
- [3] Z. Che, A. Borji, G. Zhai, S. Ling, J. Li, G. Guo and P. Le Callet, “Adversarial attack against deep saliency models powered by non-redundant priors,” *IEEE Transactions on Image Processing*, vol. 30, no. 1, pp. 1973–1988, 2021.
- [4] W. Wang, Q. Lai, H. Fu, J. Shen, H. Ling, and R. Yang, “Salient object detection in the deep learning era: An in-depth survey,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 6, pp. 3239–3259, 2022.
- [5] N. Liu, J. Han, and M.-H. Yang, “Picanet: Learning pixel-wise contextual attention for saliency detection,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3089–3098, 2018.
- [6] N. Akhtar and A. Mian, “Threat of adversarial attacks on deep learning in computer vision: A survey,” *IEEE Access*, vol. 6, pp. 14410–14430, 2018.
- [7] H. Wang, L. Wan and H. Tang, “LeNo: Adversarial Robust Salient Object Detection Networks with Learnable Noise,” arXiv preprint arXiv:2210.15392, 2022. accepted by AAAI 2023.
- [8] G. Olague, J. A. Menendez-Clavijo, M. Olague, A. Ocampo, G. Ibarra-Vazquez, R. Ochoa, and R. Pineda, “Automated design of salient object detection algorithms with brain programming,” 2022. *Applied Sciences* 2022, vol. 12, no. 20. Available: <https://www.mdpi.com/2076-3417/12/20/10686>
- [9] Y. -H. Wu, Y. Liu, L. Zhang, M. -M. Cheng and B. Ren, “EDN: Salient object detection via extremely-downsampled network,” in *IEEE Transactions on Image Processing*, vol. 31, pp. 3125–3136, 2022.
- [10] X. Qin, Z. Zhang, C. Huang, C. Gao, M. Dehghan, and M. Jagersand, “Basnet: Boundary-aware salient object detection,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7471–7481, 2019.
- [11] H. Kaiming, Z. Xiangyu, R. Shaoqing, and S. Jian, “Deep Residual Learning for Image Recognition,” in *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 770–778, 2016.
- [12] A. F. Agarap, “Deep learning using rectified linear units (relu),” arXiv preprint arXiv:1803.08375, 2018.
- [13] F. Visin, K. Kastner, K. Cho, M. Matteucci, A. Courville and Y. Bengio, “ReNet: A Recurrent Neural Network Based Alternative to Convolutional Networks,” arXiv preprint arXiv:1505.00393, 2015.
- [14] S. Liu and W. Deng, “Very deep convolutional neural network based image classification using small training sample size,” in *3rd IAPR Asian Conference on Pattern Recognition*, Kuala Lumpur, Malaysia, pp. 730–734, 2015.
- [15] N. Liu and J. Han, “Dhsnet: Deep hierarchical saliency network for salient object detection,” in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 678–686, 2016.
- [16] Y. Li, X. Hou, C. Koch, J. M. Rehg, and A. L. Yuille, “The secrets of salient object segmentation,” in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 280–287, 2014.
- [17] I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” in *3rd International Conference on Learning Representations*, 2015.
- [18] J. Su, D. V. Vargas, and K. Sakurai, “One pixel attack for fooling deep neural networks,” *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 5, pp. 828–841, 2019.
- [19] T. B. Brown, D. Mané, A. Roy, M. Abadi, and J. Gilmer, “Adversarial patch,” in *31st Conference on Neural Information Processing Systems*, vol. 6, 2017.
- [20] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk, “Frequency-tuned salient region detection,” in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1597–1604, 2009.
- [21] J. Li, M. D. Levine, X. An, X. Xu, and H. He, “Visual saliency based on scale-space analysis in the frequency domain,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 4, pp. 996–1010, 2013.

- [22] L. Wang, H. Lu, Y. Wang, M. Feng, D. Wang, B. Yin and X. Ruan, "Learning to detect salient objects with image-level supervision," in *IEEE Conference on Computer Vision and Pattern Recognition*, Honolulu, HI, USA, pp. 3796–3805, 2017.
- [23] M. S. Norouzzadeh, A. Nguyen, M. Kosmala, A. Swanson, M. S. Palmer, C. Packer, and J. Clune, "Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 115, no. 25, pp. E5716–E5725, 2018.
- [24] K. E. Doull, C. Chalmers, P. Fergus, S. Longmore, A. K. Piel, and S. A. Wich, "An evaluation of the factors affecting 'poacher' detection with drones and the efficacy of machine-learning for detection," *Sensors*, vol. 21, no. 12, 2021.
- [25] S. Kahl, C. M. Wood, M. Eibl, and H. Klink, "Birdnet: A deep learning solution for avian diversity monitoring," *Ecological Informatics*, vol. 61, p. 101236, 2021.
- [26] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimeshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, pp. 8024–8035, 2019.



Gerardo Ibarra was born in Ciudad Victoria Tamaulipas in 1985. He received the B.S. degree in communications and electronics engineering from the Universidad Autónoma de Tamaulipas, México, in 2008, and the M.Sc. and Ph.D. degrees in electronics engineering from Universidad Autónoma de San Luis Potosí, México, in 2010 and 2016, respectively. He is currently a postdoctoral researcher at the Institute for the Future of Education of Tecnológico de Monterrey, México. His lines of research are robustness evaluation of image classification models, adversarial attacks in deep learning, deep learning in image processing, evolutionary algorithms for image classification, protruding object detection and interest point detectors.



Matthieu Olague was born in Ensenada, Baja California, México in 2000. He received the B.S. degree in Mechatronics from Universidad Anáhuac Querétaro, México in 2022. Matthieu is a member of the directive board & CTO of Intra. He leads a team of talented engineers constructing an ever-growing digital ecosystem, helping companies harness the power of digital. Before Intra, Matthieu developed software solutions and applications for various Mexican companies in collaboration with Data Pulse Analytics. In his spare time, Matthieu enjoys researching subjects related to Computer Vision and Machine Learning.



Gustavo Olague (M'99–SM'17) received the B.S. and M.S. degrees in Industrial and Electronics Engineering from ITCH (Instituto Tecnológico de Chihuahua) in 1992 and 1995 respectively, and his Ph.D. in Computer Vision, Graphics, and Robotics from INPG and INRIA in France in 1998. He is a Professor in the Dept. of Computer Science at CICESE in México. He has authored over 150 conference proceedings papers and journal articles and co-edited special issues in Pattern Recognition Letters, Evolutionary Computation, and Applied Optics. Prof. Olague received numerous distinctions, among them the Talbert Abrams Award presented by the American Society for Photogrammetry and Remote Sensing (ASPRS); Outstanding Associate Editor IEEE Access 2021; Best Paper Awards at major conferences; and twice the Bronze Medal at the Humies (GECCO award). Gustavo Olague is the author of the book *Evolutionary Computer Vision* published by Springer in the Natural Computing Series.

ary Computation, and Applied Optics. Prof. Olague received numerous distinctions, among them the Talbert Abrams Award presented by the American Society for Photogrammetry and Remote Sensing (ASPRS); Outstanding Associate Editor IEEE Access 2021; Best Paper Awards at major conferences; and twice the Bronze Medal at the Humies (GECCO award). Gustavo Olague is the author of the book *Evolutionary Computer Vision* published by Springer in the Natural Computing Series.



Axel Martinez was born in Distrito Federal, México, in 1989. He received his B.S. degree in Mechatronics Engineering from Universidad Anáhuac México Norte in 2017 and M.Sc. degree in Computer Science from CICESE in 2019. He is pursuing Ph.D. degree in Computer Science, and working at EvoVisión Laboratory at CICESE. His research interests are Computer Vision, Genetic Programming, Evolutionary Algorithms, 3D Reconstruction, and he enjoys surfing and cooking.



Roberto Pineda was born in Ensenada Baja California in 1991. He received the B.S. degree from CETYS Universidad, México, in 2015, and the M.Sc. degree in computer science from CICESE, México, in 2022. Roberto works as software engineer at ONEIL. His areas of interest include animal conservation through science, particularly bird conservation through salient object detection algorithms and the positive impact these could have on vulnerable bird species.



Sambit Bakshi is currently with the Department of Computer Science and Engineering, National Institute of Technology Rourkela, India. His area of interest includes surveillance, biometric security, and digital forensics. He served as Associate Editor of IEEE Access and Expert Systems in the past. He presently serves as associate editor of Innovations in Systems and Software Engineering Springer: A NASA Journal, Expert Systems with Applications, Image and Vision Computing, Multimedia Systems, and IEEE IT Professional magazine. He is a senior member of IEEE. He is Founding Chair of IEEE Rourkela Subsection. He presently serves as a member of IEEE Computational Intelligence Society Young Professionals Subcommittee since 2020 (liaison for IEEE Young Professional for the year 2022). He is a member of Professional Activities Committee of IEEE Region 10 for the year 2022. He has published widely in more than 100 journals and conferences.



Jonathan Vargas was born in Nayarit in 1984. He received the B.S. degree in Biology from Universidad Autónoma de Nayarit, México, in 2012, and the M.Sc. degree in Marine and Coastal Sciences from Universidad Autónoma de Baja California Sur, México, in 2017. He is a Mexican biologist, conservationist and shorebird specialist and is based at Terra-Peninsular – an environmental NGO focused on conserving the biodiversity of the Baja California Peninsula.

Jonathan has spearheaded the conservation of Todos Santos Bay, on the Pacific Coast of Baja California, where he started a shorebird-monitoring project that achieved the recognition of the bay as a site of regional importance for shorebirds and other migratory birds.



Isnardo Reducindo was born in México City in 1985. He received the B.S. degree in communications and electronics engineering from the Universidad Autónoma de Zacatecas, México, in 2008, and the M.Sc. and Ph.D. degrees in electronics engineering from Universidad Autónoma de San Luis Potosí, México, in 2010 and 2016, respectively. He is currently a Professor with the Information Science Faculty of Universidad Autónoma de San Luis Potosí, México. His research interests include artificial intelligence in

document management, digital humanities, technology in education, and information processing and management.