UNIVERSITY OF GUANAJUATO

IRAPUATO - SALAMANCA CAMPUS
ENGINEERING DIVISION

Automatic Tuning of Machine Learning Systems for Novelty Detection

THESIS

TO OBTAIN THE DEGREE OF:
DOCTOR OF ELECTRICAL ENGINEERING

PRESENTED BY:

M.Eng. Marco Antonio Contreras Cruz

SUPERVISORS:

Dr. Juan Pablo Ignacio Ramírez Paredes
Dr. Víctor Ayala Ramírez

SALAMANCA, GTO. October 2020

Salamanca, Gto., a 2 de octubre del 2020.

M. en I. HERIBERTO GUTIÉRREZ MARTIN
JEFE DE LA UNIDAD DE ADMINISTRACIÓN ESCOLAR
PRESENTE.-

Por medio de la presente, se otorga autorización para proceder a los trámites de impresión, empastado de tesis y titulación al alumno(a) Marco Antonio Contreras Cruz del ***Programa de Doctorado en*** Ingeniería Eléctrica y cuyo número de ***NUA*** es: 143709 del cual soy director. El título de la tesis es: Automatic Tuning of Machine Learning Systems for Novelty Detection

Hago constar que he revisado dicho trabajo y he tenido comunicación con los sinodales asignados para la revisión de la tesis, por lo que no hay impedimento alguno para fijar la fecha de examen de titulación.

ATENTAMENTE

Dr. Juan Pablo Ignacio Ramírez Paredes NOMBRE Y FIRMA ***DIRECTOR DE TESIS*** **SECRETARIO**	**Dr. Víctor Ayala Ramírez** NOMBRE Y FIRMA ***DIRECTOR DE TESIS***
Dora **Dra. Dora Luz Almanza Ojeda** NOMBRE Y FIRMA **PRESIDENTE**	**Dr. Carlos Hugo García Capulín** NOMBRE Y FIRMA **VOCAL**
Becerra **Dr. Héctor Manuel Becerra Fermín** NOMBRE Y FIRMA **VOCAL**	Rafael Murrieta **Dr. Rafael Eric Murrieta Cid** NOMBRE Y FIRMA **VOCAL**

Acknowledgment

- To **God**, for giving me strength and support to overcome all the obstacles.
- To my mother **Gricelda**, for her help during all my life.
- To my father **Jose Antonio**, for all his support and being an example of responsibility.
- To my wife **Azucena**, my love, you are the reason I want to be a better person, thank you for everything. Every day, I love you more.
- To my sisters (**Andrea**, **Nayeli**), brother (**Juan**), nephews (**Angel**, **Jaffet**, **Mateo**) and niece (**Sofía**), thank you for your support and patience.
- To my whole family (**Tita**, **all my aunts and uncles**, **Grandma Maria**, **Doña Leyva**, **Don Montes**), thank you for being there when I need you.
- To my adopted daughter (**Sophia**), thank you for turning the quarantine into something fun. I love you, baby.
- To my sister-in-law **Esmeralda** for her strength and happiness. Keep dreaming!
- To my **supervisors**, Victor Ayala, Juan Pablo Ramirez, and Gabriela Ochoa for your advice and good talks.

Institutional Acknowledgments

UNIVERSIDAD DE
GUANAJUATO

To the University of Guanajuato, and to its Engineering Division at the Campus Irapuato-Salamanca, for the academic training provided by the Electrical Engineering Doctoral program.

To the Office of Research and Graduate Programs (Dirección de Apoyo a la Investigación y al Posgrado) DAIP, of the University of Guanajuato, for the financial support during my research visit to the University of Stirling (from June to December 2019), and the financial support for presenting the work Distributed Path Planning for Multi-Robot Teams based on Artificial Bee Colony in the Congress on Evolutionary Computation 2017.

This work was supported by the grant 237/2020 from the Institutional Awards for Scientific Research (Convocatoria Institucional de Investigación Científica 2020) of the University of Guanajuato, as part of the project Anomaly Detection in Aerial Images Using Adversarial Neural Networks (Detección de Anomalías en Imágenes Aéreas con Redes Neuronales Adversarias).

To the National Council of Science and Technology (Consejo Nacional de Ciencia y Tecnología) CONACYT, for the financial support provided during the doctoral program and during my research visit to the University of Stirling (Scotland), through the scholarship 568675/302121.

To the Secretary of Innovation, Science, and Higher Education (Secretaría de Innovación, Ciencia y Educación Superior) SICES, for the financial support in the project “Development of learning object for higher education and industrial training” of the program “ Support for Graduate Entrepreneurs 2017 (Apoyo a Emprendedores de Posgrado 2017) SICES/CON/200/2017”.

To the Laboratory of Vision, Robotics, and Artificial Intelligence (Laboratorio de Visión, Robótica e Inteligencia Artificial) LaViRIA, for the resources and support provided during the development of this research project.

Automatic Tuning of Machine Learning Systems for Novelty Detection

by

Marco Antonio Contreras Cruz
ORCID: https://orcid.org/0000-0001-8466-5787

Abstract

Novelty detection is the ability to identify test data that are different in some aspects to the usual normal data. This ability is particularly useful for applications such as fraud detection, failure detection, medical diagnosis, video surveillance, selective learning, and obstacle detection in robotics, among others. Novelty detection often uses machine learning algorithms, where their hyperparameter values define their performance. Therefore, it emerges the need for automatic configuration techniques. This thesis shows the application of automatic design configuration tools to novelty detection problems and related areas. We showed the advantage of these tools in two study cases. In the first case, we adopted the artificial bee colony algorithm for tuning novelty detectors in robotics, specifically the grow-when-required neural networks and the simple evolving connectionist systems. We trained the novelty detectors in an outdoor environment with images captured by an unmanned aerial vehicle. Then, we added some objects to the environment that should be detected as novel objects. Under this setup, we explored the performance of traditional visual features such as color histograms, GIST descriptors, and color angular indexing. We also proposed using the pre-trained MobileNetV2 as a feature extractor. Our results showed the benefits of using tuned novelty detectors with the features extracted by the MobileNetV2. Our second study proposed an automatic design methodology based on genetic programming to select and combine saliency detection algorithms using fuzzy logic combination rules, morphological operations, and image processing filters. Saliency detection is strongly related to novelty localization in images. We perform two experiments: first, we combine the response of some of the faster algorithms in the state-of-the-art, the Minimum Directional Contrast, the Minimum Barrier Salient Object Detection, the Frequency-Tuned approach, and the Histogram-based Contrast; and the second set includes four machine learning algorithms, the Deep Hierarchical Saliency Network, the Discriminative Regional Feature Integration Approach, the Dense and Sparse Labeling, and the Local Estimation and Global Search. The improvements offered by the combination models are demonstrated by comparing their performance against several state-of-the-art saliency detection methods, several classic combination models, and other evolutionary computation-based approaches on four benchmark datasets. The results were analyzed using two statistical tests: the Wilcoxon rank-sum test and the t-test. Both tests confirmed that the proposed method outperforms all of the other algorithms under test and that its performance advantage is statistically significant.

Contents

CHAPTER 1

Introduction

This thesis focuses on the automatic configuration of machine learning techniques applied to novelty detection problems. We apply the automatic configuration concepts in two study cases: the automatic tuning of online machine learning approaches for visual novelty detection and the automatic selection and combination of salient object detectors. Saliency detection is strongly related to novelty localization in images. The proposed automatic design techniques are based mainly on metaheuristics and evolutionary computation techniques.

We explore two online machine learning algorithms and different visual features extracted from images captured by an unmanned aerial vehicle for the visual novelty detection problem. We used a metaheuristic-based optimization algorithm to tune the hyperparameters of the novelty detectors. In the salient object detection problem, we propose a framework for selecting and combining the outputs of different algorithms, which include machine and non-machine learning approaches. The automatic design tool used an evolutionary computation technique.

In this chapter, we present our problem statement and its motivation; we introduce the novelty detection problem; we describe our research objectives and the target applications of this study, and finally, we show the summary of this chapter and the thesis organization.

1.1 Problem statement

The novelty detection problem has been addressed by machine learning algorithms, where a model of normality is built, and then the model is used to identify novel data. Despite the success of machine learning algorithms in classification problems, their application in new problems or new instances of the same problem is not a trivial task. The use of machine learning algorithms requires highly specialized human designers to generate high-performance machine learning applications. Many times, this implies the proper selection of the parameters of the algorithms. However, some difficulties emerge due to the wide range of design selections and the lack of orientation to obtain good selections.

Instead of creating new algorithms, recent advances have focused on studying existing algorithms and their parameters to reach higher performances. Automatic algorithm configuration is one of these areas. Automatic algorithm configuration techniques aim to reduce human designer intervention and exploit the current capabilities of computers to search for the best parameters of the algorithms. In this study, we explore automatic algorithm configuration techniques applied to novelty detection problems.

1.1.1 Motivation

The development of autonomous robots with capabilities to learn and to execute new tasks by themselves is one of the significant challenges in robotics. This characteristic would be especially useful in dangerous environments where humans can be at risk, some repetitive tasks where the fatigue can reduce the efficiency of human operators, and when it is required precision capabilities in which human operators can not satisfy the required quality.

Novelty detection can provide useful information to robots in order to achieve autonomy. In general, novelty detection consists of adding unobserved data to enrich the knowledge of the robot about the environment, i.e., the robot performs selective learning of the environment. In robotics, novelty can detect obstacle never before seen by the robot, or as stimuli to adapt its behavior in newly discovered environment conditions.

The novelty detectors are often machine learning algorithms with hyperparameters that alter their performance. Therefore, the need arises for techniques to select their hyperparameters appropriately. The use of automatic design tools helps the robot to keep its autonomy in this configuration process.

1.1.2 Novelty detection

Novelty can be defined as stimuli that differ from the usual stimuli. This ability is especially useful for animals because they can use novelty as a tool for survival, e.g., novelty can represent some potential predators or preys [1].

From the computational point of view, novelty detection consists of finding data that are different in some aspects to the known data [2]. It is a challenging problem because datasets may have many examples of the normal class, an insufficient number of examples of the novel class or no novel samples whatsoever. The area of novelty detection has gained much interest, especially in areas such as fraud detection [3], fault detection [4], medical diagnosis [5, 6], video surveillance [7], and robotics [8, 9].

In robotics, researchers have incorporated modules of novelty detection into robots to increase their degree of autonomy through providing stimulus for future actions or adding new information in selective learning of the environment. From this approach, the novelty detection implies that the robot is trained to ignore perceptions that are similar to the observed perceptions in training, such that only perceptions that are different are highlighted (potential problems, obstacles, or something new to learn). In the literature, we can find novelty detection applications in robotics in a wide variety of domains, for example classification [1], segmentation of three-dimensional maps [10], exploration and sensory fusion [8], the design of architectures based on novelty detection [11], safe human-robot interaction [12], obstacle detection in vision systems [9], among others.

1.2 Research objectives

The general objective of this thesis is to incorporate automatic configuration techniques into existing machine learning algorithms applied to novelty detection problems and related areas. The particular objectives of this study are shown as follow:

- To perform tuning of machine learning algorithms for visual novelty detection in robotics.
- To summarize the machine learning algorithms applied for novelty detection in robotics.
- To test and to evaluate the performance of the selected novelty detection algorithms.
- To contrast the effect of traditional visual features used by novelty detectors.

- To propose a new visual feature in novelty detection to increase the performance of novelty detectors.
- To develop a new methodology based on automatic design techniques to select and combine different saliency localization algorithms.
- To explore existing algorithms for saliency localization and to identify their weakness.
- To propose an automatic design technique to generate combination models that outperform existing saliency detection algorithms.
- To compare different ways to generate combination models automatically.
- To compute the effect of changing the input algorithms and fitness function into the automatic design technique.
- To provide an overview of the performance of the generated models compared with other techniques in images of the same and different domains.

1.3 Contributions

The main contributions of this study are:

- A review of automatic algorithm configuration techniques for machine learning algorithms. Among these techniques, we propose to use the artificial bee colony [13] and genetic programming [14] to explore promising regions of well-performing configurations.
- A review of existing novelty detectors for robotic applications. We adopted two novelty detectors for continuous learning, the grow-when-required neural network [15] and the simple evolving connectionist systems [16].
- We applied the artificial bee colony to tune the grow-when-required neural network and the simple evolving connectionist systems for visual exploration and inspection tasks.
- As far as we know, this is the first time that evolving connectionist systems or grow-when-required networks are applied in unmanned aerial vehicles for detecting novelties in outdoor environments

- A comparison between traditional visual features against the pre-trained MobileNetV2 [17] is performed in visual novelty detection.

- According to our knowledge, we used genetic programming for the first time to automatically generate combination models for salient object detection. Our genetic programming framework evolves combination models by using fuzzy logic operations and image processing filters embedded into it.

- We performed a study with different input algorithms and two fitness functions into genetic programming to compute their effect in the generated combination models. The first input set includes the Minimum Directional Contrast [18], the Minimum Barrier Salient Object Detection [19], the Frequency-tuned approach [20], and the Histogram-based Contrast [21]. In contrast, the second set includes four machine learning algorithms, the Deep Hierarchical Saliency Network [22], the Discriminative Regional Feature Integration Approach [23], the Dense and Sparse Labeling [24], and the Local Estimation and Global Search [25].

- We compared the automated generated models against existing saliency detection models with images of the same domain and different domains. The proposed models prevailed over preceding approaches in several performance estimation tests.

1.4 Development

In the thesis development, we focus on the automatic design of machine learning algorithms applied to novelty detection and related areas. Figure 1.1 shows the general methodology of this work. Given the target algorithm(s) with their configuration spaces, a dataset, and features extracted from the dataset, the objective is to determine the best hyperparameters of the target algorithm (hyperparameter optimization) and to find the best combination of algorithms (automatic selection and combination). A task called combined algorithm selection, and hyperparameter optimization (CASH) can be derived from this methodology, but this task was not addressed in this thesis. It is worth mentioning that the domain knowledge and the parameter values of the automatic configuration technique can improve the search process.

We applied this methodology in two study cases. The first case uses the artificial bee colony algorithm to tune the hyperparameters of the grow-when-required neural network, and the simple evolving connectionist systems to detect visual novelties in an outdoor environment. The second case applied genetic programming to select and combine saliency detection algorithms automatically. As follow, we describe in more detail both study cases.

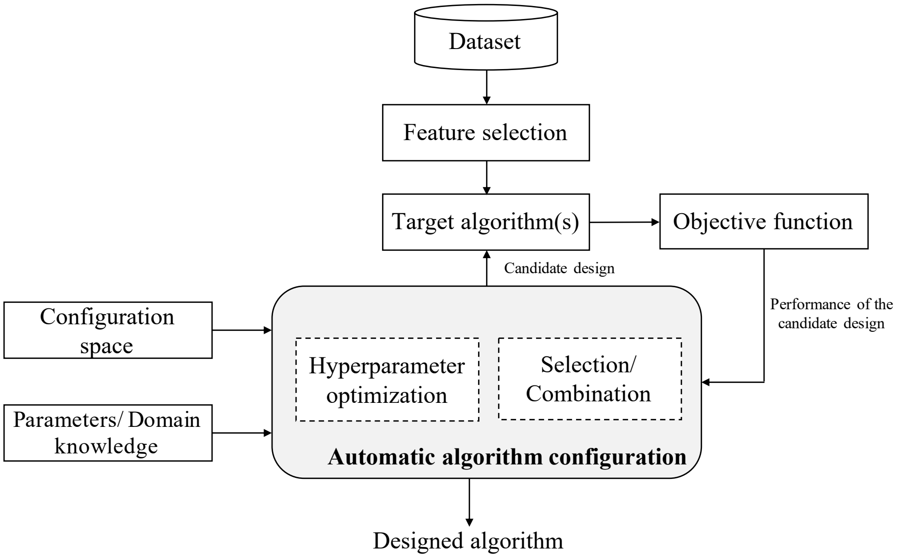


Figure 1.1: General methodology used for the automatic design of machine learning algorithms in this work.

1.4.1 Tuning online novelty detectors for specific visual exploration and inspection tasks

We propose a visual novelty detection framework based on the artificial bee colony algorithm to tune the grow-when-required neural network and the simple evolving connectionist systems. The tuned systems are used for specific exploration and inspection tasks in an outdoor environment, where the images were captured by an unmanned aerial vehicle. This study explores traditional visual features such as RGB color histograms, angular color indexing, and the GIST descriptor in the visual exploration and inspection task. We compared those traditional descriptors against the emerging area of pre-trained convolutional networks as a feature extraction techniques, specifically the MobileNetV2. The results of this work are reported in the journal article [26] and discussed in detail in Chapter 3.

1.4.2 Automatic design of combination models for salient object detection

The contribution of this work is oriented towards the automatic combination and selection of saliency detection algorithms by using genetic programming. Our approach uses a set of candidate saliency detection algorithms and a set of combination operators. The input saliency detection methods include algorithms from the state-of-the-art: machine and non-machine learning algorithms. As a domain knowledge, we include a set of combination operators typically used for the task, such as fuzzy logic combination rules, morphological operations, and image processing filters. An advantage of the proposed approach is that the output models explain and give insight into which standalone methods are essential to improve the response to the saliency detection problem. The results of this work were reported in the journal article [27] and discussed in detail in Chapter 4.

1.5 Summary

Novelty detection is a fundamental problem in a wide variety of research areas. The problem consists in recognizing data that are different in some aspect to the already known data. This ability can be useful for tasks where data only describes normal behavior, and it is not very easy to obtain novelty data for training. Some novelty detection applications can be found in medicine, fraud and fault detection, video surveillance, and robotics.

For robotics, novelty detection is particularly useful in autonomous robots because novelty can be used as stimuli to perform future action, to develop selective learning, and to identify obstacles, potential problems, or something new to learn.

As novelty detection approaches commonly use machine learning algorithms, some problems arise with their use. One of them is the proper selection of its hyperparameters. In this thesis study, we proposed using automatic design concepts into novelty detection and related areas.

We applied a bio-inspired optimization technique, called the artificial bee colony algorithm, to tune online novelty detectors for visual novelty detection in the first study case. In the second study, we proposed a genetic programming framework to automatically combine saliency detection algorithms (which include machine and non-machine learning approaches). Saliency detection is strongly related to novelty localization in images.

1.6 Thesis organization

Chapter 2 presents the related work and background in the problem of automatic design and novelty detection.

Chapter 3 describes our methodology and the results of the system for visual novelty detection. The system uses captured images by an unmanned aerial vehicle in an outdoor environment. Besides, the chapter describes the two selected online novelty detectors and the metaheuristic used to tune the behavior of the machine learning techniques.

Chapter 4 shows our framework to design combination models of saliency detection algorithms automatically. This framework uses genetic programming to select and combine saliency algorithms. Saliency detection has been used previously as a way to localize novelty regions in images.

Chapter 5 shares our main observations and conclusions. We additionally present the perspectives in this research. Finally, this chapter presents our scientific products generated during the development of this research project.

CHAPTER 2

Background

This chapter describes the problem of automatic algorithm configuration and the available tools to tune existing algorithms. It also presents the novelty detection problem, related areas, and existing methods. Finally, some works that have linked both areas are described.

2.1 Automatic algorithm configuration

Traditionally, the design and development of algorithms have been addressed manually, guided by personal experience and intuition, rule of thumb, and brute force [28, 29]. This task is undoubtedly time-consuming, fatiguing, and hard to solve when the user does not have in-depth knowledge about the algorithms.

Automatic algorithm configuration moves the design and development of algorithms from human jobs to automatic tasks that explode the current capabilities of computers. This area aims to explore complex configuration spaces by identifying the best settings of an algorithm for a given problem. Automatic algorithm configuration has recently been successfully applied for metaheuristic [28] and machine learning algorithm design [30, 31].

2.1.1 Algorithm parameters

The performance of algorithms is strongly related to their parameter values (called hyperparameters in the machine learning context); we can rarely find a parameter-free algorithm. From the automatic perspective, the design and development of an algorithm involve selecting parameters of different types [28]. The components of the algorithms can be modeled by categorical parameters, e.g., the type of kernel in the support vector data description. The parameters can be ordinals when there is a notion of order but not a notion of distance, e.g., {cold, warm, hot}. The parameters also model numeric variables, which include real and integer values. Besides, some numeric parameters can emerge in depends on the value of other parameters; these parameters are called conditional, e.g., in support vector machines with a polynomial kernel, it emerges the parameter to define the degree. Once we identified the parameters, it is convenient to establish some regions where the automatic tool should explore. In categorical and ordinal parameters, the alternatives should include the whole set, while numeric parameters should have a range with a minimum and maximum values.

2.1.2 Automatic algorithm configuration problem

The automatic algorithm configuration problem can be formulated following [28]. Let $\lambda = (\lambda_1, \ldots, \lambda_{N_p}) \in \Lambda$ be a vector of parameters that describe an algorithm, where Λ denotes the configuration space. Each parameter λ_i has an associated type t_i and a domain r_i. Let $F_{\mathcal{I}}(\lambda) : \Lambda \to \mathbb{R}$ be the objective function to measure the performance of λ in a set of instances obtained from the instance distribution $\mathcal{I}$. The automatic configuration problem is then defined as,

$$\lambda^* = \arg\min_{\lambda \in \Lambda} F_{\mathcal{I}}(\lambda) \tag{2.1}$$

$F_{\mathcal{I}}$ can be stochastic because the algorithm can incorporate stochastic decisions, or the sampling of instances can also be stochastic. Estimating $F_{\mathcal{I}}$ is computationally expensive, particularly in large datasets, because it requires training a model by using the parameters and then testing the performance on a validation set [30]. Also, in some cases, to avoid overfitting, the above function includes cross-validation [32].

Figure 2.1 shows a graphical description of the problem and how the automatic configuration tool communicates with the algorithm to be tuned.

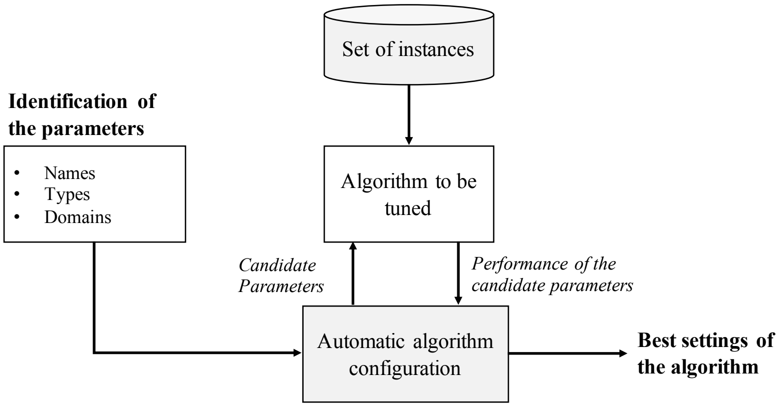

Set of instances
Identification of the parameters
Names
Types
Domains
Algorithm to be tuned
Candidate Parameters
Performance of the candidate parameters
Automatic algorithm configuration
Best settings of the algorithm

Figure 2.1: Generic description of the automatic algorithm configuration problem, adaptation from [28].

2.1.3 Automatic configuration tools

A wide variety of automatic configuration tools have been proposed in the literature [28, 32]. The classical grid search is an exhaustive way to explore the configuration space of an algorithm. It creates a grid of configurations and then evaluates all of them. The main drawback of this approach is that the number of evaluations of the objective function grows exponentially with the number of parameters. A straightforward way to reduce the number of configuration evaluations is the use of random search. Random search approaches explore configurations by selecting at random each parameter independently. Although these approaches converge faster than grid-based approaches, they explore many configurations because they do not take advantage of the knowledge about well-performing configuration regions [31]. Gradient descent can be used for some particular configuration problems, where the objective function is differentiable. However, when there is no analytic way to find the derivative of the function, the derivative can be estimated by finite differences, but this could be problematic because, in the context of automatic design, it is computationally expensive to evaluate even a single configuration [30].

An alternative to avoid the explicit computation of derivatives and to take advantage of promising regions of well-performing configurations are the derivative-free optimization algorithms, e.g., surrogate approaches, bio-inspired algorithms, and racing approaches. In the surrogate approach, the objective function is complemented with a surrogate model. This model is used to predict the performance of a configuration without evaluating the computationally expensive objective function. Examples of this approach include Bayesian optimization techniques [29]. Bio-inspired algorithms are applicable for various optimization problems because they do not assume any property about the objective function. They have been successfully applied for automatic algorithm configuration problem [33]. These techniques use the simulation of the evolution in nature or the collective behavior of some social species as ants, bees, birds, and others. In automatic design, each technique uses a particular mechanism to generate a population set of candidate configurations. During the search, the mechanism for each technique tries to improve the quality of candidate configurations until they reach a termination criterion. Bio-inspired techniques include (*i*) evolutionary computation techniques, e.g., genetic algorithms, evolutionary programming, evolution strategies, and genetic programming; and (*ii*) swarm intelligence, e.g., particle swarm optimization, ant colony optimization, firefly algorithm, cuckoo search, artificial bee colony, among others. Some other approaches use racing to find the best configuration of an algorithm. The irace package implements several iterative racing methods [34]. In iterative racing methods, the configurations are sampled from a probabilistic model, then the configurations are discarded based on statistical tests, and finally, the probabilistic model is updated based on the well-performing configurations.

2.1.4 Related areas

Different communities have addressed the algorithm configuration problem. For example, in the machine learning community, automated machine learning aims to design and develop complete machine learning pipelines automatically [32]. In metaheuristics, the automatic algorithm configuration has been used for fine-tuning and to construct new metaheuristics [28]. Another field in optimization related to automatic design is hyper-heuristics [35]. A hyper-heuristic is a search method or learning mechanism used to select or generate heuristics to solve challenging computational search problems. It explores heuristics in the search space instead of directly searching in the solution space.

2.2 Novelty detection problem

Novelty detection is the ability to identify test data that differ in some aspects to the usual data [2]. The novelty detection problem is posed as a one-class classification problem, where the objective is to differentiate the positive (normal) class from the negative (novel) class [36]. Let $\mathbf{X} = \{\mathbf{x}_i | \mathbf{x}_i \in \mathbb{R}^D, i = 1, \ldots, N\}$ be the normal training set, the goal is to construct a model $h(\mathbf{x}|\mathbf{X}, \gamma)$ to represent the normal data,

$$\begin{aligned} h(\mathbf{x}|\mathbf{X},\gamma) &= I(f(\mathbf{x}|\mathbf{X},\gamma) < \theta) \\ &= \begin{cases} 1, & \text{if } \mathbf{x} \text{ is classified as normal} \\ 0, & \text{if } \mathbf{x} \text{ is classified as novel} \end{cases} \end{aligned} \tag{2.2}$$

where $\mathbf{x} \in \mathbb{R}^D$ is a test point, I is an indicator function, f is a dissimilarity function, θ is the novelty threshold, and γ is the complexity of the method, e.g., the number of neighbors in the k-nearest neighbor algorithm, or the number of clusters in the k-means algorithm.

In this representation, f estimates the dissimilarity between the test point $\mathbf{x}$ and the target data $\mathbf{X}$. When f represents the value of similarity, the sign should be opposite ($>$). The user can specify the value of the novelty threshold θ or select a fraction of rejected target points ϵ. If ϵ is specified and f estimates dissimilarity, the optimal threshold can be found by solving

$$\begin{aligned} &\min \theta \\ &\text{s.t. } \frac{1}{N}\sum_{i=1}^{N} I(f(\mathbf{x}|\mathbf{X},\gamma) \geq \theta) = \epsilon \end{aligned} \tag{2.3}$$

2.2.1 Novelty, outlier, and anomaly detection

Novelty detection is also referred to as outlier detection and anomaly detection in some contexts. These terms appear to have the same meaning, and the methods for these problems are often shared. Novelty, anomaly, and outlier detection share the definition of finding patterns different from normal patterns [37]. However, on some occasions, the terms do not necessarily reflect the same concept.

Both anomalies and outliers often refer to undesired patterns, irregularities, noise events, artifacts, malicious activity, instrumentation error, change in the environment, and human errors [38, 39]. Gogoi et al. [40] also define outliers as aberrant data, which may affect the system in such a way that it produces incorrect results. In the above context, the novelty seems to be different. Novelty patterns are typically incorporated into the model after their detection, while the model discards outliers. Markou and Singh [41] defined outliers as a small fraction of normal data far away from the normal data in feature space, i.e., training data contains outliers. The outlier detection methods aim to discard the outliers and focus in more concentrated regions of normal data. From this perspective, it is clear that novelty is entirely different because novelty represents unobserved data during training.

In this thesis study, we consider novelty as identifying data that do not fit to the target data distribution, taking into account that the training set does not contain outliers. Besides, in our study, novelty represents new unobserved data during training, and it can be incorporated into the model to perform continuous learning.

2.2.2 Methods for novelty detection

Novelty detection can be solved by using semi-supervised and unsupervised methods. Semi-supervised methods assume that novel points are available during training to convert the novelty detection problem into the classical two-class classification problem. Then, any standard multi-class classifier can be applied to solve the problem. The first approach of semi-supervised methods consists of labeling each point in the training set and applying a two-class classifier on the labeling dataset [42]. The second approach generates artificial outliers according to a prior distribution. The outliers are positioned in regions where normal data are absent, isolated, or close to the boundaries [43, 44]. The success of artificial outlier generators depends on the quality of the outliers, and it is hard to generate outliers close to the boundaries without overlapping, particularly in high-dimensional problems [45]. Therefore, this study focuses on unsupervised approaches, which only require normal data during training.

Unsupervised novelty detection techniques are classified into five categories [2]: probabilistic, distance-based, reconstruction-based, domain-based, and information-theoretic techniques. As follow, we describe the five approaches; however, the readers are referred to review [2] for a more detailed description.

Probabilistic approaches

Probabilistic approaches estimate a model of the normal class density, and regions with higher density indicate a high probability of containing normal objects. This model is then used with a novelty threshold to decide if a test point is normal or novel. There are two families of probabilistic approaches for novelty detection: parametric and non-parametric methods.

Parametric methods use probability density functions to construct the underlying normal distribution model, such as Gaussian and mixture of Gaussians [46]. In the mixture of Gaussians, the target class is modeled by using k different Gaussians,

$$f_{mg}(\mathbf{x}) = \sum_{i=1}^{k} p_i \, g(\mathbf{x}|\mu_i, \Sigma_i) \tag{2.4}$$

where p_i are the mixture weights, and $g(\mathbf{x}|\mu_i, \Sigma_i)$ are the component Gaussian densities.

The classification of the test point $\mathbf{x}$ is defined as,

$$h_{mg}(\mathbf{x}) = \begin{cases} \text{normal}, & \text{if } f_{mg}(\mathbf{x}) \geq \theta \\ \text{novel}, & \text{if } f_{mg}(\mathbf{x}) < \theta \end{cases} \tag{2.5}$$

Standard optimization algorithms, such as Expectation-Maximization, can optimize the mixture of Gaussians (p_i, μ_i, and Σ_i). Figure 2.2 shows a mixture of Gaussians with $k = 4$ for modeling synthetic data. The gray points represent the center of the Gaussians, and the solid black lines represent the decision boundary by using a rejection rate of the target class $\epsilon = 0.01$.

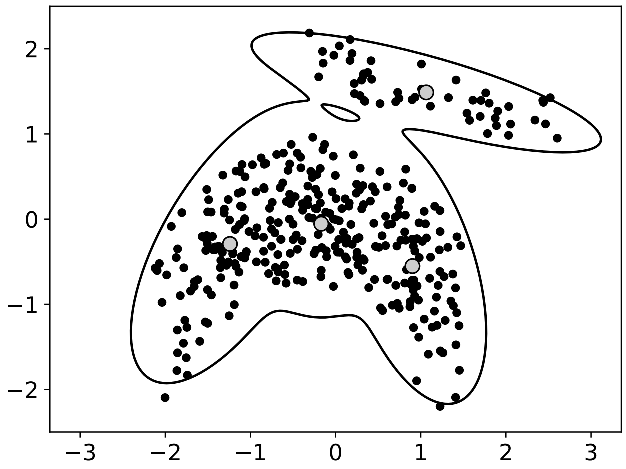


Figure 2.2: Mixture of Gaussians for novelty detection with four components ($k = 4$) on a subset of the banana dataset. Black points, gray points, and solid black lines represent the training data, the centers of the Gaussians, and the boundary ($\epsilon = 0.01$), respectively.

Non-parametric approaches use kernel density estimators, and they often refer to Parzen windows [9]. Parzen window is a more flexible density estimator because the probability density function uses a kernel function around each point in the dataset,

$$f_{parzen}(\mathbf{x}) = \frac{1}{N\beta} \sum_{i=1}^{N} K\big(\frac{\mathbf{x} - \mathbf{x}_i}{\beta}\big) \tag{2.6}$$

where $K(\mathbf{x})$ is the kernel function, e.g., Gaussian kernel, and β is the window width, also known as bandwidth.

Parzen window detects novel points using a classification function in the same form as the mixture of Gaussians. Figure 2.3 shows an example of novelty detection based on Parzen window. To build the boundary (represented by solid black lines), we selected a bandwidth $\beta = 0.25$, and a rejection rate of the target class $\epsilon = 0.01$. The plot also shows the size of the region centered at each training point represented by the dotted gray circles.

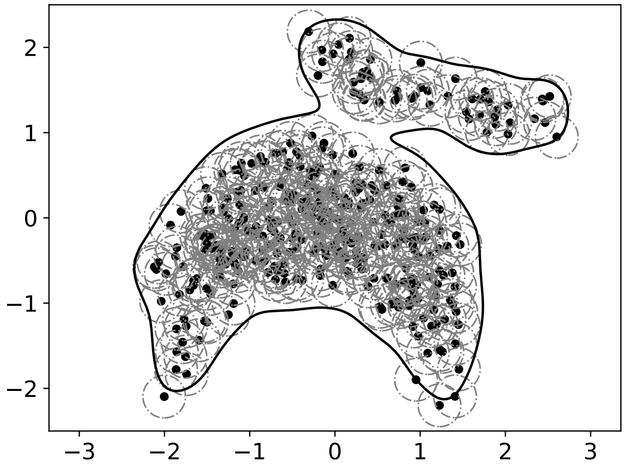


Figure 2.3: Parzen window for novelty detection with bandwidth $\beta = 0.25$ on a subset of the banana dataset. Black points represent the target data, solid black lines represent the boundary generated by the model with $\epsilon = 0.01$, and dotted gray circles represent the size of the region center at each training point.

The main drawback of probabilistic approaches resides in finding the right model to correctly describe the distribution of target data and estimate the novelty threshold correctly. In general, probabilistic approaches exhibit a limited performance when the training set is small. They also cope with difficulties in high-dimensional problems, mainly when normal data are sparse in large spaces.

Distance-based approaches

Distance-based approaches use well-established distance metrics to determine if a point is normal or novel without any assumption about the target class distribution. We can identify two distance-based approaches: nearest neighbor (NN)-based approaches and clustering-based approaches.

NN-based approaches assume that normal points have nearest neighbors in the training set, while novel points are located far away. Tax and Duin [47] proposed one of the first methods for novelty detection based on NN, named one-class classification method NN-d (or NN-d rule). The rule defines the novelty indication as the quotient between the distance from a test point $\mathbf{x}$ to its nearest neighbor $\mathrm{NN}(\mathbf{x})$ in the training set, and the distance from the nearest neighbor $\mathrm{NN}(\mathbf{x})$ to its nearest neighbor $NN(NN(\mathbf{x}))$ in the training set,

$$f_{NNd}(\mathbf{x}) = \frac{||\mathbf{x} - \mathrm{NN}(\mathbf{x})||}{||\mathrm{NN}(\mathbf{x}) - \mathrm{NN}(\mathrm{NN}(\mathbf{x}))||} \tag{2.7}$$

In the NN-d rule, L_2-norm was used for distance calculation and the test point $\mathbf{x}$ is novel if $f_{NNd}(\mathbf{x})$ is larger than $\theta = 1.0$. Figure 2.4 shows the boundaries generated by the NN-d rule with artificial data represented by black points. In the plot, the solid black lines represent the boundary, and the dotted gray circles represent the target region for each point.

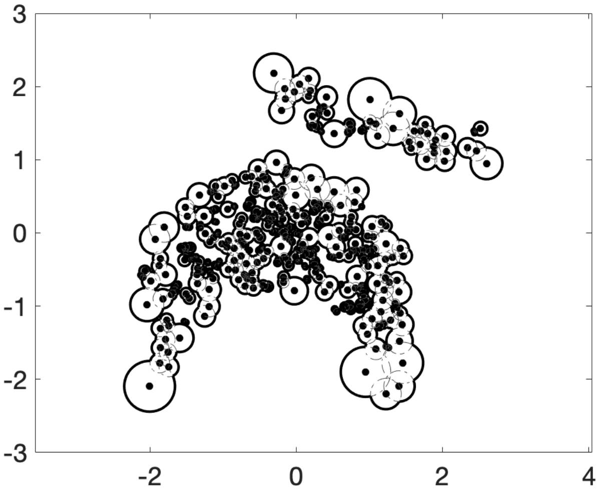


Figure 2.4: One class nearest neighbor method for novelty detection on a subset of the banana dataset. Solid lines and dotted lines represent the boundary and the aceptation region for each points, respectively.

An extension of the NN-d rule called k-NN, computes the average distance between the test point and its k nearest neighbors in the training set. If the average distance is larger than a predefined threshold, then the test point is novel. Recent advances in NN-based approaches include the Nearest Neighbor Domain description rule (NNDD) [48], which establishes that a test point is normal if the nearest neighbor distance lies within a threshold. Let r be a Minkowski metric, $r \in \{1, 2, ..., \infty\}$. The NNDD rule computes the distance d from the test point $\mathbf{x}$ to its k-nearest neighbors in the training set $\delta_k(\mathbf{x}) = (d(\mathbf{x}, \mathrm{NN}_1(\mathbf{x})), \ldots, d(\mathbf{x}, \mathrm{NN}_k(\mathbf{x})))$, where $\mathrm{NN}_k(\mathbf{x})$ represents the k-nearest neighbor of $\mathbf{x}$. The novelty indication is defined as,

$$f_{NNDD}(\mathbf{x}) = \mathrm{sign}(\theta - ||\delta_k(\mathbf{x})||_r) \tag{2.8}$$

The classical implementation of nearest neighbor approaches requires storing all the training dataset and computing the distance from the test point to the whole dataset. It is computationally expensive, particularly in high-dimensional datasets. Therefore, these methods commonly lack scalability. In order to reduce the above requirements, some methods reduce the size of storing. These methods include training-set consistency, which aims to select a training subset to classify the data correctly using the NN-d rule [49] and prototype-based nearest neighbor approaches that use prototypes to represent the training set [50]. Other exciting approaches use tree representation of the training data and apply the distance from the point to the closest edges of the tree as a novelty indication [36].

The clustering approaches also use the concept of prototypes to characterize the training dataset, and they often apply k-means or hybridization-based on k-means to find the set of prototypes [51, 52]. The k-means algorithm determines the position of k cluster centers which minimize the sum of squared distances between each point in a dataset and its nearest cluster center. Let $\mathbf{C} = \{\mathbf{c}_j | \mathbf{c}_j \in \mathbb{R}^D, j = 1, \ldots, k\}$ be the set of centers, the objective in k-means is

$$\begin{gathered}\min_{\mathbf{C}} \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{k} I_{i,j} ||\mathbf{x}_i - \mathbf{c}_j||^2 \\ I_{i,j} = \begin{cases} 1, & \text{if } ||\mathbf{x}_i - \mathbf{c}_j|| = \min\limits_j ||\mathbf{x}_i - \mathbf{c}_j|| \\ 0, & \text{otherwise} \end{cases}\end{gathered} \tag{2.9}$$

In the detection phase, clustering approaches compute the distance from the test point $\mathbf{x}$ to its closest prototype and use this distance as a novelty indication,

$$f_{kmeans}(\mathbf{x}) = \min_j ||\mathbf{x} - \mathbf{c}_j||^2 \tag{2.10}$$

If f_{kmeans} is larger than a threshold, then the test point is novel. Figure 2.5 illustrates the boundary of the k-means algorithm for novelty detection with four prototypes (represented by the gray points). We found the optimal novelty threshold by setting a rejection rate of $\epsilon = 0.01$.

Reconstruction-based approaches

Reconstruction-based approaches aim to learn a compact representation of the training data by minimizing the error between the inputs and the outputs of the system. Let $\hat{\mathbf{x}}$ be the reconstructed version of an input $\mathbf{x}$, the reconstruction function commonly adopt the form $f_{rec}(\mathbf{x}) = ||\mathbf{x} - \hat{\mathbf{x}}||$. if f_{rec} is larger than the threshold θ, then the input is novel.

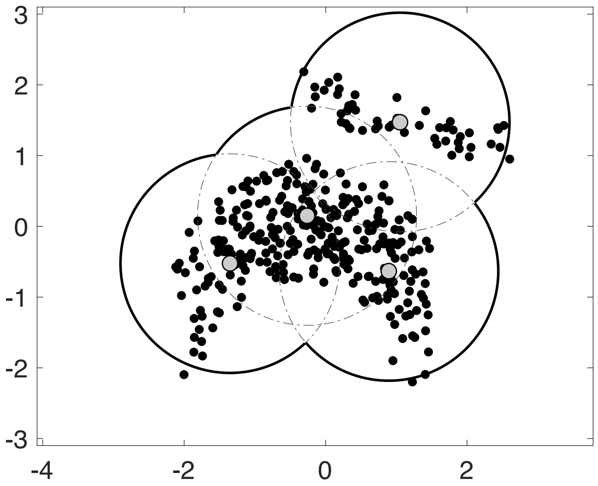


Figure 2.5: k-means algorithm for novelty detection on a subset of the banana dataset. Decision boundary with four prototypes (represented by the gray points) and a rejection rate of the target class $\epsilon = 0.01$.

Reconstruction-based approaches are classified into subspace-based methods and neural network-based methods. Subspace-based methods transform data into a lower dimension subspace, where target data can differentiate more easily from novel data. Examples of these methods include Principal Components Analysis (PCA) and variants [53, 54]. In the standard PCA, the input data are normalized and projected in a space where its orthogonal axes maximize the input data variance. For novelty detection using PCA, the k eigenvectors are used to reconstruct the input data. Let $\mathbf{U}_{(k)}$ be the set of k eigenvectors, so the test point $\mathbf{x}$ can be reconstructed by $\hat{\mathbf{x}} = \mathbf{U}_{(k)}\mathbf{U}_{(k)}^T\mathbf{x}$ and the reconstruction error is defined as,

$$f_{pca}(\mathbf{x}) = |||\mathbf{x} - \mathbf{U_{(k)}}\mathbf{U_{(k)}}^T\mathbf{x}||^2 \tag{2.11}$$

An improvement in the standard PCA, called kernel PCA [55], incorporates a function $\mathbf{\Phi}$ to map data points from input space to a high-dimensional feature space. Standard PCA can be computed such that input vectors only appear in the form of scalar products $\mathbf{x} \cdot \mathbf{y}$. In kernel PCA the scalar products appears in the form $(\mathbf{\Phi}(\mathbf{x}) \cdot \mathbf{\Phi}(\mathbf{y}))$. Kernel PCA omits the explicit transformation of a point to feature space; instead, it uses kernel functions $K(\mathbf{x}, \mathbf{y})$. It is worth mentioning that the standard PCA can be obtained as special case of kernel PCA with kernel function $K(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y})$. Figure 2.6 shows the standard PCA for novelty detection. We selected the first component of the PCA to reconstruct the input data and a rejection rate of $\epsilon = 0.01$. In the plot, solid black lines represent the boundary.

Network-based approaches include feed forward networks [56], Hopfield network

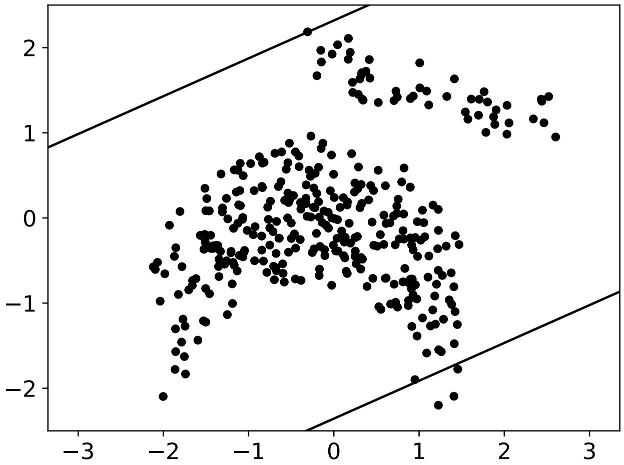


Figure 2.6: Boundary generated by the standard principal component analysis with one principal component on a subset of the banana dataset. To build the boundary, we fixed the rejection rate to $\epsilon = 0.01$.

[57], self-organizing network [58], and autoencoders [59]. These networks have restrictions in their operation because they keep a fixed network structure during the whole training. Alternatives able to address this problem are growing networks, which can add or remove nodes in the network during training, for example, grow-when-required (GWR) neural networks [15], growing neural gas [60], and evolving connectionist systems (ECoS) [61, 62].

Figure 2.7 shows an example of growing neural networks for novelty detection. This figure presents the reconstructed points by the GWR network (gray points) and its topological connections (solid gray lines) in a subset of the banana dataset. Each node of the network has habituation based on the number of times the node fires during training. In the detection phase, for the test point $\mathbf{x}$ the network finds the best matching node $\mathbf{s}$ and uses the habituation of this node (called h_s) and the reconstruction error ($a(\mathbf{x}) = \exp(-||\mathbf{x} - \mathbf{s}||^2)$) as a novelty indication. The rule to detect a novel point is defined as $h_s < h_T$ and $a(\mathbf{x}) < a_T$, where h_T and a_T are the habituation, and activation thresholds, respectively.

The reconstruction-based approaches are more flexible than probabilistic approaches because they do not assume any target data properties. In general, the models are compact because they require less storage capability, and therefore they generate fast inferences. Notably, the growing networks are suitable for continuous novelty detection, particularly useful for robotic applications. However, these kinds of networks are hard to train in high-dimensional problems, and they are also sensitive to their parameter values.

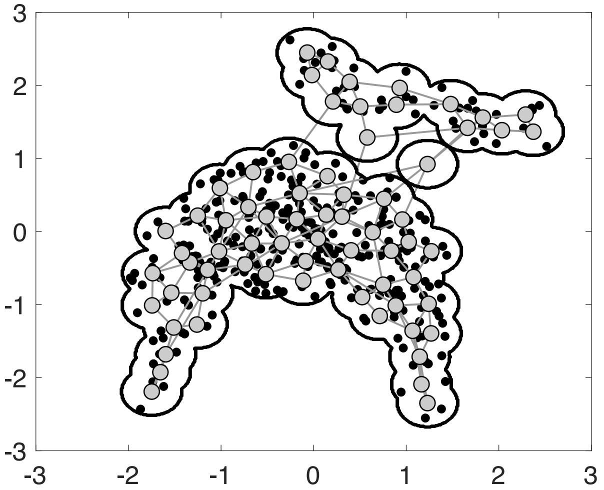


Figure 2.7: Data points reconstructed by GWR and its topological connections on a subset of the banana dataset. The gray points represent the reconstructed points; the solid gray lines represent the topological connections, and the solid black lines represent the decision boundary. GWR used three epochs, an activation threshold of 0.9, and a habituation threshold of 0.3. The network consists of 69 nodes and 148 edges.

Domain-based approaches

Domain-based approaches enclose the target data with decision boundaries maximizing the margin between classes in feature space. Variants of the support vector machine (SVM) are the most popular domain-based methods for novelty detection.

In one-class SVM [63], the objective is to find a hyperplane $\mathbf{w}$, which best separates the data $\mathbf{x}_i$ from the origin in feature space. To this end, the following quadratic problem is solved,

$$\begin{aligned} &\min_{\mathbf{w},\xi,\rho} \frac{1}{2}||\mathbf{w}||^2 + \frac{1}{vN}\sum_i \xi_i - \rho \\ &\text{s.t.} \quad \mathbf{w}\cdot\mathbf{\Phi}(\mathbf{x}_i) \geq \rho - \xi_i, \qquad \xi_i \geq 0 \end{aligned} \tag{2.12}$$

where ξ_i are slack variables, ρ is the offset of the hyperplane, $v \in (0, 1]$ is a parameter to control the fraction of data separated by the hyperplane, and $\Phi(\cdot)$ is a function which maps a data from input space to feature space.

The inequality constraint can be incorporated into the function by using Lagrange multipliers, and by fixing the partial derivatives to zero, we obtain the following dual problem,

$$\begin{aligned} &\min_{\alpha} \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \\ &\text{s.t.} \quad 0 \leq \alpha_i \leq \frac{1}{vN}, \quad \sum_i \alpha_i = 1 \end{aligned} \tag{2.13}$$

where $K(\mathbf{x}, \mathbf{y})$ is a kernel function, for instance the Gaussian kernel, and α_i are the Lagrange multipliers. Data points with corresponding $\alpha_i > 0$ are called support vectors.

The support vectors are used to compute the offset of the plane and the decision function,

$$\rho = (\mathbf{w} \cdot \mathbf{\Phi}(\mathbf{x}_i)) = \sum_j \alpha_j K(\mathbf{x}_j, \mathbf{x}_i) \tag{2.14}$$

$$f_{OCSVM}(\mathbf{x}) = \text{sign}\big(\sum_i \alpha_i K(\mathbf{x}_i, \mathbf{x}) - \rho\big) \tag{2.15}$$

where $sign(z)$ is a sign function which returns 1 when $z \geq 0$ (normal points) and -1 otherwise.

Another approach called support vector data description (SVDD) [64], uses a hypersphere instead of hyperplanes to surround all or most of the normal data, also by setting the decision boundary with only support vectors [65]. The hypersphere is characterized by a center $\mathbf{a}$ and a radius $R > 0$. The objective is to minimize the volume of the hypersphere,

$$\begin{aligned} &\min_{R, \mathbf{a}, \xi} R^2 + \frac{1}{vN} \sum_i \xi_i \\ &\text{s.t.} \ ||\mathbf{\Phi}(\mathbf{x}_i) - \mathbf{a}||^2 \leq R^2 + \xi_i, \ \xi_i \geq 0 \end{aligned} \tag{2.16}$$

where v is a parameter to control the trade-off between the volume of the hypersphere and the number of rejected points; and ξ_i are slack variables to relax the distance from $\mathbf{\Phi}(\mathbf{x}_i)$ to $\mathbf{a}$.

By introducing Lagrange multipliers in the above function and by fixing the partial derivatives to zero, we obtain the following dual problem,

$$\begin{aligned} &\min_{\alpha} \sum_{i,j} \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) - \sum_i \alpha_i K(\mathbf{x}_i, \mathbf{x}_i) \\ &\text{s.t.} \ 0 \leq \alpha_i \leq \frac{1}{vN}, \ \sum_i \alpha_i = 1 \end{aligned} \tag{2.17}$$

where α_i are the Lagrange multipliers.

In the SVDD, the center of the hypersphere is computed as the linear combination of the support vectors,

$$\mathbf{a} = \sum_i \alpha_i \mathbf{\Phi}(\mathbf{x}_i) \tag{2.18}$$

The test point $\mathbf{x}$ is considered novel, if the distance from $\mathbf{\Phi}(\mathbf{x})$ to the center of the hypersphere is higher than the radius R,

$$f_{SVDD}(\mathbf{x}) = \text{sign}\Big(R^2 - \sum_{i,j} \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) + 2\sum_i \alpha_i K(\mathbf{x}, \mathbf{x}_i) - K(\mathbf{x}, \mathbf{x})\Big) \tag{2.19}$$

and,

$$R^2 = K(\mathbf{x}_l, \mathbf{x}_l) - 2\sum_i \alpha_i K(\mathbf{x}_i, \mathbf{x}_l) + \sum_{i,j} \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) \tag{2.20}$$

where $\mathbf{x}_l$ are the support vectors with their corresponding Lagrange multiplier $0 < \alpha_l < 1/(vN)$.

Figure 2.8 shows the SVDD for novelty detection. The classifier used a rejection rate of $\epsilon = 0.01$ and generated eight support vectors in a dataset of 356 points. In the plot, gray points represent the support vectors, and solid black lines represent the decision boundary. It is worth mentioning that the OCSVM, under the same parameters, produces visible results similar to the SVDD in this particular dataset.

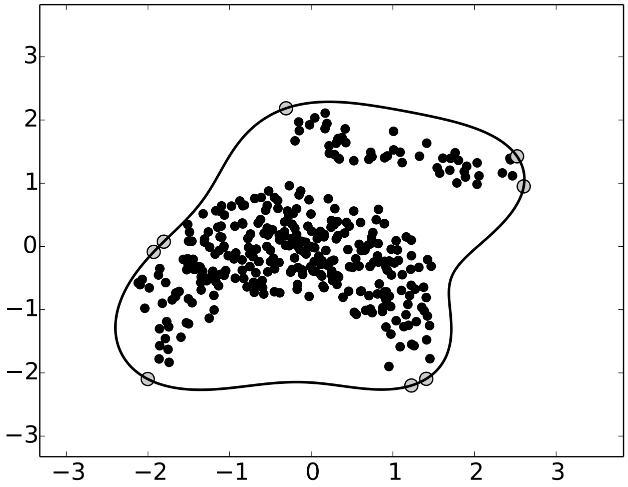


Figure 2.8: Novelty detection based on SVDD on a subset of the banana dataset. Boundary generated are marked by solid lines with a rejected rate of $\epsilon = 0.01$. Gray points represent the support vectors.

Domain-based approaches construct the decision boundary by using only points close to it in features space. The rest of the data is not considered for setting the decision boundary. These methods have drawbacks related to the complexity associated with the computation of the kernel functions, the selection of an appropriated kernel, and its parameters to control the boundaries.

Information-theoretic approaches

Information-theoretic approaches compute information measures, e.g., entropy or divergence, to obtain information about the data. These methods assume that novel data significantly contributes to the value of the information measure, and they work well when there are many novel points. However, these methods are computationally expensive and sensitive to the selection of the measure. Besides, it is not easy to compute a novelty score for a single test point. There are several examples of information-theoretic approaches for novelty detection, see [66, 67].

Other approaches

Another exciting direction for novelty detection is the use of rule-based methods. These approaches generate rules based on the behavior of the target data, and they assume that a point not explained by any rule is novel [68, 69].

Recently, deep learning algorithms have shown a new open area into novelty detection. We identify two approaches: deep networks as feature extraction techniques and deep end-to-end learning. The first approach combines the ability of deep networks to extract features with the ability of one-class classifiers to model the target class. In this category, some approaches learn the features and the classifier [7, 70] simultaneously, and other approaches use pre-trained networks for feature extraction and then apply traditional novelty detectors to learn the model of target class [71]. The second approach, deep end-to-end learning, uses deep learning for the whole process of novelty/anomaly detection, e.g., convolutional autoencoder (AE), adversarial convolutional autoencoder (AdvAET), variational autoencoder (VAEs), and generative adversarial networks (GANs) [72, 73]. These techniques are also part of the reconstruction-based approaches.

2.3 Automatic configuration of novelty detectors

The success of novelty detectors strongly depends on selecting their hyperparameter values [74, 75, 76]. Several attempts which use automatic design techniques have been proposed to face this task. Automatic design techniques require a selection criterion to optimize the hyperparameters. However, the selection criterion is hard to obtain because novelty detection is an imbalanced problem wherein most of the cases, there are not novelty data available, or they are few compared to the normal data. Existing methods to select proper hyperparameters are based on three different approaches: (*i*) selection based on the target class only, (*ii*) selection based on synthetic data, and (*iii*) supervised selection.

In selection based only on the target class, some measures extracted from the normal data estimate the quality of a candidate novelty detector. Tax and Müller [74] proposed a consistency measure that uses the error in the target class to tune a single hyperparameter. The hyperparameter determines the complexity of the classifier. Higher complexity represents a better adaptation to the target set, but with the risk of overfitting. In their algorithm, they increase the value of the hyperparameter until the classifier becomes inconsistent, i.e., when the classifier rejects a certain number of target elements. Some other works also consider the consistency measure to select and combine different novelty detectors. Krawczyk [77] proposed a firefly algorithm for ensemble pruning and weighting of one-class classifiers. They used an objective function based on the consistency measures to select the best combination model. Parhizkar and Abadi [78] implemented an automatic design technique based on a binary artificial bee colony algorithm for the construction of one-class classifier ensembles. They differ from the previous work in the selection mechanism. They proposed an objective function which weights the consistency measure and a non-pairwise diversity measure. Xiao et al. [79] proposed a Kullback-Leibler (KL) divergence measure to perform the selection. They first split the target data into edge and interior samples; then, they apply a grid search to maximize the KL-divergence between both sample sets.

In methods based on synthetic data, the methods first generate artificial outliers and then apply an optimization technique to find the best hyperparameter values. They commonly adopt a selection criterion based on the misclassification rate of both classes. Xie [80] generated artificial outliers based on a uniform hyperspherical distribution generation method. Then they applied a constrained particle swarm optimization technique to select the best hyperparameters. Wang et al. [75] proposed a mechanism to generate artificial outliers and target data efficiently. They then applied a grid search to minimize the classification error with these synthetic data. The success of this methodology depends on how well the outliers are placed in the space. Poor placement can produce a poor performance of the tuned novelty detectors [76].

What constitutes novelty is inherently application dependent [56]. Contrary to the previous approaches, supervised techniques consider specific scenarios to select the best hyperparameters. The advantage of using novelty detectors in this fashion is to face the imbalanced nature of the problem, where the supervised selection of novelty detectors has proved to obtain much better results than multi-class classification [81]. In this approach, the novelty detectors are executed several times with different hyperparameters values using only normal training data. Then, the performance of novelty detectors is estimated by using a validation set with labeled normal and novelty samples. Zhuang and Dai [82] used a grid search to select the hyperparameters of a novelty detector with an objective function that maximizes the accuracy in both normal and novel data. Haggett and Chu [56] designed novelty detectors automatically based on neural networks by using the neuroevolution of augmenting topologies. They compensated the imbalance of data using an objective function with higher weighing to the correct novel data.

2.4 Summary

Novelty detection has been posed as a one-class classification problem, where the objective is to differentiate the positive class from the negative class. Commonly, this problem has been addressed by machine learning techniques, where their performance strongly depends on their hyperparameter values.

Automatic configuration design tools provide an alternative for automatically tuning the hyperparameter values using a selection criterion and an optimization technique. However, the selection criterion is hard to obtain because novelty detection is an imbalanced problem, where there are few or none novelty data and a large number of normal data. Existing methods to select proper hyperparameter values are based on three different approaches. The first approach uses only information about the positive class by establishing error measures in the training data to find more flexible boundaries, avoiding overfitting. The second approach generates artificial data to estimate the performance of the novelty detectors. This approach depends on the artificial data position, where poor placement generates poor performance in the tuned detectors. Finally, the third approach, called supervised hyperparameter tuning, faces the imbalanced novelty problem by exploiting the few novelty samples available to generate the proper hyperparameters for specific-application scenarios.

CHAPTER 3

Tuning novelty detectors for specific robotic exploration and inspection tasks

In this chapter, we present an automatic design tool based on bio-inspired optimization techniques, specifically the artificial bee colony algorithm, for tuning online machine learning approaches in the problem of visual novelty detection. The novelty detectors are tuned to solve specific exploration and inspection tasks in an outdoor environment, where an unmanned aerial vehicle executes a path while it captures images from the environment. The novelty detectors are trained with only normal images to construct a model of the environment, and then this model is used to detect novel objects added to the environment.

3.1 Introduction

Inspired by the ability of animals to detect novelties and to respond to changes in their environment [58], researchers have tried to incorporate novelty detection methods into robots to improve their adaptation capability to dynamic environments often present in real-world robotic tasks. Nowadays, it is possible to capture useful information to perform this process using sensors incorporated into robots such as sonars, laser, cameras, GPS, and others. Among them, visual sensors are one of the most popular devices to extract information for novelty detection [83, 84, 8].

In robotics, a novelty detection module is beneficial for several applications, e.g., exploration, inspection, and surveillance. Specifically, in exploration and inspection tasks [8], the robot should explore its environment, building a normality model using the information sensed. After the model construction, the robot patrols (inspection phase) the same route of the exploration phase to detect novelties.

For the above problem, the robot needs online novelty detectors to cope with dynamic environments and approaches with fast learning capabilities to detect novelties in scenarios with a reduced amount of information. Most of the traditional one-class classifiers operate offline, which means that it is difficult to adapt these methods to dynamic environments. Meanwhile, deep learning approaches need large-scale datasets and a huge computation load to train the models. Alternatively, online approaches based on evolving connectionist systems [8] and grow-when-required neural networks [1] meet the above conditions. These methods not only build a model of normality incrementally, but they also adapt the model to dynamic changes of the input data, i.e., they can insert new information and forget old information. However, we still see challenges in applying online novelty detectors into exploration and inspection tasks based on visual information. First, current robotic applications have used low-level visual features sensitive to illumination changes, occlusion, or geometric transformations. Some visual features used in robotic applications are RGB histograms [8], color angular indexing [85], GIST descriptor [84], and others. Second, in different exploration and inspection tasks, the robots use the same parameters in the novelty detection module, without considering that the performance of the detector depends on the specific task to solve. These reasons have restricted the applications of the above online novelty detectors to indoor environments, where many conditions have been controlled.

Motivated by the previous issues, in this work, we propose to apply novelty detectors based on evolutionary connectionist systems and grow-when-required neural networks with visual descriptions drawn from deep convolutional networks for exploration and visual inspection tasks. In contrast with existing deep learning approaches for novelty detection, we propose using previously trained networks to extract visual features, instead of learning new visual features to reduce the computational load in the feature extraction phase. We prefer deep descriptions over traditional visual descriptors due to its reliability in generating robust features for classification tasks. Additionally, we propose a framework to automatically design novelty detectors via selecting the best parameters, depending on the specific robotic exploration and inspection task. This framework uses a global optimization technique as the main component to find the most appropriate parameters for the task. We verify the utility of the proposed visual novelty detection system in outdoor applications, where an Unmanned Aerial Vehicle (UAV) captures images in challenging environments, i.e., environments with illumination changes, geometric transformations in the objects of the environment, and occlusions.

The rest of this chapter is structured as follows. Section 3.2 presents the related work in novelty detection. Section 3.3 presents our visual-based novelty detection approach. Section 3.4 describes the experimental setup and compares our experimental results against traditional visual novelty detectors. In Section 3.5, we discuss the results and limitations of the work.

3.2 Related work

In robotics, several applications use novelty detection algorithms, which are particularly suitable methods for continuous learning of the environment. Marsland et al. [1] proposed a novelty detection system for mobile robots based on a GWR neural network. Neto et al. [86] applied the GWR network with visual information as input. They proposed a framework that combines a visual attention model and a visual description of the more salient points in the image based on color angular indexing and the standard deviation of the intensity. This type of description is invariant to illumination changes; however, it is infeasible to detect new objects outside the attention regions. Neto and Nehmzow [85] used the novelty detectors based on GWR and incremental PCA with two interest point detectors: the detection based on saliency and the Harris detector. They compared two ways to represent the patches in the visual input (raw pixels of the image). The first method keeps a fixed size of the patch, while the second one automatically finds the patch size. The results often showed that the fixed-size approach presents the best results. Inspired by Evolving Connectionist Systems (ECoS) and the habituation model proposed in the GWR networks, Özbielge [8] proposed a recurrent neural network for novelty detection for exploration and inspection tasks. This method predicts the next input and computes a novelty threshold value during its operation. This information is used and compared to the observed input to decide if it is novel. The system uses laser readings, motor outputs, and RGB color histograms as input information. Also, Özbielge [62] proposed a dynamic neural network for static and dynamic environments. The method computes the novelty in a similar way to the previous approach, it computes the error between the input observation and the prediction of the network, and if the error is higher than the evolved threshold, then the object is considered a novelty. Pitonakova and Bullock [87] used the GWR networks in a simulation environment with visual inputs. They compared several GWR architectures and found a trade-off between robustness in the performance of the networks based on their parameter values and the fidelity of the learned models. Besides, they proposed some improvements to balance this trade-off, e.g., variable length of input connections.

Kato et al. [84] implemented a system based on reconstruction that takes advantage of the position where the robot captured the images. The novelty detector used the GIST descriptor and a reconstruction-based approach to generate a system invariant to illumination changes. A principal limitation of their system is the absence of a threshold value to detect novelties. Gonzalez-Pacheco et al. [88] developed a novelty filter to detect new human poses. The system uses visual information of the Kinect sensor and four one-class classifiers: Gaussian Mixture Model, K-means, One-Class Support Vector Machines, and Least Squares Anomaly Detection. For this task, the Gaussian Mixture Model performs better than the other novelty detectors. However, the performance of the method depends on the number of specified Gaussians (the user defines this value in the experiment). Recently, Gatsoulis and McGinnity [11] proposed an online expandable neural network similar to the GWR network. The method uses Speeded-Up Robust Features (SURF) and an ownership vector for feature description.

All the above novelty detectors have been applied for indoor environments, and few works have been proposed for outdoor environments. For instance, Wang et al. [89] implemented an approximation to the nearest neighbor via search trees to detect novelties in indoor and outdoor environments (for the case of the outdoor environment, they used a static camera). The inputs are visual features extracted from patches, such as color histograms in the HSV space (Hue, Saturation, Value) and texture information (Gabor filters). They compared the performance of their system against the GWR network. The results showed that the proposed approach is better than the GWR network in their particular experiments. Ross et al. [9] presented a vision system for obstacle detection based on novelty for field robotics. The motivation in the use of novelty is that it is infeasible to train a system with all types of obstacles in agricultural applications. The inputs of the detector are color, texture, and position of the patches in stereo images. The system detects novelty by using the probability density estimated by a weighted version of Parzen windows.

Previous works have explored low-level visual features for image description such as color angular indexing, GIST descriptor, RGB raw values, RGB color histograms, HSV histograms, and Gabor filters. Few efforts take advantage of emerging deep convolutional neural networks for feature description in visual novelty detection. One of them is the robotic system proposed by Ritcher and Roy [59]. The objective of their work was to develop a robot with a safe navigation module. An autoencoder network composes the novelty detection module with three hidden layers that automatically find compressed representations of small input images. The goal of the network is to reconstruct the input image, and if the error between the input and the output is higher than an error tolerance, then the system will detect the novelty and use it to maintain the safety of the robot.

3.3 Materials and methods

In this section, we describe the proposed system for visual exploration and inspection tasks. In this work, we use images captured by an UAV that operates in outdoor environments. Figure 3.1 illustrates the proposed system. In the exploration phase, the UAV follows a fixed trajectory and captures images of the environment. The system represents the captured images via deep features by using a pre-trained convolutional neural network, called MobileNetV2 [17]. The novelty detector processes the feature vector and constructs a model of the environment. The user can select between two detectors: Simple Evolving Connectionist Systems (SECoS) or GWR network. Finally, in the inspection phase, the UAV again executes its path and searches for novel objects. The UAV uses the above model to identify novelties. Then, we describe in more detail the components of the proposed visual novelty detection system.

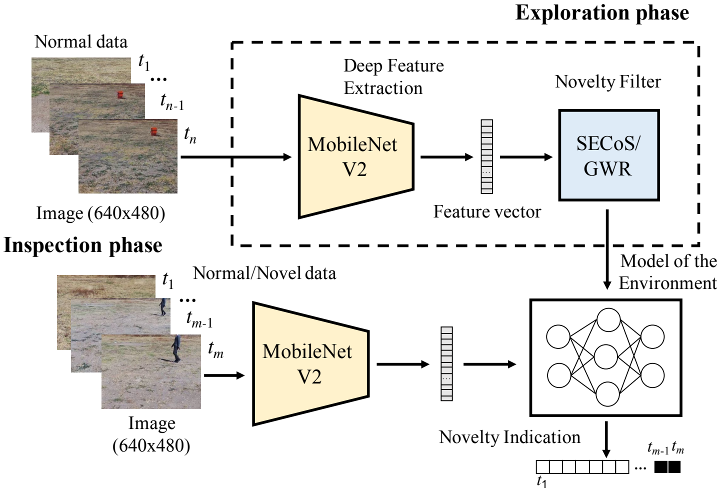


Figure 3.1: Graphical description of the proposed system for visual exploration and inspection tasks.

3.3.1 Visual feature extraction

One way to represent the images is via visual feature vectors. Among the visual features, traditional features as RGB color histograms [8], color angular indexing [83], and the GIST descriptor [84] have been applied for visual novelty detection in robotics. However, traditional visual features are highly sensitive to illumination changes, noise, occlusion, or geometric transformations. Recently, convolutional neural networks have been applied successfully as powerful tools to extract features from images [90], having robust performances in a wide variety of classification tasks.

Motivated by the success of convolutional neural networks as feature extraction methods, we propose to apply a convolutional neural network to extract features from images for the task of visual novelty detection in robotics. In this work, we selected MobileNetV2 [17] because it is the network with the lowest number of parameters in the Keras API and the TensorFlow engine. In our implementation, we used the pre-trained network with the weights trained on the ImageNet dataset. In order to extract the visual features, we resize the input image to the default size in the Keras API of 224×224 pixels. We also deactivate the classification layer and activate the average pooling mode for feature extraction. We obtain visual feature vectors of 1280 elements.

3.3.2 Novelty detectors

We have selected two online novelty detection methods that are used as the base to develop exploration and inspection tasks with real robots [8, 83, 1]. Both techniques are constructive and can evolve the structures of the models and their parameters during their operation. We selected the SECoS and the GWR network.

Simple evolving connectionist systems

The ECoS, proposed by Kasabov [91], are a type of neural network that can evolve their parameters and their structure over time. Below, we show the characteristics of the ECoS that make them attractive to address the problem of visual novelty detection in robotics [61]:

- Fast learning capabilities (one-pass learning).
- Online learning and incremental adaptation to new data.
- The model is evolved to adapt to the input information, and the examples are added to the model when they are different in some aspects to the current model of the data.

The SECoS conserve these characteristics [16], but they present two advantages concerning the other ECoS implementations. The SECoS are easy to implement because they have a low number of layers to learn the input data, and they work directly on the input space. Figure 3.2 shows a graphical description of the SECoS network. Three layers compose the network: the input layer which transfers the inputs to the nodes of the next layer; the hidden layer (evolving layer) which incorporates new nodes to represent novel data; and the output layer which uses saturation linear activation functions to compute the output. We can observe that in SECoS network, there exist two connection layers: the connections between the nodes of the input layer and the nodes of the evolving layer (incoming connections), and the connections between the nodes of the evolving layer and the nodes of the output layer (outcoming connections).

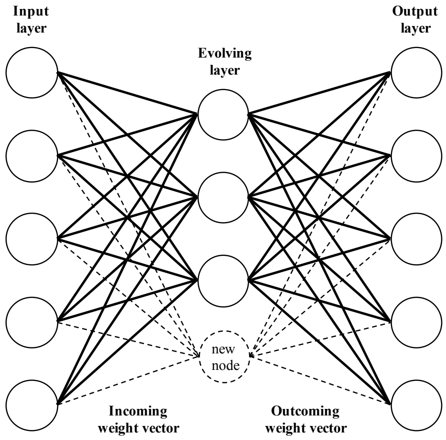


Figure 3.2: Graphical description of the SECoS network. Adaptation of the general ECoS representation from Watts [61].

In this work, we use the SECoS learning algorithm proposed by Watts and Kasabov [16]. The algorithm receives as input the weights of the connections in the network, the input features, and the desired output. The proposed approach uses a SECoS implementation with the same number of nodes in the input layer and the output layer. The objective of the approach is to generate a system able to reconstruct the input vector. When the model generated by the SECoS implementation is not able to represent an input, it should add a new node in the evolving layer with the incoming weight values equal to the input vector and the outcoming weight values equal to the desired output. Also, it should add a new node to the model when the reconstructed output is significantly different from the desired output, i.e., the Euclidean distance between the desired output and the current output of the network is greater than the threshold E_{thr}. When the model can represent a given input successfully, the SECoS implementation only updates the model (updating of the connection weights) to represent better the input data. The parameters of this learning model include the learning coefficients (η_1, η_2), the sensitivity threshold (S_{thr}), and the error threshold (E_{thr}). For more details about this learning algorithm, the readers can refer to the work by Watts and Kasabov [16].

Grow-when-required neural network

GWR is an online self-organized neural network proposed to solve the novelty detection problem [15]. Figure 3.3 shows a graphical representation of the GWR neural network. A clustering layer of nodes and a single output node compose the network. The nodes in the clustering layer use weight vectors to represent the centers of the clusters. The GWR network can add and remove nodes to its structure, specifically in the clustering layer, to adapt to the changes of the inputs. The connection synapses to the clustering layer in the network are subject to a habituation model, which is a reduction in response to similar inputs.

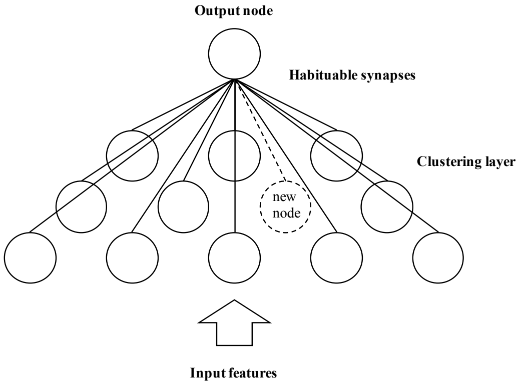


Figure 3.3: Graphical representation of the GWR neural network. Adaptation of the network architecture presented by Neto et al. [86].

In the proposed framework, we use the algorithm of the GWR network for novelty detection described by Neto [83]. The network starts with two dishabituated nodes with weight vectors initialized to the positions of the first two input vectors. At the beginning, there are no topological connections between both nodes. From the third input vector, the best matching node s and the second best matching node t of the clustering layer are found, i.e., the nearest nodes to the input vector. If there exists a topological connection between both nodes, its age is set to zero; otherwise, the connection between both nodes is created with age zero. The GWR network uses the activation and habituation level of the node s to decide if the input is novel or not. If the input vector is novel, a new node in the clustering layer is created with its weight vector initialized to the average position between the input vector and the best matching node. Also, the topological connections of the nodes in the clustering layer are updated by removing the connection between the best matching nodes and inserting new connections between the best matching nodes and the created node. Then, the best matching node and its topological neighbors update their positions in the direction of the input vector and also update their habituation levels. Finally, all the connections increase their ages and all connections with ages higher than the maximum age are removed. When a node has no topological connections is also removed (ability to forget).

The parameters that impact the behavior of the network are the parameters of the habituation model, the activation threshold (a_T), the habituation threshold (h_T), the proportionality factor (η), and the learning rate (ϵ). A detailed description of the learning algorithm of the GWR neural network can be found in [83].

3.3.3 Automatic configuration of novelty detectors

One of the main problems in the application of novelty detectors is the proper selection of their parameters in order to obtain the best results regarding the accuracy in the detection. With this in mind, we propose a framework to tune the novelty detectors automatically for a specific task, see Figure 3.4. Our optimization approach not only searches for parameters of the novelty detectors but also finds the best size of the visual feature vector.

In this work, we propose the use of Artificial Bee Colony algorithm (ABC) [13] as the optimization tool. It is worth noting that although in this work we show the use of the ABC algorithm, in the proposed framework we can incorporate different algorithms to find the more appropriate parameters of the filters to solve specific tasks. The ABC algorithm offers a population-based approach for numeric optimization. In the ABC algorithm, artificial bees update their position during the time to find the best food sources. This algorithm has shown to be better than or competitive with other bio-inspired optimization techniques. Besides, we can find applications of the ABC algorithm for a wide variety of engineering problem such as image processing, data mining, control, mobile robotics [13]. The implementation details of the algorithm can be found in Mernik et al. [92]. In the proposed methodology, we use an implementation with termination condition based on the number of iterations, also known as ABC$_{imp1}$.

In our implementation of the ABC algorithm, each food position represents a set of parameter values of the novelty detector. Table 3.1 shows the parameters that should be adjusted by using the ABC algorithm. The search range of all the decision variables is in $[0, 1]$. In the case of the GWR novelty filter, we set the parameters of the habituation model to the default values, and also we keep as constant the maximum age value. The details about the ABC algorithms are as follow. We use a population of 20 food positions and a total number of 100 iterations.

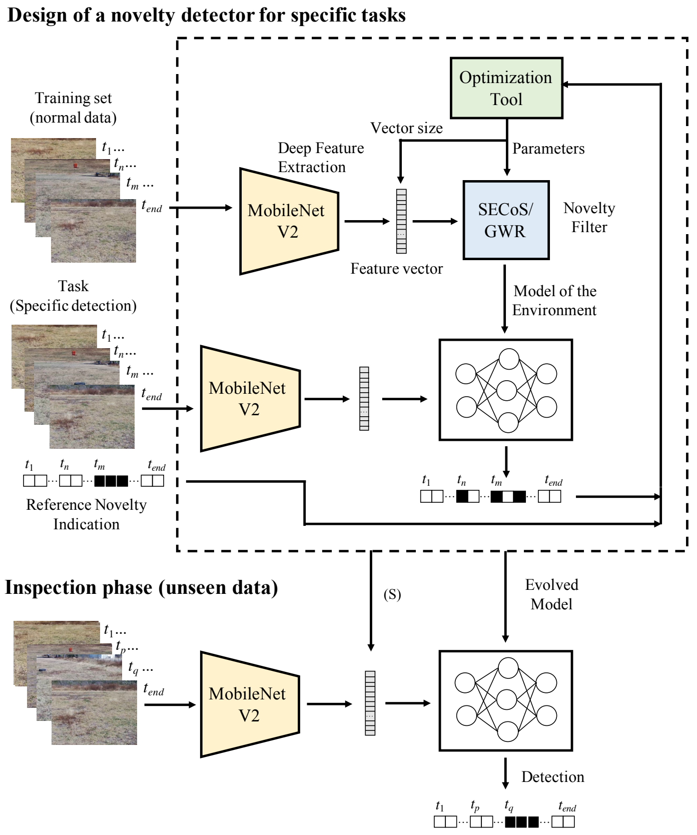


Figure 3.4: Flowchart of the visual novelty detection for specific tasks. In the training phase, the novelty filter learns to detect a specific object. In the inspection phase, the evolved model is used to detect the object(s) in the environment.

3.4 Experimental preparation

We validate the performance of the proposed methods by using images captured by a real robot in outdoor environments. We construct the datasets using these images to train and to test the novelty detection systems. We designed an experiment to compare the deep visual feature extraction technique against largely used visual

Table 3.1: Parameters to be tuned for each novelty detector.

Novelty detector	Parameter	Description
SECoS	η_1	Learning rate 1
	η_2	Learning rate 2
	S_{thr}	Sensitivity threshold
	E_{thr}	Error threshold
GWR	a_T	Activation threshold
	h_T	Habituation threshold
	η	Proportionality factor
	ϵ	Learning rate

features for the problem of visual exploration and inspection. In summary, in this section, we describe the datasets, the methods for comparison, the experimental setup and the evaluation metrics.

3.4.1 Datasets

In this work, we construct the dataset with images captured by the visual sensor of an UAV. For this purpose, we use a Parrot Bebop 2 Drone with a 14 Mpx flight camera. The captured images have a dimension of 1920×1080 pixels, but we constrained the search in the center region of the images with a reduced field-of-view of 640×480 pixels. Figure 3.5 shows the UAV used for data acquisition. It is worth noting that the novelty detector systems receive images of the environment every 250 ms.

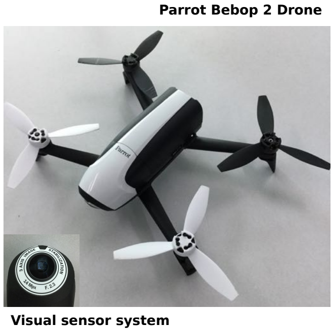


Figure 3.5: Parrot Bebop 2 Drone with a 14 Mpx flight camera. In the bottom-left corner, we show its visual sensor system.

Figure 3.6 illustrates the outdoor environment used in this experiment. The UAV executes its default execution control module to flight over the environment in a rectangular shape. In order to generate the datasets, the UAV should execute the same path several times with different environment setups.

In the first set of experiments, the UAV flew at 2 m above ground with morning light conditions (around 11:00 and 12:00). The original environment contains an

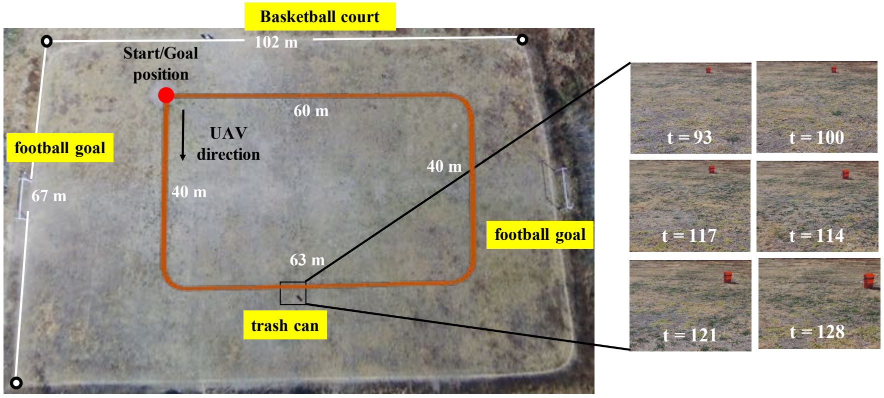


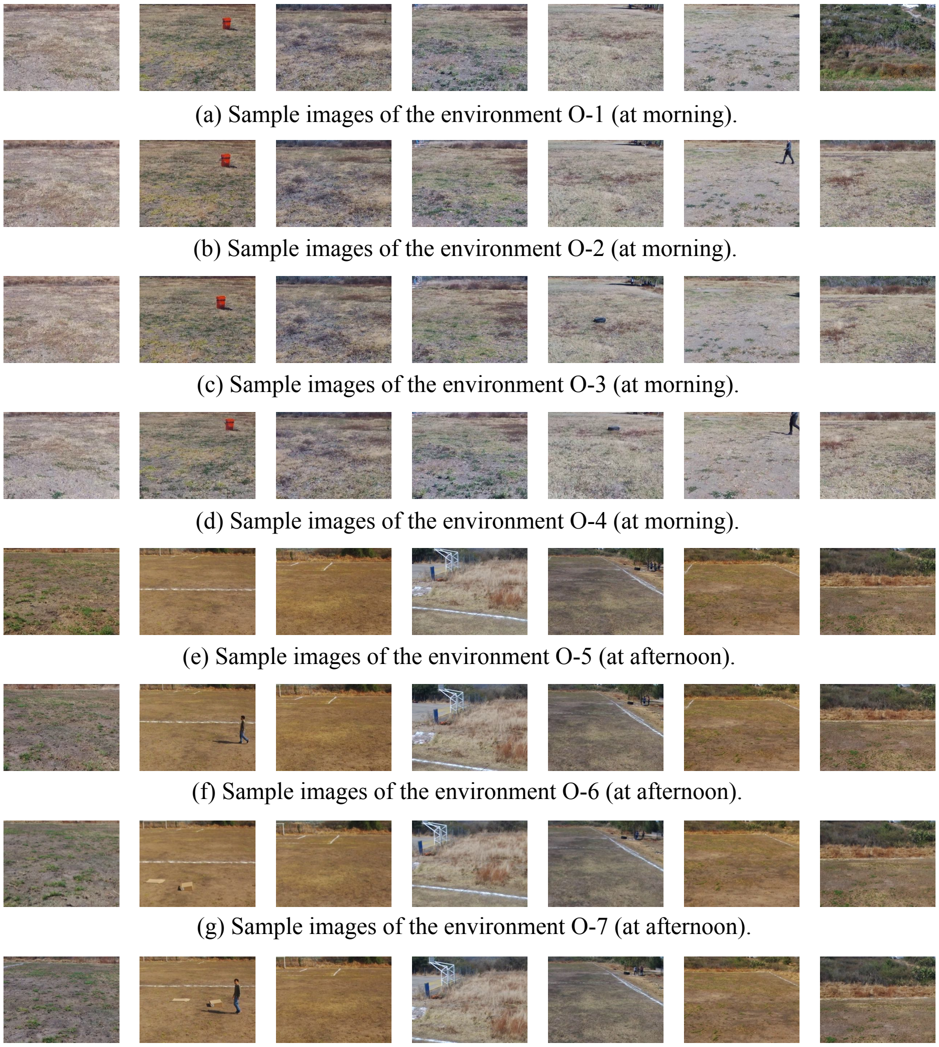
Figure 3.6: Experimental setup: the outdoor environment, and some sample captured images.

orange trash can (we called O-1 to this environment). First, the UAV explored the O-1 environment, executing two times its path. The UAV captured a total of 896 images, 448 for each execution. Then, it executed the inspection phase and captured another 896 images. In this inspection phase, a person appeared in the environment (we denote this new environment as O-2). The sequence contains 60 frames with the person. In the second experiment, we added a tire to the O-1 environment (we denote this environment as O-3). The UAV captured a total of 896 images. The tire is present in 58 frames. Finally, the UAV executed its path in the environment with the person and the tire at the same time. The UAV captured another 896 images in its two path executions. In total, the person is present in 37 frames, and the tire is present in 64 frames. We identify this environment as O-4.

We developed a second set of experiments to test the robustness of the proposed method, considering different scales, types of occlusions, novel objects, and light conditions. In this new set, the UAV flew at 4 m above ground with afternoon light conditions (around 16:00 and 17:00). The methodology to capture the image sequences was similar to the first set of experiments, but with some differences in the settings of the environments. We introduced environment O-5, where the orange trash can was removed. We designed another environment with a person in a different position, and named it O-6. To test the robustness of the proposed method, we added inconspicuous novel objects to the environment O-5 (brown boxes). We denote this environment as O-7. Finally, we set a new environment O-8, where the UAV could visualize how the person occludes the boxes in the environment.

Figure 3.7 shows some sample images of the above environments. Table 3.2 summarizes the environments used for novelty detection, and Table 3.3 reports the data partition of the environments to perform the training and test phases.

In all the experiments, the novelty detectors use the images of the training en-



(a) Sample images of the environment O-1 (at morning).

(b) Sample images of the environment O-2 (at morning).

(c) Sample images of the environment O-3 (at morning).

(d) Sample images of the environment O-4 (at morning).

(e) Sample images of the environment O-5 (at afternoon).

(f) Sample images of the environment O-6 (at afternoon).

(g) Sample images of the environment O-7 (at afternoon).

(h) Sample images of the environment O-8 (at afternoon).

Figure 3.7: Sample images captured by the UAV in the environments: (a) original at morning (O-1), (b) the person at morning (O-2), (c) the tire at morning (O-3), (d) the person and the tire at morning (O-4), (e) empty environment at afternoon (O-5), (f) the person at afternoon (O-6), (g) the boxes at afternoon (O-7), and (h) the person and the boxes at afternoon (O-8).

Table 3.2: Summary of the environments used in the experiments for novelty detection.

Environment	Description	#Normal	#Novel
O-1	Original setup of the environment (at morning).	896	0
O-2	A person in the O-1 environment (at morning).	836	60
O-3	Inclusion of a tire to the O-1 environment (at morning).	838	58
O-4	A person and tire in the O-1 environment (at morning).	795	101
O-5	Empty environment (at afternoon).	896	0
O-6	A person in the O-5 environment (at afternoon).	822	74
O-7	Inclusion of brown boxes to the O-5 environment (at afternoon).	835	61
O-8	A person and boxes in the O-5 environment (at afternoon).	825	71

Table 3.3: Data partition for novelty detection.

Dataset	Exploration	Inspection	Test case (novelty)
D-1	O-1	O-2	A dynamic object (person).
D-2	O-1	O-3	A small conspicuous object (black tire).
D-3	O-2	O-4	A conspicuous object in a dynamic environment (black tire).
D-4	O-3	O-4	A dynamic object in an environment with a static object (person).
D-5	O-1	O-4	Multiple novel objects (person and tire).
D-6	O-5	O-7	Inconspicuous objects (brown boxes).
D-7	O-6	O-8	Occlusion of inconspicuous objects (brown boxes).

vironment of both loops for exploration while only using one loop of the test environment for inspection. The other loop of the test environment is used to evolve the novelty detectors.

3.4.2 Evaluation metrics

To measure the performance of the novelty detectors, we use the confusion matrix shown in Table 3.4. *TP* represents the number of true positives (normal data labeled as normal), *TN* represents the number of true negatives (novel data labeled as novel), *FP* represents the number of false positives (novel data labeled as normal), and *FN* represents the number of false negatives (normal data labeled as novel).

Table 3.4: Confusion matrix to evaluate the performance of the novelty detectors.

Class / Prediction	Normal	Novel
Normal	*TP*	*FN*
Novel	*FP*	*TN*

Different metrics have been proposed to reflect in a single quantity the performance reached by the novelty detectors. Three of the most commonly adopted are F_1 score, accuracy (ACC), and Matthews Correlation Coefficient (MCC). As Özbielge [8], we use these three metrics to evaluate the performance of the novelty detectors. These metrics are defined as,

$$F_1 = \frac{2 \cdot TP}{2 \cdot TP + FP + FN} \tag{3.1}$$

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.2}$$

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{3.3}$$

We also incorporate two additional metrics the True Negative Rate (TNR) and the True Positive Rate (TPR).

3.4.3 Experimental setup

All the algorithms for novelty detection under study can operate online. However, to compare the detectors, we use the same data partition shown in Table 3.3. We implement the SECoS, GWR, and ABC algorithms in the C++ Programming Language. The developed ABC library uses the Mersenne Twister pseudo-random generator of 32-bit numbers. In the case of the deep feature extraction technique, we use the pre-trained MobileNetV2 available in the Keras API and the TensorFlow engine. The experiments were developed in a computer with an Intel Core i5 processor, running at 2.9 GHz and with 16 GB of RAM.

To verify the performance of the detectors, we use three traditional visual feature extraction techniques: the RGB color histograms used by Özbilge [8], the color angular indexing used by Neto [83], and the GIST descriptor used by Kato et al.[84]. We compare the performance of the detectors with these feature extraction techniques against the features extracted by the MobileNetV2 network. In this experiment, the system for automatic design uses the two image sequences in the exploration phase as training and one sequence of the inspection phase as a validation. The goal of the optimization process is to maximize the performance of the detector concerning the F_1 score, the ACC, and the MCC. Therefore, we use the fitness function

$$f = 1 - \frac{1}{3}\Big(F_1 + ACC + \frac{1 + MCC}{2}\Big) \tag{3.4}$$

with $f \in [0, 1]$, $f = 1$ represents the worst case with no data classified correctly and $f = 0$ indicates that the novelty detector under study classifies all the data from the validation correctly. In this experiment, we execute 30 simulations for each novelty detector, and we report the average results to perform the comparison.

3.5 Results and discussion

This section shows and discusses the results of the experiments. We design the specific novelty detectors for each visual feature independently. We find the most suitable size of the feature vector and the parameters of the novelty detection methods for the particular visual exploration and inspection task. In the first part of this section, we compare the results of the proposed feature extraction technique against the well-established feature extraction techniques in the problem of visual novelty detection. Then, we present an analysis of the optimization process of the novelty detectors that use the MobileNetV2 feature extractor. We also show some sample novelty detectors generated by the proposed framework and their visual results. Finally, we discuss some limitations of the proposed methodology.

3.5.1 Deep and traditional features in novelty detection

We use as reference the RGB color histograms used by Özbilge [8], the color angular indexing applied by Neto [83], and the GIST descriptor implemented by Kato et al. [84]. Table 3.5 reports the average performance of the novelty detectors in the inspection phase for each dataset, where CAI represents the color angular indexing technique, hRGB represents the RGB color histograms, and MNF represents the feature extraction method based on MobileNetV2. In the table, we also report the average vector size of the features (*VSize*), and the average size of the learned models of the environment (*MSize*), i.e., the average number of nodes in the models. Note that the CAI descriptor produces feature vectors of four elements. In the rest of the descriptors, the optimization process can produce feature vectors of different sizes. In the table, we mark the best performing method for each metric, according to the specific detector and the particular dataset.

On D-1 dataset, the objective is to learn a model of the original environment O-1, and to detect a dynamic object represented by a person. In this dataset, the feature extraction technique MNF shows the best performance compared to all other visual extraction techniques. The detectors that use the MNF descriptor can generate compact models of the environment and keep higher performance. They show accuracies upper 98%, and MCC near to 0.9. On the second dataset (D-2), the novelty detectors should learn a model of the environment O-1, and identify the black tire as the new object. The proposed method achieved the best performance over all others in this dataset, see the ranking on D-2 dataset in Table 3.5. The average ACC by using both detectors with the MNF technique is around 98%, and the MCC is 0.87. D-3 dataset presents a more challenging situation because the detector should learn a model of the environment with the person and detect the black tire. The environment in the inspection phase includes both the person and the black tire. Under this situation, the novelty detectors that use the MNF also achieve the best performance with ACC values around of 96% (for both detectors), and MCC values of 0.79, and 0.76, for the SECoS and GWR detectors, respectively. On D-4 dataset, the objective is to learn a model of the environment with the tire. In the inspection phase, the person represents the novel object and the black tire represents a normal object. The results show that the MNF technique is the second best (the first is the GIST descriptor) with 96% ACC, and 0.6 MCC for both detectors. On D-5 dataset, the novelty detectors should learn a model of the O-1 environment and detect multiple novel objects (both the tire and the person). The MNF description achieved the best performance with ACC values around 97% for both novelty detectors and MCC values of 0.89 and 0.88 for the detectors SECoS and GWR, respectively.

On the above datasets, the novelty detectors were tested with novel objects that are highly different from the environment. This could facilitate their detection. In the following, we test the detectors in more challenging situations. To this end, we use the D-6 and D-7 datasets generated by the UAV at a different height (4 m) and with a different light condition (images captured at afternoon). In the inspection phase on dataset D-6, we use inconspicuous brown boxes to represent the novel objects. In this dataset, the detectors with MNF feature extraction are the best methods to detect novelties with a ranking of 1.2. Finally, we show the results of the detectors on D-7 dataset. The objective in this dataset is to learn a model of an environment with a person and tire and to detect the brown boxes that can be occluded in some frames by the person. The results show the superiority of the MNF descriptor for novelty detection with MCC values above of 0.9 and ACC values around of 98%, for both detectors.

We then compare the average CPU time to generate the visual features per image on all the datasets. The average time excludes the reading of the image and the post-processing of the visual features. The post-processing only consists in reducing the vector size to the size found by the optimization process. The reduction is through the average of sectors of equal elements. Figure 3.8 shows the average time to generate visual features in all the datasets. hRGB is the fastest method mainly because it only needs to count the number of pixels that belong to a given intensity value. The CAI method is the second fastest method because its computation consists of simple image operations as average, standard deviation, inverse cosine, and dot product. Meanwhile, the GIST descriptor involves more advanced operations. It includes convolution between the image and Gabor filters at different scales and orientations. The MNF is the slower feature extraction technique because it includes

Table 3.5: Average results in the inspection phase over the 30 runs.

Dataset	Detector	Descriptor	***VSize***	***MSize***	***TPR***	***TNR***	$\mathbf{F_1}$	***ACC***	***MCC***
D-1	SECoS	CAI	4.0	17.5	0.9692	0.2750	0.9607	0.9258	0.2865
		hRGB	305.0	12.4	0.9738	0.4571	0.9689	0.9415	0.4673
		GIST	350.5	47.5	0.9867	0.8571	0.9886	0.9786	0.8312
		MNF	169.1	7.1	**0.9922**	**0.9000**	**0.9928**	**0.9865**	**0.8859**
	GWR	CAI	4.0	29.1	0.9520	0.3238	0.9532	0.9127	0.2530
		hRGB	357.3	20.5	0.9757	0.2393	0.9628	0.9297	0.2317
		GIST	398.1	46.7	**0.9900**	0.8452	0.9898	0.9810	0.8418
		MNF	153.3	13.8	0.9899	**0.8869**	**0.9912**	**0.9835**	**0.8646**
D-2	SECoS	CAI	4.0	13.6	**0.9879**	0.0155	0.9620	0.9271	0.0076
		hRGB	337.0	25.3	0.9734	0.0857	0.9567	0.9179	0.0655
		GIST	384.9	37.7	0.8444	0.8333	0.9084	0.8438	0.4295
		MNF	143.3	16.6	0.9806	**0.9976**	**0.9901**	**0.9817**	**0.8729**
	GWR	CAI	4.0	2.4	0.9943	0.0000	0.9649	0.9321	-0.0104
		hRGB	365.4	2.0	**1.0000**	0.0000	0.9677	0.9375	0.0000
		GIST	334.3	79.8	0.8300	0.7821	0.8976	0.8270	0.3758
		MNF	180.8	23.7	0.9852	**0.9548**	**0.9910**	**0.9833**	**0.8729**
D-3	SECoS	CAI	4.0	11.8	0.9426	0.6086	0.9561	0.9195	0.4765
		hRGB	427.2	29.3	0.9642	0.1452	0.9507	0.9075	0.0914
		GIST	269.0	50.2	0.9019	0.6022	0.9323	0.8812	0.3742
		MNF	184.1	27.8	**0.9788**	**0.8484**	**0.9836**	**0.9698**	**0.7881**
	GWR	CAI	4.0	6.5	0.9905	0.1118	0.9632	0.9297	0.1111
		hRGB	445.0	6.1	**0.9922**	0.0118	0.9602	0.9244	0.0024
		GIST	353.4	152.2	0.9117	0.4645	0.9317	0.8807	0.2852
		MNF	216.0	34.2	0.9723	**0.8710**	**0.9812**	**0.9653**	**0.7653**
D-4	SECoS	CAI	4.0	16.9	0.9790	0.0157	0.9703	0.9424	-0.0072
		hRGB	303.2	29.5	0.9745	0.3000	0.9733	0.9489	0.3008
		GIST	315.0	2.2	**0.9912**	**0.8706**	**0.9930**	**0.9866**	**0.8259**
		MNF	147.2	15.8	0.9729	0.8098	0.9825	0.9667	0.6585
	GWR	CAI	4.0	6.1	**0.9947**	0.0000	0.9780	0.9570	-0.0105
		hRGB	289.0	51.9	0.9552	0.3176	0.9633	0.9310	0.2046
		GIST	334.0	15.3	0.9690	**0.9039**	0.9821	0.9665	**0.7279**
		MNF	173.8	15.4	0.9770	0.7784	**0.9840**	**0.9695**	0.6578
D-5	SECoS	CAI	4.0	7.5	0.9765	0.0778	0.9356	0.8802	0.0945
		hRGB	276.5	40.4	**0.9823**	0.1299	0.9414	0.8910	0.1976
		GIST	306.7	24.6	0.9536	0.5764	0.9512	0.9132	0.5516
		MNF	180.0	26.9	0.9813	**0.9472**	**0.9874**	**0.9776**	**0.8916**
	GWR	CAI	4.0	7.8	0.9749	0.1049	0.9361	0.8817	0.1565
		hRGB	331.4	36.6	0.9833	0.0660	0.8978	0.8850	0.0795
		GIST	385.9	46.7	0.9305	0.6403	0.9420	0.8994	0.5475
		MNF	221.8	26.9	**0.9917**	**0.8681**	**0.9880**	**0.9784**	**0.8847**
D-6	SECoS	CAI	4.0	7.9	0.9560	0.0344	0.9439	0.8943	-0.0206
		hRGB	213.2	6.0	0.9270	0.8900	0.9580	0.9246	0.6233
		GIST	245.4	9.5	0.8352	**0.9167**	0.9059	0.8407	0.4707
		MNF	150.2	20.4	**0.9750**	0.8911	**0.9834**	**0.9693**	**0.7950**
	GWR	CAI	4.0	11.6	0.9761	0.0200	0.9535	0.9121	-0.0045
		hRGB	304.3	61.3	0.8946	0.8622	0.9388	0.8924	0.5277
		GIST	289.0	11.5	0.7977	**0.9111**	0.8825	0.8053	0.4194
		MNF	210.5	16.0	**0.9796**	0.8878	**0.9857**	**0.9734**	**0.8107**
D-7	SECoS	CAI	4.0	8.5	0.9730	0.0192	0.9487	0.9028	−0.0085
		hRGB	276.6	6.3	0.9482	**0.9939**	0.9731	0.9516	0.7605
		GIST	444.9	9.9	**0.9867**	0.9364	0.9907	0.9829	0.8831
		MNF	165.2	17.6	0.9855	0.9848	**0.9921**	**0.9855**	**0.9065**
	GWR	CAI	4.0	2.8	**0.9982**	0.0000	0.9609	0.9247	−0.0021
		hRGB	255.7	12.2	0.9369	0.8424	0.9607	0.9300	0.6307
		GIST	370.7	14.4	0.9654	0.9030	0.9783	0.9608	0.7757
		MNF	175.7	8.1	0.9862	**0.9960**	**0.9929**	**0.9869**	**0.9162**

more complex operations in the image, i.e., it is a deep structure with different convolutional layers. However, all the feature extraction techniques in this work can generate visual features in less than 200 ms, time that is susceptible to the proposed visual exploration and inspection tasks.

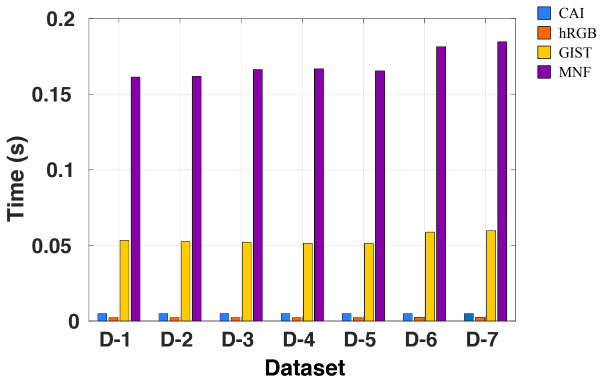


Figure 3.8: Average time (s) to generate the visual features using different descriptors on all the datasets.

Overall, MNF has balanced results in contrast with the baseline methods. The models found by the MNF descriptor and the novelty detectors are compact, with no more than 35 nodes. In most cases, it works better in detecting novelties than the traditional visual descriptors. Besides, we found that traditional visual features need a low number of nodes to represent the environment. However, their low performance concerning the ACC and the MCC indicates that the extracted features are insufficient to differentiate the image in the sequences.

3.5.2 Analysis of the configuration process

Figure 3.9 presents the average fitness value of the best-evolved novelty detectors per iteration in the 30 runs on D-2 dataset. We show the optimization processes of both novelty detectors that use the MNF feature extraction technique. In this figure, we also present the standard deviation of the fitness values through bars. At the begin, the best detectors in the different runs have more variations among them, and this variation is reduced according to the increment in the number of iterations. Analyzing the curve, we can observe that detectors evolve easily on the dataset because they reach fitness values near to the perfect score (zero values), i.e., the optimization process found the appropriate parameter values of the detector for the specific novelty detection task. For the GWR, from the initial to the final iteration, it had a decrement of 0.2591 in the average fitness. The more notable change occurred in the first 20 iterations with a change of 0.2532. For the SECoS detector, the optimization process showed a decrease of 0.3386 in the average fitness from the initial to the final iteration. The more significant change occurred in the first 14 iterations with a change in the average fitness of 0.3341. For the rest of the datasets, the results showed similar behaviors in the optimization process.

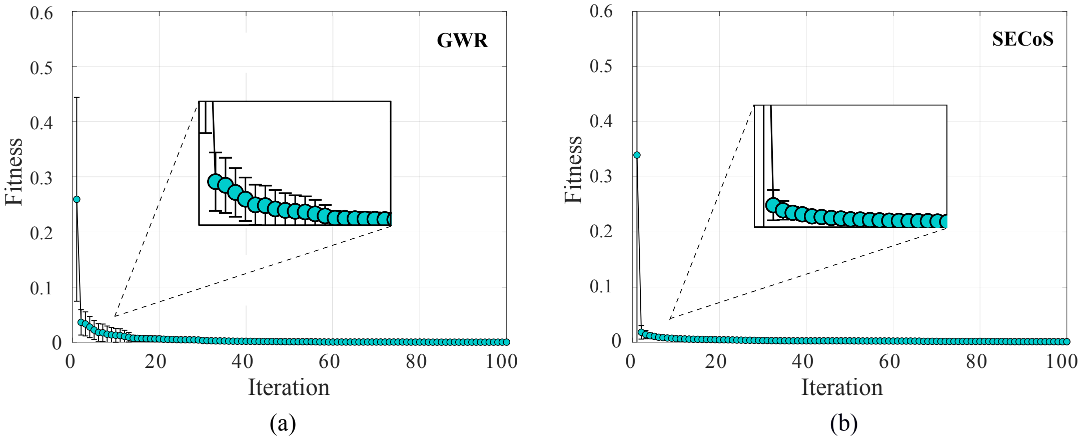


Figure 3.9: Average fitness value of the best-evolved detectors by using the ABC algorithm in the 30 independent runs on the D-2 dataset. The detectors used the MNF feature extraction technique, (a) GWR detector, and (b) SECoS detector.

3.5.3 Automatically designed novelty detectors

We use a SECoS detector with deep features on D-3 dataset to illustrate the effects of task-specific novelty detectors. The designed detector has the following characteristics: $\eta_1 = 0.0183574$, $\eta_2 = 0.4830270$, $A_{thr} = 0.4651190$, $E_{thr} = 0.7776980$, and $VSize = 256$. The proposed tuning technique obtained these parameters. In D-3 dataset, the training of the detector consists of generating a model of the O-2 environment (an environment with a person) and the objective is to detect a black tire in an environment with the tire and the person (this new environment is called O-4).

Figure 3.10 presents the exploration and inspection phases by using the evolved SECoS novelty detector. In the exploration phase, the detector constructs the model of the environment finding the most relevant information as the football goal, the orange trash can, the basketball court, and the person. It is commonly adopted for novelty detectors that the first input will be part of the learned model. The image to the left of the football goal in Loop 1 represents the first input image. We use two loops of the same normal environment (O-2) to train the detector. The evolved detector found a model of 18 nodes to represent the O-2 environment. In the inspection phase, the detector uses this model on the environment O-4 to detect novelties. In this new environment, the detector found in almost all the cases the tire as the novel object with a single false novelty detection. The performance of this particular detector was $TPR = 0.9976$, $TNR = 0.9677$, $F_1 = 0.9976$, $ACC = 0.9955$, and $MCC = 0.9653$.

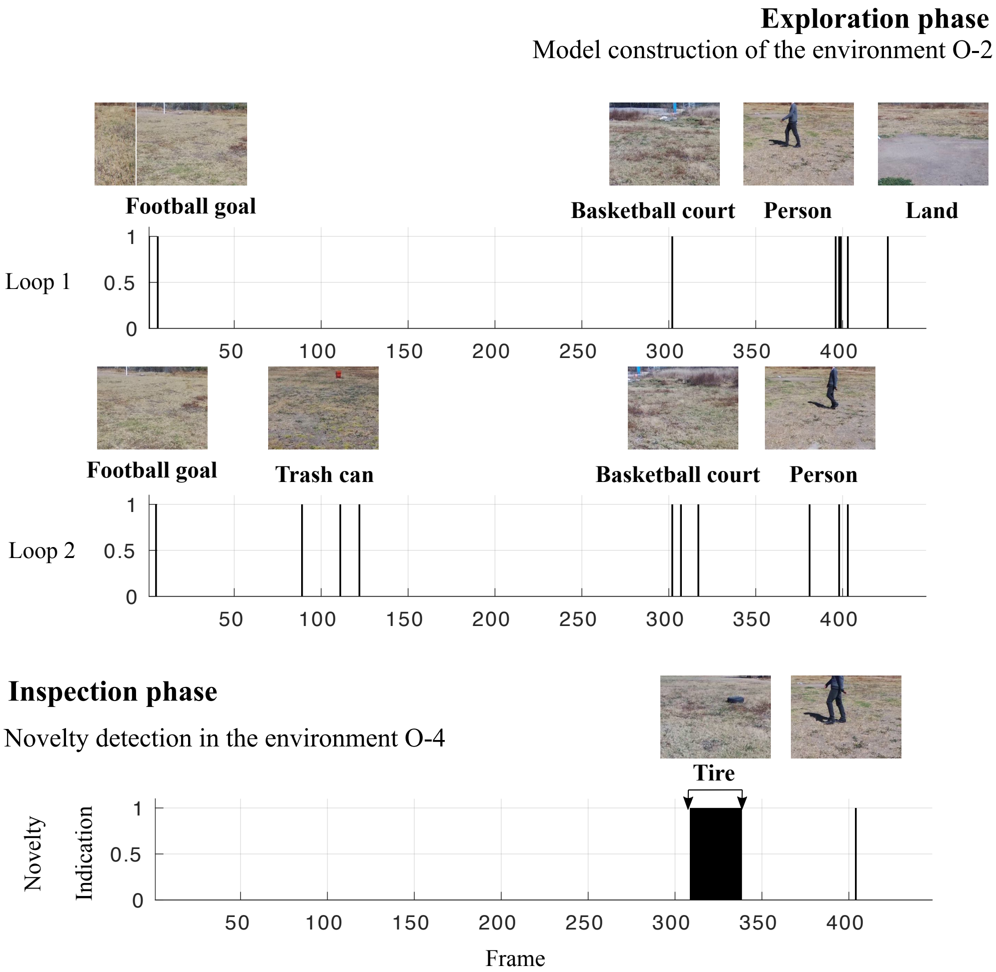


Figure 3.10: Illustration of the visual exploration and inspection task on dataset D-3 to detect the black tire as the novel object. In the exploration phase, the SECoS detector constructs a model of the environment with the person. In the inspection phase, the detector uses this model to detect the black tire.

Table 3.6 presents a set of sample novelty detectors generated by the proposed framework for each dataset. We show the parameter values of η_1, η_2, S_{thr}, and E_{thr} for the SECoS detectors, and the parameter values of a_T, h_T, η, and ϵ for the GWR detectors. The table also reports the found vector size of the deep features for each detector.

Table 3.6: Set of sample evolved detectors generated by the proposed global optimization framework on all the datasets.

Detector	**Dataset**	η_1	η_2	$\mathbf{A}_{thr}$	$\mathbf{E_{thr}}$	***VSize***
SECoS	D-1	0.2002440	0.2428720	0.0545579	0.4578700	170
	D-2	0.2045570	0.2697980	0.5078360	0.2089900	74
	D-3	0.0183574	0.4830270	0.4651190	0.7776980	256
	D-4	0.1456960	0.3827950	0.1083890	0.2627370	75
	D-5	0.0000000	0.0109682	0.2810780	0.5950580	242
	D-6	0.0000000	0.0000000	0.6285940	0.5863690	144
	D-7	0.6577200	0.2922090	0.1750940	0.4164290	96
		$\mathbf{a_T}$	$\mathbf{h_T}$	η	ϵ	***VSize***
GWR	D-1	0.6827340	0.6826510	0.0706664	0.0490785	152
	D-2	0.7888710	0.2963600	0.3931710	0.0000000	101
	D-3	0.5653500	0.3496060	0.4179080	0.0631437	249
	D-4	0.5521850	0.4037900	0.2024040	0.0000000	216
	D-5	0.5756130	0.8404430	0.0000000	0.0000000	256
	D-6	0.7806360	0.7388830	0.2143130	0.0000000	67
	D-7	0.5295850	0.6676220	0.0790152	0.7237070	135

In Table 3.7, we report the performance of the above-evolved detectors. We can observe that the SECoS detectors have similar behavior than the GWR detectors concerning the novelty detection (see the *TNR* values), except on dataset D-5, where the SECOS detector outperforms the GWR. Besides, on datasets D-1, D-3, D-4, D-6 and D-7, the SECoS detectors exceed the GWR concerning the *TPR* values.

Table 3.7: Results in the inspection phase (unseen data) of the sample evolved detectors. Bold values indicate the best result for each metric.

Dataset	**Detector**	***MSize***	***TPR***	***TNR***	$\mathbf{F_1}$	***ACC***	***MCC***
D-1	SECoS	6	**0.9976**	0.9643	**0.9976**	**0.9955**	**0.9619**
	GWR	9	0.9952	0.9643	0.9964	0.9933	0.9440
D-2	SECoS	12	0.9929	1.0000	0.9964	0.9933	0.9470
	GWR	27	0.9929	1.0000	0.9964	0.9933	0.9470
D-3	SECoS	18	**0.9976**	0.9677	**0.9976**	**0.9955**	**0.9653**
	GWR	21	0.9856	0.9677	0.9915	0.9843	0.8900
D-4	SECoS	11	**0.9930**	0.7647	**0.9919**	**0.9844**	**0.7802**
	GWR	19	0.9861	0.7647	0.9884	0.9777	0.7118
D-5	SECoS	23	0.9975	**0.9792**	**0.9975**	**0.9955**	**0.9767**
	GWR	32	0.9975	0.9375	0.9950	0.9911	0.9527
D-6	SECoS	17	**0.9952**	0.9000	**0.9940**	**0.9888**	**0.9094**
	GWR	14	0.9904	0.9000	0.9916	0.9844	0.8770
D-7	SECoS	9	**0.9952**	1.0000	**0.9976**	**0.9955**	**0.9687**
	GWR	4	0.9928	1.0000	0.9964	0.9933	0.9540

Now, we introduce some visual results of the evolved detectors in the environments at morning. In Figure 3.11, the novelty detectors learned a model of the original environment O-1, and detected the person as the novel object. The figure shows the novelty indication of both methods, an image frame in the exploration phase (picture in the upper left corner), and a picture at the same time step in the inspection phase. In a yellow ellipse, we mark the novel object. This figure also presents some successful novelty detections on its right. From these samples, we can observe the advantage of the evolved detectors, which is that they can detect the person at different scales, perspectives, and occlusion levels.

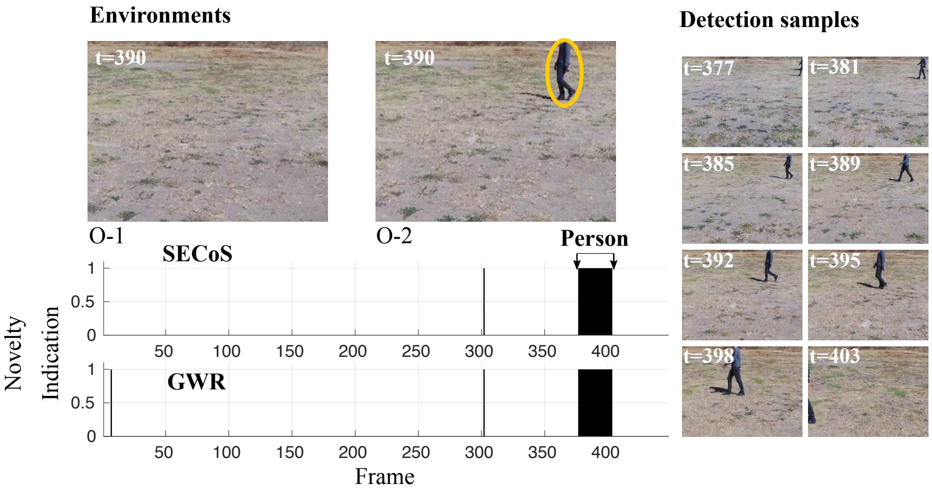


Figure 3.11: Visual results in novelty detection on dataset D-1 (the person as the novel object).

Figure 3.12 shows other example of visual exploration and inspection task. The task consists in learning a model of the original environment O-1 and to detect the black tire in the inspection phase on environment O-3. The detectors found the tire as the novel object in all cases, even the methods can detect novelties with occlusion, see the last detection sample ($t = 334$) where the tire is almost incomplete.

A more challenging example is presented in Figure 3.13. In this figure, the detectors should indicate that the black tire is the novel object and the person is the normal object. In almost all the cases, the methods can detect the novel object. However, some false novelty detections appear with the person. The SECoS showed to be less sensitive to this phenomenon than the GWR. Another challenging problem is to detect the person as the novel object and the tire as the normal object. Figure 3.14 illustrates the performance of both detectors in this situation. Like the above example, the methods can detect the person in almost all the cases and discover false novelties in the tire.

We then present the visual results in detecting both the tire and the person as the novel objects multiple novel object detection). In this case, both methods can identify the tire and the person with only one false novelty detection, see Figure 3.15.

While the previous cases showed results on novel objects that are different from

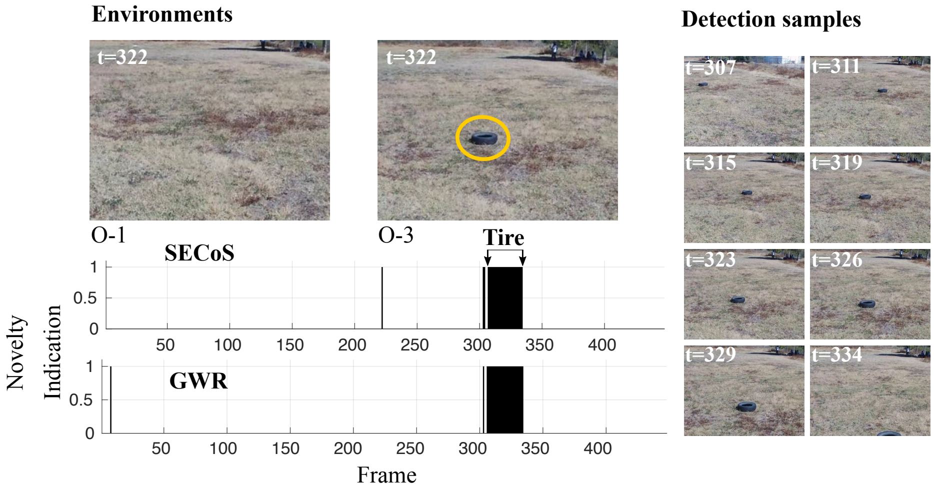


Figure 3.12: Visual results in novelty detection on dataset D-2 (the tire as the novel object).

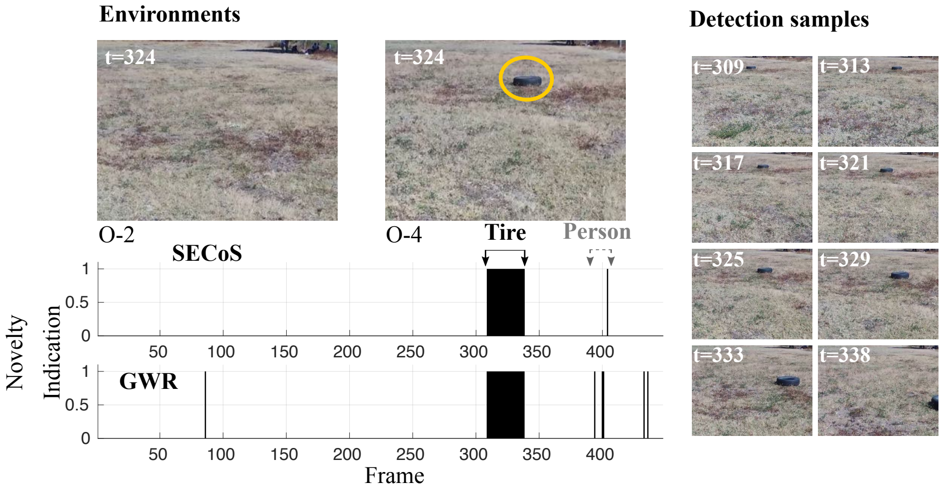


Figure 3.13: Visual results in novelty detection on dataset D-3 (the tire as the novel object, and the person as the normal object).

the environment, the next cases show visual exploration and inspection tasks with inconspicuous novel objects, brown boxes in this experiment. To capture the image frames, the UAV flew at 4 m of height with afternoon light conditions. In Figure 3.16, the problem is to detect the images with the brown boxes through a learned model of the empty environment at afternoon (called O-5 environment). We can observe that the evolved detector in almost of the cases detect the brown boxes with only two false novelty indications.

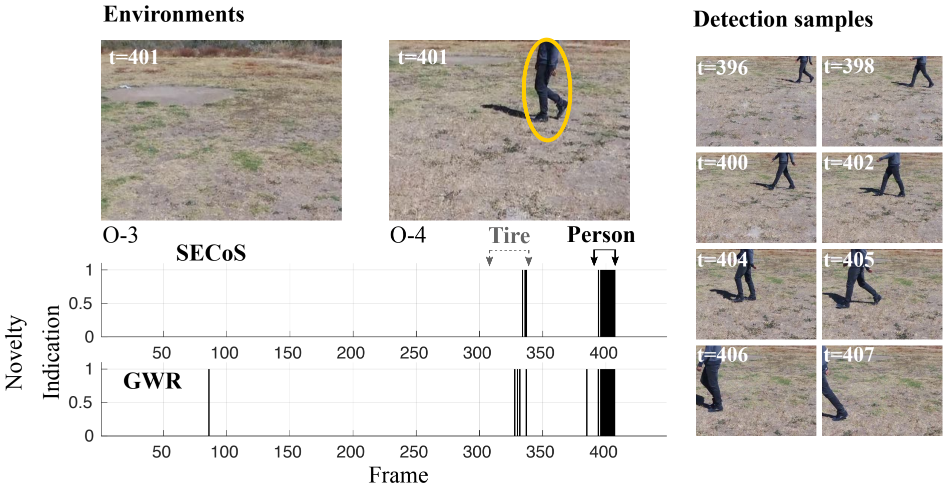


Figure 3.14: Visual results in novelty detection on dataset D-4 (the person as the novel object and the tire as the normal object).

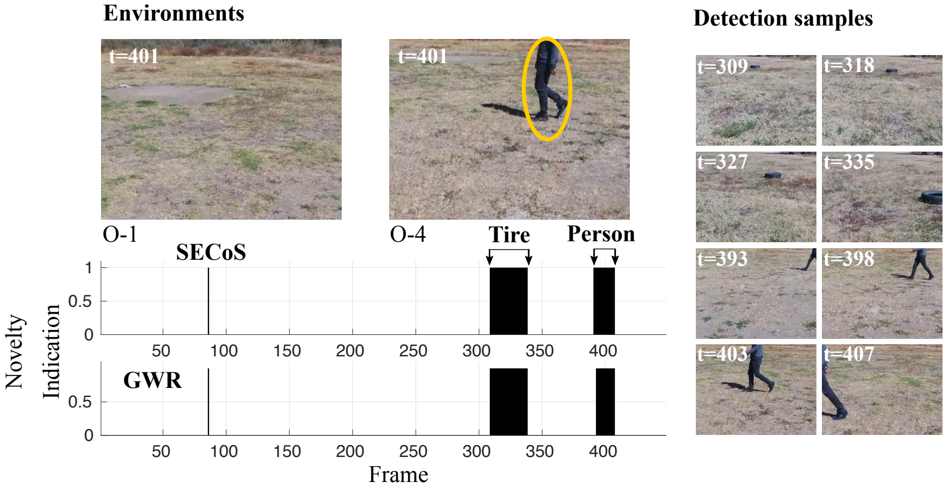


Figure 3.15: Visual results in novelty detection on dataset D-5 (the person and the tire as the novel objects).

Finally, we show the results of the evolved detectors when a person occluded the brown boxes. Figure 3.17 presents this situation. The results show that the evolved detectors learned a model of the environment with the person and detect the images with the brown boxes, even if the person occludes them.

In summary, the visual results show that the evolved detectors can identify in almost all the cases the novelty. The detectors present some false novelty detections. However, it is more critical in this type of problems to detect the novelties than to

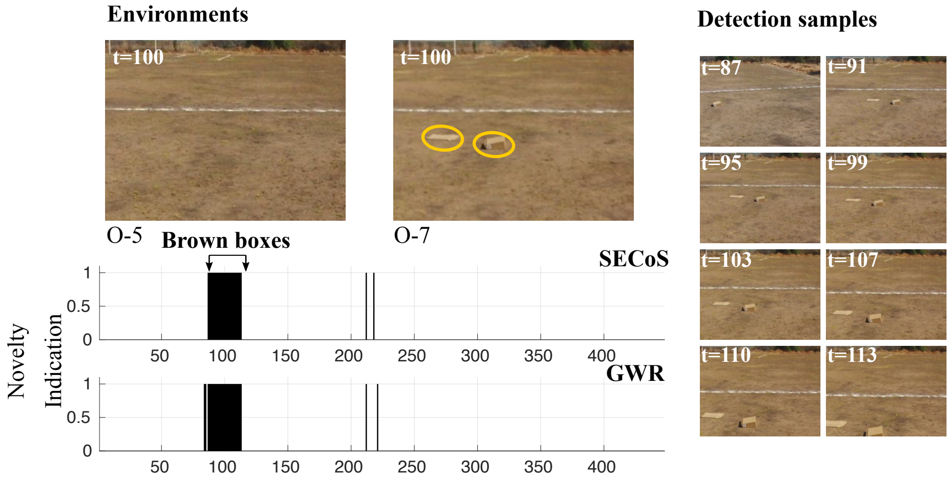


Figure 3.16: Visual results in novelty detection on dataset D-6 (the brown boxes as the novel objects).

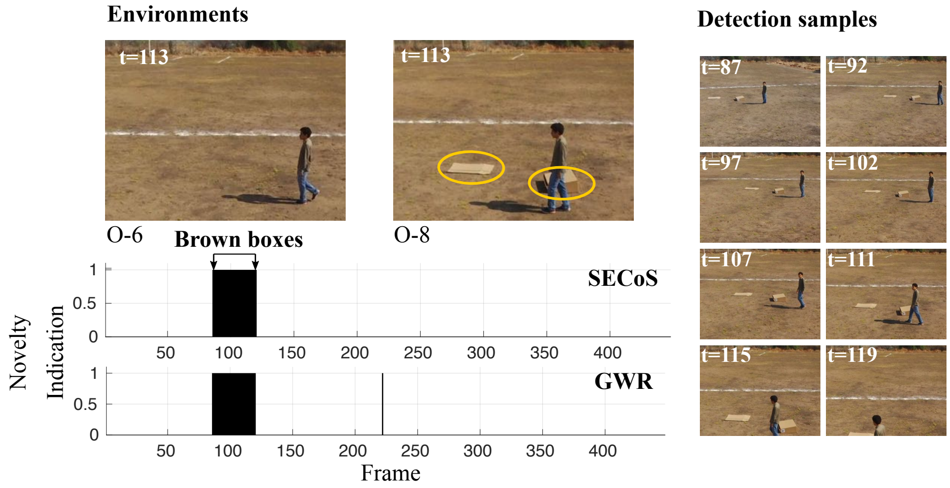


Figure 3.17: Visual results in novelty detection on dataset D-7 (the brown boxes as the novel objects).

miss the novelties and detect all the normal data. Furthermore, the proposed detectors have excellent capabilities in challenging scenarios with illumination changes, scales, and occlusions.

3.5.4 Limitations

The proposed framework addresses the visual novelty detection in exploration and inspection tasks. Although our proposal shows results robust to illumination changes, scale, and occlusion, the evolved detectors present some issues with abrupt perspective changes in images, induced by the flight control of the UAV.

Figure 3.18 shows some failure samples of novelty detections. In the first row, we present some sample images for the training of the evolved novelty detector (GWR in this case). In the second row, we show some sample images in the inspection phase with a change in the perspective induced by the flight control of the UAV. In the exploration phase, the GWR system builds a model of normality of the environment with the tire (Environment O-3). In the inspection phase, the system should detect the person as the novelty in the environment with the tire and the person (Environment O-4). Due to the change in perspectives in the image frames in the inspection phase induced by the flight control module of the UAV, these frames are encoded by information that is not currently represented in the learned model of normality. Therefore, the system detects them as novelty. A possible solution to the problem is to evolve the novelty detectors online to adapt to dynamic changes in the environment. Another possible solution is to learn ad-hoc visual features for the problem. Also, we can explore the incorporation of information from several sensors of the UAV to complement the visual information. With this new information, we can detect new ways of novelties, as novelty based on the position of the objects. All these issues will be subject of future studies.

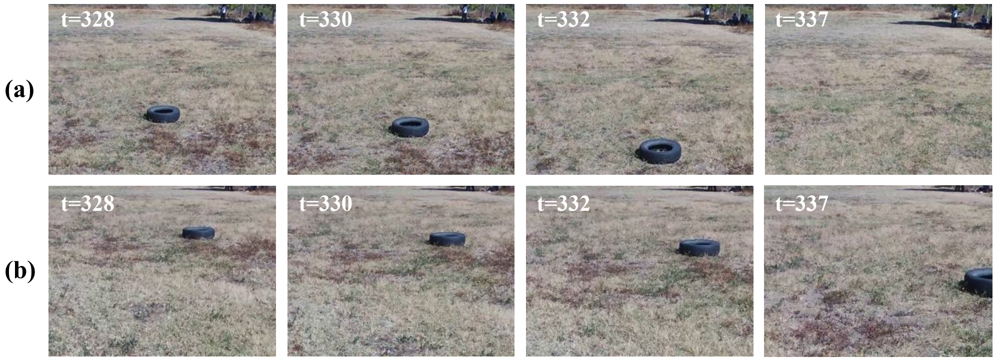


Figure 3.18: Failure cases in the evolved GWR detector on D-4 dataset, (a) sample image frames in the exploration phase, and (b) false novelty indications in the inspection phase. In the exploration phase, the UAV explores the environment O-3. Then, it should detect the person as the novelty in the environment O-4. In the inspection phase, due to changes in perspective in the frames induced by the UAV flight, some false novelty detections are presented because the information of the frame encoding is too different to the learned model.

CHAPTER 4

Automatic design of combination models for salient object detection

This chapter presents an automatic design tool based on genetic programming to select and combine saliency detection algorithms, including machine and non-machine algorithms. Saliency detection is strongly related to novelty localization in images [85, 93, 71].

4.1 Introduction

As different saliency detection algorithms perform well on different sets of images [94], a straightforward strategy leads to a search for a combination model that could improve their responses. Frequently, combination models for saliency detection follow a fixed structure [95, 96]. They combine the input maps through a weight feature integration approach: first, it normalizes the input maps and then uses a pre-established weighted function as a combination operation. In this type of methodology, all of the input maps are commonly used to compute the final saliency map, although some maps have redundant information. That is one of the main reasons why combination models are computationally expensive.

To exclude redundant information, it would be useful to have a combination mechanism that simultaneously finds the algorithms and generates the combination model without being restricted to the well-known combination structures. Motivated by the problem above, we propose to use Genetic Programming (GP) as a tool capable of exploiting both aspects, given that it can be simultaneously used to select the input maps and to generate suitable combination models. In our methodology, GP uses the input maps generated by SOD algorithms and generates new combination operators via fuzzy logic operations and image processing filters. To measure the quality of the candidate models into GP, we propose two fitness functions to represent the goals of useful saliency detection algorithms. After GP finishes its evolutionary process, it generates the best combination model, which is composed of the selected input maps and the more appropriate combination operator. Finally, we validate the performance of the evolved models against state-of-art saliency detection algorithms and other combination strategies. The results show that the proposed methodology can generate competitive combination models that outperform the state-of-the-art algorithms in four well-established benchmark datasets.

The rest of this chapter is structured as follows. Section 4.2 describes our related work. Section 4.3 introduces the proposed methodology to design combination models for saliency detection. Section 4.4 describes the experimental setup. Section 4.5 shows the results of the designed models compared with other state-of-the-art models.

4.2 Related work

Salient object detection (SOD) consists of identifying the object(s) in an image that grabs the attention of humans. Figure 4.1 shows input images of the MSRA-B dataset [97, 23] with their corresponding bounding box in the salient object (first row), and the reference map (second row), typically selected by humans.

In the past few years, researchers have proposed a wide variety of saliency detection models [98]. The most frequently used feature to detect salient objects is the color contrast because they are considered some of the fastest methods in the state-of-the-art [20, 21, 18]. Contrast-based saliency models can be classified into local and global approaches. In the local analysis, a small part of neighborhoods is used to measure the difference between regions. In contrast, in global analysis, the complete image is used to compute the saliency values. In general, global contrast-based methods perform better than local approaches [18]. Contrast-based methods work well in images with a simple background, but they tend to fail with more complex input images, where there are textured regions or changes in light intensity.

An approach to cope with these problems is to apply prior knowledge about the salient objects, e.g., center prior [99, 100], background, and boundary priors

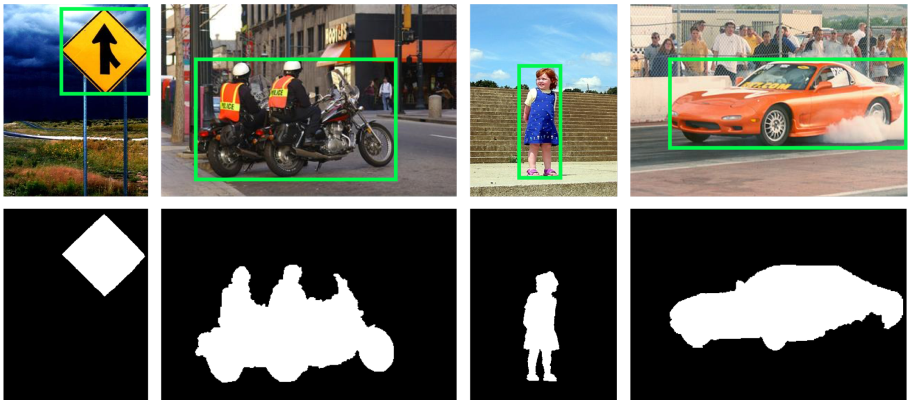


Figure 4.1: Input image with a bounding box in the salient object (first row), and image annotations of the salient object (second row).

[101, 19], high-level priors [102], among others. In the center prior, the methods work with the hypothesis that salient objects lie at the center of the image [99, 100]. In the background and boundary priors; typically, it is considered that the image boundary is part of the background [23]. These methods may fail when a salient object touches even slightly the image boundary. Zhu et al. [101] proposed the boundary connectivity measure to address the above problem. In this approach, the region is assigned as background only if it is densely connected to the image boundary. Other high-level priors have been used to implement SOD models [102], e.g., faces, colors, etc. In high-level priors, the assumption is that people pay more attention to faces or some particular colors. In all the approaches above, the knowledge prior is linked to an assumption about the contents of the image. Therefore the prior is only valid for images satisfying the specific hypothesis.

Another approach to enhance the quality of saliency maps is to perform segmentation processes either in a pre-processing phase [100, 99] or in a post-processing phase [20, 18]. Some widely used segmentation methods are: superpixels [100, 103], graph-based segmentation [23], watershed-like methods [99, 18], GraphCut [21], mean shift algorithm [20], among others. These segmentation methods considerably improve the saliency estimation, but they also increase the computation cost.

Machine learning has gained particular interest for the community, and methods such as the random forest regressor [23], the bootstrap learning algorithm [104], the extreme learning machine [105], and convolutional neural networks [106] are examples of the utility of machine learning in solving the SOD problem. Deep learning is considered a new wave in SOD modeling [107]. The deep learning-based models can generate multi-level and multi-scale features without prior knowledge to find salient regions. Borji et al. [107] classify the deep learning approaches into two categories.

The first type of model works with image patches processed by a Convolutional Neural Network to extract high-level features. After that, the features are used by a multi-layer perceptron (MLP) to detect saliency. However, this type of model can not preserve spatial information. The second type of model, called Fully Convolutional Network (FCN), operates at pixel-level. This model overcomes the problem of spatial information preservation, and it finds the boundaries of salient objects accurately [108]. Despite their success, FCN models still fail in images with low contrast between foreground and background, and images with complex background.

4.2.1 Combination of saliency detection algorithms

During this time, the computer vision community has developed new algorithms to detect salient objects. However, the performance of the algorithms varies with the type of image. Borji et al. [98] stated that no individual SOD algorithm works well for every image in the benchmark datasets. This phenomenon motivated a less explored area into the problem, the combination of saliency algorithms. The main objectives of this area are to generate a consistent performance model regardless of the type of image and exploit the different perspectives captured by existing algorithms to improve the detection [95].

In Fig. 4.2, we can observe the response of two state-of-the-art models for saliency detection, the Minimum Directional Contrast (MDC) [18], and the Minimum Barrier Salient Object Detection (MBS) [19]. Concerning the Mean Absolute Error (MAE), we can see that the best response is obtained using the MDC algorithm for the input image in the first row. On the other hand, for the input image in the second row, the best algorithm is the MBS. The responses of a combination model are shown in the last column to demonstrate qualitatively that a combination helps to improve the performance.

Now, let us introduce some notation into the problem of combining saliency maps. Let $\{M_1, M_2, \ldots, M_m\}$ be a set of m saliency detection algorithms and let $\{S_1, S_2, \ldots, S_m\}$ be the corresponding saliency maps resulting from applying the model to an input image I. The saliency map S_i resulting from the algorithm M_i is an image where the element $S_i(x)$ represents the saliency value for pixel x. The saliency values are normalized, i.e., $S_i(x) \in [0,1], i \in \{1, 2, \ldots, m\}$, to make their aggregation into an overall saliency map S feasible. In the following, we present some works that combine different saliency maps.

Borji et al. [94] proposed one of the first works related to the combination of saliency detection algorithms. In their approach, they define $P(s_x = 1)$ as the probability that a pixel x belongs to the salient object, and $P(s_x = 1|S_i)$ as the estimation of $P(s_x = 1)$ using only the evidence provided by the saliency map S_i.

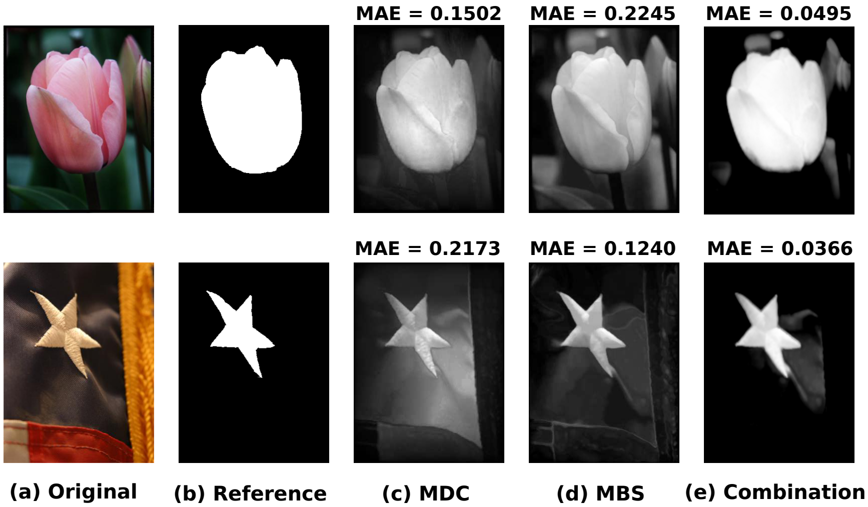


Figure 4.2: Motivation of the combination models for saliency detection. (a) input image, (b) ground truth, (c)-(d) responses of the two state-of-the-art algorithms: MDC [18], MBS [19], respectively, and (e) response a combination model.

They assumed independence between the individual algorithms, and then combined the different saliency maps through:

$$S(x) = P(s_x = 1|\mathbf{f}_x) \propto \frac{1}{Z}\sum_{i=1}^{m} \zeta(S_i(x)) \tag{4.1}$$

where $\mathbf{f}_x = (S_1(x), S_2(x), \ldots, S_m(x))$ is the vector composed of the saliency values for pixel x, and $\zeta(\cdot)$ is a real valued function which can take the following form:

$$\zeta_1(z) = z;\ \zeta_2(z) = exp(z);\ \zeta_3(z) = -\frac{1}{log(z)}$$

Inspired by Borji et al. [94], Mai et al. [95] proposed a logistic regression-based aggregation model, as follows:

$$P(s_x = 1|\mathbf{f}_x;\ \lambda) = \sigma\left(\sum_{i=1}^{m} \lambda_i S_i(x) + \lambda_{m+1}\right) \tag{4.2}$$

with $\lambda = \{\lambda_i | i \in \{1, 2, \ldots, m+1\}\}$ being the set of model parameters and $\sigma(z) = 1/(1 + exp(-z))$.

Furthermore, Naqvi et al. [96] have extended the models above into a general form that considers the importance of normalization, weighting, and integration steps for the aggregation process. They propose a weighted feature combination model template described by:

$$S = \circ w_i \zeta(S_i) \tag{4.3}$$

where $\circ$ is the integration function, w_i is the weight associated to the saliency map S_i, and $\zeta(\cdot)$ is a normalization function.

The objective of the integration function is to combine the individual maps to generate the overall saliency map S. Naqvi et al. [96] suggest several functions including:

$$\sum_{i=1}^{m} w_i \zeta(S_i), \ \prod_{i=1}^{m} w_i \zeta(S_i), \ \left(\frac{1}{m} \sum_{i=1}^{m} \frac{1}{w_i \zeta(S_i)} \right)^{-1}$$

The normalization function serves to apply some conditioning in the response of the individual methods. To this purpose, the authors use the following set of functions:

$$\zeta(z) \in \left\{ z, exp(z), -\frac{1}{log(z)}, \frac{1}{1+exp(-z)}, z \cdot (Z_{max} - \bar{z})^2, z + z * \mathcal{D}\mathrm{o}\mathcal{G} \right\}$$

where Z_{max} is the global maximum of z, $\bar{z}$ is the average of local maxima in Z and $\mathcal{D}\mathrm{o}\mathcal{G}$ is a difference of Gaussians.

All of the approaches discussed above share the potential problem of ignoring the interaction between neighbor pixels. To address this, Mai et al. [95] have also proposed a system based on Conditional Random Fields (CRF) to capture the relationships among neighbor pixels. The parameters of this model are optimized using a training step, and the saliency value for each pixel is the *a posteriori* probability resulting from the trained CRF.

More recently, researchers have proposed new techniques to combine different saliency detection algorithms. Chen et al. [109] tested different combination strategies applied to saliency detection: linear aggregation, non-linear aggregation (e.g., combination based on the median), and naive Bayes classification. Wei et al. [110] proposed the use of Dempster-Shafer Theory for fusing saliency maps. Alternatively, Wang et al. [111] proposed transformation-based fusion techniques to combine different saliency maps. The transformation consists of two steps: 1) the normalization of the input maps and 2) the application of fusion rules. In the normalization, they generate input saliency maps with zero mean and unit standard deviation. In applying fusion rules, they proposed three fusion rules based on their performances: the sum rule, the min rule, and the max rule. Apart from the models above, there exist techniques that generate combination operators according to the image type [112, 90].

4.3 Materials and methods

Previous approaches use a specific structure to combine different saliency algorithms. These approaches commonly require to compute all of the input maps to obtain the final saliency map. Instead, this work proposes to construct combination models without defining a priori the combination model or specifying the saliency algorithms to be combined. In our methodology, the GP evolves models that simultaneously select the input algorithms and generate an appropriate combination structure by using simple operators embedded into the GP.

GP has been successfully applied to solve image processing and computer vision applications, e.g., the design of interest point detectors [113], the design of edge detectors [114], the combination of change detection algorithms [115], the automatic generation of image descriptors for image classification [116], and others. Applications based on GP more related to the proposed work include the design of visual attention models to detect particular objects in images [117], and object detectors [118].

Previous models differ from our approach in one key aspect. Whereas the previous models search for a model to segment a specific object from the background, the proposed method searches for a model that segments all foreground regions from the background. It is worth mentioning that foreground regions can include several types of objects.

4.3.1 Overall system

Figure 4.3 shows a graphical overview of the proposed combination framework (GP-framework in short). GP-framework uses a dataset to evolve the combination models. The system splits the dataset into training, validation and testing subsets that include 50%, 10%, and 40% of the total number of the images in the dataset. The dataset needs to provide the binary ground truth of the salient objects in the images. For the training phase, the GP-framework randomly selects a few instances of the training set. Then, the system computes the response of the saliency algorithms for each image in the training subset. Both the resultant saliency maps and the ground truth are fed to a GP process. The evolutionary process finds the best-fitted individual for each run. This individual represents the program of a combination model. The GP runs r times to obtain r combination models. The selection criteria to chose the overall best combination model can be based either on the tests on the validation subset or on specific criteria from the performance results on the training subset.

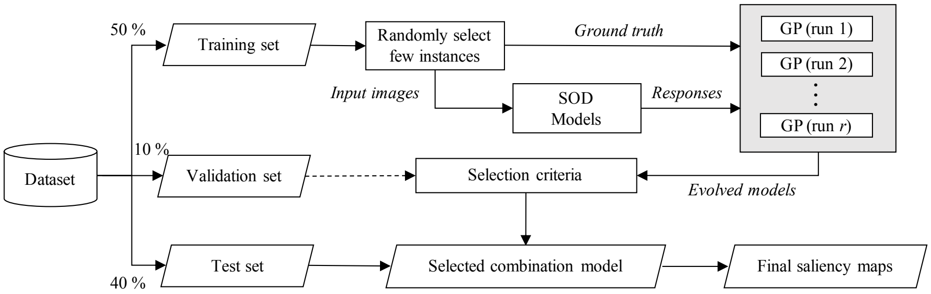


Figure 4.3: Graphical description of the proposed GP-framework to combine responses of different SOD models.

4.3.2 Evolutionary process based on genetic programming

In our proposal, we use a conventional tree-based GP implementation. Each individual in the population encodes a combination model for SOD. When the evolutionary process starts, the initial population is generated using a ramped half-and-half initialization method with a set of terminals $\mathcal{T}$ and a set of functions $\mathcal{F}$. The terminal nodes are the saliency maps of the input saliency detection algorithms, and the function nodes include well-known image processing operations and combination operations typically used for multiple criteria decision making.

At each GP generation, the population of combination models is evaluated by applying all of the training instances as input and by measuring the similarity of the responses of the model to the corresponding ground truth. A fitness function is used to assign a score to each combination model. The next step is to select a subset of combination models from the population and to apply to them evolutionary operators (reproduction, crossover, and mutation). The GP selects the combination models using roulette-wheel selection and applies the genetic operators following the original scheme proposed by Koza [14]. The GP evolves the models during a maximum number of generations. Finally, we extract the best-evolved model.

Next, we describe more details about the set of terminals and functions and the fitness functions used by our proposed GP-framework.

4.3.3 Terminal set $\mathcal{T}$

A large number of SOD algorithms exist in current literature. Nevertheless, we have chosen to use some of the fastest methods available in the state-of-the-art for computing the input saliency maps. The main reason is to avoid an excessive computational cost to obtain the output of the individual SOD algorithms. Despite this, we do not restrict to our GP-framework from using other input saliency algorithms.

We have chosen the Frequency-Tuned approach (FT) [20], the Histogram-based Contrast (HC) [21], the MDC [18], and the MBS [19] to construct the terminal set. In Figure 4.4, we have an example of the output of these SOD algorithms for an input image. The first three methods are color contrast approaches, and the last one is a method that exploits background and boundary priors. It is worth mentioning that we only combine the raw saliency maps generated by executing the program versions provided by the authors.

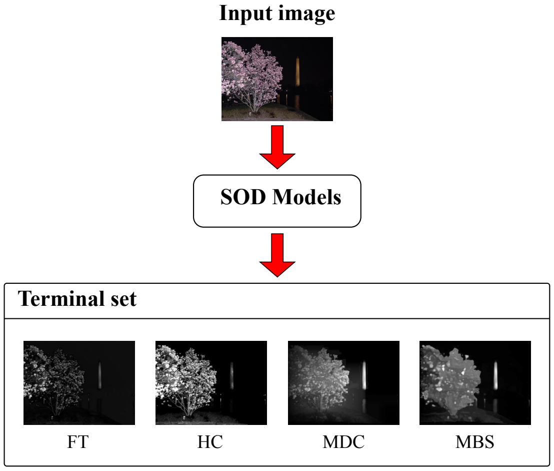


Figure 4.4: Responses from different SOD models used to construct the terminal set.

4.3.4 Function set $\mathcal{F}$

The function set is an essential component for the GP to be able to generate combination models for saliency detection. The proposed function set includes 11 operations: 6 using a single argument, 4 using two arguments and one using three arguments. The function set is composed of: 1) well known image processing operators, for example, low-pass filters and morphological operations; and 2) combination operators used in multiple criteria decision analysis, for example, fuzzy logic operations including the triangular norm (a generalization of the logical conjunction) and, the triangular co-norm (a generalization of the logical disjunction).

All the functions operate with single-channel images to guarantee the closure property, particularly the type consistency. That is, the input to each operation is one or more single-channel images, as required by the operation, and the output is also a single channel image having the same size as the input images. Regarding evaluation safety, all of the proposed functions satisfy this property.

Inspired by the functions proposed by Bianco et al. [115] for the combination of change detection algorithms, we propose the function set shown in Table 4.1. The main difference in using these functions is that the proposed method combines the saliency responses of SOD models with values in $[0,1]$, instead of combining change detection algorithms with binary images.

Table 4.1: Function set $\mathcal{F}$ used in the GP evolutionary process.

Name	**Arity**	**Description**	**Type**
PTN	2	Product t-norm: $I_1 \cdot I_2$	Fuzzy logic conjunction
PTC	2	Product t-conorm: $I_1 + I_2 - I_1 \cdot I_2$	Fuzzy logic disjunction
LTN	2	Lukasiewicz t-norm: $\max(I_1 + I_2 - 1, 0)$	Fuzzy logic conjunction
LTC	2	Lukasiewicz t-conorm: $\min(I_1 + I_2, 1)$	Fuzzy logic disjunction
AMT	3	Average of the majority, $th = 0.5$	Decision rule
NOT	1	Not operation: $1 - I_1$	Fuzzy logic negation
MB	1	Median blur: 5×5 kernel	Low-pass filter
BL	1	Blur (average filter): 5×5 kernel	Low-pass filter
GB	1	Gaussian blur: 5×5 kernel	Low-pass filter
ERO	1	Erosion operation: 3×3 kernel	Morphological operation
DIL	1	Dilation operation: 3×3 kernel	Morphological operation

Pixel-wise operations include implementations of the fuzzy logic operations, both for the product and Lukasiewicz approaches. The interest of using this kind of operations instead of using the binary logic operations is because they can handle continuous values in $[0,1]$ that result from the normalization of the saliency maps.

The average of the majority operation (AMT) tries to emulate a binary majority voting decision rule for continuous values in $[0,1]$. Let us take I_1, I_2, I_3 as the input images and let us define a threshold value th. For each pixel x, we define three sets: $\mathcal{A}(x) = \{I_1(x), I_2(x), I_3(x)\}$; $\mathcal{B}(x) = \{a \in \mathcal{A}(x) \,|\, a \geq th\}$; and $\mathcal{C}(x) = \mathcal{A}(x) \setminus \mathcal{B}(x)$.

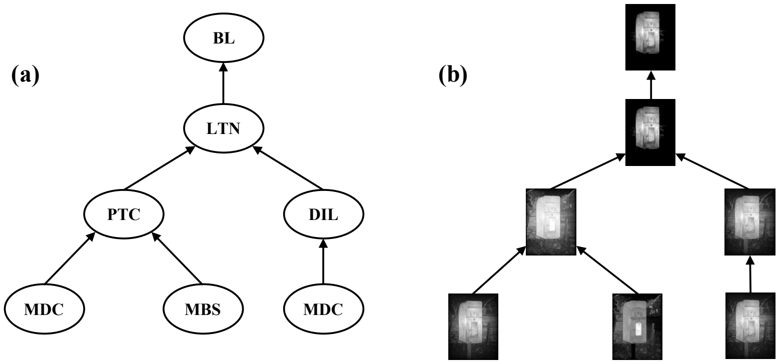
If we denote O as the output image, then the rule to obtain $O(x)$ is given by:

$$O(x) = \begin{cases} \dfrac{1}{n(\mathcal{B}(x))} \displaystyle\sum_{i=1}^{n(\mathcal{B}(x))} B_i(x) & \textit{if } n(\mathcal{B}(x)) \geq n(\mathcal{C}(x)) \\ \dfrac{1}{n(\mathcal{C}(x))} \displaystyle\sum_{i=1}^{n(\mathcal{C}(x))} C_i(x) & \textit{Otherwise} \end{cases} \tag{4.4}$$

where $n(\cdot)$ is the set cardinality operator, $B_i(x)$ is the i-th element of $\mathcal{B}(x)$ and $C_i(x)$ is the i-th element of $\mathcal{C}(x)$.

Region-based operations are included to take into account the interactions of each pixel with its neighborhood. For this end, the set of functions include two types of image processing operations: blurring operations and morphological operations. The blurring operations are implemented using a 5×5 pixels kernel. These three operations act as low pass filters that spread the intensity of the center pixel into their neighbors. Morphological operations like erosion and dilation are useful to reduce and grow, respectively the contour of a region in the image according to a structuring element. In this work, we applied 3×3 rectangular structuring elements.

Figure 4.5 shows an example of the use of the function set to represent a possible evolved combination model. Figure 4.5(a) describes the operators and their relationships, needed to compute the final saliency map. Figure 4.5(b) shows the step-by-step execution of the corresponding evolved structure.



Expression: BL(LTN(PTC(MDC, MBS), DIL (MDC)))

Expanded expression: BL(max(MDC + MBS – MDC · MBS + DIL(MDC) - 1, 0))

Figure 4.5: Illustration of a combination model using the proposed approach: (a) structure, and (b) step-by-step execution.

4.3.5 Fitness functions

In order to evaluate the GP individuals, we need to consider that the output of each candidate combination model is a continuous-valued saliency map. Then a natural fitness measure is to evaluate the fitness score of the evolved model by using an image similarity metric between the output of the model (the GP individual) and its corresponding ground truth image. One of the most used metrics in works using GP is the *F-measure*. Nevertheless, this metric is commonly adopted in classification tasks where the images to be compared are binary images. In the proposed approach, we have a different case: output images are composed of values in $[0, 1]$, and ground truth images are composed of binary pixels $\{0, 1\}$.

This work proposes two alternatives for fitness evaluation:

1) A fitness function based on the Squared Euclidean Distance (SED) of the two images. Let S be the output obtained from the execution of the candidate model and G be the ground truth available for the image under test. The fitness function f_1 is then defined as follows:

$$f_1 = 1 - \frac{||G - S||^2}{s} \tag{4.5}$$

where s is the number of pixels in S (or G, given that both images are the same size).

For the function f_1, a value of 0 represents the worst case, while a value of 1 indicates that both images are identical. The main drawback of this fitness function is the indistinguishability of the cases where the same value is generated either by a small error spread over a large number of pixels; or by a significant error on a small region in the image. The former situation is difficult to handle because it labels non-salient regions that are kept as salient during the evolutionary process.

2) The second option for the fitness evaluation is to use a thresholding step to convert the output image from a candidate combination model into a binary image, and a further step for applying a similarity metric for binary images. For the thresholding step, we use a well-accepted method in saliency detection, known as image-dependent adaptive thresholding [20]. This methods defines the threshold t_α as two times the average value of the saliency map: $t_\alpha = 2 \times \text{mean}(S)$.

In SOD problems, the foreground region is typically smaller than the background non-salient region. That can cause problems during the GP process because of this cardinality unbalance. A metric particularly useful in unbalanced classification problem is the *MCC*. The MCC takes values ranging form -1 to 1, where -1 is the anti-correlation case and 1 is the perfect correlation case. Using this, the second fitness function f_2 is defined as:

$$f_2 = \frac{1 + MCC}{2} \tag{4.6}$$

f_2 can take values in $[0, 1]$, with 0 indicating the matching of no pixels in both images, and 1 indicating the perfect matching of both images.

Worth noting is that for each candidate combination model, the GP process computes the fitness score as the average of the fitness function over the training set. For all the two fitness functions described above, we have performed independent tests in order to determine which one is best suited for the SOD problem.

4.4 Experimental setup

In this section, we present the public datasets used to evaluate the performance of the evolved combination models. We also discuss the evaluation metrics, the baseline models, the description of the experiments and the parameter settings.

4.4.1 Benchmark datasets

We chose four public benchmark datasets to evaluate the performance of the combination models generated by the proposed GP-framework, including: **MSRA-B** [97, 23], **ECSSD** [119], **iCoSeg** [120], and **SED2** [121]. We take into account two different aspects to select the datasets: 1) datasets widely used in SOD modeling and 2) datasets that cover a wide variety of images for SOD.

MSRA-B is a dataset typically used to compare different SOD models. It contains 5,000 images, extending the 1,000 images provided by [20]. The images of this dataset commonly contain a single salient object and simple background. **ECSSD** dataset incorporates more complex background structures compared with the previous one. In total, it includes 1,000 images recovered from the Internet. We also evaluate the capability of the models to detect multiple salient objects in the same image. To this end, we select **SED2** and **iCoSeg** datasets. **SED2** dataset contains 100 images with two salient objects for each image. **iCoSeg** dataset includes some images with more than two salient objects. In this dataset, users segmented foreground and background regions through automatic recommendations. In total, it contains 643 images. All the aforementioned public benchmark datasets provide pixel-wise binary annotations. Figure 4.6 shows some representative images and their corresponding ground truth for each dataset.

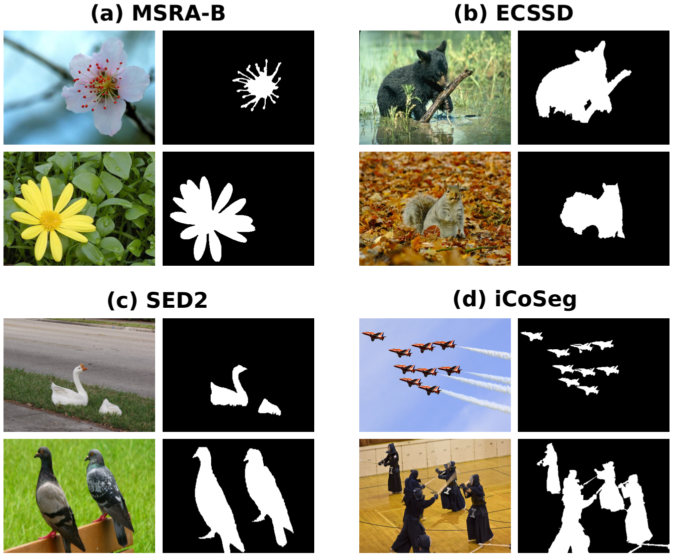


Figure 4.6: Representative images extracted from the four benchmark datasets: (a) **MSRA-B**, (b) **ECSSD**, (c) **SED2** and (d) **iCoSeg**.

4.4.2 Evaluation measures

We use three standard measures for evaluating SOD models. The first two evaluation metrics use the intersection regions between an estimated saliency map (its binary version) and the ground truth annotations. The evaluations include the *Precision-Recall* curve and the *F-measure* curve. As our third evaluation metric, we use the *MAE* to bring a more direct comparison between an estimated saliency map and the ground truth.

Precision-Recall curve

Given a saliency map S, we can convert it into a binary mask S^* by using a fixed threshold and compute the *Precision* and *Recall* values by comparing the mask S^* with the binary ground truth G:

$$Precision = \frac{|S^* \cap G|}{|S^*|}, \quad Recall = \frac{|S^* \cap G|}{|G|} \tag{4.7}$$

where $|\cdot|$ operator counts the number of non-zero elements, $Precision \in [0,1]$, and $Recall \in [0,1]$.

We can draw the *Precision-Recall* curve by computing n pairs of average precision and recall values with a set of thresholds that vary from 0 to 255. Each image generates a pair of precision and recall values per threshold. The final curve is the average of the pairs of precision and recall values for all images in a dataset. Note that the recall value increases when the threshold value decreases.

F-measure curve

For a more comprehensive evaluation, we use the *F-measure* that combines the precision and the recall values. *F-measure* is defined as the weighted harmonic mean of precision and recall:

$$F_\beta = \frac{(1+\beta^2)\cdot Precision \cdot Recall}{\beta^2 \cdot Precision + Recall} \tag{4.8}$$

where $\beta^2 = 0.3$ is used to increase the importance of precision more than recall [20], and $F_\beta \in [0,1]$.

The reason behind $\beta^2 = 0.3$ is that precision is more important than recall in SOD modeling because it is easy to obtain a $Recall = 1$ assigning the whole image as foreground region [98]. We can draw the *F-measure* curve in a similar way to the *Precision-Recall* curve. For this curve, we have to consider that a point consists of a threshold (that varies from 0 to 255) and an average *F-measure* value. In our experiments, we apply a step of 1 in the threshold to draw both curves.

Mean absolute error

Neither *Precision-Recall* curve nor *F-measure* curve consider the true negative values, i.e., these metrics favor the methods that correctly mark salient regions as salient, but they do not favor the methods that mark non-salient regions as non-salient. To consider all the cases, we develop a direct comparison between the saliency estimation and the ground truth with the *MAE*. The *MAE* is defined as,

$$MAE = \frac{1}{s}\sum_{x=1}^{s}|S(x) - G(x)| \tag{4.9}$$

where s is the number of image pixels, $S(x) \in [0,1]$ is the saliency value at the pixel x, and $G(x) \in \{0,1\}$ is the value of the pixel x in the ground truth.

4.4.3 Baseline methods

Typically, the combination models for saliency detection are evaluated by comparing their final responses against the responses of the input algorithms [110, 90]. In this paper, we use two sets of baseline methods. The first one includes saliency detection methods that are fast to produce their outputs. We use the FT [20], HC [21], MDC [18], and MBS [19] models as reference. The second set includes four machine learning-based saliency detection algorithms with higher performance, DHSNet [22], DRFI [23], DSL [24], and LEGS [25]. The response of each input algorithm is compared to the combination model obtained from the proposed approach.

We also include in the tests four classic combination models applied to the set of input algorithms, and three EC-based combination techniques. For the classic combination models [111] , we use: the sum rule (AVG), the min rule (MIN), the max rule (MAX) and the top 2 fusion method (TOP2) [122]. To combine the maps of the input algorithms with these fusion strategies, the maps are first normalized [111]:

$$\mathcal{N}(S_i) = \frac{S_i - \mu}{\sigma} \tag{4.10}$$

where S_i is the saliency map of the input algorithm M_i, μ is the mean saliency value of S_i, and σ is the standard deviation of the saliency values in S_i.

TOP2 model is defined as the average between the outputs of the two best input algorithms, in the first group of input algorithms, the MDC-MBS, and in the second group the DSHNet-DSL algorithms.

For the case of EC-based combination techniques, we use the GA-based approach for feature combination [96], the Constrained Particle Swarm Optimization (CPSO) [123], and the Particle Swarm Optimization (PSO) [124]. These techniques use the same input algorithms to compare their performances.

4.4.4 Description of the experiments

The proposed method aims to find a combination model that outperforms the input algorithms and the other combination strategies. For this purpose, we develop two experiments described as follows.

In the first experiment, we use the first set of input algorithms: FT, HC, MDC, and MBS. For all the EC-based methods, we select a subset of instances from the training set proposed by Jiang et al. [23] to learn the combination models. Jiang et al. [23] split **MSRA-B** into training, validation, and test sets. We select at random the training instances and use the same instances for all the EC-based methods. We train the methods using a limited number of training instances to demonstrate that it is possible to evolve good combination models under this consideration. In our experiment, we use 140 training instances. The objective of the experiment is to investigate two aspects: 1) the effectiveness of the evolved models using test images from the same domain, and 2) the generalization capabilities of the resultant models using images from a different domain. In the first case, we evaluate the performance of the evolved models in the test subset of **MSRA-B**. In the second case, we test the evolved models in all the images provided by the datasets: **ECSSD**, **SED2**, and **iCoSeg**.

In the second experiment, we replace the input algorithms with the second set of baseline methods. These new algorithms are machine learning-based models with the highest performance in the state-of-the-art: DHSNet, DRFI, DSL, and LEGS. The goal of this experiment is to verify if the proposed methodology is capable of improving the best algorithms in the state-of-the-art. We develop the training and test in the **ECSSD** dataset, due to most of the authors provide the precomputed saliency maps in this dataset. We use the same number of training instances than the above experiment and evaluate the performance of all the models in the rest of the images.

In both experiments, we resize the input saliency maps and the ground truth to 200 × 200 pixels to reduce the computational cost in the training phase. We adopt this technique based on the works [96, 124]. However, in the test phase, all the evaluations are performed considering the original size of the input images. We run the experiments 31 times due to the stochastic nature of the EC-based techniques. All the experiments were developed in an Intel Core i5 computer, with a processor running at 2.7 GHz and 8 GB of RAM. We implement the GP-framework and the other EC-based methods using the C++ Language. We take into account the information provided by the authors to reproduce, as faithfully as possible, the combination strategies used in these baseline EC methods.

4.4.5 Parameter settings

For the experiments, we use the baseline SOD models with their default parameter values. Concerning the classic combination models, these methods have no parameters to be tuned. All the EC-based techniques employ the same population size and stopping criterion to develop a fair comparison between them. We select a population size of 100 individual and a maximum number of 50 generations [124]. The specific parameters for each technique were adjusted based on the information provided by the original works.

For the GP-framework, we generate an initial population by using the ramped half-and-half method. We implement crossover, mutation, and reproduction with probabilities of 0.90, 0.05, 0.05, respectively. The selection method is roulette wheel selection. A maximum initial depth restricts the initial population to 3 levels. The population should never exceed a maximum of 10 levels during the evolution. We adopt elitism of the best individual to keep the best solution found in the evolutionary process. It is worth mentioning that these GP settings are similar to the typical values used in the literature [115, 113].

4.5 Results

This section discusses the results of the previously described experiments. First, we analyze the evolved combination models generated by the proposed GP-framework. After that, we present the experiments to compare the performance of the evolved models against the baseline methods. We compare the methods based on the quality in detecting saliency.

4.5.1 Analysis of the evolved combination models

We present some typically used statistics to describe the evolution in the GP process. These statistics include the evolution of the best individuals according to the fitness function, the depth of the trees, and the size of the models. To construct these graphs, we use the data of the 31 models generated during training. Figure 4.7 shows the average curves of these evolved models. This figure presents the two proposed types of models, the one generated by the SED-based fitness function and the MCC-based fitness function. Figure 4.7(a) illustrates how the fitness of the best individuals increases smoothly with the generations. Analyzing the curves, we can observe that the training instances are complex because both types of evolved models cannot find their corresponding perfect fitness scores. We present two additional graphs, shown in Figure 4.7(b) and Figure 4.7(c), to analyze the evolution in the structural complexity of the best individuals. For the case of the SED-based evolved models, from the initial generation to the end of the evolutionary process, there is an average increment of 3.84 in the depth of the trees, and an average increment of 10.87 in the number of nodes. For the MCC-based evolved models, the depth has an average increase of 3.7 and an average increase of 7 nodes.

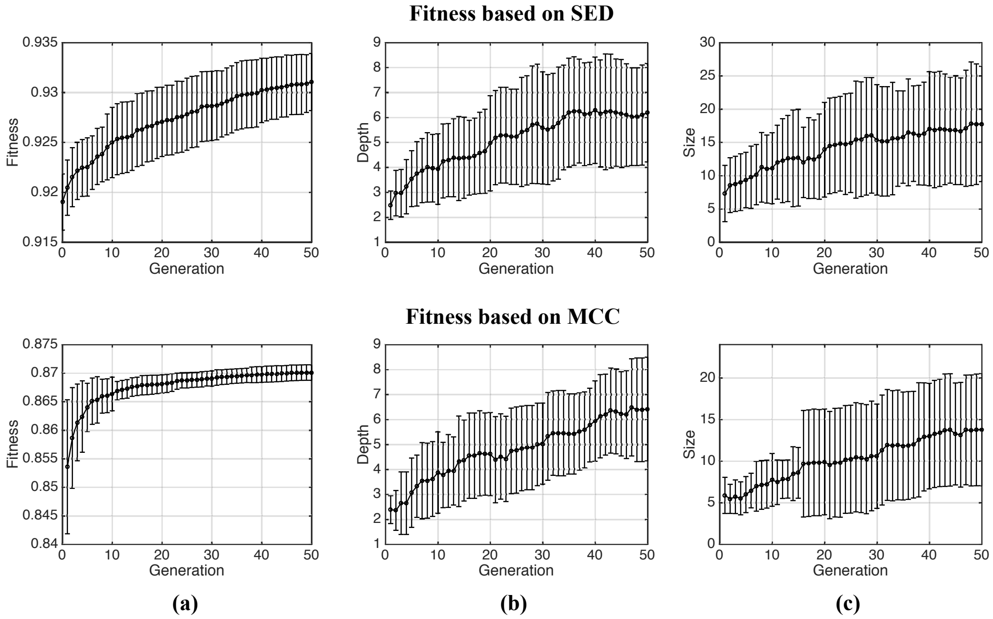


Figure 4.7: Convergence of the best-evolved models in the training phase, across the runs. Average and standard deviation curves for (a) the fitness score, (b) the depth of the tree, and (c) the size of the tree. From top to bottom, the evolution of the SED-based models, and the evolution of the MCC-based models.

Now we present a structural analysis of the best-evolved models. To this end, we registered the components that are common between the 31 models, considering each type of evolved model independently. Figure 4.8 shows the appearance frequency of the terminal and the function nodes. We can note that the evolved models for both types of fitness functions have the MDC and MBS models as essential terminals.

These contribute with 87.21% of the SED-based models, and 98.50% of the MCC-based models, as shown in Figure 4.8(a). In the case of the operators, we can observe similar normalized histograms in both types of model. The combination operators based on Fuzzy-logic are the most used for the best-evolved models, these contribute with 54.94% of the operations in the SED-based models, and 63.02% in the MCC-based models. The less used operators are the Gaussian filters and the two morphological operations. All of these operations contribute with 4.35% and 1.51% for the SED-based models, and MCC-based models, respectively.

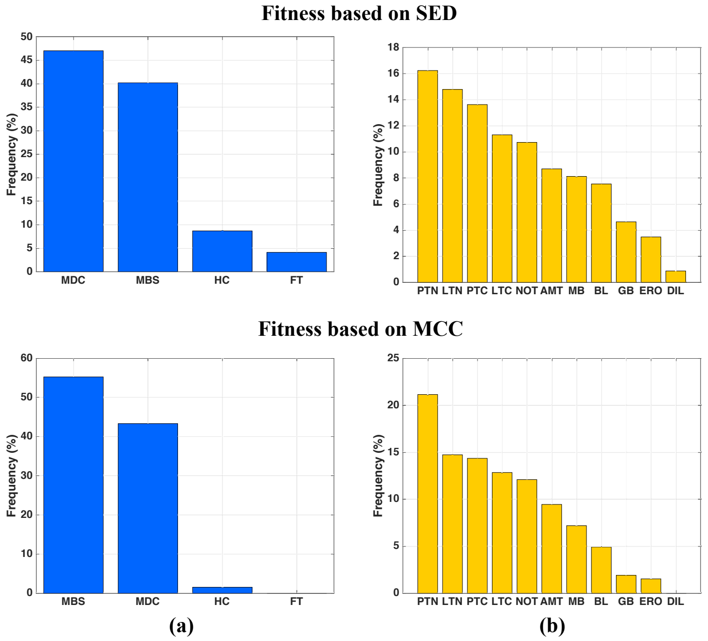


Figure 4.8: Appearance frequency of the common components of the 31 best-evolved models: (a) terminal nodes, and (b) function nodes. From top to bottom, the SED-based models, and the MCC-based models.

The previous description gives us a general overview of the evolved models. Now, we present two examples of evolved models. We apply the selection scheme based on the median to select the models [124]. Figure 4.9 shows the two selected models, one for fitness function, the SED-based model (GPSED for short) and the MCC-based model (GPMCC for short). GPSED is composed of only ten nodes, which include three combination operators, specifically: Lukasiewicz t-norm, and the Lukasiewicz t-conorm. Two morphological operations are used to dilate and to erode the salient regions, from the MBS response and the MDC response, respectively. A box filter is used to smooth the MBS response. The last nodes include the terminal nodes from the two different models (MDC and MBS). GPMCC model presents a simpler structure. It consists of 7 nodes: two nodes are the combination operations Lukasiewicz t-norm, and Product t-conorm; two nodes are low pass filters to smooth the MDC response, and the last three nodes are its input responses.

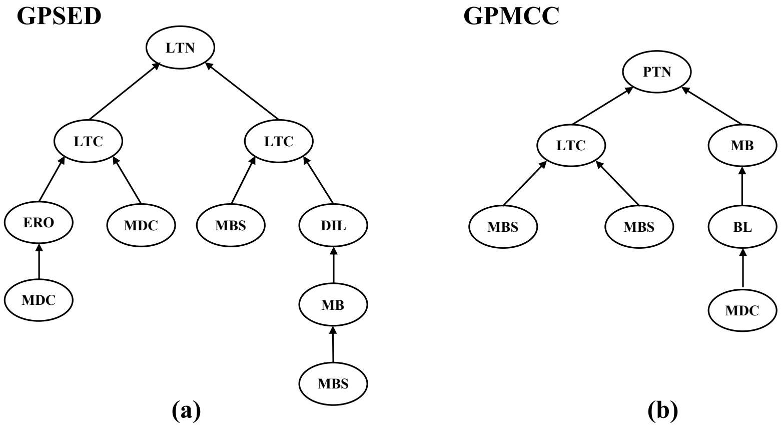


Figure 4.9: Evolved models generated by the proposed method: (a) GPSED, and (b) GPMCC.

4.5.2 Ranking the performance of the saliency models

In this section, we rank the performance of all the models under comparison by using the average ranking to compare through a single value all the models across different metrics and different datasets [115]. A ranking close to one represents a better model. Let us consider a set of models to be compared, denoted as $\mathcal{M} = \{M_1, M_2, M_{m+d}\}$, where m is the number of input algorithms, and d is the number of combination models; a set of test images denoted as $\mathcal{T}_I$, and a set of P performance measures. We can compute the average ranking of a saliency model M_i as,

$$R_i = \frac{1}{P}\sum_{j=1}^{P} rank\Big(M_i; measure_j(M_k(\mathcal{T}_I)), \, \forall k \neq i\Big) \tag{4.11}$$

where $rank(M_i; \cdot)$ computes the rank of the saliency model M_i considering the results of the rest of the models in the measure $measure_j$. To obtain a single number to represent the quality of the model, we perform the average ranking across all the test datasets.

Table 4.2 reports the average ranking of the models in the 31 runs. In this table, GP1 represents the results of the evolved models based on SED, and GP2 the evolved models based on MCC. We can observe that all evolutionary-based techniques are best positioned compared to all the input algorithms, it confirms that the combination is useful to improve the performance of the input algorithms. Also, the evolutionary-based techniques have a better performance compared to the AVG, MIN, and MAX rules, it proves that a learning-based algorithm is more appropriate to combine than the fusion rules. Note that among the evolutionary based-techniques, the proposed approaches outperform the rest of the models for a wide margin. For example, the GP1 models outperform the models generated by the best evolutionary technique (GA) by a difference of 2.43 points in the average ranking. In the case of classic combination models, the GP1 outperforms by 3.89 points the TOP2 model, and for the input algorithms, the GP1 is better than the best input algorithm by 5.39 points.

Table 4.2: Average ranking in the 31 runs across test datasets.

Saliency models	**Average ranking**
GP1	**2.05**
GP2	**2.40**
GA	4.48
PSO	4.72
TOP2	5.94
CPSO	6.71
MBS	7.44
AVG	7.82
MIN	8.04
MDC	8.15
HC	9.73
MAX	11.25
FT	12.28

To verify if the evolved models are statistically different, we apply the Wilcoxon rank-sum test on the results from Table 4.2 at a significance level $\alpha = 0.01$, and with critical values computed by using the Bonferroni correction. The results showed that all the models are significantly different from the other models except the pairs of models: GP1-GP2, and GA-PSO. Additionally, we run a t-test at a significance level $\alpha = 0.001$ without assuming that the data comes from the same variance. These results matched with the results obtained in the Wilcoxon rank-sum test. Both statistical tests confirm that the proposed models are significantly different from the rest of the models. Therefore, we conclude that the proposed models are the best among the models under comparison.

4.5.3 Precision-recall analysis

We show in detail the results of the proposed models concerning the *Precision-Recall* curve. For this purpose, we use the median solution for the 31 runs. The median solution is selected instead of the mean solution because it is more robust to outliers [124]. In the case of the GA-based approach, we select the solution closest to the mean solution in the parameter space as proposed in the original work. Figure 4.10 presents the *Precision-Recall* curves of all the models under comparison. We denoted **MRSA-BTest** to the test set from the **MSRA-B** dataset. In the figure, we rank the models based on the maximum *F-measure*, denoted as F_{max}.

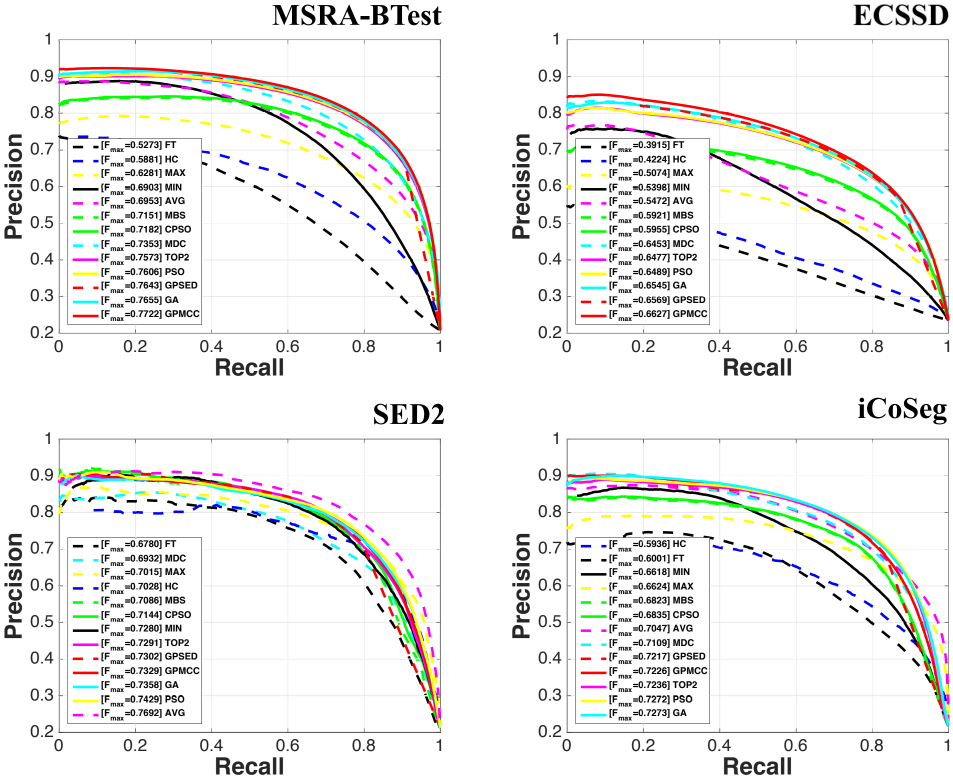


Figure 4.10: Quantitative comparison between the proposed models and the rest of the models, according to the *Precision-Recall* curve. For a better interpretation of the curves, the reader is referred to the web version of this article.

When we compared the proposed models in images from the same dataset (**MSRA-BTest**), it is clear that they outperform the input algorithms (FT, HC, MDC, and MBS). The same behavior appeared when we tested the proposed models in images from a different domain. For example, in **MSRA-BTest**, the GPMCC model is around 3% better than the best input algorithm in terms of the F_{max}, and in **iCoSeg** the GPMCC is 1.17% better than the MDC. Additionally, we can compare the results of the proposed models against classic combination models. From this analysis, we can observe that the performance of the proposed models, in test images from the same dataset, is better than the classic combination models. The GPMCC is 1.43% better than the TOP2 regarding the F_{max}. In one dataset (**iCoSeg**) the

TOP2 outperforms the GPMCC. However, both models are very close in performance (0.1% of difference). Finally, we analyze the results of the proposed models compared to the rest of evolutionary-based techniques. We can observe that the GPMCC model is the best in datasets with a single salient object (**MSRA-BTest** and **ECSSD**). In the case where the datasets contain multiple salient objects (**SED2** and **iCoSeg**) is more difficult to the proposed models outperform to the GA-based model and PSO-based model. However, the results are still competitive in images with multiple salient objects with a maximum difference of 1%.

4.5.4 *F*-measure analysis

Figure 4.11 shows the *F-measure* curves of all the models under comparison. In this figure, we rank the models according to their Area Under the Curve (AUC), denoted as AUC_F. More area represents that the model is less sensitive to the threshold value in detecting salient objects. In this figure, we can observe that the proposed model GPSED is the best-ranked model.

An in-depth analysis can be done to show the advantages of the proposed models. The GPSED model is better in terms of the AUC_F than the best input algorithm by 20.48%, 20.28%, 4.85% and 13.36% in the datasets **MSRA-BTest**, **ECSSD**, **SED2**, and **iCoSeg**, respectively. With respect to the best classic combination model, the GPSED obtains a maximum difference of 21.71% in the **MSRA-BTest** and a minimum diference of 12.32% in the **iCoSeg**. Also, among the EC-based techniques, the GPSED is the best. The GPSED outperforms the best EC-based model (considering only the baseline models) with a maximum difference of 21.38% and a minimum difference of 15.45%. The GPSED is less sensitive to the threshold value compared to the GPMCC model; however, the last model is competitive, it ranks second in the datasets with a single salient object, fifth in the **SED2** dataset, and third in the **iCoSeg** dataset.

To further examine the behavior of the proposed models, we also report the average F_α values using the image-dependent adaptive threshold [20]. Table 4.3 presents the F_α values of all the models. In this table, we mark the three best performing models with a triangle, a diamond, and a star, respectively. The proposed evolved models show higher F_α values compared to the input algorithms, the classic combination models and the EC-based models. The GPMCC and GPSED are the best two models according to the F_α values.

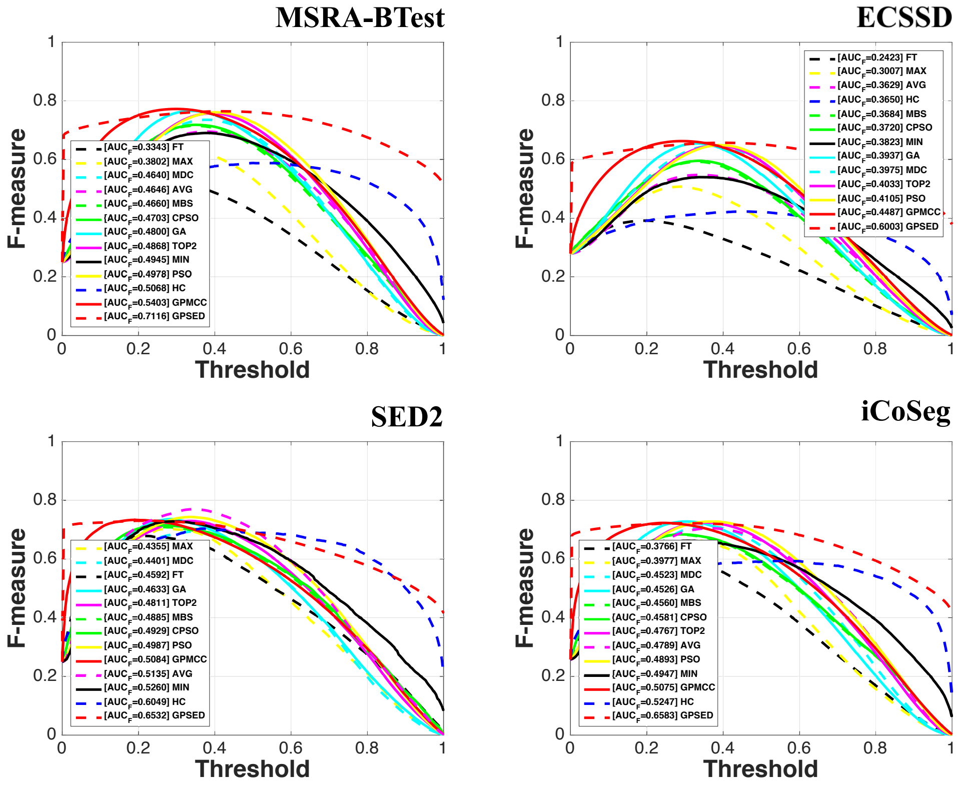


Figure 4.11: Quantitative comparison between the proposed models and the rest of the models, according to the *F-measure* curve. For a better interpretation of the curves, the reader is referred to the web version of this article.

Table 4.3: Comparison of the models regarding the F_α values using the image-dependent adaptive threshold. We mark the best three models with a triangle, a diamond, and a star, respectively.

Dataset	**FT**	**HC**	**MDC**	**MBS**	**MIN**	**MAX**	**AVG**
MSRA-BTest	0.5319	0.5663	0.7241	0.7116	0.6762	0.6105	0.6751
ECSSD	0.3775	0.3894	0.6127	0.5570	0.5177	0.4344	0.4994
SED2	0.6250	0.6079	0.6074	0.6654	0.6833	0.6106	0.6971★
iCoSeg	0.5545	0.5471	0.6348	0.6253	0.6210	0.5587	0.6105
Dataset	**TOP2**	**CPSO**	**GA**	**PSO**	**GPMCC**	**GPSED**	
MSRA-BTest	0.7423	0.7137	0.7579★	0.7455	0.7711▲	0.7662◆	
ECSSD	0.5979	0.5588	0.6200★	0.5988	0.6592▲	0.6592◆	
SED2	0.6658	0.6653	0.6914	0.6701	0.7148◆	0.7340▲	
iCoSeg	0.6556	0.6235	0.6678★	0.6557	0.6865◆	0.7157▲	

4.5.5 Mean absolute error analysis

Regarding the *MAE*, the proposed models exhibit better values compared to the input algorithms, classic combination models, and EC-based models, see Table 4.4. We can observe that the best model is the GPSED and the second best model is the GPMCC.

First, let us analyze the results of the proposed models compared against the input algorithms. The GPSED is around 8% better than the MBS in the same domain images. In case of the evaluation with images from a different domain, the GPSED is around 8%, 4%, and 5% better than the best input algorithm for **ECSSD**, **SED2**, and **iCoSeg**, respectively.

Considering the comparison of the proposed models versus the classic combination models, we can observe the GPSED is around 9% better than the TOP2 in the evaluation using the same domain images. In case of the evaluation with images from a different domain, the GPSED is around 8%, 5%, and 6% better than the best classic combination model for **ECSSD**, **SED2**, and **iCoSeg**, respectively.

Finally, we compare all the evolutionary-based approaches. The GPSED is around 8% better than the GA-based model in the evaluation using the same domain images. In case of the evaluation with images from a different domain, the GPSED is around 7%, 4%, and 5% better than the best EC-based model for **ECSSD**, **SED2**, and **iCoSeg**, respectively.

Table 4.4: Comparison of the *MAE* with the four input saliency algorithms, and the nine combination models. We mark the best three models with a triangle, a diamond, and a star, respectively.

Dataset	**FT**	**HC**	**MDC**	**MBS**	**MIN**	**MAX**	**AVG**
MSRA-BTest	0.2407	0.2397	0.2064	0.1983	0.2215	0.2336	0.2202
ECSSD	0.2893	0.3299	0.2520	0.2572	0.2758	0.2885	0.2824
SED2	0.2047	0.1966	0.2124	0.1898	0.2069	0.2065	0.1971
iCoSeg	0.2293	0.2292	0.2107	0.2021★	0.2228	0.2257	0.2153
Dataset	**TOP2**	**CPSO**	**GA**	**PSO**	**GPMCC**	**GPSED**	
MSRA-BTest	0.2033	0.1985	0.1953★	0.2022	0.1552◆	0.1112▲	
ECSSD	0.2569	0.2578	0.2486★	0.2576	0.2085◆	0.1709▲	
SED2	0.2003	0.1892★	0.2016	0.1983	0.1713◆	0.1440▲	
iCoSeg	0.2063	0.2023	0.2047	0.2050	0.1751◆	0.1460▲	

4.5.6 Visual comparison

In Figure 4.12, there are depicted visual comparisons between the baseline models (input algorithms, classic combination models and EC-based combination models) and the proposed models. In the figure, the last row represents the ground truth (GT). In all the images, the first and second columns present images with a simple background and a single salient object. The third and fourth columns provide images with a cluttered background. The fifth and sixth columns present images that contain two salient objects. Finally, the last two columns show input images with multiple salient objects (more than two). We select these images to show the qualitative results of all the models with different image types. In the figure, for images with a simple background and a single salient object, the proposed models easily distinguish

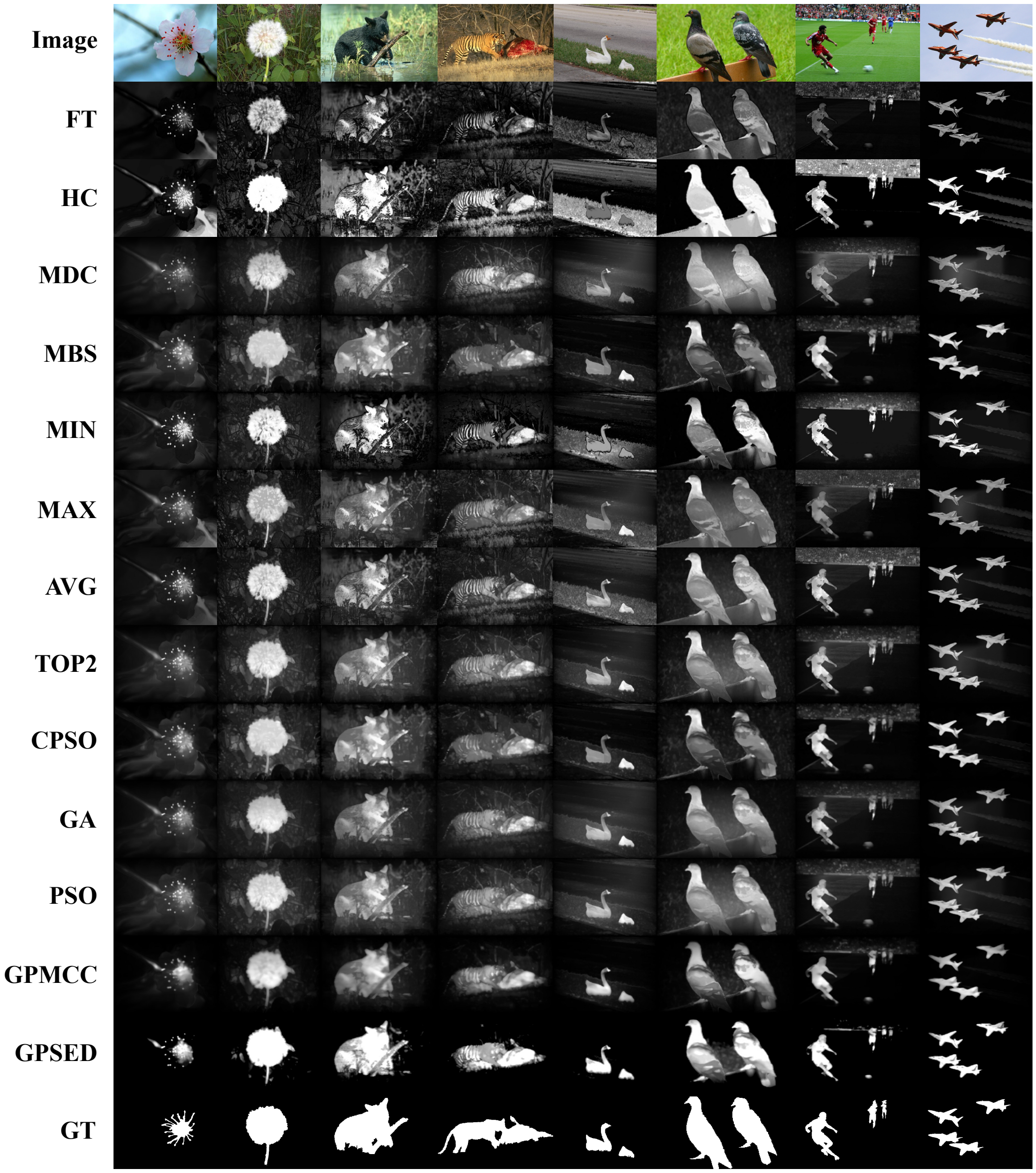


Figure 4.12: Visual comparison between the baseline models and the evolved combination models. GT represents the ground truth.

between salient regions and background. Even if the image presents a cluttered background, our models uniformly highlight salient objects while the baseline models tend to fail. Also, the proposed models can detect multiple salient objects in the same image in contrast to some baseline models that highlight more to some salient objects than others. Analyzing the visual results, we can note that the GPSED and

GPMCC models can reduce the noise in the background regions. Conversely, the rest of the models present more noise in these regions.

4.5.7 Run-time analysis

We analyze the computational cost of all the models to compute their corresponding saliency maps. Table 4.5 reports the time in the operation phase of the four input algorithms, and the nine combination models. We can observe in Table 4.5, that the MDC is the fastest input algorithm, and the MBS is the second one. The MDC model is approximately three times faster than the proposed GPSED model. However, the GPSED model keeps a low operation time (20 ms). In return to the increase in computation time to generate the final saliency map, the GPSED model improves the MDC response concerning the *MAE* around 9% in **MSRA-BTest**. Also, it improves the F_α value 4.21% in the same dataset. The proposed models exhibit a very similar figure than the fastest combination model (TOP2). There is a difference of 1 ms between the TOP2 model and GPSED model. However, the GPSED and GPMCC is 9.21%, and 4.81% better than the TOP2 model for the *MAE* values, respectively. The operation time of the other classic combination models is higher than the proposed models because these include all the input saliency responses to combine. Finally, we can note that the proposed models outperform all the EC-based approaches. The proposed combination models selected the best input algorithms (MDC and MBS models) from the available terminal set, this is the reason why the methods are faster. Let us give an example of the superiority of the proposed models. The GPSED and GPMCC models can generate a final response eight times faster than the PSO model in the **SED2** dataset.

Table 4.5: Average operation time [ms] of the SOD models. We mark the best three combination models with a triangle, a diamond, and a star, respectively.

Dataset	**FT**	**HC**	**MDC**	**MBS**	**MIN**	**MAX**	**AVG**
MSRA-BTest	24.61	248.65	6.76	11.52	290.28	292.63	290.72
ECSSD	25.20	247.74	6.60	11.95	290.86	288.96	292.74
SED2	20.51	124.01	6.68	10.61	161.47	160.85	161.02
iCoSeg	31.30	370.81	6.51	11.34	417.41	417.67	417.72
Dataset	**TOP2**	**CPSO**	**GA**	**PSO**	**GPMCC**	**GPSED**	
MSRA-BTest	18.88▲	290.60	299.84	287.73	20.22★	19.53◆	
ECSSD	18.11▲	338.29	289.72	314.59	20.12★	19.60◆	
SED2	16.11▲	177.74	160.76	140.99	17.43◆	17.70★	
iCoSeg	18.23▲	492.78	418.86	387.21	21.03★	19.18◆	

4.5.8 Failure cases

Figure 4.13 shows some failure cases of the evolved combination models. We can observe that in all cases the input algorithms fail in detecting salient regions. Due to our combination models work with these responses as inputs, the resultant saliency maps fail to highlight salient regions. These limitations in the input algorithms undoubtedly affect the performance of the proposed combination models.

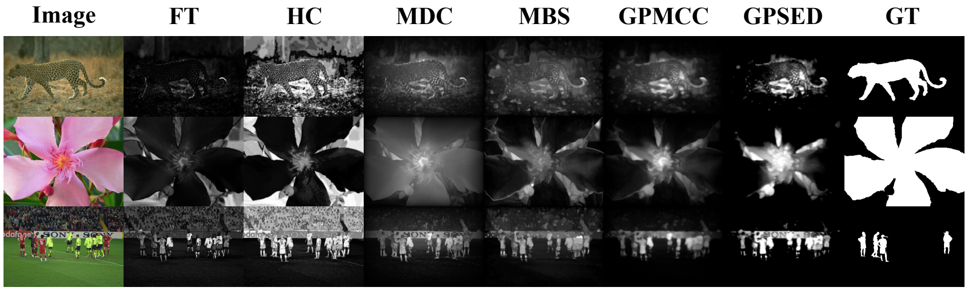


Figure 4.13: Failure cases of the input algorithms and the proposed combination models.

4.5.9 Further analysis

We further investigate the proposed methodology by using a new group of input algorithms. We select four machine-learning based saliency detection algorithms with the highest performance, DHSNet [22], DRFI [23], DSL [24], and LEGS [25]. To evaluate the performance of the combination models, we perform the experiments over the **ECSSD** dataset. Under these input algorithms, the new TOP2 model should combine the DHSNet and DSL algorithms. The GP-framework and the rest of the EC-based approaches use the same number of training instances of the previous experiment with the same input algorithms. Finally, we evaluate the performance of the saliency models with the rest of the images in the dataset.

Table 4.6 reports the average ranking of the models considering the results of the 31 runs in all the metrics. We executed the Wilcoxon rank-sum test on the results in the table at a significance level $\alpha = 0.01$ and critical values computed by the Bonferroni correction to verify if the evolved models are statistically different, from the other models. The results show that there are only two pairs of models that are the same in the statistical analysis: CPSO-GA and GP1-TOP2. Additionally, we execute a t-test at $\alpha = 0.01$ without assuming that the data come from the same variance. These results matched with the previous statistical test. Both statistical tests conclude that the GP2 model is significantly different from the rest of the models, and therefore that the proposed GP2 is the best-ranked model. Table 4.6

shows that the GP2 models outperform the best input algorithm, the best classic combination model, and the best EC-based approach by 1.66, 2.38, and 3.54 points, respectively; while the rest of the combination models fail to improve the performance of the best input algorithm. In the rest of the section, we will observe why this second experiment is more challenging for the combination strategies compared to the first experiment.

Table 4.6: Average ranking of the input algorithms, classic combination models, and EC-based approaches.

Saliency models	Average ranking
GP2	**1.69**
DHSNet	3.35
CPSO	4.07
GA	4.08
TOP2	5.23
GP1	5.35
PSO	5.75
MIN	8.00
DSL	8.40
AVG	9.07
LEGS	11.25
MAX	11.75
DRFI	13.00

Figure 4.14 presents two sample programs evolved by the GP-framework. The programs are the median solutions from the 31 runs. Both programs contain the best input algorithm (DSHNet), and the GPSED also contains the second best input algorithm (DSL). Figure 4.15 illustrates the performance of these evolved models and all the baseline models according to the *Precision-Recall*, and the *F-measure* curves. Figure 4.15(a) shows that the proposed GPSED model ranks second and the GPMCC ranks fifth concerning the F_{max}, and for the AUC_F the GPMCC model ranks first while the GPSED ranks fourth, see Figure 4.15(b). We also compared the models according to the F_α and the MAE metrics, see Table 4.7. The results show that the proposed GPMCC model is the best model in both metrics. In the F_α, the GA-based combination ranks second, and the DHSNet ranks third. For the MAE, the DHSNet ranks second, and the CPSO ranks third.

Now, let us analyze the results of the input algorithms. From the results above, the best input algorithm (DHSNet) outperforms the rest of the input algorithms by a large margin. This difference in performance converts the combination problem in a more challenging task, due to there being the risk that the worst input algorithms degenerate the results of the best one. For this reason, the combination models tend to fail. However, the proposed GPMCC is the only combination model capable of improving the best input algorithm and being the best-ranked model in the experiments.

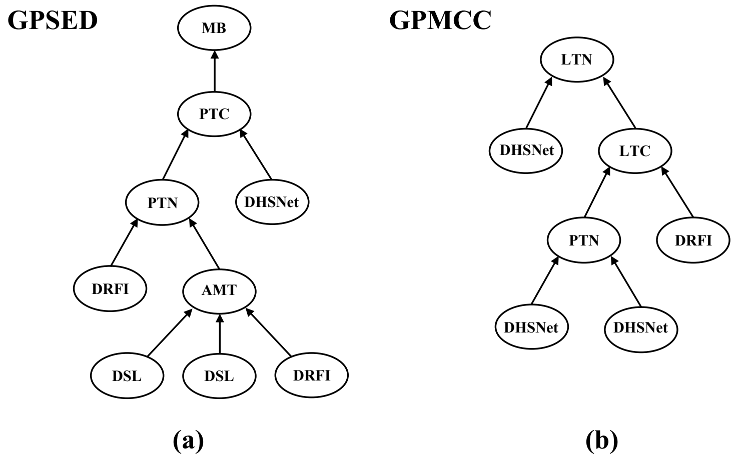


Figure 4.14: Evolved models, (a) GPSED, and (b) GPMCC for the second group of input algorithms.

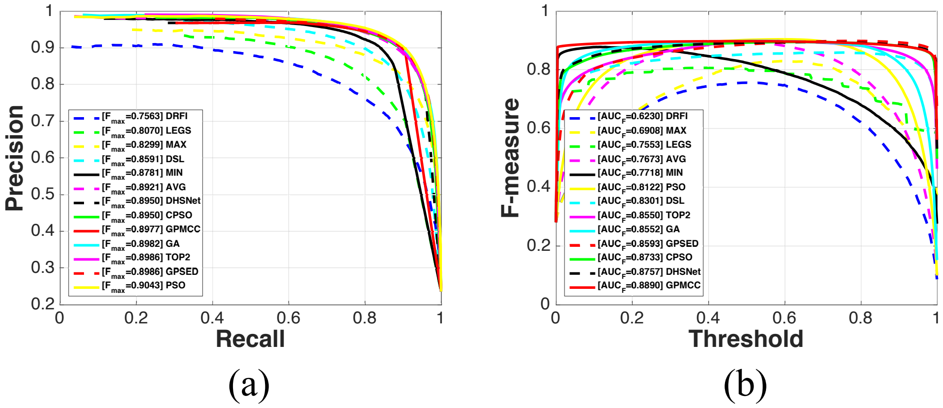


Figure 4.15: Comparison of the models on the ECSSD dataset according to the (a) *Precision-Recall* curve, and (b) *F-measure* curve for the second group of input algorithms. For a better interpretation of the curves, the reader is referred to the web version of this article.

Table 4.7: Comparison of the F_α and the MAE measures with the four input saliency algorithms, and the nine combination models. We mark the best three models with a triangle, a diamond, and a star, respectively.

Dataset	Metric	Saliency detection models						
ECSSD	F_α	DHSNet	DRFI	DSL	LEGS	MIN	MAX	AVG
		0.8675★	0.7253	0.8302	0.7851	0.8620	0.7562	0.8451
		TOP2	CPSO	GA	PSO	GPMCC	GPSED	
		0.8529	0.8658	0.8779◆	0.8590	0.8854▲	0.8469	
	MAE	DHSNet	DRFI	DSL	LEGS	MIN	MAX	AVG
		0.0586◆	0.1770	0.0814	0.1178	0.1084	0.1513	0.1106
		TOP2	CPSO	GA	PSO	GPMCC	GPSED	
		0.0704	0.0597★	0.0702	0.0899	0.0569▲	0.0630	

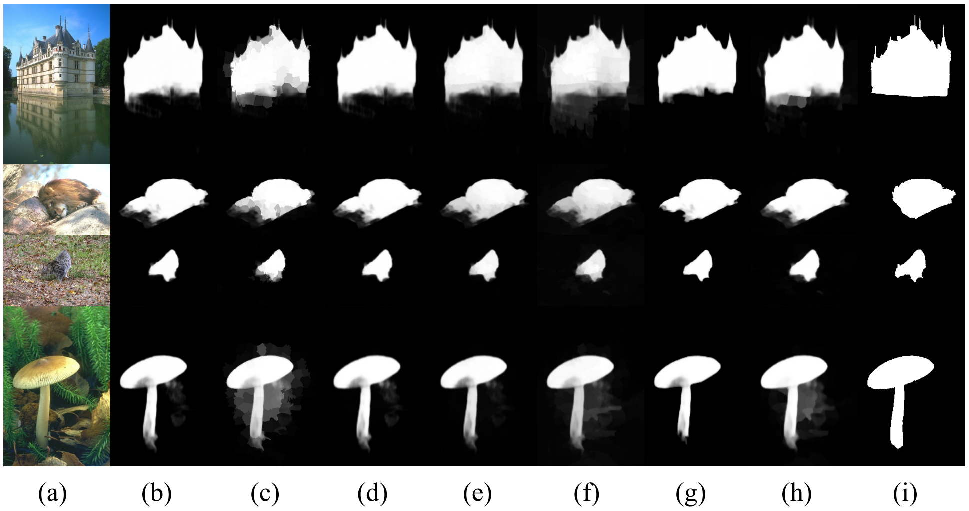


Figure 4.16: Visual comparison between the best seven models on ECSSD dataset. (a) input images; saliency maps generated by the models (b) DHSNet, (c) TOP2, (d) CPSO, (e) GA, (f) PSO, (g) GPMCC, and (h) GPSED; (i) GT.

Finally, we present a visual comparison between the seven best-ranked models according to the results of Table 4.6. Figure 4.16 shows the outputs of the best-ranked models in four test images from the **ECSSD** dataset. We can observe that almost all the models are capable of distinguishing between the salient object and the non-salient regions. However, the GPMCC is the model that produces the most similar images to the ground truth. On the other hand, the worst model is the PSO, followed by the GPSED, and TOP2 models. The visual responses of above models match with the position in the average ranking table.

CHAPTER 5

Conclusions and perspectives

In this thesis, we proposed using automatic design configuration tools for novelty detection problems and related areas. We performed the automatic configuration in two directions. In the first case, we used a bio-inspired optimization technique for tuning two machine learning algorithms used for novelty detection in robotics. We selected novelty detectors for continuous learning, such as grow-when-required networks and evolving connectionist systems. We tested the novelty detectors by using visual information on an outdoor environment with images captured by an unmanned aerial vehicle. We considered different novelties to verify the performance of the proposed methodology that included conspicuous or inconspicuous novel objects, static or dynamic novel objects, and multiple novel objects. We also considered two different light conditions in the outdoor environment (morning and afternoon), and two different flight heights of 2 m and 4 m. We observed the effects of using traditional visual features such as color histograms, the GIST descriptor, the color angular indexing. Besides, we proposed to used deep features extracted by a pre-trained neural convolutional network to enhance the performance of the novelty detectors. The results showed that the pre-trained network is competitive or even better than these traditional features. We observed that the designed detectors are robust to illumination changes, scale changes, and occlusion levels based on the results. Although they presented some problems with perspective changes produced by the flight control module of the unmanned aerial vehicle, the proposed evolved methods can detect in almost all the cases the novelties, which is a desirable characteristic of the novelty detection methods.

In the second study case, we use automatic algorithm configuration concepts to select and combine different saliency detection algorithms, including machine and non-machine algorithms. Saliency detection has shown in the literature a strong relation to novelty localization in images. We observed the weaknesses of existing saliency detection algorithms, and proposed a methodology based on genetic programming to automatically design combination models for salient object detection to obtain more consistent results. The central contribution of this work is that the methodology can select the input saliency algorithms and simultaneously generate the proper combination model for salient object detection. Another significant contribution of the work is that we embedded simple combination operators at a pixel-wise level (based on fuzzy logic) into the genetic programming, and operators that consider the relationship between neighboring pixels (image processing functions and morphological operators). We have also explored two fitness functions to guide the evolutionary process, to find the most suited for salient object detection. As a characteristic of our methodology, the resulting combination models are easy to interpret. The tree nodes help us identify the selected input algorithms for the combination, and the operations explain how to compute the final saliency map. This representation enables us to implement the evolved combination models quickly. We developed a comparison between our evolved models against the fastest algorithms in the state-of-the-art and other combination models considering datasets from the same and a different domain. The results showed that the evolved models are the best-ranked models concerning all the performance measures under study and all the test datasets. We verified the results using statistical tests, which confirmed that the proposed models are significantly different from the models generated by the rest of the approaches. We then replaced the input algorithms by four machine learning techniques with higher performance in the state-of-the-art. We found that the proposed methodology based on the Matthews Correlation Coefficient is the best-ranked technique. We validated the results through two statistical tests. It is essential to mention that our approach remained the best-ranked, although the considerable difference in performance between the best input algorithm and the rest of the input algorithms made the combination a more challenging problem.

5.1 Perspectives

The automatic design configuration problem is challenging because of the wide range of design selections and the lack of orientation to obtain good selections. In the context of novelty detection, the problem becomes even more challenging due to the imbalanced nature of the problem, where only a few or no novelty samples are available. We observed three different paths of this thesis that we enlist with their corresponding subtopics.

- Automatic design configuration of novelty detectors. We intend to explore an automatic design configuration for novelty detectors based on normal data only. We want to observe the effect of consistency measures and other metrics to tune novelty detectors. Besides, in supervised approaches, we are currently working on racing approaches for tuning novelty detectors. As the above approaches are global optimization techniques, we consider it essential to generate online control strategies to adjust the hyperparameter values of novelty detectors while they are working.

- Novelty detection tasks. We are exploring and developing an end-to-end learning approach to reduce human intervention in selecting and designing visual features. For example, the application of generative adversarial networks has shown excellent results in novelty/anomaly detection. Also, a pending task is sensor fusion to detect novelties when it is difficult to do so with visual information only.

- Automatic design of combination models for salient object detection. More research is needed to determine a strategy to select the number of training images and the characteristics we need to select them. Another direction of the work is to model the saliency detection as a multi-objective problem. This last idea implies to evolve combination models with several goals, e.g., optimizing all the metrics used in the literature to evaluate salient object detectors. Finally, we want to integrate the studied saliency object detection algorithms and novelty detectors to perform novelty localization. We developed some preliminary studies, and we have observed some promising results in this area, see Figure 5.1.

5.2 Final thoughts

In this section, I want to share some personal thoughts about the development of this thesis study. The first aspect is related to the topic addressed in this thesis. We observed that automatic design can automatically create and tune algorithms, reducing human intervention, and avoiding personal experience bias. I consider that more advances will emerge in this area thanks to the current capabilities of the computers. This will help to explore the configuration space of algorithms faster

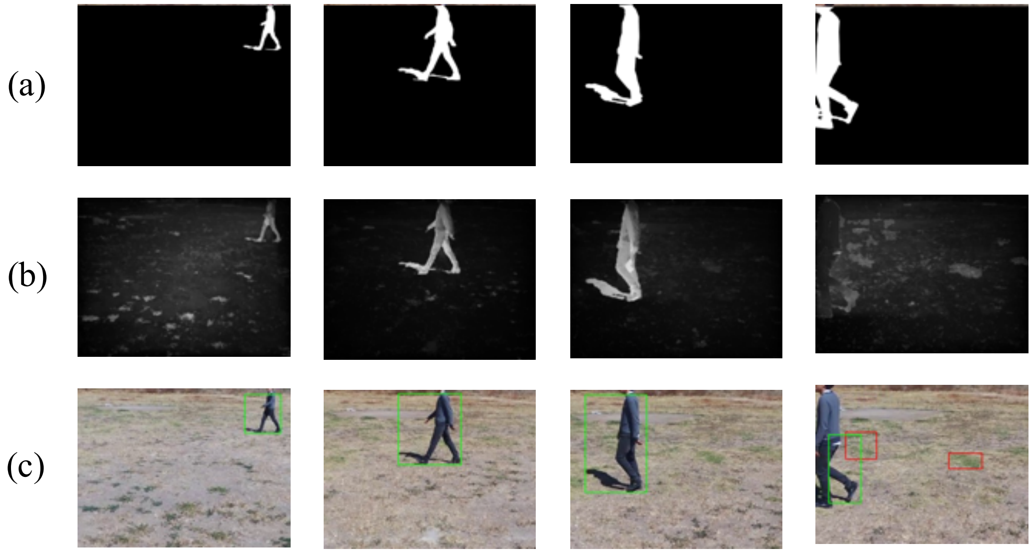


Figure 5.1: Novelty localization based on saliency detection. (a) Reference novelty indication, (b) Saliency detection with the Minimum Directional Contrast algorithm, and (c) Patches detected as novel objects by the grow-when-required network (green bounding boxes).

and be beneficial for the research community to achieve fair comparisons between algorithms.

Another aspect is the importance of research. I believe that research is crucial to reach the technological development in the country. With the development of this thesis, I acquired research experience by publishing both journal and conference papers. In my opinion, the publication and interaction with other researchers are essential to transmit the knowledge and contribute to the area.

Finally, I want to close my thoughts with my motivation letter of my admission process. At that moment, I wrote that “my future vision always has been to be a professor with deep knowledge in my area. I see myself involved in projects that help society. I want to develop technological tools that help to spread scientific and technical expertise among youth people. Because in that way, I believe we can contribute to the development of the country.” With the development of this thesis and my participation in projects during this period, I kept this vision, and now, I feel a little further advance in this direction.

5.3 Derived publications

This section shows the publications generated during the development of this thesis study.

Journal papers:

- Contreras-Cruz, M. A., Ramirez-Paredes, J. P., Hernandez-Belmonte, U. H., & Ayala-Ramirez, V. (2019). Vision-based novelty detection using deep features and evolved novelty filters for specific robotic exploration and inspection tasks. *Sensors*, 19(13), 2965. DOI: https://doi.org/10.3390/s19132965.

- Contreras-Cruz, M. A., Martinez-Rodriguez, D. E., Hernandez-Belmonte, U. H., & Ayala-Ramirez, V. (2019). A genetic programming framework in the automatic design of combination models for salient object detection. *Genetic Programming and Evolvable Machines*, 20(3), 285-325. DOI: https://doi.org/10.1007/s10710-019-09345-5.

- Lopez-Alanis, A., Lizarraga-Morales, R. A., Contreras-Cruz, M. A., Ayala-Ramirez, V., Sanchez-Yanez, R. E., & Trujillo-Romero, F. (2020). Rule-based aggregation driven by similar images for visual saliency detection. *Applied Intelligence*, 50(6), 1745-1762.
DOI: https://doi.org/10.1007/s10489-019-01582-6.

- Lopez-Alanis, A., Lizarraga-Morales, R. A., Sanchez-Yanez, R. E., Martinez-Rodriguez, D. E., & Contreras-Cruz, M. A. (2019). Visual saliency detection using a rule-based aggregation approach. *Applied Sciences*, 9(10), 2015. DOI: https://doi.org/10.3390/app9102015.

- Lopez-Perez, J. J., Hernandez-Belmonte, U. H., Ramirez-Paredes, J. P., Contreras-Cruz, M. A., & Ayala-Ramirez, V. (2018). Distributed multirobot exploration based on scene partitioning and frontier selection. Mathematical Problems in Engineering, 2018. DOI: https://doi.org/10.1155/2018/2373642.

Conference papers:

- Contreras-Cruz, M. A., Ochoa, G., Ramirez-Paredes, J.J. Synthetic vs. real-world continuous landscapes: a local optima Networks View. To be published in the Proceedings of the 9th International Conference on Bioinspired Optimisation Methods and Their Applications, Springer, 2020.

- Contreras-Cruz, M. A., Lopez-Perez, J. J., & Ayala-Ramirez, V. (2017, June). Distributed path planning for multi-robot teams based on artificial bee colony. In *2017 IEEE Congress on Evolutionary Computation (CEC)* (pp. 541-548). IEEE. DOI: https://doi.org/10.1109/CEC.2017.7969358.

- Goranova, M., Contreras-Cruz M.A., Hoyle, A., Ochoa, G. (2020, July). Optimising antibiotic treatments with multi-objective population-based algorithms. In *2020 IEEE Congress on Evolutionary Computation (CEC)* (pp. 1-7), IEEE. DOI: https://doi.org/10.1109/CEC48606.2020.9185489.

- Martinez-Rodriguez, D. E., Contreras-Cruz, M. A., Hernandez-Belmonte, U. H., Bereg, S., & Ayala-Ramirez, V. (2018, April). Saliency improvement through genetic programming. In Proceedings of the 3rd International Workshop on Interactive and Spatial Computing (pp. 29-38). DOI: https://doi.org/10.1145/3191801.3191809.

Bibliography

[1] S. Marsland, U. Nehmzow, and J. Shapiro, "On-line novelty detection for autonomous mobile robots," *Robotics and Autonomous Systems*, vol. 51, no. 2-3, pp. 191–206, 2005.

[2] M. A. Pimentel, D. A. Clifton, L. Clifton, and L. Tarassenko, "A review of novelty detection," *Signal Processing*, vol. 99, pp. 215–249, 2014.

[3] E. Stripling, B. Baesens, B. Chizi, and S. vanden Broucke, "Isolation-based conditional anomaly detection on mixed-attribute data to uncover workers' compensation fraud," *Decision Support Systems*, 2018.

[4] A. Ziaja, I. Antoniadou, T. Barszcz, W. J. Staszewski, and K. Worden, "Fault detection in rolling element bearings using wavelet-based variance analysis and novelty detection," *Journal of Vibration and Control*, vol. 22, no. 2, pp. 396–411, 2016.

[5] N. Mohammadian Rad, T. van Laarhoven, C. Furlanello, and E. Marchiori, "Novelty detection using deep normative modeling for IMU-based abnormal movement monitoring in parkinson's disease and autism spectrum disorders," *Sensors*, vol. 18, no. 10, p. 3533, 2018.

[6] P. Burlina, N. Joshi, S. Billings, I.-J. Wang, and J. Albayda, "Deep embeddings for novelty detection in myopathy," *Computers in biology and medicine*, 2018.

[7] M. Ribeiro, A. E. Lazzaretti, and H. S. Lopes, "A study of deep convolutional auto-encoders for anomaly detection in videos," *Pattern Recognition Letters*, vol. 105, pp. 13–22, 2018.

[8] E. Özbilge, "On-line expectation-based novelty detection for mobile robots," *Robotics and Autonomous Systems*, vol. 81, pp. 33–47, 2016.

[9] P. Ross, A. English, D. Ball, B. Upcroft, and P. Corke, "Online novelty-based visual obstacle detection for field robotics," in *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pp. 3935–3940, IEEE, 2015.

[10] P. Drews-Jr, P. Núñez, R. P. Rocha, M. Campos, and J. Dias, "Novelty detection and segmentation based on Gaussian mixture models: A case study in 3D robotic laser mapping," *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1696–1709, 2013.

[11] Y. Gatsoulis and T. M. McGinnity, "Intrinsically motivated learning systems based on biologically-inspired novelty detection," *Robotics and Autonomous Systems*, vol. 68, pp. 12–20, 2015.

[12] R. Hornung, H. Urbanek, J. Klodmann, C. Osendorfer, and P. Van Der Smagt, "Model-free robot anomaly detection," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3676–3683, IEEE, 2014.

[13] D. Karaboga, B. Gorkemli, C. Ozturk, and N. Karaboga, "A comprehensive survey: artificial bee colony (ABC) algorithm and applications," *Artificial Intelligence Review*, vol. 42, no. 1, pp. 21–57, 2014.

[14] J. R. Koza, "Human-competitive results produced by genetic programming," *Genetic Programming and Evolvable Machines*, vol. 11, no. 3-4, pp. 251–284, 2010.

[15] S. Marsland, J. Shapiro, and U. Nehmzow, "A self-organising network that grows when required," *Neural networks*, vol. 15, no. 8-9, pp. 1041–1058, 2002.

[16] M. Watts and N. Kasabov, "Simple evolving connectionist systems and experiments on isolated phoneme recognition," in *Combinations of Evolutionary Computation and Neural Networks, 2000 IEEE Symposium on*, pp. 232–239, IEEE, 2000.

[17] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," *arXiv preprint arXiv:1801.04381*, 2018.

[18] X. Huang and Y. Zhan, "300 FPS salient object detection via minimum directional contrast," *IEEE Transactions on Image Processing*, 2017.

[19] J. Zhang, S. Sclaroff, Z. Lin, X. Shen, B. Price, and R. Mech, "Minimum barrier salient object detection at 80 FPS," in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1404–1412, 2015.

[20] R. Achanta, S. Hemami, F. Estrada, and S. Susstrunk, "Frequency-tuned salient region detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1597–1604, IEEE, 2009.

[21] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. Torr, and S.-M. Hu, "Global contrast based salient region detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 3, pp. 569–582, 2015.

[22] N. Liu and J. Han, "DHSNet: Deep hierarchical saliency network for salient object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 678–686, 2016.

[23] H. Jiang, J. Wang, Z. Yuan, Y. Wu, N. Zheng, and S. Li, "Salient object detection: a discriminative regional feature integration approach," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2083–2090, 2013.

[24] Y. Yuan, C. Li, J. Kim, W. Cai, and D. D. Feng, "Dense and sparse labeling with multi-dimensional features for saliency detection," *IEEE Transactions on Circuits and Systems for Video Technology*, 2016.

[25] L. Wang, H. Lu, X. Ruan, and M.-H. Yang, "Deep networks for saliency detection via local estimation and global search," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3183–3192, 2015.

[26] M. A. Contreras-Cruz, J. P. Ramirez-Paredes, U. H. Hernandez-Belmonte, and V. Ayala-Ramirez, "Vision-based novelty detection using deep features and evolved novelty filters for specific robotic exploration and inspection tasks," *Sensors*, vol. 19, no. 13, p. 2965, 2019.

[27] M. A. Contreras-Cruz, D. E. Martinez-Rodriguez, U. H. Hernandez-Belmonte, and V. Ayala-Ramirez, "A genetic programming framework in the automatic design of combination models for salient object detection," *Genetic Programming and Evolvable Machines*, vol. 20, no. 3, pp. 285–325, 2019.

[28] T. Stützle and M. López-Ibáñez, "Automated design of metaheuristic algorithms," in *Handbook of metaheuristics*, pp. 541–579, Springer, 2019.

[29] J. Snoek, H. Larochelle, and R. P. Adams, "Practical Bayesian optimization of machine learning algorithms," in *Advances in neural information processing systems*, pp. 2951–2959, 2012.

[30] G. I. Diaz, A. Fokoue-Nkoutche, G. Nannicini, and H. Samulowitz, "An effective algorithm for hyperparameter optimization of neural networks," *IBM Journal of Research and Development*, vol. 61, no. 4/5, pp. 9–1, 2017.

[31] J. S. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for hyper-parameter optimization," in *Advances in neural information processing systems*, pp. 2546–2554, 2011.

[32] M. Zöller and M. F. Huber, "Survey on automated machine learning," *CoRR*, vol. abs/1904.12054, 2019.

[33] A. Godínez-Bautista, L. C. Padierna, A. Rojas-Domínguez, H. Puga, and M. Carpio, "Bio-inspired metaheuristics for hyper-parameter tuning of support vector machine classifiers," in *Fuzzy Logic Augmentation of Neural and Optimization Algorithms: Theoretical Aspects and Real Applications*, pp. 115–130, Springer, 2018.

[34] M. López-Ibáñez, J. Dubois-Lacoste, L. P. Cáceres, M. Birattari, and T. Stützle, "The irace package: iterated racing for automatic algorithm configuration," *Operations Research Perspectives*, vol. 3, pp. 43–58, 2016.

[35] E. K. Burke, M. R. Hyde, G. Kendall, G. Ochoa, E. Özcan, and J. R. Woodward, "A classification of hyper-heuristic approaches: revisited," in *Handbook of Metaheuristics*, pp. 453–477, Springer, 2019.

[36] P. Juszczak, D. M. Tax, E. Pe, R. P. Duin, *et al.*, "Minimum spanning tree based one-class classifier," *Neurocomputing*, vol. 72, no. 7-9, pp. 1859–1869, 2009.

[37] E. R. Faria, I. J. Gonçalves, A. C. de Carvalho, and J. Gama, "Novelty detection in data streams," *Artificial Intelligence Review*, vol. 45, no. 2, pp. 235–269, 2016.

[38] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: a survey," *ACM computing surveys (CSUR)*, vol. 41, no. 3, pp. 1–58, 2009.

[39] V. Barnett and T. Lewis, *Outliers in Statistical Data.* Wiley Series in Probability and Mathematical Statistics: Applied Probability and Statistics, Wiley and Sons, 1994.

[40] P. Gogoi, D. Bhattacharyya, B. Borah, and J. K. Kalita, "A survey of outlier detection methods in network anomaly identification," *The Computer Journal*, vol. 54, no. 4, pp. 570–588, 2011.

[41] M. Markou and S. Singh, "A neural network-based novelty detector for image sequence analysis," *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 10, pp. 1664–1677, 2006.

[42] N. Görnitz, M. Kloft, K. Rieck, and U. Brefeld, "Toward supervised anomaly detection," *Journal of Artificial Intelligence Research*, vol. 46, pp. 235–262, 2013.

[43] C. Désir, S. Bernard, C. Petitjean, and L. Heutte, "One class random forests," *Pattern Recognition*, vol. 46, no. 12, pp. 3490–3506, 2013.

[44] C. Surace and K. Worden, "Novelty detection in a changing environment: a negative selection approach," *Mechanical Systems and Signal Processing*, vol. 24, no. 4, pp. 1114–1128, 2010.

[45] X. Ding, Y. Li, A. Belatreche, and L. P. Maguire, "Novelty detection using level set methods," *IEEE transactions on neural networks and learning systems*, vol. 26, no. 3, pp. 576–588, 2014.

[46] S. Ntalampiras, I. Potamitis, and N. Fakotakis, "Probabilistic novelty detection for acoustic surveillance under real-world conditions," *IEEE Transactions on Multimedia*, vol. 13, no. 4, pp. 713–719, 2011.

[47] D. M. Tax and R. P. Duin, "Data description in subspaces," in *Proceedings 15th International Conference on Pattern Recognition. ICPR-2000*, vol. 2, pp. 672–675, IEEE, 2000.

[48] F. Angiulli, "Condensed nearest neighbor data domain description," *IEEE transactions on pattern analysis and machine intelligence*, vol. 29, no. 10, pp. 1746–1758, 2007.

[49] F. Angiulli, "Fast nearest neighbor condensation for large data sets classification," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 11, pp. 1450–1464, 2007.

[50] F. Angiulli, "Prototype-based domain description for one-class classification," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 6, pp. 1131–1144, 2012.

[51] J. de Andrade Silva, E. R. Hruschka, and J. Gama, "An evolutionary algorithm for clustering data streams with a variable number of clusters," *Expert Systems with Applications*, vol. 67, pp. 228–238, 2017.

[52] L. Chomatek and A. Duraj, "Multiobjective genetic algorithm for outliers detection," in *2017 IEEE International Conference on Innovations in Intelligent SysTems and Applications (INISTA)*, pp. 379–384, IEEE, 2017.

[53] Y. Xiao, H. Wang, W. Xu, and J. Zhou, "L1 norm based KPCA for novelty detection," *Pattern Recognition*, vol. 46, no. 1, pp. 389–396, 2013.

[54] H. V. Neto and U. Nehmzow, "Incremental PCA: An alternative approach for novelty detection," *Towards Autonomous Robotic Systems*, 2005.

[55] H. Hoffmann, "Kernel PCA for novelty detection," *Pattern recognition*, vol. 40, no. 3, pp. 863–874, 2007.

[56] S. J. Haggett and D. F. Chu, "Evolving novelty detectors for specific applications," *Neurocomputing*, vol. 72, no. 10-12, pp. 2392–2405, 2009.

[57] P. Crook, G. Hayes, *et al.*, "A robot implementation of a biologically inspired method for novelty detection," in *Proceedings of Towards Intelligent Mobile Robots Conference*, 2001.

[58] S. Marsland, U. Nehmzow, and J. Shapiro, "Detecting novel features of an environment using habituation," in *Proc. Simulation of Adaptive Behavior*, 2000.

[59] C. Richter and N. Roy, "Safe visual navigation via deep learning and novelty detection," in *Proceedings of Robotics Science and Systems (RSS)*, Robotics: Science and Systems Foundation, 2017.

[60] O. Fink, E. Zio, and U. Weidmann, "Novelty detection by multivariate kernel density estimation and growing neural gas algorithm," *Mechanical Systems and Signal Processing*, vol. 50, pp. 427–436, 2015.

[61] M. J. Watts, "A decade of kasabov's evolving connectionist systems: a review," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 39, no. 3, pp. 253–269, 2009.

[62] E. Özbilge, "Detecting static and dynamic novelties using dynamic neural network," *Procedia Computer Science*, vol. 120, pp. 877–886, 2017.

[63] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural computation*, vol. 13, no. 7, pp. 1443–1471, 2001.

[64] D. M. Tax and R. P. Duin, "Support vector data description," *Machine learning*, vol. 54, no. 1, pp. 45–66, 2004.

[65] A. Rekha, M. S. Abdulla, and S. Asharaf, "Lightly trained support vector data description for novelty detection," *Expert Systems with Applications*, vol. 85, pp. 25–32, 2017.

[66] L. Livi, A. Sadeghian, and W. Pedrycz, "Entropic one-class classifiers," *IEEE transactions on neural networks and learning systems*, vol. 26, no. 12, pp. 3187–3200, 2015.

[67] S. Wu and S. Wang, "Information-theoretic outlier detection for large-scale categorical data," *IEEE transactions on knowledge and data engineering*, vol. 25, no. 3, pp. 589–602, 2011.

[68] D. T. Nguyen and K. J. Cios, "Rule-based OneClass-DS learning algorithm," *Applied Soft Computing*, vol. 35, pp. 267–279, 2015.

[69] Q.-F. Zhou, H. Zhou, Y.-P. Ning, F. Yang, and T. Li, "Two approaches for novelty detection using random forest," *Expert Systems with Applications*, vol. 42, no. 10, pp. 4840–4850, 2015.

[70] S. M. Erfani, S. Rajasegarar, S. Karunasekera, and C. Leckie, "High-dimensional and large-scale anomaly detection using a linear one-class svm with deep learning," *Pattern Recognition*, vol. 58, pp. 121–134, 2016.

[71] T. Dang, S. Khattak, C. Papachristos, and K. Alexis, "Anomaly detection and cognizant path planning for surveillance operations using aerial robots," in *2019 International Conference on Unmanned Aircraft Systems (ICUAS)*, pp. 667–673, IEEE, 2019.

[72] T. Schlegl, P. Seeböck, S. M. Waldstein, G. Langs, and U. Schmidt-Erfurth, "f-AnoGAN: fast unsupervised anomaly detection with generative adversarial networks," *Medical image analysis*, vol. 54, pp. 30–44, 2019.

[73] H. Zenati, M. Romain, C.-S. Foo, B. Lecouat, and V. Chandrasekhar, "Adversarially learned anomaly detection," in *2018 IEEE International Conference on Data Mining (ICDM)*, pp. 727–736, IEEE, 2018.

[74] D. M. Tax and K.-R. Muller, "A consistency-based model selection for one-class classification," in *Proceedings of the 17th International Conference on Pattern Recognition, (ICPR).*, vol. 3, pp. 363–366, IEEE, 2004.

[75] S. Wang, Q. Liu, E. Zhu, F. Porikli, and J. Yin, "Hyperparameter selection of one-class support vector machine by self-adaptive data shifting," *Pattern Recognition*, vol. 74, pp. 198–211, 2018.

[76] H. Trittenbach, K. Böhm, and I. Assent, "Active learning of svdd hyperparameter values," *arXiv preprint arXiv:1912.01927*, 2019.

[77] B. Krawczyk, "One-class classifier ensemble pruning and weighting with firefly algorithm," *Neurocomputing*, vol. 150, pp. 490–500, 2015.

[78] E. Parhizkar and M. Abadi, "BeeOWA: A novel approach based on abc algorithm and induced OWA operators for constructing one-class classifier ensembles," *Neurocomputing*, vol. 166, pp. 367–381, 2015.

[79] Y. Xiao, H. Wang, and W. Xu, "Hyperparameter selection for gaussian process one-class classification," *IEEE transactions on neural networks and learning systems*, vol. 26, no. 9, pp. 2182–2187, 2014.

[80] L. Xie, "Swarm intelligent tuning of one-class ν-svm parameters," in *International Conference on Rough Sets and Knowledge Technology*, pp. 552–559, Springer, 2006.

[81] B. Raskutti and A. Kowalczyk, "Extreme re-balancing for svms: a case study," *ACM Sigkdd Explorations Newsletter*, vol. 6, no. 1, pp. 60–69, 2004.

[82] L. Zhuang and H. Dai, "Parameter optimization of kernel-based one-class classifier on imbalance learning," *Journal of Computers*, vol. 1, no. 7, pp. 32–40, 2006.

[83] H. V. Neto, "On-line visual novelty detection in autonomous mobile robots," *Introduction to Mordern Robotics*, vol. 2, pp. 241–265, 2011.

[84] H. Kato, T. Harada, and Y. Kuniyoshi, "Visual anomaly detection from small samples for mobile robots," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, pp. 3171–3178, IEEE, 2012.

[85] H. V. Neto and U. Nehmzow, "Visual novelty detection with automatic scale selection," *Robotics and Autonomous Systems*, vol. 55, no. 9, pp. 693–701, 2007.

[86] H. V. Neto and U. Nehmzow, “Real-time automated visual inspection using mobile robots,” *Journal of Intelligent and Robotic Systems*, vol. 49, no. 3, pp. 293–307, 2007.

[87] L. Pitonakova and S. Bullock, “The robustness-fidelity trade-off in grow when required neural networks performing continuous novelty detection,” *Neural Networks*, vol. 122, pp. 183–195, 2020.

[88] V. Gonzalez-Pacheco, A. Sanz, M. Malfaz, and M. A. Salichs, “Using novelty detection in HRI: Enabling robots to detect new poses and actively ask for their labels,” in *Humanoid Robots (Humanoids), 2014 14th IEEE-RAS International Conference on*, pp. 1110–1115, IEEE, 2014.

[89] X. Wang, X. L. Wang, and D. M. Wilkes, “An automated vision based on-line novel percept detection method for a mobile robot,” *Robotics and Autonomous Systems*, vol. 60, no. 10, pp. 1279–1294, 2012.

[90] T. V. Nguyen and M. Kankanhalli, “As-similar-as-possible saliency fusion,” *Multimedia Tools and Applications*, vol. 76, no. 8, pp. 10501–10519, 2017.

[91] N. Kasabov, “Ecos: Evolving connectionist systems and the eco learning paradigm.,” in *Iconip*, vol. 98, pp. 1232–1235, 1998.

[92] M. Mernik, S.-H. Liu, D. Karaboga, and M. Črepinšek, “On clarifying misconceptions when comparing variants of the artificial bee colony algorithm by offering a new implementation,” *Information Sciences*, vol. 291, pp. 115–127, 2015.

[93] Y. Shi, Y. Liu, Q. Zhang, Y. Yi, and W. Li, “Saliency-based abnormal event detection in crowded scenes,” *Journal of Electronic Imaging*, vol. 25, no. 6, p. 061608, 2016.

[94] A. Borji, D. N. Sihite, and L. Itti, “Salient object detection: a benchmark,” in *Proceedings of the European Conference on Computer Vision*, pp. 414–429, Springer-Verlag Berlin Heidelberg, 2012.

[95] L. Mai, Y. Niu, and F. Liu, “Saliency aggregation: a data-driven approach,” in *Proceedings of the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1131–1138, 2013.

[96] S. S. Naqvi, W. N. Browne, and C. Hollitt, “Genetic algorithms based feature combination for salient object detection, for autonomously identified image

domain types," in *Proceedings of the IEEE Congress on Evolutionary Computation*, pp. 109–116, IEEE, 2014.

[97] T. Liu, Z. Yuan, J. Sun, J. Wang, N. Zheng, X. Tang, and H.-Y. Shum, "Learning to detect a salient object," *IEEE Transactions on Pattern analysis and machine intelligence*, vol. 33, no. 2, pp. 353–367, 2011.

[98] A. Borji, M.-M. Cheng, H. Jiang, and J. Li, "Salient object detection: a benchmark," *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5706–5722, 2015.

[99] Q. Yan, L. Xu, J. Shi, and J. Jia, "Hierarchical saliency detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1155–1162, 2013.

[100] W. Tan and B. Yan, "Salient object detection via multiple saliency weights," *Multimedia Tools and Applications*, pp. 1–17, 2017.

[101] W. Zhu, S. Liang, Y. Wei, and J. Sun, "Saliency optimization from robust background detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2814–2821, 2014.

[102] Z. Jiang and L. S. Davis, "Submodular salient region detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2043–2050, 2013.

[103] F. Perazzi, P. Krähenbühl, Y. Pritch, and A. Hornung, "Saliency filters: contrast based filtering for salient region detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 733–740, IEEE, 2012.

[104] N. Tong, H. Lu, X. Ruan, and M.-H. Yang, "Salient object detection via bootstrap learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1884–1892, 2015.

[105] L. Zhang, J. Li, and H. Lu, "Saliency detection via extreme learning machine," *Neurocomputing*, vol. 218, pp. 103–112, 2016.

[106] G. Li and Y. Yu, "Visual saliency detection based on multiscale deep CNN features," *IEEE Transactions on Image Processing*, vol. 25, no. 11, pp. 5012–5024, 2016.

[107] A. Borji, M.-M. Cheng, Q. Hou, H. Jiang, and J. Li, "Salient object detection: Aasurvey," *Computational Visual Media*, vol. 5, no. 2, pp. 117–150, 2019.

[108] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3431–3440, 2015.

[109] S. Chen, L. Zheng, X. Hu, and P. Zhou, "A comparative study of saliency aggregation for salient object detection," in *Proceedings of the International Conference on Image and Graphics*, pp. 30–42, Springer, 2015.

[110] X. Wei, Z. Tao, C. Zhang, and X. Cao, "Structured saliency fusion based on Dempster–Shafer theory," *IEEE Signal Processing Letters*, vol. 22, no. 9, pp. 1345–1349, 2015.

[111] J. Wang, A. Borji, C.-C. J. Kuo, and L. Itti, "Learning a combined model of visual saliency for fixation prediction," *IEEE Transactions on Image Processing*, vol. 25, no. 4, pp. 1566–1579, 2016.

[112] S. S. Naqvi, W. N. Browne, and C. Hollitt, "Feature quality-based dynamic feature selection for improving salient object detection," *IEEE Transactions on Image Processing*, vol. 25, no. 9, pp. 4298–4313, 2016.

[113] G. Olague and L. Trujillo, "Evolutionary-computer-assisted design of image operators that detect interest points using genetic programming," *Image and Vision Computing*, vol. 29, no. 7, pp. 484–498, 2011.

[114] W. Fu, M. Johnston, and M. Zhang, "Genetic programming for edge detection: a gaussian-based approach," *Soft Computing*, vol. 20, no. 3, pp. 1231–1248, 2016.

[115] S. Bianco, G. Ciocca, and R. Schettini, "Combination of video change detection algorithms by genetic programming," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 6, pp. 914–928, 2017.

[116] H. Al-Sahaf, M. Zhang, A. Al-Sahaf, and M. Johnston, "Keypoints detection and feature extraction: A dynamic genetic programming approach for evolving rotation-invariant texture image descriptors," *IEEE Transactions on Evolutionary Computation*, vol. 21, no. 6, pp. 825–844, 2017.

[117] L. Dozal, G. Olague, E. Clemente, and D. E. Hernández, "Brain programming for the evolution of an artificial dorsal stream," *Cognitive Computation*, vol. 6, no. 3, pp. 528–557, 2014.

[118] Y. Liang, M. Zhang, and W. N. Browne, “Genetic programming for evolving figure-ground segmentors from multiple features,” *Applied Soft Computing*, vol. 51, pp. 83–95, 2017.

[119] J. Shi, Q. Yan, L. Xu, and J. Jia, “Hierarchical image saliency detection on extended CSSD,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 4, pp. 717–729, 2016.

[120] D. Batra, A. Kowdle, D. Parikh, J. Luo, and T. Chen, “icoseg: Interactive co-segmentation with intelligent scribble guidance,” in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pp. 3169–3176, IEEE, 2010.

[121] S. Alpert, M. Galun, R. Basri, and A. Brandt", “"image segmentation by probabilistic bottom-up aggregation and cue integration.",” in *"Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition"*, "June" "2007".

[122] O. Le Meur and Z. Liu, “Saliency aggregation: Does unity make strength?,” in *Asian Conference on Computer Vision*, pp. 18–32, Springer, 2014.

[123] N. Singh, R. Arya, and R. Agrawal, “A novel approach to combine features for salient object detection using constrained particle swarm optimization,” *Pattern Recognition*, vol. 47, no. 4, pp. 1731–1739, 2014.

[124] S. Afzali, B. Xue, H. Al-Sahaf, and M. Zhang, “A supervised feature weighting method for salient object detection using particle swarm optimization,” in *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 1–8, IEEE, 2017.