Evolutionary Computation for Feature Manipulation in Salient Object Detection

by

Shima Afzali Vahed Moghaddam

A thesis
submitted to the Victoria University of Wellington
in fulfilment of the
requirements for the degree of
Doctor of Philosophy
in Computer Science.

Victoria University of Wellington
2020

Abstract

The human visual system can efficiently cope with complex natural scenes containing various objects at different scales using the visual attention mechanism. Salient object detection (SOD) aims to simulate the capability of the human visual system in prioritizing objects for high-level processing. SOD is a process of identifying and localizing the most attention grabbing object(s) of a scene and separating the whole extent of the object(s) from the scene. In SOD, significant research has been dedicated to design and introduce new features to the domain. The existing saliency feature space suffers from some difficulties such as having high dimensionality, features are not equally important, some features are irrelevant, and the original features are not informative enough. These difficulties can lead to various performance limitations. Feature manipulation is the process which improves the input feature space to enhance the learning quality and performance.

Evolutionary computation (EC) techniques have been employed in a wide range of tasks due to their powerful search abilities. Genetic programming (GP) and particle swarm optimization (PSO) are well-known EC techniques which have been used for feature manipulation.

The overall goal of this thesis is to develop feature manipulation methods including feature weighting, feature selection, and feature construction using EC techniques to improve the input feature set for SOD.

This thesis proposes a feature weighting method utilizing PSO to explore the relative contribution of each saliency feature in the feature combination process. Saliency features are referred to the features which are extracted from different levels (e.g., pixel, segmentation) of an image to

compute the saliency values over the entire image. The experimental results show that different datasets favour different weights for the employed features. The results also reveal that by considering the importance of each feature in the combination process, the proposed method has achieved better performance than that of the competitive methods.

This thesis proposes a new bottom-up SOD method to detect salient objects by constructing two new informative saliency features and designing a new feature combination framework. The proposed method aims at developing features which target to identify different regions of the image. The proposed method makes a good balance between computational time and performance.

This thesis proposes a GP-based method to automatically construct foreground and background saliency features. The automatically constructed features do not require domain-knowledge and they are more informative compared to the manually constructed features. The results show that GP is robust towards the changes in the input feature set (e.g., adding more features to the input feature set) and improves the performance by introducing more informative features to the SOD domain.

This thesis proposes a GP-based SOD method which automatically produces saliency maps (a 2-D map containing saliency values) for different types of images. This GP-based SOD method applies feature selection and feature combination during the learning process for SOD. GP with built-in feature selection process which selects informative features from the original set and combines the selected features to produce the final saliency map. The results show that GP can potentially explore a large search space and find a good way to combine different input features.

This thesis introduces GP for the first time to construct high-level saliency features from the low-level features for SOD, which aims to improve the performance of SOD, particularly on challenging and complex SOD tasks. The proposed method constructs fewer features that achieve better saliency performance than the original full feature set.

Dedication

I would like to dedicate this thesis to my lovey mother for her unconditional love, support, and encouragement.

Acknowledgments

I would like to express my sincerest gratitude to those without whom this thesis would not be possible.

To be begin with, I would like to thank my supervisors, Assoc. Prof. Bing Xue, Prof. Mengjie Zhang, Dr. Christopher Hollitt, and Dr. Harith Al-Sahaf for their expertise, guidance, and academic support. They spent countless hours to go through my thesis and provided constructive feedback. My special thanks go to Prof. Mengjie Zhang for being approachable and responsive despite his busy schedule. Prof. Mengjie Zhang is a perfect example of tenacity and leadership. His work ethics and discipline inspired me to achieve more every day. I am indebted to Dr. Christopher Hollitt, and Dr. Harith Al-Sahaf in more ways than they can imagine. They got the best out of me through difficult but necessary questions and lengthy reviews. They challenged my thinking and helped me broaden my understanding of my research area. Dr. Harith Al-Sahaf contributed immensely to my thesis through his brilliance, intuition, and well rounded approach. It was my absolute pleasure and a privilege to work with my supervisors.

I am especially thankful to Victoria University of Wellington for awarding me the Victoria Doctoral Scholarship and the Victoria Doctoral Submission Scholarship. I appreciate my supervisors for supporting various applications and grants over the course of my PhD.

Finally, I wish to express my greatest gratitude to my friends, colleagues, and my dearest family for their love, patience, and support.

List of Publications

- Shima Afzali, Bing Xue, Harith Al-Sahaf, and Mengjie Zhang, "A supervised feature weighting method for salient object detection using particle swarm optimization," *IEEE Symposium Series on Computational Intelligence*, Hawaii, USA, 2017. pp. 1-8.

- Shima Afzali, Harith Al-Sahaf, Bing Xue, Christopher Hollitt, and Mengjie Zhang, "Foreground and background feature fusion using a convex hull based center prior for salient object detection," *In Proceedings of the 30th International Conference on Image and Vision Computing New Zealand*, IEEE Press. Auckland, New Zealand, 2018, pp. 1-9.

- Shima Afzali, Harith Al-Sahaf, Bing Xue, Christopher Hollitt, and Mengjie Zhang, "A genetic programming approach for constructing foreground and background saliency features for salient object detection," *In Proceedings of the 31st Australasian Joint Conference on Artificial Intelligence*, *Lecture Notes in Computer Science*, Springer. Wellington, New Zealand, 2018, pp. 209-215.

- Shima Afzali, Harith Al-Sahaf, Bing Xue, Christopher Hollitt, and Mengjie Zhang, "Genetic programming for feature selection and feature combination in salient object detection," *In Proceedings of the 22th European Conference on Applications of Evolutionary Computation, Lecture Notes in Computer Science*. Leipzig, Germany, 2019. pp. 308-324.

- Shima Afzali, Harith Al-Sahaf, Bing Xue, Christopher Hollitt, and Mengjie Zhang, "An automatic feature construction method for salient object detection: A genetic programming approach," *Expert Systems with Applications*. (Under review).

Contents

List of Tables

List of Figures

Chapter 1

Introduction

This chapter provides the problem statement, motivations, goals, contributions, and organization of the thesis. Firstly, the problem statement gives an introduction to this thesis. The motivations part discusses limitations of existing research works. The research goals describes the main objectives of this thesis. Finally, the chapter ends with a brief discussion on the thesis organization.

1.1 Problem Statement

The human visual system uses visual attention to efficiently interpret complex natural scenes containing visually distinctive objects. In computer vision, a visual attention mechanism called "*visual saliency*" has been widely investigated to simulate the capability of the human visual system in prioritizing objects for high-level processing [102]. The task of identifying foreground object(s) in a scene for visual attention is described as *salient object detection* (SOD) [27]. The goal of a SOD method is to correctly separate foreground objects as a whole and suppress background sufficiently. Comprehensive surveys can be found in [27, 57, 86]. SOD can be helpful to relieve computational demand in complex vision problems such as scene understanding by detecting and separating salient objects from background

in the image [27]. Objects apparently catch more attention than background regions such as grass, sea, and sky. Therefore, if generic objects can be detected early in a machine vision pipeline, scene understanding could be performed more effectively in the subsequent stages. SOD therefore serves as an important pre-processing step for many tasks [27], such as image classification and image retargeting [147]. Many applications benefit from saliency detection such as image retrieval [16], image and video compression [72], object recognition [144], object tracking [202], content-aware image resizing [18], and object segmentation [179].

Based on biological, computational, and mathematical concepts, SOD methods can broadly be classified into two groups, bottom-up and top-down methods [201]. Bottom-up methods [75, 119, 149] are data-driven that attempt to extract multiple low-level features such as intensity, color, location, and texture. Due to the absence of high-level knowledge, the majority of bottom-up methods attempt to find unusual areas of an input scene. Top-down methods [77, 112, 189] are task-dependent and usually utilize domain-specific/prior knowledge [105].

Both top-down and bottom-up SOD methods rely largely on saliency features that are extracted from different scales to compute a final saliency map or binary mask. Therefore, many studies have developed a rich set of saliency features including heuristic features [27], hand-crafted local features [75], global features [76, 138], and hybrid [28], and indicated the importance of powerful feature representations for SOD. A detailed review of these features can be found in [25]. Since the number of saliency features has been increasing in recent years [25], more effort is required for the process of evaluating and comparing the effectiveness of the existing saliency features on different types of images, and designing new informative features. Feature manipulation techniques, including feature selection, feature weighting, and feature construction, can improve the quality of the feature set in order to improve the SOD performance. Feature selection selects a subset of original features and feature construction generates

novel features from the original features [128]. Feature weighting assigns a weight to each feature based on its relevant importance.

In principle, more features means more discriminative power [33]. However, in practice excessive features cause unjustified computational demands. Finding an optimal feature set is difficult not only because of a large feature space, but also feature interaction problems. Feature interactions can be two-way, three-way or complex multi-way among features [55]. A saliency feature, which is weakly relevant to the target concept by itself, could significantly improve the performance if it is used together with some complementary features. In contrast, an individually relevant saliency feature may become redundant when used together with other saliency features. The removal or selection of features may miss the optimal feature subset(s). Therefore, feature selection has the potential to improve the performance of a system and computational cost by selecting a set of useful and complementary features from a large number of original features, and removing irrelevant and redundant features [87].

Since different image types, e.g., images with large salient object, small salient object, multiple salient objects, and cluttered background have diverse properties of salient objects and background regions [27], different feature sets should be exploited to have effective and efficient detection results [180]. Exploring and finding different effective feature sets based on image types is a challenging task, since it is time-consuming and requires domain knowledge. Feature selection has the potential to select suitable features from a wide range of saliency features for different image types. However, feature selection tasks often suffer from having a large and complicated search space [154]; utilizing a powerful search technique such as genetic programming (GP) [82] and particle swarm optimization (PSO) [78] will be more efficient to find better solutions.

Feature weighting (FW) is one of the important feature manipulation techniques that provides the relative importance of different features when combining features [136]. Weighting features gives a chance to pri-

oritize the features based on their importance degrees on different image types. Some SOD studies [45, 76, 119] have manually designed optimization frameworks for weighting features. However, it is not easy to manually investigate the priority of each feature in relation to the other features in the feature set, or the importance of each feature in the feature set based on the different image types. Hence, the difficulties in manual feature prioritization increases the motivation to develop an automatic feature weighting method.

Since feature selection does not generate new features, it will not be helpful if the original features are not informative enough to achieve good performance, hence, feature construction can be helpful [128]. Feature construction is a means to enhance the representation quality of the data, where the original features may not provide enough discrimination for learning algorithms [128]. In this case, feature construction aims at combining sets of features to obtain new features with stronger discriminating power than the original ones. Therefore, the capability of a learning algorithm can be improved. Moreover, as the constructed features are combinations of the original features generated by linear or non-linear constructive operators (e.g. addition and division), they consider the interactions of the original features.

Saliency features have thus far been dominated by hand-crafted and low-level features which are often effective in simple scenarios, but they are not always robust in some challenging situations [194]. High-level saliency features, on the other hand, can capture high-level information in challenging and complex images. However, designing high-level saliency features is a challenging task that requires expertise in both image analysis and task-domain. An automatic feature construction method can help to tackle with mentioned difficulties and consequently improve the final result. As the potential of newly constructed high-level features from the original features have not been widely studied in the SOD field, it is worthwhile to study different feature construction approaches.

1.2 Motivations

In SOD, significant research has been dedicated to design and introduce new features to the domain. However, there has not been much investigation on the existing saliency features and how to use them in new SOD methods.

The majority of feature spaces have difficulties and a saliency feature space is not an exception. The difficulties include high dimensionality, features that are not equally important, some features that are irrelevant, redundant or even noisy, the original features are not informative enough, and the features are not linearly separable [109]. These difficulties can lead to performance degradation or longer computational time.

Using the existing saliency features stimulates some possible questions. For example, whether the selected feature is informative for different image types (e.g., images with little color variation, having cluttered background, and having multiple objects), how it effects other features, whether it is a duplicate feature in the domain, which types of features it can complement, and how can we effectively use the new feature in different application scenarios. It is very difficult to provide a priori answers for such questions. We therefore aim to investigate these questions and explore answers.

There are some potential ways to answer the aforementioned questions. One plausible way is to use a domain expert, and this way has some difficulties such as requiring domain knowledge of the task, time-consuming, costly, and no guarantee to have comprehensive knowledge. Another way is developing a heuristic method, which is very popular in the literature [138, 188]. However, it becomes more and more challenging to design heuristic methods that are able to fully explore the potential of the existing features [27], when the SOD datasets become more difficult by including more complex images. Another plausible and effective solution is to develop automatic, domain-independent methods for feature

manipulation including feature weighting, feature selection, and feature construction. The feature manipulation methods will widely explore characteristics of the existing and newly created features and find the relationship between them. In addition, they have the ability to select informative and non-redundant features that complement each other for different image types.

The number of the existing saliency features is large which leads to a complex feature space, because of the diversity of saliency features. For high-dimensional and complex feature spaces, feature manipulation is usually required in order to avoid the curse of dimensionality and reduce the risk of overfitting. A few related researches for feature manipulation on SOD has been reported to date. This thesis aims to fill the gap and investigate the possibility of feature manipulation to improve SOD methods.

Generally, feature manipulation can be performed by three approaches: filter, wrapper and embedded approaches [118]. Filter methods evaluate candidate solutions based on the general characteristics of the training data rather than the feedback of a learning algorithm [146]. Wrapper methods employ a learning algorithm to evaluate the goodness of candidate solutions [159]. Finally, embedded methods do feature selection and build a learning model in one step [46]. Although wrapper-based methods typically result in better performance in comparison to filter based methods, they usually have a high computational cost and the goodness of the solution depends on the performance of the inductive algorithm [5]. Embedded feature manipulation methods can combine the advantages of the two other approaches, since it employs an inductive algorithm to evaluate features and avoid the high computational cost. The GP-based embedded approaches can also provide better understanding of the interactions between features due to making a link between the feature manipulation (e.g. feature construction) and the inductive algorithm [56].

Evolutionary computation (EC) is a sub-field of artificial intelligence

(AI) that includes well-known algorithms for optimization and learning tasks [20]. For decades, many researchers have been using EC techniques for different purposes [8, 48, 58, 167]. Similar to other studies [5, 8, 48, 128, 130, 167], this thesis aims at using EC techniques for feature manipulation, e.g., feature selection, feature weighting, and feature construction in SOD. Although these studies [5,8,48,128,130,167] may use EC techniques for the similar purposes (e.g. PSO for feature selection), they apply EC techniques to their problems (problems which are different from SOD) in different ways. Different research areas (such as biomarker identification, edge detection and image classification) aim to address different problems having datasets (e.g. Mass spectrometry datasets, and image datasets) with specific characteristics in terms of instances, features and data types (e.g. continuous, discrete). For example, in high-dimensional classification problems, a dataset includes thousands to tens of thousands of features and most of the high-dimensional datasets have continuous values, while in SOD problems, a dataset contains 2D images and features are low-level (e.g color, intensity [75]), hand-crafted (e.g. objectness of image window [13]), and high-level (e.g. Face, people [77]). Although EC techniques have been used for the domains such as image classification and edge detection with image-based datasets, the goal and characteristics of the SOD problem is different from those domains. Therefore, EC techniques are needed to be investigated and evolved in a way more specific and suitable to this field.

EC techniques have been widely employed for feature manipulation based on the following reasons [183]: 1) They do not make any assumption about the problem, such as whether it is linearly or non-linearly separable, and differentiable, so that they are widely applicable, 2) They do not need domain-specific knowledge, 3) They keep a population of initially randomised solutions, which makes them robust, which is particularly critical for problems with many local optima, and 4) In feature manipulation, because of the highly complex feature interaction issues, it is extremely chal-

lenging to predict which features working together can achieve the best performance, even for domain experts. EC techniques have the potential to find solutions that are even better than the best solution designed by human experts [183].

1.2.1 Feature Manipulation (FM)

1.2.1.1 Feature Selection (FS)

Feature selection is the task of selecting a small subset of relevant features from an original feature set. By eliminating unnecessary and redundant features, feature selection can reduce the dimensionality of the data, speed up and improve the learning process, simplify the learnt model, and/or increase the performance. In SOD, a large number of different features from different levels of information have been developed [113, 138, 203]. However, the existing saliency features are not all essential. Some features are redundant or not complementary to the other features. Using the full set of possible features would negatively affect the performance of an algorithm. The following reasons motivate us to study feature selection for SOD.

- In SOD, if using only a few features gives better or comparable results compared to a large number of features, we prefer to employ a few features. For instance, in some simple images when salient object and background are both homogeneous and the salient object has a high contrast with background, a feature such as color spatial distribution [113] is often enough to produce a correct saliency map.

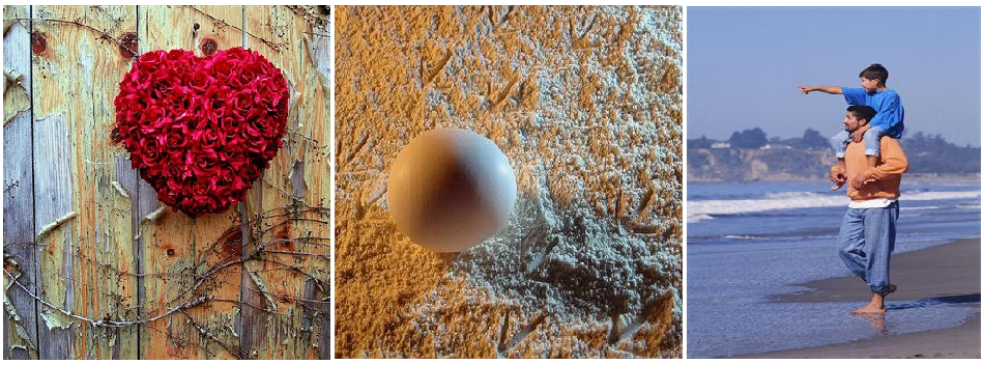
- In order to have a precise saliency map, features are required to be complementary to each other well. Some features can complement each other, while some features may negatively be in conflict with other features in a feature combination. For example, when a

salient object and its background have similar colors, the combination of global contrast [119] and compactness [138] features may fail to highlight foreground regions uniformly, whilst local contrast [65] and compactness features can appropriately complement each other [69] in this case. Another example, when salient object and background are both homogeneous and there is a high contrast between them, the combination of three features including multi-scale contrast, center-surround histogram and color spatial-distribution [113] will normally segment salient regions from the background. However, the combination of the mentioned features may not properly suppress the background regions when the background is cluttered. In this case, the combination with a good background feature may help to address the problem.

- Exploring a large space of features and selecting the informative ones which can complement each other is a challenging task. Obtaining domain knowledge and expertise to achieve feature reduction is hard and time-consuming. Hence, it will be favourable to have a method which can automatically reduce the feature space without human intervention.

1.2.1.2 Feature Weighting (FW)

In the majority of SOD methods, the final saliency map is produced by combining different selected/extracted features. In this combination, considering the relative contribution of each feature is important [113]. For example, for the three images shown in Figure 1.1, different features will be more meaningful in the salient object detection task. It can be considered that in Figure 1.1(a), there is a high contrast between the salient object and the background, therefore, the color feature can be given higher importance than the texture feature. In Figure 1.1(b), the texture feature will be more informative than a color feature, since the texture between the



(a) (b) (c)

Figure 1.1: Example images

salient object and background is different, while in Figure 1.1(c) both color and texture features are informative, however, finding a suitable balance between them is important to achieve the best performance. Assigning a level of importance to each feature for a dataset containing different types of images is a very challenging task.

1.2.1.3 Feature Construction (FC)

Low-level and hand-crafted features can handle simple SOD scenarios, but they usually have difficulties in challenging cases [194]. Moreover, the majority of low-level and hand-crafted features have been manually designed to capture and focus on some aspects or parts of images. In order to tackle the drawbacks of the limited capability of these features, high-level features have been more recently introduced [66]. A good high-level feature aims to capture the general concept along with the details of salient objects. Recently, efforts to develop deep convolutional neural networks (CNNs) for SOD have achieved good results. However, most CNN-based methods unavoidably drop the location information and low-level fine details (e.g. edges and corners) of salient objects, leading to unclear/blurry

boundary predictions [66, 98].

The majority of the reported high-level saliency features have been manually developed/designed by domain experts [27]. However, manually exploring and extracting/designing powerful and high-level features is a difficult task, especially in complex image types, e.g., images with cluttered backgrounds or low-contrast between the foreground object and background. The manually feature designing process suffers from three potential difficulties: 1) obtaining domain knowledge is difficult and time-consuming, 2) domain experts are not always available and are expensive to employ, and 3) designing high-level features is required to be done carefully to ensure robustness to different image types. Therefore, an automatic method to generate high-level features would have many advantages.

Feature construction is a helpful means to automatically construct new high-level features from the original features that are expected to better capture the target saliency [109]. In producing new features, a feature construction technique explores the relationship between the original features, so it will produce new features which are more beneficial than the original ones [168].

In this thesis, the difference between "feature combination" and "feature construction" is the final output of each of these two operations/tasks. The former will produce the final "saliency map" whereas the latter will produce only "a feature" that needs to be combined with other features to produce the saliency map. In other words, feature construction is a preprocessing step of the feature combination task.

1.2.2 EC Techniques for SOD

In the literature, EC techniques have been employed in a wide range of tasks due to having powerful search techniques [40]. Feature manipulation has been successfully performed by different EC techniques such as

PSO [19, 106, 120], GP [130], and GAs [135]. A few studies have utilized EC methods for SOD and achieved promising results [124–126, 151]. However, the usage of EC techniques is still relatively new in SOD, and these techniques have potential for further investigation.

1.2.2.1 Why PSO for Feature Weighting?

Feature weighting needs to optimise weights (continuous values) of all features simultaneously, which requires a powerful continuous optimization method. PSO is an effective and efficient global search technique [40]. The following advantages of PSO makes it a suitable algorithm to address the feature weighting problem in SOD.

- PSO provides an appropriate representation for feature weighting, where each particle is a complete solution (i.e. potential weight vector) and each entry of a particle's position is used to represent a weight for a feature;
- Compared to the other EC methods (e.g., GA, GP), PSO is easier to implement, it converges quickly and has few parameters to tune [40]. PSO can perform as effective as GA, but is computationally more efficient than GA [59]; and
- PSO has shown promise in feature weighting in machine learning tasks [136, 150].

1.2.2.2 Why GP for Feature Selection and Feature Construction?

GP is an EC technique which explores a search space and automatically evolves computer programs (solutions) [82]. GP has the ability to solve various complex problems in many research areas such as classification [41, 152, 158], object detection [107, 193], and high-dimensional data [164, 165]. GP has been widely applied for feature selection [117, 129], feature

construction [5, 34, 130, 131], and both feature selection and construction simultaneously [6, 164, 165].

- One of the advantages of GP over other EC techniques is the ability to perform multiple tasks simultaneously. For example, Lensen et al. [94] developed a GP approach to simultaneously select regions, extract features, and perform binary classification on a given image. Another example, Al-Sahaf et al. [9] showed that GP has the ability to automatically extract features from raw pixels, perform feature selection, and finally image classification. Moreover, GP makes the proposed approaches free from any requirement for human intervention or domain-specific knowledge. Similar to the other image-based problems, GP can be applied to do multiple tasks simultaneously in SOD. For example, GP can be used for saliency feature selection and combination tasks simultaneously;

- GP is well-known for having a flexible program representation which is an important and powerful property of GP [82, 83, 89]. This characteristic of GP makes it suitable for saliency feature construction process in SOD. Although both PSO and GA as EC methods have straightforward representations, neither of them is suitable for feature construction. GA and PSO do not have a flexible representation and they are not good at combining features using mathematical operations;

- SOD with a large variety of features has a complex feature space. Complex search spaces often cause the problem of becoming stuck in local optima. So it needs a global search technique. EC methods are well-known for their ability to search globally [185]. GP as an EC algorithm is able to effectively search large and complex spaces to find optimal or near-optimal solutions;

- Recently, GP has been successfully used for feature manipulation

tasks, particularly feature construction [6, 115, 141, 166] in image-based and non-image-based problems. Unlike traditional feature construction algorithms (e.g., principle component analysis) coming with certain assumptions and constraints and are limited to certain types of transformation, GP has the ability to build a variety of transformations without being bounded to any predefined templates [128];

- GP can usually handle tasks with a very small number of instances/images [12], which provides an opportunity to work on the datasets with a small number of images. Unlike GP, deep learning based CNN methods have difficulties when the dataset has a small number of images;

- Unlike the majority of SOD methods which are manually designed, GP can automatically generate solution which is not thought about by domain experts [7].

1.3 Research Objectives

The overall goal of this thesis is to develop feature manipulation methods using EC techniques to improve the quality of the feature space based on existing saliency features to enhance the SOD performance. This goal can be broken down to:

1. *Develop a feature weighting method using particle swarm optimization to automatically find a suitable weight vector to improve the salient object detection performance.*

   In the majority of the existing SOD methods, the final saliency map is produced by combining features. The relative contribution of each feature is important in this combination. A suitable weight must to be assigned to each feature to reflect the importance of each feature.

There are a few studies that consider weighting features, however, they have been manually designed [45, 76, 119]. While, the feature weighting task is favored to be independent of domain knowledge and human intervention.

This thesis aims to develop a new PSO-based method to automatically generate a suitable weight vector for combining features.

The new method will be compared to the benchmark methods on three benchmark datasets. We will also investigate whether the generated weight vector is capable of showing the importance of different features in a feature set. Moreover, we will compare the combination of the weighted features to the combination of the non-weighted features based on precision-recall curves on three different benchmark datastes.

2. *Develop a bottom-up salient object detection method for manually constructing new informative foreground and background features and a feature combination framework.*

   Saliency images generally contain two parts, foreground object(s) and background. Hence, an SOD problem can be decomposed into two tasks, identifying foreground object and background. Among the existing features, some features are good at identifying the foreground object(s), while some others perform well on reflecting background. Saliency features may not be individually informative and strong enough to completely capture the foreground object(s) or background, but those features might be effective and informative when appropriately combined with other saliency features.

   This thesis aims to construct informative foreground and background features using the existing features. The constructed features are combined using a newly introduced feature combination framework in this work. Previous studies employed image center prior to assign higher saliency to the regions near the image center [162, 163].

However, this principle becomes invalid when the objects are placed far from the image center. To avoid this problem here, the feature combination framework is designed based on foreground object center prior for assigning saliency values to image pixels.

This thesis will analyse and compare the constructed features to the individual saliency features using precision-recall curves and qualitative results. Moreover, it will investigate whether the constructed saliency features and the newly designed combination framework can improve the performance of SOD. The proposed bottom-up SOD method will be compared to the other competitive SOD methods on three benchmark datasets regarding three evaluation criteria such as precision-recall (PR) curve, F-measure, receiver operating characteristic (ROC) curve, running times, and statistical significance test based on the area under the PR curve (AUCPR).

3. *Develop a genetic programming based method for automatically constructing new informative foreground and background saliency features.*

   Manually selecting saliency features from the existing features and constructing new features is an expensive and challenging task. This process relies on domain knowledge and expertise and becomes increasingly difficult as the complexity of candidate models increases.

   To relieve human intervention and domain knowledge, this thesis aims to develop an automatic GP-based feature construction method. As for the work for the second objective, foreground and background features are constructed to complement each other, so that each can improve the other's shortcomings. However, unlike the second objective, the whole process from selecting, combining, and constructing features is automatically achieved using GP. Two different fitness functions are used to guide GP to construct different foreground and background features, since these two features have different targets.

The automatically constructed GP-based foreground and background features will be compared to the manually constructed ones in the previous objective. The thesis will investigate whether the automatically constructed features improve the performance of SOD. The obtained results will be compared with the results of other SOD methods on four benchmark datasets based on quantitative and qualitative results.

4. *Develop an automatic genetic programming based method for salient object detection*

   Employing more informative and diverse saliency features can enhance the power of SOD methods to capture the distinctive information between the foreground object(s) and background in challenging images. However, large feature spaces cause difficulties such as increasing the complexity of feature interaction, and being computationally expensive. Therefore, a suitable feature selection method is required to select effective features. Mostly popular SOD methods manually design or select features from existing features and design a framework to combine the features to produce the final saliency map [27].

   This thesis aims to develop a new GP-based method to automatically select saliency features and generate a mathematical function to combine those features to produce the final saliency map. The new GP-based method considers the complementary characteristics of the selected features for the combination stage. An appropriate fitness function will be designed to evaluate GP solutions. The new fitness function will measure the difference between two probability distributions of the GP output and ground truth, since the saliency distribution of GP output is required to be similar to the ground truth.

   This thesis will evaluate the proposed GP-based method using four

datasets of varying difficulties to test the generalizability property of this method. It will explore whether the proposed method can select effective feature subsets for complex image types. Moreover, the performance of the proposed method will be compared to that of seven hand-crafted SOD methods drawn from the literature to test whether those automatically evolved programs have the potential to achieve comparable or better performance to the domain-expert designed ones.

5. *Develop a genetic programming based method for automatically constructing high-level saliency features*

   The performance of a SOD method mainly relies on saliency features which are extracted from different levels. Low-level and hand-crafted features are often effective in simple scenarios, but they are not always robust in some challenging cases. Moreover, the majority of low-level and hand-crafted features are manually designed/extracted by domain experts and they often focus on detecting some parts of the image. To tackle the drawbacks of the limited capability of these features, high-level features have been recently introduced [66]. Recently, deep convolutional neural networks (CNNs) methods developed high-level and semantic features which result in good progress in SOD. However, CNN-based constructed high-level features unavoidably drop the location information and low-level fine details (e.g., edges and corners) of salient object(s), leading to unclear/blurry boundary predictions [66, 98]. Meanwhile, manually designing high-level saliency features is a challenging task and requires domain knowledge.

   The thesis aims to develop a GP-based method to automatically construct new high-level features for SOD. Similar to the previous objective, the proposed method will take low-level and hand-crafted features as input to construct high-level features. Here, the final result

is high-level constructed feature, unlike the previous objective (objective 4) where the final result of the GP algorithm was a saliency map. Moreover, this objective aims to reduce the complexity of the input saliency feature space with a feature grouping method.

This thesis will investigate whether the GP-based high-level saliency feature(s) can obtain better results than the low-level and hand-crafted features. We will provide analysis on the evolved GP programs and selected saliency features by those programs. We will provide visual examples of the constructed features and discuss how they impact different regions of the image and complement each other in the combination stage. Moreover, we will show the statistical test results based on AUCPR to show how the new method significantly outperforms the benchmark SOD methods on the different benchmark datasets.

1.4 Major Contributions

This thesis contributes to the following important aspects in the fields of evolutionary computation and computer vision, specifically in salient object detection.

1. This thesis shows how feature weighting can be helpful in identifying the relative contributions of each feature in the feature combination process for SOD. The thesis develops a PSO-based method that can improve the results of feature combination by automatically weighting saliency features. The proposed method is a supervised learning method and does not require any assumptions or domain knowledge. Experimental results show that employing the weighted features has better performance than non-weighted features, thus it has a positive influence on the SOD performance.

   Shima Afzali, Bing Xue, Harith Al-Sahaf, and Mengjie Zhang, "A

supervised feature weighting method for salient object detection using particle swarm optimization," *IEEE Symposium Series on Computational Intelligence*, Hawaii, USA, 2017. pp. 1-8.

2. This thesis shows how to separately consider foreground object(s) and background in saliency images and construct corresponding saliency features that provide better representation for each. This thesis investigates how different saliency features can be informative on different regions of the image and how they are required to be combined with each other to improve the detection result. The thesis develops an unsupervised SOD method that does not require any ground truth for the saliency images. We provide a discussion regarding the complementary characteristic of saliency features and the impact of features on highlighting or suppressing background. The quantitative and qualitative results demonstrate that the foreground and background saliency features outperform each individual feature.

   Shima Afzali, Harith Al-Sahaf, Bing Xue, Christopher Hollitt, and Mengjie Zhang, "Foreground and background feature fusion using a convex hull based center prior for salient object detection," *In Proceedings of the 30th International Conference on Image and Vision Computing New Zealand*, IEEE Press. Auckland, New Zealand, 2018, pp. 1-9.

3. This thesis shows for the first time how GP can be utilized for automatically constructing saliency features for SOD. This thesis addresses the problem of involving domain knowledge and human intervention in feature construction by developing a GP-based method. The new method can automatically and implicitly handle feature interaction and build a suitable relation among the features using mathematical operations. The results of the experiments reveal the potential of this method to significantly outperform domain-

expert hand-crafted features (e.g. the constructed features in the second contribution) and improve the performance of SOD.

Shima Afzali, Harith Al-Sahaf, Bing Xue, Christopher Hollitt, and Mengjie Zhang, "A genetic programming approach for constructing foreground and background saliency features for salient object detection," *In Proceedings of the 31st Australasian Joint Conference on Artificial Intelligence, Lecture Notes in Computer Science*, Vol. 11320. Springer. Wellington, New Zealand, 2018, pp. 209-215.

4. This thesis shows how GP can be utilized for automatically producing a saliency map for a given image in SOD. As mentioned before, saliency map is the final result of a SOD method, while saliency features are feature maps which saliency map is produced by. The thesis also shows how the proposed GP-based method can handle a large and complex feature space of the input saliency features and select the distinctive features. The proposed GP-based method has a good generalizability over different types of images as it can evolve a solution that is suitable for the majority of the images in a given dataset. The proposed method can incorporate any additional features and usually select the features that are complementary to each other. It has the ability to explore a wide range of saliency features which are extracted from different segmentation levels and search for various mathematical expressions for the feature combination stage.

   Shima Afzali, Harith Al-Sahaf, Bing Xue, Christopher Hollitt, and Mengjie Zhang, "Genetic programming for feature selection and feature combination in salient object detection," *In Proceedings of the 22th European Conference on Applications of Evolutionary Computation, Lecture Notes in Computer Science*. Vol.11454, Leipzig, Germany, 2019. pp. 308-324.

5. To improve SOD performance, especially on challenging/complex saliency images, this thesis introduces GP for the first time to con-

struct high-level features from low-level and hand-crafted features. Unlike the third contribution which in GP constructs two specific features which separately focus on foreground and background detection, GP constructs a general high-level feature without any constraints in this contribution. This thesis also proposes a feature subset preparation method to provide different input feature sets for GP to insure the diversity in constructing high-level saliency features. Compared with the low-level and hand-crafted features, the constructed high-level saliency features are more informative and have better generalizability. The GP-based constructed features have the ability to capture salient regions and suppress background regions over the whole image. Moreover, the GP-based constructed features have better interpretability compared to CNN-based features. The final saliency map produced by the combination of the high-level constructed features is more accurate compared to the methods which only employ low-level and hand-crafted features. The reason is that the constructed high-level features can capture high-level knowledge from the foreground object and background of a given image.

Shima Afzali, Harith Al-Sahaf, Bing Xue, Christopher Hollitt, and Mengjie Zhang, "An automatic feature construction method for salient object detection: A genetic programming approach," Submitted to: *Expert Systems with Applications*.

1.5 Organization of the Thesis

The remainder of this thesis is organized as follows. The literature of related work is reviewed in Chapter 2. The main contributions of the thesis are presented in Chapters 3-7. Chapter 8 concludes this thesis.

Chapter 2 presents an introduction to salient object detection and evolutionary computation techniques such as GP and PSO. This chapter gives

a detailed description about existing different types of saliency features. EC-based and non-EC-based SOD methods are reviewed with their shortcomings highlighted, which form the motivation for the work presented in this thesis.

Chapter 3 proposes a PSO-based feature weighting method for computing the level of importance for each saliency feature. A new fitness function is defined to evaluate the evolved weight vectors. The performance of the proposed method is examined on three different datasets and evaluated using precision-recall curves, ROC curves, average precision, recall, F-measure, and AUCPR.

Chapter 4 designs two new informative saliency features and a combination framework for computing saliency maps. This chapter provides detailed description of how to evolve the foreground and background saliency features, and also the feature combination framework. The manually constructed foreground and background features are investigated whether they perform better than the individual hand-crafted features using precision-recall curves and qualitative results.

Chapter 5 extends Chapter 4 by automatically constructing those foreground and background saliency features. It discusses the capability of GP in selecting and constructing new saliency features using GP. In this chapter, two different fitness functions are developed for constructing foreground and background features separately. The automatically constructed features are investigated to show how they perform better than the manually constructed features in Chapter 4 and also individual ones (e.g. low-level features).

Chapter 6 proposes a novel GP-based method to produce a saliency map for a given image. This chapter discusses the capability of GP in exploring a large search space of saliency features from different segmentation levels and finding a suitable combination of them. The further analysis is provided for the evolved GP individuals.

Chapter 7 proposes a GP-based method for automatically constructing

different high-level features. This chapter shows how different GP individuals (solutions) constructed from different types of features can produce diverse features that can perform differently on various image types.

Chapter 8 concludes this thesis and summarises the major contributions of the thesis. This chapter also provides and discusses some possible future research directions.

Chapter 2

Literature Survey

This chapter reviews the literature and provides the basic concepts and terminology of computer vision, salient object detection, and related research areas. This chapter also provides detailed description of well-know saliency features. Moreover, feature manipulation techniques including feature weighting, feature selection, feature construction, and feature extraction are presented. In addition, evolutionary computation techniques such as GP, PSO, and GAs are discussed. This chapter then reviews related works of GP for image analysis, and existing SOD methods, including EC-based, machine learning based, non-EC-based, and deep learning based SOD methods.

2.1 Computer Vision

If vision is defined as a means to know the world by looking, computer vision is the same concept except it acquires the knowledge by a computational instrument rather than the brain [92]. Computer vision can be considered from two points of view, biological science and engineering [67]. In the former view, computer vision aims to devise with computational models of the human visual system. In the latter view, computer vision refers to build autonomous systems which can perform some of the

tasks which the human visual system can perform [67]. More specifically, computer vision is a process of acquiring, processing, analyzing, and understanding useful information from a single image or a sequence of images [148]. Some examples for sub-domains of computer vision are object pose estimation, scene reconstruction, event detection, object tracking, object recognition, motion estimation, and image restoration [156]. There are some other examples which are most closely related to computer vision, image processing, image analysis, and machine vision [156].

2.2 Salient Object Detection

In the context of visual attention, Tsotsos et al. [169] proposed the term *saliency*. Visual saliency is a fundamental research problem in neuroscience, psychology and computer vision. Saliency detection is a process of identifying and localizing the most attention grabbing region(s) of an image or viewpoint. Saliency detection was originally known as a task of predicting eye-fixation on images [75]. Most earlier methods focused on human eye fixation prediction and they have presented the basic principles of saliency detection. Recently, it is extended to detect region(s) including object(s) which demand more attention than others and then segments the whole salient object, called salient object detection (SOD) [113].

In the SOD domain, the final result of a saliency detection process for a given image is returned either as a saliency map or a binary map. The *saliency map* is a 2-D matrix where each cell corresponds to a pixel in the image. The value of each cell represents the likelihood of that pixel to be salient. A higher value in the saliency map indicates that the corresponding image pixel is more likely to be salient, and vice versa. However, the *binary map* is a 2-D map which is made up of binary values (0 or 1). A pixel is salient if its value is 1 and it is non-salient if the value is 0. Figure 2.1 shows an example of an input image, its corresponding binary map and saliency map.

Input image Binary map Saliency map

Figure 2.1: An example of saliency map and binarized mask for a given image.

SOD methods use different saliency features including low-level, hand-crafted, and high-level saliency features [198]. Low-level features are basic features that are extracted automatically from an image without having any information about existing shapes in the image, such as intensity, color, and texture [134]. By contrast, hand-crafted features are designed by domain experts using low-level features, assumptions, and prior knowledge [76, 99, 119, 138, 142, 162, 188, 203]. For example, objectness of image windows [13], which is computed based on four features: multi-scale, color contrast of a window from its surrounding regions, the edge density inside a window, and the number of superpixels that have their pixels both inside and outside the boundary of the window. High-level features concern descriptive and abstract concepts [64, 93, 97, 98, 100, 110, 174–176, 195, 196, 200] and refer to those features which contain information about shapes and components of objects occurring in an image. These components and shapes could be eyes, nose, and ears in a face detection system [62] or wheels, headlights and tail-lights in a vehicle detection system [95]. These features are typically used for domain-specific tasks such as object classification.

Saliency methods can be classified into two groups, *bottom-up* and *top-down* based on biological, computational and mathematical concepts [197]. In bottom-up methods, multiple low-level features are extracted from an image. The extracted features are then normalized and combined into a

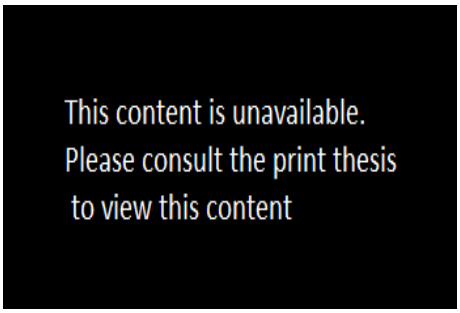


Figure 2.2: The traditional computational model of visual attention proposed by Itti et al. [75]

saliency map. In top-down methods, prior knowledge of the visual system is used to develop an approach targeted to a particular task. These methods are often integrated with bottom-up methods. Top-down methods usually learn models from training examples with manually labelled ground truth.

Figure 2.2 shows the traditional bottom-up computational model of visual attention introduced by Itti et al. [75]. It uses three steps to find the final saliency map. In the first step, saliency features along with choice of feature related parameters such as the number of layers in the image pyramid, types of image pyramids, number and parameters of orientation filters, designation of centre and surround levels, and feature importance weights are extracted. In the second step, a normalization function is used to emphasize features that are likely to contain meaningful saliency information. In the third step, features are combined to produce the final saliency map. Most proposed methods that are inspired by the Itti's model use human encoded parameters and region based feature computation at step 1, employ a fixed normalization function at step 2, do not consider relative feature importance and apply a heuristic combination function at step 3. Therefore, many existing SOD methods suffer from poor generalizability, due to the fixed and heuristic design choices [123].

The terms attention, saliency, and gaze are often used interchangeably,

but each one has its own definition [29]. Attention refers to a general concept that covers all factors that affect selection mechanisms, whether they are bottom-up or top-down. Saliency intuitively identifies some parts of a scene (which could be objects or regions) that appear to be stand out relative to their neighboring regions. Gaze, a coordinated motion of the eyes and head, has often been used as a proxy for attention in natural behavior [60].

2.3 Other Related Research Areas to SOD

There exist some close research areas to SOD such as fixation prediction, image segmentation, object detection, object classification, and object recognition. We provide the definition of these research areas as follows.

Eye fixation refers to the preferential fixation on conspicuous or meaningful regions in a scene at a first glance by human viewers [161, 178] and those regions are correspond with important objects and their relationships [143]. During fixation events, the eyes remain nearly stationary, since the human visual look at details in selected locations. This property makes eye movements a valuable proxy for understanding human attention.

Image segmentation is the process of dividing an image to several homogeneous regions, where pixels within a region are similar based on a homogeneity criterion, while pixels in different regions are heterogeneous [103]. This process is helpful to find regions of interest, therefore, images become easier to manipulate and more meaningful for following higher-level tasks [80].

Object detection is the task of detecting instances of semantic objects of a certain class (such as humans, buildings, or cars) considering the background [155]. Face detection and pedestrian detection are examples of well-researched domains for object detection.

The task of assigning a class label to each instance is defined as object classification [172]. Object classification aims to group similar instances

in one group and identifies each instance as a single object. The object classification task aims to predict the existence of objects within images, whereas the object detection targets localizing the objects.

Object recognition is the task of detecting and identifying objects in an image. Object recognition is the process of performing both object detection and classification simultaneously [17]. The object recognition problem denotes the more general problem of identifying all the objects present in the image and providing accurate location information of the respective objects [17]. There also exist alternative approaches for classifying the various levels of the recognition problem. For example, [63] discerns five levels of tasks of increasing difficulty in the recognition problem, verification, detection and localization, classification, naming, and description.

2.4 Machine Learning

Machine learning is a subfield of AI concerned with the question of how to construct computer programs that automatically improve with experience [133]. Learning by observing examples and making changes to improve their performance is the main characteristic of machine learning algorithms.

Suppose there is a function fc, and the task of the learner is to predict what it is. hp denotes our hypothesis about the function to be learned. Both fc and hp are functions of a vector-valued input $\boldsymbol{XV} = (xv_1, xv_2, ..., xv_{nc})$ which has nc components. It is assumed that hp as being implemented by a device that has $\boldsymbol{XV}$ as input and $hp(\boldsymbol{XV})$ as output. We assume a prior that the hypothesized function, hp, is selected from a class of functions H. We select hp based on a training set TS, of mp input vector examples. There are two major settings in which a function is learned, supervised learning and unsupervised learning [133].

In supervised learning, we have knowledge about the values of fc for the mp samples in the training set TS. It is assumed that if we can find

a hypothesis hp that closely agrees with fc for the members of TS, then this hypothesis will be a good approximation to fc, especially if TS is large [133]. Some supervised learning methods include neural networks, decision trees, support-vector machines, and bayesian filtering [96].

In unsupervised learning, a training set of vectors is given without their corresponding function values. In this case, the training set is divided into subsets, $TS_1, ..., TS_R$, in some appropriate way [133]. Some examples of unsupervised algorithms are clustering algorithms and non-negative matrix factorization [96].

2.5 Evolutionary Computation

Evolutionary computation is a subfield of artificial intelligence (AI) that comprises of nature inspired algorithms. EC algorithms have a similar framework. Generally, EC algorithms typically start with a population of randomly generated candidate solutions and evaluate them using a fitness function as a guide to search for better solutions. Then a termination criterion is checked. If the termination criterion is not met, certain candidates are selected and employed for creating a new generation. Finally, the algorithm returns the best solution of the population, when the termination criterion is satisfied.

EC algorithms can be categorized into two main groups [128]: evolutionary algorithms and swarm intelligence. Evolutionary algorithms search for an optimal solution by employing Darwinian principles of natural selection and applying genetic operators such as reproduction, crossover, and mutation to evolve better solutions. Some algorithms belong to this group are evolutionary programming (EP), evolutionary strategies (ESs), genetic algorithms (GAs), and genetic programming (GP). Swarm intelligence refers to algorithms that are inspired by the collective intelligence of social insects, which utilize interactions among candidates and between candidates and the environment [24]. Typical examples of

this group are ant colony optimization (ACO) and particle swarm optimization (PSO). EC algorithms have been successfully applied to many problems in computer vision such as salient object detection [126], classification [9], edge detection [47], and image segmentation [104].

2.5.1 Genetic Programming (GP)

GP is an EC technique which explores a search space and automatically evolves solutions in the form of computer programs [82]. GP follows the concept of "Survival of the Fittest" where a number of GP individuals (computer programs) with good performance survive during the evolutionary process. Algorithm 1 presents the process of GP algorithm. First, the GP algorithm starts by randomly generating a predefined number (δ) of individuals using a function set ($\digamma$) and a terminal set (τ). To measure the goodness of each individual (ξ), a fitness value is computed using a fitness function (Δ_ξ). To produce the individuals of the next generation (Ξ_{i+1}), GP applies operations including crossover, mutation, and reproduction on the individuals of the current generation (Ξ_i). The individual with better fitness values have higher chance to be selected for generating the next population. The algorithm stops when the generations counter (i) reaches a predefined number of generations (β) or the best fitness value so far (λ) meets the ideal fitness value (γ). At the end, the GP algorithm returns the best solution so far (ϑ).

2.5.1.1 GP Program Representation

A GP program, typically, has a tree-based representation [82]. An evolved program (individual) is made up of a root node, a number of internal nodes, and some leaf nodes. The terminal nodes (leaves) are variables or constants. The internal nodes consist of functions which are usually arithmetic operators (e.g. summation, multiplication, protected division, and subtraction). Functions can represent simple arithmetic operators or com-

Algorithm 1 The GP Evolutionary Process

1: **procedure** GP ($\tau, \digamma, \delta, \beta, \gamma$) ▷ Terminal ($\tau$) and function sets ($\digamma$), population size ($\delta$), number of generations (β), and ideal fitness value (γ)
2: $i \leftarrow 0$ ▷ The generations counter
3: $\lambda \leftarrow +\infty$ ▷ Best fitness so far
4: $\vartheta \leftarrow null$ ▷ Best solution so far
5: $\Xi_0 \leftarrow Generate(\tau, \digamma, \delta)$ ▷ Randomly generate the initial population
6: **repeat**
7: **for all** $\xi \in \Xi_i$ **do**
8: $\Delta_\xi \leftarrow Fitness(\xi)$ ▷ Fitness of the current individual
9: **if** $\Delta_\xi < \lambda$ **then** ▷ If the fitness is better than the best so far
10: $\lambda \leftarrow \Delta_\xi$ ▷ Report the fitness as the best so far
11: $\vartheta \leftarrow \xi$ ▷ Make the individual as the best so far
12: **end if**
13: **end for**
14: $\Xi_{i+1} \leftarrow Populate(\Xi_i)$ ▷ Populating subsequent generation
15: $i \leftarrow i + 1$ ▷ Increment the generations counter
16: **until** ($i = \beta$ or $\lambda = \gamma$) ▷ Check if a termination criterion is met
17: **end procedure**

plex functions such as loop structure and domain specific. An example of a GP individual is demonstrated in Figure 2.3, the mathematical formula of the GP tree is $(\sin(X) \times Z) + (Y - Z)$. The internal and root nodes containing functions $\{+, -, \times, \sin\}$ which are applied on the terminals (inputs) $\{X, Y, Z\}$ or the outputs of the other functions.

The output of a GP program can be produced in different types such as a single numerical/boolean/string value, vector, or matrix, since different applications require different types of solutions. For instance, conditional functions such as if-then-else can be employed as a function in the GP tree to make different decisions. Thus, extending conventional GP to consider

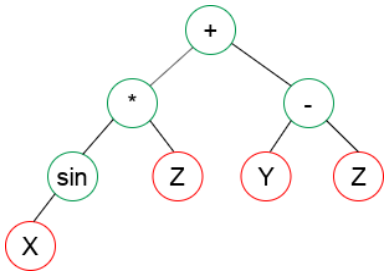


Figure 2.3: An example of a GP individual.

different types has been investigated by different research studies such as grammar-based GP and strongly-typed GP [82, 121, 139].

2.5.1.2 Initialization Methods

GP initiates the evolutionary process by randomly generating individuals. Koza specified three different techniques including *grow*, *full*, and *ramped-half-and-half* for generating the individuals [139]. In the grow method, nodes are randomly selected from the whole primitive set (i.e. functions and terminals) until a terminal is selected or a predefined maximum depth is reached. When the predefined maximum depth is reached, only terminals may be chosen. In the full method, nodes are randomly taken from the function set until the predefined maximum depth is reached, then only terminals can be chosen. The ramped-half-and-half method is a combination of grow and full, which can provide individuals vary in shape and size [82].

2.5.1.3 Evaluation

To evaluate the goodness of each individual (program), an evaluation measure, called a fitness function (e.g., accuracy or AUC for classification, mean squared error for regression) is used [82].

2.5.1.4 Selection Methods

GP system considers a selection method to choose individuals to produce new generation. In this procedure, better individuals (the individuals having good fitness value) are more likely to be chosen than inferior individuals [139]. There are different selection methods such as Roulette wheel selection, truncation selection and tournament selection [31]. In the Roulette wheel selection, the probability of selection of each individual is proportional to its fitness. A program with better fitness value has a higher chance for the selection. The tournament selection method is the most commonly used one [103] and it has two steps. In the first stage, a set of individuals are randomly selected from the population. In the second stage, the selected individuals are compared and the best one is chosen to be in the mating pool.

2.5.1.5 Genetic Operators

To produce the population of the subsequent generation, some genetic operators are employed to create new individuals (children) from the current individuals (parents). There are three genetic operators: reproduction (elitism), crossover, and mutation [23]. These operators are applied based on a defined probability and the summation of all probabilities is required to be 1. Unlike other EC techniques which can be applied sequentially, the GP operators are mutually exclusive [82].

1. Crossover: Figure 2.4 shows an example of the crossover operation. As can be seen, the parent individuals are selected using one of the selection methods. Then, a crossover point is randomly selected in each parent tree. Next, the offspring are created by swapping the sub trees at crossover points.

2. Mutation: Figure 2.5 shows an example of the mutation operation. Firstly, a mutation point is selected on the parent individual and

then the root of the sub tree is replaced with another randomly generated sub tree. There is an important difference between crossover and mutation, mutation allows the system to generate new building blocks, but crossover allows the system to explore different combination of the existing building blocks. Meanwhile, crossover generates two individuals, while mutation generates one individual.

3. Reproduction: In reproduction, a selected individual is simply copied to the next generation. Elitism is a similar operator to the reproduction. In reproduction, the copy operation is applied based on a predefined probability, while elitism selects a predefined number of the best individuals having good fitness values and copies them to the next generation. Hence, elitism avoids degrading the performance and keeps the achieved level of performance during the evolutionary process. To give more chance to the system for exploring the solution space, the probability of the elitism operator is set to a low value.

2.5.1.6 GP Settings

Most GP set-up parameters follow Koza's default parameter settings of GP from Koza's works [82–84]. The GP parameters are summarized in Table 2.1. As can be seen, the population size is 1024 and the number of generations is 51. The minimum and maximum depth of initial GP individuals are 2 and 6, respectively. However, the maximum depth of the individuals can not exceed 17 in the evolutionary process. The rates of mutation and crossover operators are 0.1 and 0.9, respectively. The tournament selection method is employed to choose GP individuals for reproduction. In the tournament selection method, individuals are randomly selected from the population to create tournaments with size of 7. The initilization method is *ramped half-and-half* which is based on the combination of the full and grow methods [82].

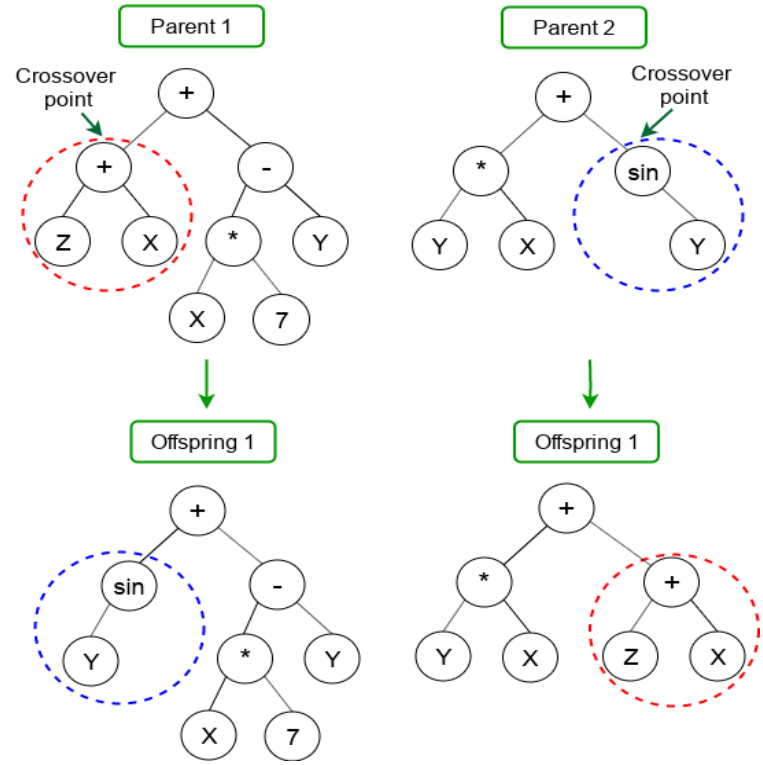


Figure 2.4: Crossover operation.

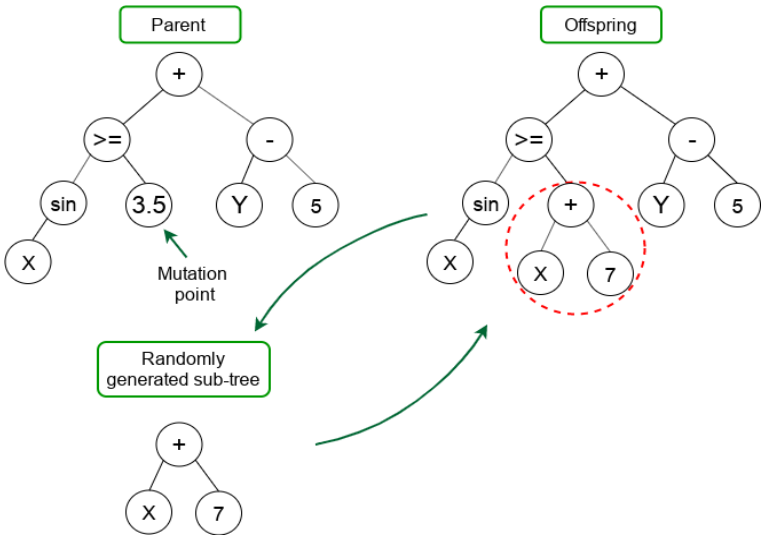


Figure 2.5: Mutation operation.

Table 2.1: Koza's GP settings.

Parameter	Setting	Parameter	Setting
Population Size	1024	Generations	51
Minimum Depth	2	Maximum Depth	6
Mutation Rate	0.10	Crossover Rate	0.90
Reproduction	Keep the single best	Selection Type	Tournament
Initial Population	half-and-half	Size of Tournament	7

2.5.2 Genetic Algorithm (GA)

Genetic algorithms are a popular EC technique introduced by John Holland [21]. Similar to the other EC techniques, GA applies biological evolution during its process. GA is mostly used for optimization problems. GA uses a representation of fixed-length bit string and includes genetic operators such as crossover, mutation, and reproduction for producing new generations or evolving individuals [52].

2.5.3 Particle Swarm Optimization (PSO)

Kennedy and Eberhart [79] developed PSO based on social behaviors of bird flocking or fish schooling. Algorithm 2 presents the process of the PSO algorithm. In PSO, each candidate solution is represented by a particle and a swarm is represented as a population of particles. In a swarm, each particle has its own position and velocity. The position of each particle i is encoded as a *D*-dimensional vector $xp_i = (xp_{i1}, xp_{i2}, ..., xp_{iD})$, where *D* is the dimensionality of the search space. The velocity of each particle i is also represented by a *D*-dimensional vector $vp_i = (vp_{i1}, vp_{i2}, ..., vp_{iD})$. In the search space, particles move around by utilizing their position and velocity to find the optimal solution. The velocity of each particle is limited by a predefined maximum velocity, vp_{max} and $vp_{id}^t \in [-vp_{max}, vp_{max}]$. During the movement, each particle records its previous best position which is called *pbest* and the best position of all particles so far which is called *gbest*. The position and the velocity of each

Algorithm 2 The PSO Algorithm

```
1: procedure PSO(Population_size, MaxIt) output:P_gbest
2:     Population_size ← 0
3:     P_gbest ← 0                                   ▷ set parameters
4:     it = 0                                        ▷ iteration number
5:     for pop ← 1, Population_size do
6:         P_velocity ← randomvelocity(Population_size)
7:         P_position ← randomposition(Population_size)
8:         P_pbest ← P_position
9:         Compute fitness function (P_position)
10:        if cost (P_pbest) ≤ cost(P_gbest) then
11:            P_gbest ← P_pbest
12:        end if
13:        pop = pop + 1;
14:    end for
15:    while it ≤ MaxIt do
16:        it = it + 1
17:        for pop ← 1, Population_size do
18:            P_velocity ← updatevelocity(P_velocity, P_gbest, P_pbest)
19:            P_position ← updateposition(P_velocity, P_position)
20:            Compute fitness function (P_position)
21:            if cost (P_position) ≤ cost(P_pbest) then
22:                P_pbest ← P_position
23:                if cost (P_pbest) ≤ cost(P_gbest) then
24:                    P_gbest ← P_pbest
25:                end if
26:            end if
27:            pop = pop + 1;
28:        end for
29:
```

particle are updated by the following equations.

$$xp_{id}^{t+1} = xp_{id}^{t} + vp_{id}^{t+1} \tag{2.1}$$

$$vp_{id}^{t+1} = w \times vp_{id}^{t} + c_1 r_{i1} \times (p_{id} - xp_{id}^{t}) + c_2 r_{i2} \times (p_{gd} - xp_{id}^{t}) \tag{2.2}$$

where *t* represents the t^{th} iteration, $d \in D$ denotes the d^{th} dimension, w is a predefined constant interia weight, c_1 and c_2 are acceleration constants, r_{i1} and r_{i2} are random values uniformly generated in [0,1], and p_{id} and p_{gd} denote *pbest* and *gbest* in the d_{th} dimension, respectively.

2.6 Feature Manipulation

This section provides definitions of feature manipulation techniques including feature selection, feature construction, feature weighting, and feature extraction.

2.6.1 Feature Selection

In the literature, there are many definitions for feature selection based on different criteria, but most of them follow a similar intuition and/or content [37]. Many image processing tasks, such as SOD, rely on integrating information drawn from constituent features. Feature selection is a process that aims to find a minimal subset of features to achieve similar or better performance than using all the original features by eliminating noisy and irrelevant features [185]. The process of selecting informative and relevant features not only reduces the dimensionality, which can make the learning method faster, but also improves the performance of the method. In addition, it is easy to interpret the learned method with a smaller number of features [128].

There are four major aspects in a general procedure for feature selection [37]: the initialization procedure, candidate feature subset generation, feature subset evaluation, stopping criteria and a validation procedure.

- The initialization procedure is the first step of a feature selection algorithm where the number of original features is taken as the dimensionality of the search space.

- The candidate feature subset generation is known as search procedure [90], which can start with no features, all features, or a random subset of features. In this step, many search techniques such as EC techniques, can be employed to explore the best feature subset.

- The generated candidates are evaluated based on a criterion which is called a fitness function in EC-based feature selection. The fitness function will measure the goodness of each candidate feature subset, so it has an important role in guiding the algorithm to find an optimal solution.

- The feature selection algorithm will be stopped when the stopping criterion is satisfied. The generation procedure and evaluation function can be used to determine the stopping criterion. For example, the algorithm can stop when a predetermined maximum number of iterations have been reached, or a predefined number of features have been selected.

- Validation procedure aims to check whether the subset of features can achieve good performance.

2.6.2 Feature Construction

Feature construction is a process which combines original features to construct new high-level features [199]. Feature construction aims to improve the quality of representation by transforming the original representation space, i.e., features, into a new one in which the capability of a learning algorithm can be improved [129]. Constructed features are mathematical expressions of the original features. In order to enhance the performance

of a method, the original feature set can be augmented or replaced by the new constructed features [85]. Feature construction include the following four steps.

- Feature selection: New features are constructed by combining the selected original features using mathematical operators. The key point is to select appropriate features and operators, so the newly constructed features will have a higher discriminating ability than the original ones.

- Feature evaluation: To guide the search algorithm, the constructed features are evaluated by means of a fitness function in EC-based FC similar in feature selection methods.

- Stopping criterion: Similar to feature selection methods, when a stopping criterion is met, the best constructed features will be returned.

- Validation procedure: This step is similar to feature selection methods. The constructed new features are required to be checked whether they can achieve a good performance.

2.6.3 Feature Weighting

Feature weighting aims to assign a weight to each feature based on its degree of relevance to the target concept. Feature weighting gives high weights to the relevant features and low weights to the irrelevant features. In the SOD task, weighting features is important during the feature combination process. Relief is an example for feature weighting methods that use distance measures to evaluate the degree of feature relevance [88].

2.6.4 Feature Extraction

Feature extraction is a process that extracts a set of new features from the raw data using some functional mapping [182]. Feature extraction can transfer the input data into a different domain or reduce representation set (e.g. aggregating a set of low-level features and calculating a single value out of it). Moreover, this operation can be used to reduce the amount of irrelevant information in the data. To find good transformations, an intensive search is required [128]. Feature extraction aims to create a minimum set of new features via some transformation according to a certain performance measure [122].

2.6.5 Evaluation Criteria

The evaluation methods can generally be divided in to, filters, wrappers and embedded methods [55, 184].

2.6.5.1 Filter Approaches

Filter methods evaluate feature subsets based on the intrinsic characteristics of the training data rather than the feedback of a learning algorithm [146] (shown in Figure 2.6). Filter methods can use different types of measures such as distance measure, information measure, dependence measure, and consistency measure [38]. Filter approaches have low computational cost and they are fast due to the avoidance of the inductive algorithms. However, evaluating the subsets in the search process is a challenging issue without relying on the inductive algorithms , since they are often not optimized to be used with any specific learning algorithm [103]. They usually have lower classification performance than other feature selection methods, e.g., wrapper, on a particular learning algorithm, as the prediction performance of the selected features or constructed features on a learning algorithm is not considered in the filter methods [146].

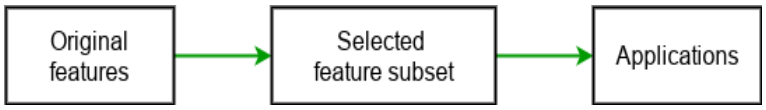


Figure 2.6: A filter method for feature selection.

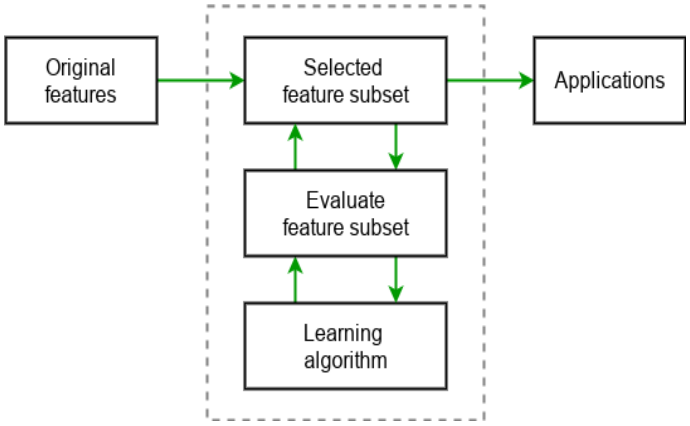


Figure 2.7: A wrapper method for feature selection.

2.6.5.2 Wrapper Approaches

Unlike filter-based methods, wrapper-based methods employ an inductive algorithm to evaluate the goodness of the selected features [159] (shown in Figure 2.7). As compared to filter-based methods, wrapper-based methods result more effective feature subsets, but the computational complexity is usually higher. This types of approaches is claimed to be less general than the filter approaches, since the selected feature or constructed feature subsets are mainly relied on the predetermined learning algorithm. For example, the best feature subset evaluated by one learning algorithm(e.g. decision tree (DT)), may not improve the prediction performance of another learning algorithm (e.g. support vector machine) [32].

2.6.5.3 Embedded Approaches

Unlike the wrapper-based methods, the embedded-based methods make an interaction between the learning algorithms and the feature selection or

construction approaches. Embedded methods determine the features and the learning algorithm (e.g. classifier) simultaneously during the training process [56]. For example, in DT, the tree is built by partitioning the data according to the importance of the features to the classification accuracy. Another example is GP which has an intrinsic capability of selecting or constructing features, and it improve the performance of the method [137]. The selected features by the embedded methods are more effective than those generated by the filter methods. Moreover, they have less computational cost than the wrapper methods [46].

2.7 Saliency Features

2.7.1 General Concept

In this section, we provide three well-known and basic categories of saliency features including [44]: photometric, geometrical, and spatial.

- **Photometric Features**

  Photometric features rely on luminance properties of the image, such as contrast and color.

  Contrast

  The most widely used assumption about the properties of objects and backgrounds is that the appearance contrasts between objects and their surrounding regions are high. This is called contrast feature and is used almost in all saliency methods [1,51,113,119,138].

  1. Local Contrast

     Local contrast has been regarded as the key factor in many studies for visual-attention evaluation. Local contrast refers to an image pattern or distinct structure which differs from its immediate neighbourhood by considering some properties such

as intensity, color, and texture. Local contrast can be shown by points, edges, or small patches. A good local feature is made by some properties including repeatability, distinctiveness, locality, quantity, accuracy, and efficiency. Local contrast which gives information in pixel-level can be more effective to show the contrast in boundaries [171].

2. Global Contrast

   Global contrast mainly concentrates on color uniqueness in terms of global statistics. Global contrast describes the image as a whole, and tries to generalize a whole object with a single vector [171].

Color

Based on daily experiences, some studies [149] have found that the human visual system grab more attentions from warm colors like yellow and red than cold colors like green and blue. Zhang et al. [191] proposed a method to model color features. A color feature concentrates on color of pixels and gives high saliency value to warm colors and low saliency value to cold colors.

- **Geometrical Features**

  Geometrical features such as compactness and objectness measures provide geometrical assumptions about the object with respect to the background [44].

  Compactness

  Salient objects are generally more compact in spatial domain, while background regions are usually spread over the entire image. In other words, salient regions have low spatial distribution compared to the background regions [69].

  Objectness

  Objects are stand-alone things with well-defined closed boundaries and centers (e.g. cars and cows), as opposed to amorphous background regions (e.g. sky, and grass) [14]. To define the objectness measure, [14] considers any object has at least one of the three distinctive characteristics: 1) a well-defined closed boundary in space, 2) a different appearance from its surroundings, 3) it is often unique within the image and stands out as salient. This measure gives the highest score to windows fitting an object tight, lower score to windows covering partly an object and partly the background, and the lowest score to windows containing only the background.

- **Spatial Features**

  Spatial features make assumptions about salient object and background locations [44].

  Backgroundness

  Backgroundness features mostly focus on background regions in turn lead to better foreground detection. Boundary feature and connectivity feature as background features have been used in some salient object detection methods. In the boundary feature, the image

boundary is mostly supposed to be background. In the connectivity feature, most of the image regions (e.g. superpixels) in the background are supposed to have a connectivity to each other. In some kind of images, some backgrounds such as grass and and sky are homogeneous by themselves, but some parts in this context can be hardly connected [99, 180, 188, 203].

Location

There are various works [3, 42, 51] in the literature to investigate location information. Location feature gives some information about the region that salient object is located.

Central

Based on Gestalt psychology [81], Tong et al. [163] assume that most salient objects are appeared near to the center of the image, so they involve center feature to their bottom-up saliency measure by computing spatial distance between superpixels and image center. Some of methods assign higher saliency value to the regions near the center by assuming salient objects are mostly located in the center. This may cause incorrect detection when salient object is far from the center. To alleviate this problem, Yang et al. [187] use a convex-hull to enclose salient regions, therefore the center of convex-hull is defined as the center of salient object.

2.7.2 Feature Extraction Levels

In this thesis, three levels of feature extraction including region level, superpixel level, and cluster level have been studied as follows.

- **Region Level**

  To compute region level segmentation, a graph-based image segmentation approach [43] is applied to generate multiple segmentations of a given image I using m groups of different parame-

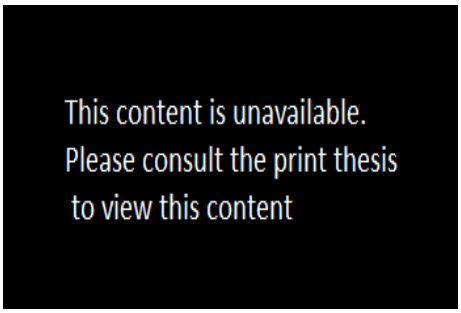


Figure 2.8: Some sample segmented images containing superpixels with size 64, 256, and 1024 pixels using SLIC [2].

ters. An image I is segmented to a set of m-level segmentations $L = \{L_1, L_2, \ldots, L_m\}$, where each segmentation level L_m is a decomposition of the image I.

- **Superpixel Level**

  Algorithms such as *simple linear iterative clustering* (SLIC) [2] segment an image by performing a local clustering of pixels in the 5-D space defined by the $L^*a^*b^*$ values of the CIE$L^*a^*b^*$ color space and the x, y pixel coordinates. The SLIC algorithm computes superpixels by clustering pixels based on their color similarity and proximity in the image. Figure 2.8 shows some examples of segmented images using SLIC with different segmentation size including 64, 256, and 1024.

- **Cluster Level**

  The cluster level segmentation [45] is inspired by the global contrast methods [36, 39]. In these methods, the feature channels of pixels are quantized into the histogram format to compute the spatial contrast dissimilarity, and evaluate the saliency of the pixel with respect to the other pixels in the entire image. However, the estimated feature distributions based on histogram are discontinuities at the bin edges. Hence, [45] utilized clustering to avoid the discontinuities at the bin edges of histograms, and result a highly cohesive global

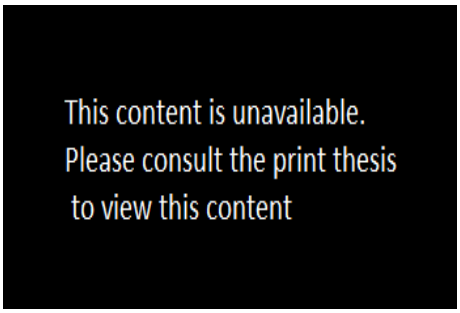


Figure 2.9: A visual example for two-layer clustering including intra-image (single image) and inter-image (multiple images) layers [45].

constraint. In [45] , K-means is used for clustering image pixels. Figure 2.9 shows an example of a clustering method considering inter-image (multiple images) and intra-image (single image) clustering.

2.7.3 The Employed Saliency Features in This Thesis

In the literature, a large number of primitive features have been designed and introduced for SOD. In this thesis, we study and explore some of well-known saliency features which are summarized in Table 2.2. In this work, we introduce all of the following features in the notation of their source papers.

In the literature, well-known and informative regional saliency features contains three groups of regional features including regional contrast, regional backgroundness, and regional property.

The regional contrast group is a 29-dimensional feature vector contains color and texture features. The color features are extracted from three different color spaces including RGB (red, green, and blue), HSV (hue, saturation, and value), and $L^*a^*b^*$.

The texture features are local binary patterns (LBP) [61] and responses of Leung-Malik (LM) filter bank [76].

The regional backgroundness group is a 29-dimensional feature vector

Table 2.2: Four groups of saliency features.

$\boldsymbol{fg}^1$ (regional contrast)	Notation	$\boldsymbol{fg}^2$ (regional backgroundness)	Notation
average RGB values	$f_1^1 \sim f_3^1$	average RGB values	$f_1^2 \sim f_3^2$
RGB histogram	f_4^1	RGB histogram	f_4^2
average HSV values	$f_5^1 \sim f_7^1$	average HSV values	$f_5^2 \sim f_7^2$
HSV histogram	f_8^1	HSV histogram	f_8^2
average $L^*a^*b^*$ values	$f_9^1 \sim f_{11}^1$	average $L^*a^*b^*$ values	$f_9^2 \sim f_{11}^2$
$L^*a^*b^*$ histogram	f_{12}^1	$L^*a^*b^*$ histogram	f_{12}^2
absolute response of LM filters	$f_{13}^1 \sim f_{27}^1$	absolute response of LM filters	$f_{13}^2 \sim f_{27}^2$
max response histogram of the LM filter	f_{28}^1	max response histogram of the LM filter	f_{28}^2
histogram of the LBP feature	f_{29}^1	histogram of the LBP feature	f_{29}^2
$\boldsymbol{fg}^3$ (regional property)	Notation	$\boldsymbol{fg}^4$ (hand-crafted)	Notation
average norm x coordinates	f_1^3	multi-scale contrast	f_1^4
average norm y values	f_2^3	color spatial distribution	f_2^4
10^{th} percentile of the norm x values	f_3^3	center-surround hist	f_3^4
10^{th} percentile of the norm y values	f_4^3	convex-Hull-based center	f_4^4
90^{th} percentile of the norm x values	f_5^3	cluster-based contrast	f_5^4
90^{th} percentile of the norm y values	f_6^3	cluster-based spatial	f_6^4
norm perimeter	f_7^3	background weighted contrast	f_7^4
aspect ratio of the bounding box	f_8^3	uniqueness	f_8^4
variances of the RGB values	$f_9^3 \sim f_{11}^3$	distribution	f_9^4
variances of the $L^*a^*b^*$ values	$f_{12}^3 \sim f_{14}^3$	SUSAN edge detector	f_{10}^4
variances of the HSV values	$f_{15}^3 \sim f_{17}^3$		
variances of the response of the LM filters	$f_{18}^3 \sim f_{32}^3$		
variances of the LBP	f_{33}^3		
norm area	f_{34}^3		
norm area of the neighbor regions	f_{35}^3		

extracted by computing the difference between each region and a pseudo-background region as a reference. Similar to the regional contrast group, the color and texture features are used to compute the regional backgroundness features.

The regional property group is 35-dimensional feature vector computed by considering the generic properties of a region such as appearance and geometric features.

2.7.3.1 Regional Contrast Features

Regions with high contrast to their surroundings are more likely to get more attention. Jiang et al. [76] presented a contrast feature vector representing the differences of feature vectors of regions. Each region $rg_i \in l_m$ is described by a feature vector which consists of color and texture features, represented as $\boldsymbol{v}^{rg_i}$. The regional contrast value rv_k^c derived from the k^{th} feature channel is computed by considering rg_i against all other regions,

$$rv_k^c(rg_i) = \sum_{j=1}^{N_m} \alpha_j rw_{ij} D_k(\boldsymbol{v}^{rg_i}, \boldsymbol{v}^{rg_j}) \tag{2.3}$$

where N_m is the number of regions in l_m. α_j is involved to account for the irregular shapes of regions, defined as the normalized area of the region. $rw_{ij} = \exp^{-} \frac{||P_i^m - P_j^m||^2}{2\sigma_s^2}$ presents spatial weighting, P_i and P_j are the mean positions of rg_i and rg_j, respectively. σ_s is used to control the strength of the spatial weighting effect. $D_k(\boldsymbol{v}^{rg_i}, \boldsymbol{v}^{rg_j})$ denotes the difference between feature vectors $\boldsymbol{v}^{rg_i}$ and $\boldsymbol{v}^{rg_j}$ in the k^{th} channel.

The details of the regional contrast features are given in Table 2.2. In Table 2.2, $d(\boldsymbol{X}_1, \boldsymbol{X}_2) = (|x_{11} - x_{21}|, \ldots, |x_{1ne} - x_{2ne}|)$, where ne is the number of elements in the vectors $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$.

$$\chi^2(\boldsymbol{h}_1, \boldsymbol{h}_2) = \sum_{i=1}^{b} \frac{2(h_{1i} - h_{2i})^2}{h_{1i} + h_{2i}} \tag{2.4}$$

b is the number of histogram bins.

2.7.3.2 Regional Backgroundness Features

The backgroundness features are presented to determine the backgroundness degree (accordingly the saliency degree) of a region. Similar to the regional contrast features, the details of the regional backgroundness features are given in Table 2.2. The backgroundness feature vector for each region is computed using a pseudo-background region as a reference. The

pseudo-background region B is defined as the 15-pixel wide narrow border region of the image. The backgroundness value rv_k^b of the region rg_i on the k^{th} feature channel is denoted as

$$rv_k^b(rg_i) = D_k(\boldsymbol{v}^{rg_i}, \boldsymbol{v}^B) \tag{2.5}$$

2.7.3.3 Regional Property Features

The regional property feature vector is presented by considering the generic properties of a region, including appearance and geometric features. The appearance features aim to compute the distribution of colors and textures in a region to characterize the common properties of salient and background regions. For example, the background regions usually have homogeneous color distribution or similar texture pattern. The geometric features consider the size and position of a region which are helpful to define the spatial distribution of saliency and background regions. For example, unlike the background which usually scatters over the entire image, the salient objects are usually appeared close to the center of the image. The details of regional property features are given in Table 2.2.

2.7.3.4 Hand-crafted Features

Here, 10 widely employed hand-crafted saliency features are reviewed. Table 2.2 shows a summary of these features.

Liu et al. [113] proposed three features including multi-scale contrast, center-surround histogram, and color spatial distribution to study a salient object at three different scales, locally, regionally, and globally.

1) **Multi-scale contrast:** the multi-scale contrast feature as a local feature aims to highlight the boundaries with high contrast by assigning low values to the homogeneous regions inside the salient object [113]. Here, the multi-scale contrast feature $f_c(p, I)$ is defined as a

linear combination of contrasts in the Gaussian image pyramid [113]:

$$f_c(p, I) = \sum_{l=1}^{lp} \sum_{p' \in WN(p)} \|I^l(p) - I^l(p')\|^2 \tag{2.6}$$

where I^l is the l^{th} level of image I in the pyramid and lp is the number of pyramid levels. p and $p^{'}$ are two adjacent pixels. $WN(p)$ is a 9×9 window. The feature map $f_c(\cdot, I)$ is normalized to a fixed range [0,1].

2) **Center-surround histogram:** the center-surround feature as a regional feature attempts to find the location of the salient object in the image [113].

Suppose R is a rectangle which encloses the salient object and R_S is a surrounding contour with the same area as R. To compute how distinct the salient object in the rectangle is with respect to its surroundings, the distance between R and R_S can be measured using different features such as color, texture, and intensity. The χ^2 distance between histograms of RGB color is used [113]. Histograms have some benefits like being robust global description of appearance, insensitive to small changes in size, shape, and viewpoint. Moreover, [140] introduced an integral histogram to compute histogram of a rectangle with any location and size very quickly. Here, the intensity histogram and oriented gradient histogram are not employed, since the intensity histogram is redundant with the color histogram and the oriented gradient histogram is not a good measure because the texture distribution in a semantic object is usually not coherent. In [113], five templates are used with different aspect ratios 0.5, 0.75, 1.0, 1.5, 2.0 to handle varying aspect ratios of the object. The most distinct rectangle $R^*(p)$ centered at each pixel p is found by varying the size and aspect ratio:

$$R^*(p) = \arg\max_{R(p)} \chi^2(R(p), R_S(p)) \tag{2.7}$$

The size range of the rectangle $R(p)$ is set to [0.1, 0.7]$\times \min(W, H)$, where W and H are the image width and height, respectively. Then, the center-surround histogram feature $f_h(p, I)$ is defined as a sum of spatially weighted distances:

$$f_h(p, I) \propto \sum_{\{p'|p \in R^*(p')\}} w_{xx'} \chi^2(R^*(p'), R_S^*(p')) \tag{2.8}$$

where $R^*(p')$ denotes the rectangle centered at p' and containing the pixel p. The weight $w_{xx'} = \exp(-0.5\sigma_{p'}^{-2}\|p-p'\|^2)$ indicates a Gaussian falloff weight with variance $\sigma_{p'}^2$, which is set to one third of the size of $R^*(p')$. The feature map $f_h(\cdot, I)$ is also normalized to the range [0,1].

3) **Color spatial distribution:** the color spatial-distribution feature as a global feature is obtained by computing the spatial variance of each color in the image [113].

First, all colors in the image are represented by Gaussian mixture models (GMMs) $\{w_c, \mu_c, \sum_c\}_{c=1}^{C}$, where $\{w_c, \mu_c, \sum_c\}$ is the weight, the mean color and the covariance matrix of the c^{th} component. Each pixel is assigned to a "color component" with the probability:

$$P(c|I_p) = \frac{w_c \mathcal{N}(I_p|\mu_c, \sum_c)}{\sum_c w_c \mathcal{N}(I_p|\mu_c, \sum_c)} \tag{2.9}$$

Then, the horizontal variance $V_h(c)$ of the spatial position for each color component c is:

$$V_h(c) = \frac{1}{|X|_c} \sum_p p(c|I_p) \cdot |x_h - M_h(c)|^2 \tag{2.10}$$

$$M_h(c) = \frac{1}{|X|_c} \sum_p p(c|I_p) \cdot x_h \tag{2.11}$$

where x_h is x-coordinate of the pixel p, and $|X|_c = \sum_p p(c, I_p)$. The vertical variance $V_v(c)$ is similarly defined. The spatial variance of a component c is $V(c) = V_h(c) + V_v(c)$. $\{V(c)\}_c$ is normalized to the

range [0,1]. Finally, the color spatial-distribution feature $f_s(p, I)$ is defined as a weighted sum:

$$f_s(p, I) \propto \sum_c p(c|I_p) \cdot (1 - V(c)) \tag{2.12}$$

The feature map $f_s(\cdot, I)$ is also normalized to the range [0,1].

$$f_s(p, I) \propto \sum_c p(c|I_p) \cdot (1 - V(c)) \cdot (1 - D(c)) \tag{2.13}$$

where $D(c) = \sum_p p(c|I_p)d_p$ represents a weight which assigns less importance to colors nearby image boundaries and is also normalized to [0,1], similar to $V(c)$. d_p is the distance from pixel p to the image center.

Yang et al. [187] introduced a convex-Hull-based center feature to make the center-based feature (center feature) more accurate and more robust to the location of salient object.

4) **Convex-hull-based center:** the convex-hull feature enclosing interesting points to estimate the location of salient regions [187]. The centroid of the obtained convex-hull is used as the center of object [187]. Supposing (x_0, y_0) is the image center, the saliency of a superpixel sp is defined as:

$$f_{ce}(sp) = \exp(-\frac{\|x_{sp} - x_0\|^2}{2\sigma_x^2} - \frac{\|y_{sp} - y_0\|^2}{2\sigma_y^2}) \tag{2.14}$$

where x_{sp} and y_{sp} represent the mean horizontal and vertical positions of the superpixel sp. σ_x^2 and σ_y^2 respectively indicate the horizontal and vertical variances and they are set $\sigma_x^2 = \sigma_y^2 = 0.15$ with pixel coordinates normalizes to [0,1] based on our empirical tuning and according to [187]

Fu et al. [45] introduced cluster-based contrast and cluster-based spatial features to measure the cluster-level saliency.

5) **Cluster-based contrast:** the cluster-based contrast feature represents the visual feature uniqueness on the image. The main property of the cluster-based method is that features appear on the cluster level rather than the individual pixel-level [45].

The contrast feature $f_{cc}(k)$ of cluster C^k is defined using its feature contrast to all other clusters:

$$f_{cc}(k) = \sum_{i=1, i\neq k}^{K} (\frac{n^i}{N_t} \|\mu^k - \mu^i\|_2) \tag{2.15}$$

where a L_2 norm is used to compute the distance on the feature space, n^i is the number of pixels of cluster C^i, and N_t denotes the number of all pixels. μ^k is the prototype (cluster center) associated with the cluster C^k, and K denotes the number of clusters. This definition favours large clusters to play more influence [45].

6) **Cluster-based spatial:** the cluster-based spatial feature measures a global spatial distribution of the cluster [45]. The spatial feature $f_{cs}(k)$ of cluster C^k is defined as:

$$f_{cs}(k) = \frac{1}{n^k} \sum_{j=1}^{M} \sum_{i=1}^{N_j} [\mathcal{N}(\|z_i^j - o^j\|^2 | 0, \sigma^2) \cdot \delta[b(p_i^j) - C^k] \tag{2.16}$$

where Gaussian kernel $\mathcal{N}(\cdot)$ computes the Euclidean distance between pixel z_i^j and the image center o^j, $\delta(\cdot)$ denotes the Kronecker delta function, the variance σ^2 represents the normalized radius of images, and the normalization coefficient n^k is the pixel number of cluster C^k.

Zhu et al. [203] introduced a background measure, called boundary connectivity. Unlike some methods which assume the image boundary is background or an image region is background if it can easily be connected to the image boundary, the proposed measure in [203] describes an image region is background only when the region is heavily connected to the image boundary.

7) **Background weighted contrast:** the background weighted contrast feature [203] is the extension of region contrast [138]. The region contrast is computed as the summation of region's appearance distance to all other regions, weighted by their spatial distances.

In this case, a superpixel's contrast can be written as:

$$Ctr(sp) = \sum_{i=1}^{N_s} d_{app}(sp, sp_i) w_{spa}(sp, sp_i) \tag{2.17}$$

where $w_{spa}(sp, sp_i) = \exp(-\frac{d^2_{spa}(sp,sp_i)}{2\sigma^2_{spa}})$. $d_{spa}(sp, sp_i)$ is the distance between the centers of superpixel sp and sp_i, and $\sigma_{spa} = 0.25$ as in [138]. The background weighted contrast is obtained by introducing a background probability w_i^{bg} as a new weighting term. The probability w_i^{bg} is mapped from the boundary connectivity value of superpixel sp_i. When the boundary connectivity is large, it will be close to 1 and when it is small, it will be close to 0. The definition is:

$$w_i^{bg} = 1 - \exp(-\frac{BndCon^2(sp_i)}{2\sigma^2_{bndCon}}) \tag{2.18}$$

σ_{bndCon} is empirically set to 1 in [203]. The results are insensitive to this parameter when $\sigma_{bndCon} \in [0.5, 2.5]$. The boundary connectivity is computed as:

$$BndCon(sp) = \frac{Len_{bnd(sp)}}{\sqrt{Area(sp)}} \tag{2.19}$$

The length along the boundary is defined as:

$$Len_{bnd}(sp) = \sum_{i=1}^{N_s} S(sp, sp_i) \cdot \chi(sp_i \in Bnd) \tag{2.20}$$

where N_s is the number of superpixels. $\chi(\cdot)$ is 1 for superpixels on the image boundary and 0 otherwise. The spanning area of each superpixel sp is defined as:

$$Area(sp) = \sum_{i=1}^{N_s} S(sp, sp_i) = \sum_{i=1}^{N_s} \exp(-\frac{d^2_{geo}(sp, sp_i)}{2\sigma^2_{clr}}) \tag{2.21}$$

$Area(\cdot)$ computes an area of the region that sp belongs to. There is an undirected weighted graph which connects all adjacent superpixels (sp, sq) and assigns their weight $d_{app}(sp, sq)$ as the Euclidean distance between their average colors in the CIE$L^*a^*b^*$ color space. The geodesic distance between any two superpixels $d_{geo}(sp, sq)$ is defined as the accumulated edge weights along their shortest path on the graph:

$$d_{geo}(sp, sq) = \min_{sp_1=sp, sp_2, \ldots, sp_n=q} \sum_{i=1}^{N_s-1} d_{app}(sp_i, sp_{i+1}) \tag{2.22}$$

For convenience, $d_{geo}(sp, sp) = 0$. Using Ctr and w^{bg}, the background weighted contrast is defined as follow

$$wCtr(sp) = \sum_{i=1}^{N_s} d_{app}(sp, sp_i) w_{spa}(sp, sp_i) w_i^{bg} \tag{2.23}$$

According to $wCtr(\cdot)$, the object regions receive high w_i^{bg} from the background regions and their contrast is enhanced. On the contrary, the background regions receive small w_i^{bg} from the object regions and the contrast is attenuated. Therefore, the contrast difference between the object and background regions enlarges under this asymmetrical behavior.

Perazzi et al. [138] developed two contrast features, uniqueness and distribution on images which are segmented by an adaptation of SLIC superpixels [2].

8) **Uniqueness:** the uniqueness feature is described as the rarity of a segment sp_i with position $\boldsymbol{P}_{sp_i}$ and color $\boldsymbol{c}_{sp_i}$ in comparison to all other segments sp_j [138].

$$U_{sp_i} = \sum_{sp_j=1}^{N_s} \|\boldsymbol{c}_{sp_i} - \boldsymbol{c}_{sp_j}\|^2 \cdot \underbrace{w(\boldsymbol{P}_{sp_i}, \boldsymbol{P}_{sp_j})}_{w^{(p)}_{sp_i sp_j}} \tag{2.24}$$

By considering $w^{(p)}_{sp_i sp_j}$, global and local contrast estimation are combined with control over the influence radius of the uniqueness operator. Evaluating uniqueness is expensive and needs $O(N^2)$ operations, where N is the number of superpixels. For a Gaussian weight $w^{(p)}_{sp_i sp_j} = \frac{1}{Z_{sp_i}} \exp(-\frac{1}{2\sigma_p^2} \|\boldsymbol{P}_{sp_i} - \boldsymbol{P}_{sp_j}\|^2)$, uniqueness can be evaluated in linear time $O(N)$. σ_p controls the range of the uniqueness operator and Z_{sp_i} is the normalization factor ensuring $\sum_{sp_j=1}^{N_s} w^{(p)}_{sp_i sp_j} = 1$.

9) **Distribution:** the distribution feature for a segment is defined using the spatial variance of its color in the entire image [138]. This feature is computed as:

$$DB_{sp_i} = \sum_{sp_j=1}^{N_s} \|\boldsymbol{P}_{sp_j} - \mu_{sp_i}\|^2 \underbrace{w(\boldsymbol{c}_{sp_i}, \boldsymbol{c}_{sp_j})}_{w^{(c)}_{sp_i sp_j}} \tag{2.25}$$

where $w^{(c)}_{sp_i sp_j}$ denotes the similarity of color c_{sp_i} and color c_{sp_j} of segments sp_i and sp_j, respectively, $\boldsymbol{P}_{sp_j}$ is the position of segment sp_j, and $\mu_{sp_i} = \sum_{sp_j=1}^{N_s} w^{(c)}_{sp_i sp_j} \boldsymbol{P}_{sp_j}$ describes the weighted mean position of color c_{sp_i}. Here, another equation is defined to evaluate DB_i in linear runtime, since distributation has quadratic runtime complexity. Therefore, the color similarity is chosen to be Gaussian $w^{(c)}_{sp_i sp_j} = \frac{1}{z_{sp_i}} \exp(-\frac{1}{2\sigma_c^2} \|c_{sp_i} - c_{sp_j}\|^2)$. The permutohedral lattice [4] is used as a linear approximation to the Gaussian filter in the $L^*a^*b^*$ space. σ_c controls the color sensitivity of distribution. $\sigma_c = 20$ is used in all the experiment [138]. DB_{sp_i} is defined as:

$$DB_{sp_i} = \sum_{sp_j=1}^{N_s} \boldsymbol{P}_j^2 w^{(c)}_{sp_i sp_j} - \mu^2_{sp_i} \tag{2.26}$$

Smith et al. [153] introduced Smallest Univalue Segment Assimilating Nucleus (SUSAN) principle for edge detection. The proposed edge detection algorithm takes an image and uses a predetermined window centred at each pixel in the image, applying a locally acting

set of rules to give an edge response. This response is then processed to give as the output a set of edges.

10) **SUSAN edge:** the SUSAN technique as a edge detector is used to highlight boundaries and identify rapid color changes in the image [153].

2.8 Related Work

This section includes four parts: the first part provides a brief background on GP for image analysis, the second part discusses related works on EC-based SOD methods, the third part discuses non EC-based SOD methods, and the fourth part provides deep learning based SOD methods.

2.8.1 GP for Image Analysis

Among EC techniques, GP has the ability to solve various complex problems in many research areas such as feature extraction [11, 94], classification [70], and object detection [107, 192]. Lensen et al. [94] developed a GP approach to simultaneously select regions, extract histogram of oriented gradients (HOG) features and perform binary classification on a given image. Al-Sahaf et al. [9] showed that GP has the ability to automatically extract features, perform feature selection and image classification. Later on, Al-Sahaf et al. [10], used multi-tree GP representation to automatically evolve image descriptors. Unlike existing hand-crafted image descriptors, their proposed method [10] automatically extracts feature vectors using a few instances of each class.

Ain et al. [6] developed a GP-based method to do feature selection and feature construction for skin cancer image classification. The authors observed that the GP-based selected and constructed features helped the classification algorithms to produce effective solutions for the real-world

skin cancer detection problem. Fu et al. [48] used GP to construct low-level edge detectors by automatically searching pixels in natural images without adopting the window approach. The goal achieved by applying a shifting function instead of using a fixed window to evolve edge detectors. After selecting pixels by the GP edge detectors, linear and second-order filters are used to check the goodness of the extracted pixels. To reduce the number of the selected pixels, they employed merge operation in the GP edge detector. Fu et al. [49] also developed a Gaussian-based edge detection method using GP. The main goal of the edge detector model was to automatically set parameters of the Gaussian filters and investigate different combinations of them.

In the mentioned studies, GP made the proposed approaches free from any requirement for human intervention or domain-specific knowledge. GP is popular for being employed in feature manipulation tasks, specifically, GP has shown to be effective in feature construction tasks, due to its flexible representation and a population-based search [165].

Unlike the other image related research areas, GP has not been studied and investigated in salient object detection field, while the potential of GP for solving different complex problems has been proved.

2.8.2 Salient Object Detection using EC Methods

Naqvi et al. [127] introduced a new GA-based method to detect salient object. In [127], different types of images are autonomously grouped into k clusters based on their distances to the other images in a feature space. Then multiple GAs are employed to learn multiple parametric such as normalization, integration schemes, and feature weightings, to improve generalization of the system. Thus, different optimal parameters are obtained for different image types. In another work [124], they proposed a new bottom-up SOD method to predict human fixation using GA. The final saliency map is produced by a linear combination of three weighted

feature maps including color, intensity and orientation. GA is used to search the optimal values for some parameters including aspect ratio, standard deviation, wavelength of the Gabor filter, four local orientations, two phases and three weight values.

Iqbal et al. [69] investigated a learning classifier system to develop SOD method. To this end, they implemented a GA and two XCS-based classifier systems including linear combination functions (XCSCA) and code-fragment actions (XCSRCFA) to compute the saliency map. To address the drawbacks of learning only a single weight vector, LCS techniques can learn weight sets based on image types. Among these three approaches, the linear combination XCS based method is found to have the best performance and consistent results. Since the code-fragment based system tries to compute saliency map using only one feature, it can not be efficient or robust than the linear combination XCS based System which combines different features in different classifiers. The experiment results indicate that GA suffers from over-fitting, while XCS-based techniques avoid this problem using niche mechanism.

2.8.3 Salient Object Detection using non-EC Methods

Here, we provide related research works to the concepts which have been widely studied in the literature. These concepts are hand-crafted saliency features, different level of feature extraction, foreground and background, boundary connectivity, objectness, feature weighting with more detail as follows.

2.8.3.1 Hand-crafted Features

As a pioneer, Itti et al. [75] proposed a bottom-up saliency method using center-surround differences across multi-scale features. The earliest SOD work [1] developed a frequency-tuned method to estimate center-surround contrast using color and luminance features. Later on, Achanta

et al. [3] proposed a maximum symmetric surround method for saliency detection using symmetric surround filtering near image borders.

Cheng et al. [119] introduced an automatic global contrast based SOD method without any need to a prior knowledge or assumptions. Two contrast based methods, histogram-based contrast method (HC) and region-based contrast method (RC) have been proposed to extract saliency maps. Experiment results show that RC performs better than HC based on precision and recall rates, but RC is more expensive. The proposed algorithm in [119] has limited performance in images with multiple objects and specially in cluttered scenes.

Zhou et al. [201] developed a SOD method by integrating compactness and local contrast features. This study aims to address the shortcomings of combining global contrast and compactness features by considering local contrast. Here, local contrast can detect some salient regions that ignored by compactness. In this work, they employ a diffusion process based on manifold ranking to propagate saliency information. Due to using only color information, this study has some drawbacks in images with lack of color variation, especially when foreground and background objects have similar colors.

2.8.3.2 Different Level of Feature Extraction

To investigate feature extraction in cluster level, Fu et al. [45] proposed a single image saliency method and a Co-saliency detection method using three cluster-based features including contrast, spatial, and corresponding to compute saliency of each cluster. The first two features are used in weighting of both single image and multi-image saliency. The corresponding feature is adopted to find the common objects belonging to the multiple images. This method aims to consider features at the cluster level rather than pixel level. The final co-saliency is generated by multiplication function of the aforementioned three features. The authors observed that the cluster-based spatial feature helps to get better results in images

with complex background. The visual results of this work show that the proposed method obtains good results in images with texture, complex background, and large salient objects.

To investigate feature extraction in pixel level, Lin et al. [105] introduced a method to predict salient object by extracting multiple features like local contrast, global contrast, and background contrast in different levels like pixel-level, region-level and object-level. The final saliency map is computed by integrating background priors, refined global contrast, and local contrast. However this method may fail to highlight salient object completely when the foreground object is not homogeneous.

To investigate feature extraction in superpixel level, Fan et al. [42] considered the importance of three factors including isolation, distribution, and location prior measure in saliency detection. The superpixel isolation map is measured by finding the shortest path of each superpixel to a virtual background node. The superpixel distribution measure is employed to evaluate the distribution of a feature of the superpixel in the image. The superpixel location prior is used to emphasizes the superpixel close to the centroid of the saliency map as the foci of attention. Finally, the saliency map is obtained by integrating those three feature maps. This method fails when a salient object does not have much contrast (e.g. color) with the background. The proposed method does not perform well when the background is cluttered or complex, as it may wrongly choose some parts of background as salient object. Since this study considers color feature as contrast measure, it may fail in some psychological patterns (e.g. different shape) [42]. This problem may be addressed by using some additional methods such as sparse coding based method [42].

To investigate feature extraction in multi-layer, Filali et al. [44] developed an algorithm which employed multi-scale graph ranking and local-global saliency refinement to detect salient object. In order to get better description of objects, in construction of multi-layer saliency graphs, region information, boundary information and spatial information are combined.

The aim of applying multi-layer graph was to detect salient object boundaries more accurately, as different size of superpixels can affect the generated saliency maps. In the next stages of the algorithm, some refinement methods are considered such as spatial saliency refinement, considering feature relevance and boundary information, using random forests.

2.8.3.3 Foreground and Background

Yang et al. [188] proposed a bottom-up salient object detection method within manifold ranking framework. In this study, the final saliency map is computed by linearly combining background-based and foreground-based saliency maps obtained using color and texture features. The authors observed that their method uniformly highlights the salient regions and preserves finer object boundaries than the other methods [188]. However, [188] have difficulties in a challenging dataset such as DUT-OMRON [188].

Tong et al. [163] utilized the benefits of top-down and bottom-up methods to deal with complex training process of top-down methods and the limitation of feature utilization of bottom-up methods. In spite of previous works, they proposed a coding-based measure to combine multiple features without any requirement to use ground truth or human interaction. To achieve this goal, they consider global low-level features from the bottom-up method and a locality-constrained coding top-down method. In this study, bottom-up method tends to produce more integrate salient object and top-down method aims to obtain more reliable background [163]. This method fails when the background region and salient object have a similarity in color and texture features, since it only considers the above mentioned two features [163].

2.8.3.4 Boundary Connectivity

Perazzi et al. [138] developed a contrast-based filtering method for SOD. Although local contrast feature can successfully detect boundaries of the salient object, it does not give much information about the object interior. This problem which is called "object attenuation" can be found in all local and some global features. Although it is alleviated in global features, these features still have some limitations in highlighting salient objects completely [180].

Later, Wei et al. [180] proposed two background features, boundary and connectivity to address the problem of "object attenuation" by studying background regions. [180] suggested removing background clutters would help to have better foreground detection.

Zhu et al. [203] introduced a new boundary connectivity feature and a principled optimization framework. The boundary connectivity feature measures the portion of the connectivity between a region and image boundary, so a large connectivity belongs to the background regions and vie versa. Three low-level features including background, foreground and smoothness are integrated to define a cost function for saliency optimization. This method can perform well in images with no salient object. They observed that their method is not sensitive to image appearance variations. The results show that boundary connectivity performs better than GS (geodesic saliency) and the proposed background weighted contrast improves the background contrast as well.

2.8.3.5 Objectness

Although some of the previous studies [76, 119, 138] which utilize the bottom-up methods show good results in detecting salient objects, they may fail in the images with complex structures. Some recent methods use statistical features [76, 149, 180, 203, 205] to accurately distinguish the boundaries of the salient object regions and suppress background regions,

whereas these methods fail to identify salient objects when there is low contrast between salient and non-salient regions, heterogeneous objects and cluttered background. The salient object detection methods which consider objectness measures by randomly sampling methods require to know the object size to perform more efficiently. By considering the mentioned problems, Huo et al. [68] developed a method to perform object-level saliency detection by fusing two types of features, low-level global features and high-level objectness features. Object candidates are extracted by quantizing color attributes. In the high-level stage, color component straddling and candidate compactness are investigated to obtain candidate objectness. In the low-level stage, color focusness and color spatial distribution are considered to find the global saliency map. In [68], authors observed that combining high-level objectness with global low-level global cues makes their method more suitable for processing images with complex background [68].

Srivatsa et al. [157] proposed a new saliency detection method by employing objectness proposals and a foreground connectivity measure. They adapt the magnitude of norm of image gradient (NG) which is utilized in binarized normed gradients (BING) to obtain objectness score of each window. In the next step, they proposed "*foreground connectivity*" as a saliency measure to evaluate the connectivity of a superpixel to the estimated foreground. In order to combine foreground weights and background weights, an optimization framework [203] is employed.

2.8.3.6 Feature Weighting

Gopalakrishnan et al. developed a graph based method which attempts to assign equal weights to all features [53]. Since this approach does not consider the aforementioned relevancy, it suffers from noise caused by irrelevant features in the graph structure [54].

Liu et al. [113] developed a supervised SOD method based on the combination of local, regional and global features. The local feature is iden-

tified by considering the contrast information of a pixel in a local neighbourhood at multiple scales. In order to obtain regional features, a center-surround histogram is computed. The global feature is made up by a color spatial-distribution map which is represented by Gaussian mixture models (GMM). In [113], a linear weight vector is obtained through condition random field (CRF) learning method using maximum likelihood estimation (MLE) criterion. Some ideas like non-rectangular shapes for salient objects, non-linear combination of features, and more sophisticated visual features may improve the performance [113].

2.8.4 Deep Learning based SOD Methods

In recent years, due to the development of deep learning networks, deep learning based SOD method have made good progress. Compared with traditional methods that use hand-crafted features, Convolutional Neural Network (CNN) based methods that adaptively extract high-level semantic information from raw images have shown good results in predicting saliency maps [195]. Wang et al. [177] reviewed deep SOD algorithms from different perspectives including network architecture, level of supervision, learning paradigm and object level detection. In [177], the authors also summarized the existing SOD datasets, metrics and provided a comprehensive comparison and analysis for deep SOD methods.

Zhao et al. [200] introduced a multi-context deep learning framework for SOD. They used global context and local context and then integrated into a unified multi-context deep learning framework. The proposed method in [200] showed that their method has the ability to coherently highlight the salient object regions, and it has a better prediction especially in complex scene with complex background regions.

Lee et al. [93] considered both hand-crafted features and high-level features extracted from CNNs. To combine the features together, they designed a unified fully connected neural network to compute saliency

maps. In [93], the authors show that their method has a short training time and testing time compared to other methods, this is the result of sharing of their high-level features which only required to be computed once for a whole image. The qualitative results show that [93] obtains good performance on images with low-contrast salient objects and complicated backgrounds, and also works well on other difficult scenes.

Recently, Hou et al. [64] developed a CNN method which combines both low-level and high-level features from different scales. In [64], the authors developed a series of short connections from deeper side outputs to shallower ones for two reasons, 1) deeper side outputs encodes high-level knowledge and can locate salient objects; 2) shallower side outputs capture rich spatial information. Their proposed method could successfully improve the current best maximum F-measure by 1 point on the ECSSD and SOD datasets. It achieves achieves a more than 1-point decrease in terms of mean absolute error on the MSRA-B and PASCAL datasets. They categorized their failure cases in three groups: 1) it may fail completely segmenting out the salient objects and leaving a small part of the salient object missed and this is one of the common defect in CNN-based SOD methods, 2) it may not extract the main body of the salient object or it may highlight non-salient regions, 3) it may fail in images with transparent objects.

Liu et al. [111] proposed a pixel-wise contextual attention network (PiCANet) to learn both global and local context. They formulate PiCANet in two forms, global and local, to incorporate contexts with different scopes. In [111], the quantitative results show that when PiCANets are gradually employed to incorporate global and multi-scale local contexts selectively, the performance of their proposed method can be progressively boosted. The qualitative results show that the global PiCANet helps to better discriminate the foreground object from backgrounds, whereas the local PiCANet improves the feature map to be more homogeneous, which makes the whole foreground object highlighted more uniformly [111].

CNN-based methods have the ability to learn multi-level features from the given images during the training process. CNN-based features are more semantically informative compared to the hand-crafted features [204]. CNN-based SOD methods [110, 175] have reported good performances, due to employing high-level features in saliency detection. However, the CNN-based methods still have some important problems. Firstly, downsampling operations such as pooling and convolution decrease the resolution of the image, hence, the details of the image such as corners, boundaries are degraded [204]. Secondly, many CNN-based SOD methods [176, 190, 195] have introduced overloaded layers to combine multi-level features which may cause features cluttered, consequently, it results in inaccurate saliency detection [204]. Thirdly, the lack of structural supervision of CNN-based methods makes SOD an extremely challenging problem in complex images [194]. Fourthly, the top CNN methods require non-trivial steps such as generating object proposals, applying post-processing, enforcing smoothness through the use of superpixels or defining complex network architectures [114]. Finally, CNN-based SOD methods mainly focus on either changing the training data, or stacking more network layers. Although it helps to achieve a better performance, the impact of the semantic information has not been adequately studied [132].

2.9 Standard SOD Datasets

In the literature, researchers have introduced a large number of SOD datasets which differ in number of images, number of salient objects in each image, image resolution and annotation form (bounding box or accurate region mask). Hence, it is important to evaluate the proposed SOD methods based on different SOD datasets to come up with a fair comparison. A good SOD method is expected to keep consistency and have reasonable performance over the different datasets. To compare SOD methods, there exist some benchmark datasets including: MSRA10K [119],

THUR15K [35], ECSSD [186], Judd-A [25], DUT-OMRON [188], SED [15], and PASCAL [101].

In this thesis, from the existing datasets, we choose four benchmark SOD datasets based on the following reasons to evaluate the performance of the proposed methods. These datasets are selected based on the following criterion: 1) being widely-used, 2) containing both large and small number of images, and 3) having different biases (e.g. number of salient objects, image clutter, center-bias). Since we mostly focused on developing supervised algorithms in this thesis, we chose the datasets containing annotated saliency ground truth instead of picking any unlabeled image dataset from the computer vision domain. To make a fair comparison with the existing SOD methods, we chose the commonly used benchmark SOD datasets. The majority of the SOD datasets do not have a massive number of images compared to the other image datasets (e.g. CIFAR-10 with 80 million), since the manually annotating process or identifying the ground truth in SOD images is an expensive and time-consuming task [170]. The process will cost time, money and human effort due to asking people to label salient object for a large number of images. Moreover, the process has the potential to have the problem of labelling inconsistency (subjectivity) due to labelling by different people based on their understanding of salient object for a particular image (e.g. whether the reflection of a salient object on water surface should be considered salient or not). In this thesis, we attempted to investigate the effectiveness of the proposed algorithms on both small (e.g SED1 dataset) and large (e.g., ASD and ECSSD) datasets. In SOD, a dataset with thousand number of images such as ECSSD considered as a large dataset. The performance of some SOD methods are affected by changing the number of images used for training/building a model, e.g., deep CNN-based methods. Hence, considering both small and large SOD datasets can be helpful to make unbiased conclusions.

The SED1 (*single-object database*) dataset[1] is a subset of the SED dataset

[1]Download from: https://drive.google.com/file/d/

which includes SED1 and SED2. The SED1 dataset has 100 images containing only one salient object in each image as depicted in Figure 2.10. Pixel-wise ground truth annotations for the salient objects in SED1 are provided. The SED1 dataset contains images with different sizes such as 300 $\times$ 400, 300 $\times$ 200, and 300 $\times$ 170.

The ASD dataset[2] [1] is a subset of the MSRA10K dataset. The MSRA10K dataset provides bounding boxes manually drawn around salient regions by nine users. However, a bounding box-based ground truth is far from being accurate for SOD tasks. Thus, Liu et al. [113] created an accurate object-contour based ground truth dataset of 1000 images. Each image is manually segmented into foreground and background. Most images have only one salient object and strong contrast between objects and backgrounds. Figure 2.11 shows some example images of ASD dataset. The ASD dataset comprises of 1000, 300 $\times$ 400 and 400 $\times$ 300 pixel images.

The ECSSD dataset[3] contains 1000 semantically meaningful but structurally complex images as shown in Figure 2.12. The ECSSD dataset is recently collected to overcome the weakness of existing saliency datasets such as MSRA, in which background structures are simply and smooth. Ground truth masks are provided by 5 subjects. The ECSSD dataset contain images with different sizes, e.g., 267 $\times$ 400, 400 $\times$ 300, 267 $\times$ 400.

The PASCAL dataset [4] contains 850 images. This dataset contains images with multiple objects and cluttered background. Figure 2.13 shows some example images. This dataset has images with different sizes such as 500 $\times$ 319, 461 $\times$ 307, and 500 $\times$ 375.

Early SOD datasets have some limitations such as the number of im-

---

0BxNhBO0S5JCRbFVmM0gwWUNVUWM/view

[2]Download from: https://mmcheng.net/msra10k/

[3]Download from: https://drive.google.com/file/d/0BxNhBO0S5JCRbGVCeFFVSXpBWkU/view

[4]Download from: https://drive.google.com/file/d/0BxNhBO0S5JCRREJGOTI2N3JxWWM/view

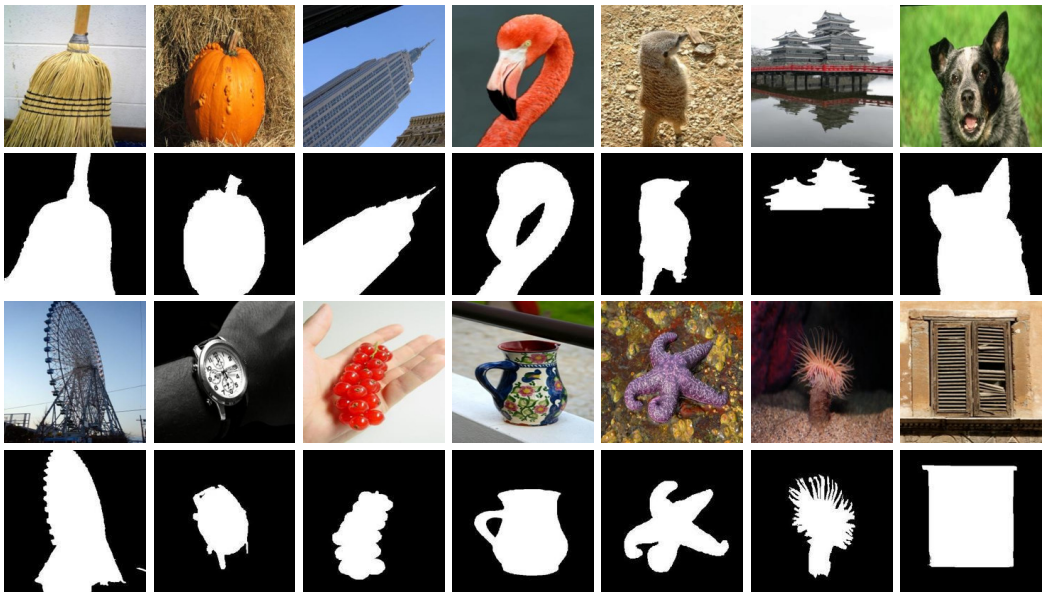


Figure 2.10: Samples of images and their corresponding ground truth from the SED1 dataset.

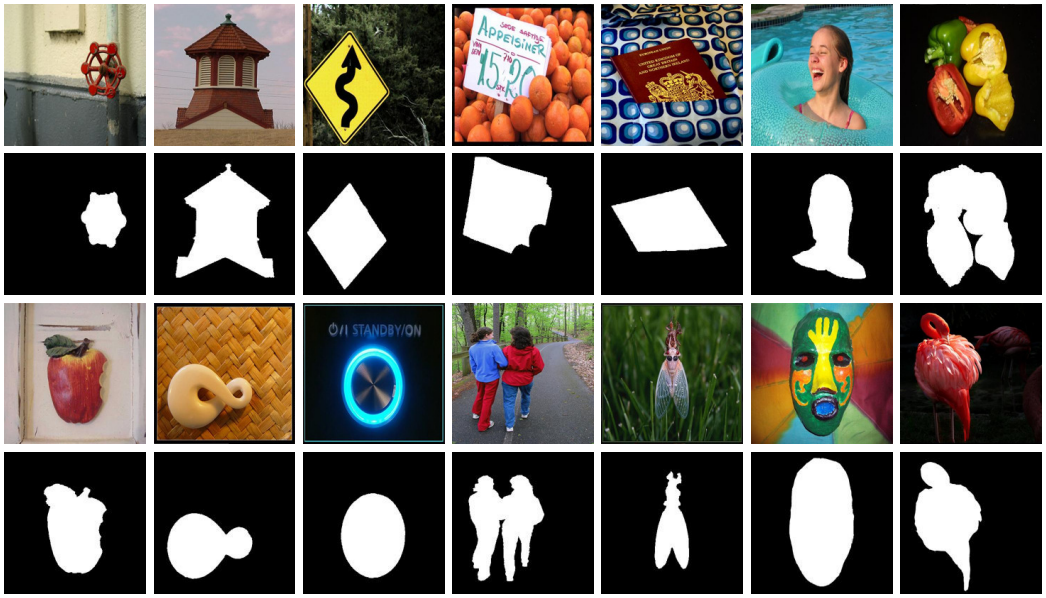


Figure 2.11: Samples of images and their corresponding ground truth from the ASD dataset.

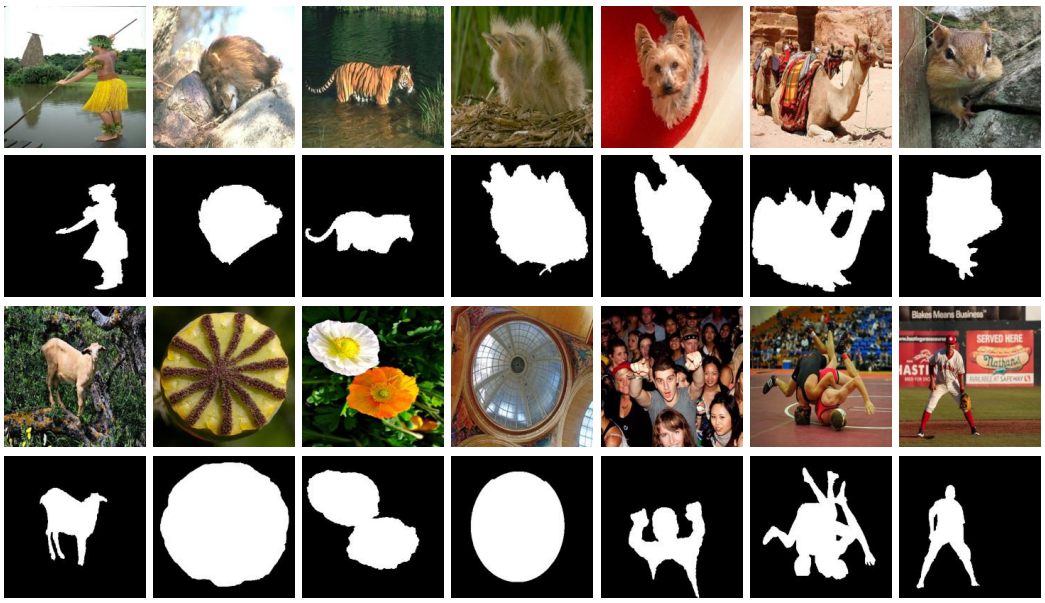


Figure 2.12: Samples of images and their corresponding ground truth from the ECSSD dataset.

Figure 2.13: Samples of images and their corresponding ground truth from the PASCAL dataset.

ages and their coarse annotation of salient objects. For example, MSRA-A [113] and MSRA-B [113] datasets include salient objects which are roughly annotated in the form of bounding boxes. The ASD, SED1 and MSRA10K datasets mostly contain only one salient object in each image, while the SED2 dataset contains two salient objects with small number of images. Recently, researchers attempted to collect datasets containing images with multiple objects and also considering images with complex and cluttered background to improve the quality of datasets. For example, DUT-OMRON [188], ECSSD [186], Judd-A [25], and PASCAL have been introduced to the domain. These datasets have been improved in terms of annotation quality and the number of images, compared to the previous datasets.

2.10 Chapter Summary

This chapter reviewed the main concepts of computer vision, saliency, saliency features, evolutionary computation, machine learning, and feature manipulation techniques. In addition, the related works of using GP for images, EC-based and non-EC based SOD methods, deep learning based SOD methods have been discussed. In this chapter, the limitations of the existing works have been highlighted, which can be summarised as follows.

- In SOD, GA as an EC technique has been used to search for the optimal values for different parameters such as feature weighting, normalization, integration scheme. However, other EC techniques such as PSO can be more suitable for the mentioned tasks due to having favourable characteristics such as being easy to implement, computationally more efficient, and performing effectively.
- The majority of the low-level features have been designed to partially tackle the SOD problem. In other words, the proposed features

are often not informative enough and they can not be expected to detect the whole salient object or suppress the background adequately in challenging images.

- The vast majority of SOD methods have manually selected or designed informative features and manually designed a feature combination framework to produce the final saliency map. However, manually performing the mentioned tasks has some difficulties such as being expensive, time-consuming, and error prone.

- Designing high-level saliency features is a challenging task, and will be even more challenging to design them based on assumptions or domain knowledge. Recently, CNN-based high-level (semantic) features have been introduced to the domain. CNN-based features often capture high-level concepts of salient objects, and they are required to be combined with some low-level fine details (e.g. edges and corners).

- The majority of the existing SOD methods have not been tested in terms of the generalizability over different types of images. The complexity and variety of saliency images make it difficult to provide generalization ability in the proposed SOD methods.

- The existing SOD datasets do not contain enough samples of complicated and challenging images. Thus, most of the SOD methods face some limitations in terms of training the proposed method with different samples during the learning process.

- The majority of the existing SOD methods are focused on designing new features and paid less attention to the generalizability of the new features over different image types.

Chapter 3

Particle Swarm Optimization for Weighting Features

3.1 Introduction

In the majority of the existing SOD methods, the final saliency map is produced by combining different selected primitive features. Considering the relative contribution of each feature is important in this combination. For example, when there is a high contrast between a salient object and background, the color contrast features can be given higher importance than the texture features. To reflect the importance of each feature in the combination process, a suitable weight is assigned to each feature.

In the literature, the feature weighting task has been done in different ways. Some studies [113, 116, 186] used conditional random field (CRF) under the maximum likelihood estimation (MLE) criterion to weight and combine multiple saliency features. MLE is a well-known parameter estimation technique and it provides a consistent and asymptotically efficient approach for parameter estimation [15]. However, MLE can be heavily biased for small samples and is highly sensitive to the choice of starting values. MLE is a gradient based approach where the function has an analytical form. Furthermore, the algorithm is not guaranteed to converge

(find optimal solution) and is usually non-trivial for the numerical estimation.

There are several works [45,76,119] that manually designed optimization frameworks for weighting features. Although the designed frameworks can be more efficient than CRF-based methods, they are restricted to the type of features and their combination scheme.

In this chapter, a PSO-based feature weighting method is developed to evolve a weight vector for different saliency features. PSO can be a good method due to the following characteristics which make it suitable for solving optimization problems [22, 145]: 1) PSO provides an appropriate representation for feature weighting, where each particle is a complete solution (i.e. potential weight vector) and each entry of a particle's position is used to represent a weight for a feature, 2) Compared to the other EC methods (e.g., GA, GP), PSO is easier to implement, it converges quickly and has few parameters to tune [40]. PSO can perform as effective as GA, but is computationally more efficient than GA [59], 3) a derivative-free method, and 4) PSO has shown promise in feature weighting in machine learning tasks [136,150]. The problems of MLE can be overcome by using PSO. However, the usage of PSO is still relatively new in SOD.

3.1.1 Chapter Goals

This chapter aims to develop a PSO-based method to generate a suitable weight vector for combining features in SOD. Specifically, this chapter aims to fulfil the following objectives:

- Develop a supervised PSO-based method and employ an appropriate fitness function that can guide PSO to find a suitable weight vector for constructing saliency from constituent features;
- Evaluate and compare the performance of the proposed method and other SOD methods on the benchmark datasets;

- Investigate whether the generated weight vector by the proposed method is capable of showing the importance of different features in a feature set; and
- Investigate whether the combination of the weighted saliency features can perform better than the non-weighted features.

3.1.2 Chapter Organization

The remainder of the chapter is organized as follows. Section 3.2 details the proposed method. Section 3.3 provides the experiment design. Section 3.4 presents and discusses the results. Section 3.5 presents the summary of this chapter.

3.2 PSO-based Feature Weighting

In this section, the structure of the proposed PSO-based saliency feature weighting method for SOD (named wPSOSOD), including the encoding scheme, the employed saliency features, and the fitness function, are explained.

3.2.1 The Overall Algorithm

An overview of the PSO-based algorithm is shown in Figure 3.1. First, saliency features are extracted from the training images, and then normalized (described in Section 3.2.4). The PSO algorithm starts with a predefined number of particles (weight vectors) and it takes the saliency features and ground truth of the training images as inputs. The training stage returns the evolved particle or the weight vector which is a median result as the final solution. In this experiment, the median solution is selected as the representative solution. In [127], they chose the mean of solutions as the representative solution, but mean is highly affected by outliers. We

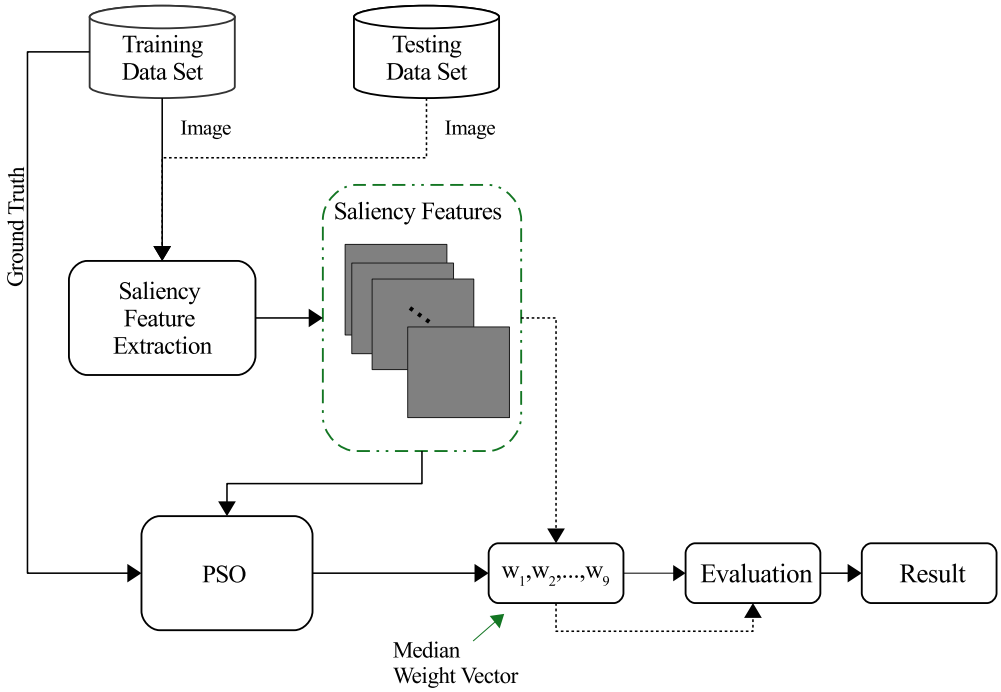


Figure 3.1: Overview of the proposed method.

choose the median one, since the median does not have the mentioned problem and it can give us the "*middle*" solution of the 31 solutions. The test stage takes the median evolved weight vector and used it on the extracted saliency features of the test images.

3.2.2 Encoding Scheme

Each particle is known as a weight vector and each cell/element in the particle denotes the corresponding weight for a feature. The weight values are continuous numbers in the range [0,1]. As the number of the employed features is 9, the dimension of the weight vectors is set to $D = 9$. In addition to the value and the velocity, the weight vector (particle) has a fitness value (described in Section 3.6) to evaluate the goodness of the weight vector.

3.2.3 Employed Features

To consider different levels of saliency information, features extracted from pixel, superpixel, and cluster levels are chosen in this thesis. The selected features are popular and robust features in the SOD domain. Moreover, different aspects including local, global, and heuristic are also considered for selecting those features. For each image in the dataset, the following nine saliency features are extracted (see Table 2.2 in Section 2.7.3.1 on page 52). Figure 3.2 shows some sample images, their ground truth, and the nine saliency feature maps. Here, we provide definition of the nine features as follows.

1. **Multi-scale contrast f_1:** a contrast feature has the ability to simulate the human visual receptive fields, thus it is one of the popular local features for attention detection [75,108]. Without knowing the size of the salient object, a contrast feature is usually computed at multiple scales [113]. Multi-scale contrast as a local feature aims to highlight the boundaries having high contrast with background [113].

2. **Center-surround histogram f_2:** center-surround as a regional feature attempts to find the location of the salient object in the image. To compute this feature [113], it is assumed that the salient object is enclosed by a rectangle R. A surrounding contour R_S is constructed with the same area of R. The idea is to measure how distinct the salient object is in the rectangle with respect to its surroundings. Thus, the distance between R and R_S is computed using various features such as color, intensity, and texture [113].

3. **Color spatial distribution f_3:** color spatial distribution as a global feature is obtained by computing the spatial variance of each color in the image. The idea is that a specific color with high spatial variance (more distributed) over the image has a high probability of being non-salient object and vice versa. The spatial distribution of a spe-

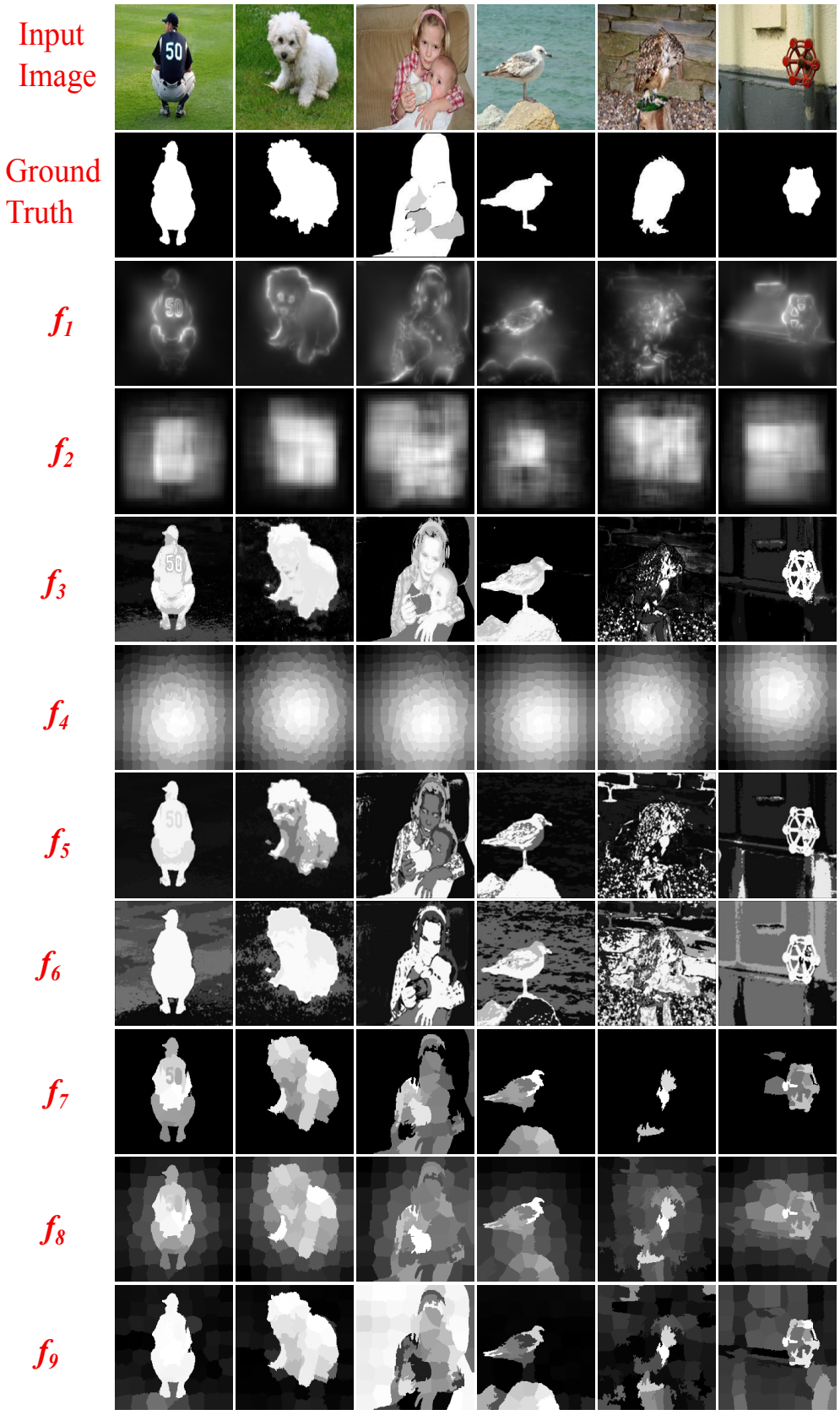


Figure 3.2: Some samples for the nine saliency features.

cific color is simply computed by considering the spatial variance of the color [113].

4. **Convex-hull-based center f_4:** previous studies [162, 163] employed center prior to assign higher saliency to the regions near the image center. However, this principle becomes invalid when the objects are placed far from the image center. Here, to avoid the mentioned problem, a convex hull enclosing interesting points is computed to estimate the location of salient region [187]. The centroid of the obtained convex-hull is used as the center of the foreground object to get the convex-hull-based center feature.

5. **Cluster-based contrast f_5:** contrast feature represents the visual feature uniqueness on the image. The main property of the cluster-based methods is that features appear on the cluster-level rather than the individual pixel-level [45].

6. **Cluster-based spatial f_6:** human visual system is more attracted by the regions near the image center than the other regions [39]. The attention gain will decrease when the distance between the object and the image center increases. This scenario is called as "*central bias rule*" in SOD [160]. This concept is extended to the cluster-based method, which measures a global spatial distribution of the cluster [45].

7. **Background weighted contrast f_7:** background weighted contrast [203] is the extend of region contrast [119, 138]. The region contrast is computed as the summation of region's appearance distance to all other regions, weighted by their spatial distances.

8. **Uniqueness f_8:** uniqueness is described as the rarity of a superpixel with a position and a specific color compared to all other superpixels in the image [138].

9. **Distribution $\boldsymbol{f}_9$:** the distribution measure for a superpixel is defined using the spatial variance of the superpixel's color on the entire image. Having low variance indicates a spatially compact object which is considered more salient than spatially widely distributed one [138].

3.2.4 Normalization

Since the employed features have different ranges of values, they are normalized in the range [0,1]. Here, each feature is presented as a 2-dimensional feature map. The normalized feature map $\boldsymbol{f}_d^{'}$ is computed as

$$\boldsymbol{f}_d^{'} = \frac{\boldsymbol{f}_d - min(\boldsymbol{f}_d)}{max(\boldsymbol{f}_d) - min(\boldsymbol{f}_d)} \tag{3.1}$$

where $min(\boldsymbol{f}_d)$ and $max(\boldsymbol{f}_d)$ present the minimum and maximum values of the d^{th} feature map.

3.2.5 Fitness Function

Algorithm 3 shows the process of computing fitness value for a specific weight vector (solution) generated by PSO. In this algorithm, first, saliency map for each image is computed and then normalized. Edge information of the normalized saliency map is obtained. Then, an adaptive threshold is used to compute a binarized mask of each saliency map. After that, *error* between ground truth and the binarized mask is computed. Finally, the average error over all the training images is returned as a fitness value of each particle.

Let $\boldsymbol{w} = \{w_1, w_2, ..., w_9\}$ be the weight vector, i.e., a particle, and D be the number of features ($D = 9$). The saliency value $\boldsymbol{S}$ of each pixel p is computed as

$$\boldsymbol{S}(p, \boldsymbol{w}) = \sum_{i=1}^{D} w_i \times \boldsymbol{f}_i^{'} \tag{3.2}$$

Algorithm 3 Fitness Function

procedure WPSOSOD ($P_{position}(\{w_1, ..., w_9\})$, features $\{\boldsymbol{f}_1, \boldsymbol{f}_2, ..., \boldsymbol{f}_9\}$, ground truth ($GT$) set of the training images, the number of training images ($imgn$))
  for $ii \leftarrow 1$ to $imgn$ **do**
    Compute saliency map $\boldsymbol{S}(p, \boldsymbol{w})$ using Equation (3.2)
    Normalize saliency map
    Compute edge information $\boldsymbol{E}(p, \boldsymbol{w})$
    $\boldsymbol{E}(p, \boldsymbol{w}) \leftarrow edge(\boldsymbol{S}_{nr}(p, \boldsymbol{w})), Canny)$
    Compute threshold $\tau(\boldsymbol{w})$ using Equation (3.4)
    if $\boldsymbol{S}_{nr}(p, \boldsymbol{w}) \geq \tau(\boldsymbol{w})$ **then**
      $\boldsymbol{M}(p, \boldsymbol{w}) \leftarrow 1$
    else
      $\boldsymbol{M}(p, \boldsymbol{w}) \leftarrow 0$
    end if
    Compute $Error(\boldsymbol{M}, GT)$ using Equation (3.7)
  end for
  Compute $Fitness_{pso}$ using Equation (3.6)
  return $Fitness_{pso}$

The produced saliency map $\boldsymbol{S}$ is normalized between [0,1].

3.2.5.1 Threshold

After normalizing the final saliency map, we need to use a threshold to classify each pixel into one out of the two groups, i.e., background pixel or foreground pixel. One of the popular ways to classify pixels is to use a fixed threshold. However, a fixed threshold may not be appropriate for all saliency maps. Therefore, based on [151], we use an adaptive threshold that is dependent on the saliency map. This adaptive threshold takes two steps. In the first step, to generate the object's silhouette, a Canny edge

operator is applied to the normalized saliency map $\boldsymbol{S}_{nr}$. For each pixel p, the edge information $\boldsymbol{E}$ is computed as

$$\boldsymbol{E}(p, \boldsymbol{w}) = edge(\boldsymbol{S}_{nr}(p, \boldsymbol{w})) = \begin{cases} 1 & p \text{ is an edge pixel} \\ 0, & \text{otherwise} \end{cases} \tag{3.3}$$

In the second step, a threshold τ is computed by averaging the saliency values which present at the object's silhouette.

$$\tau(\boldsymbol{w}) = \frac{\sum_{p \in \boldsymbol{P}} \boldsymbol{E}(p, \boldsymbol{w}).\boldsymbol{S}_{nr}(p, \boldsymbol{w})}{\sum_{p \in \boldsymbol{P}} \boldsymbol{E}(p, \boldsymbol{w})} \tag{3.4}$$

By considering the obtained threshold for each image, the final binarized mask $\boldsymbol{M}$ is generated

$$\boldsymbol{M}(p, \boldsymbol{w}) = \begin{cases} 1 & \boldsymbol{S}_{nr}(p, \boldsymbol{w}) \geq \tau(\boldsymbol{w}) \\ 0, & \text{otherwise} \end{cases} \tag{3.5}$$

where foreground pixels and background pixels can be denoted by giving the value of 1 and 0 to $\boldsymbol{M}$, respectively.

3.2.5.2 Fitness

After finding the binarized mask, we compute error between the mask $\boldsymbol{M}$ and the corresponding ground truth $\boldsymbol{G}$ of the image i using Equation (3.7). To compute the fitness value of each weight vector (particle), we take the average of error values over all training images which is computed as

$$Fitness_{pso} = \frac{1}{n} \sum_{i=1}^{n} Error(\boldsymbol{G}_i, \boldsymbol{M}_i) \tag{3.6}$$

$$Error(\boldsymbol{G}_i, \boldsymbol{M}_i) = \frac{FP + FN}{TP + TN + FP + FN} \tag{3.7}$$

where n is the number of images in the training set. TP (true positive) is the number of foreground pixels that are correctly detected as foreground,

TN (true negative) is the number of background pixels that are correctly detected as background, FP (false positive) is the number of background pixels that are incorrectly detected as foreground, and FN (false negative) is the number of foreground pixels that are incorrectly detected as background.

3.3 Experiment Design

3.3.1 Datasets

We evaluate the performance of the proposed method over three widely used benchmark datasets including SED1, ASD, and ECSSD (see Section 2.9 on page 71). Each dataset is randomly divided into two sets, training set and test set which contain 70% and 30% of the image dataset, respectively. From a statistical point of view, optimization methods often require much larger test sample sizes than the final evaluation in order to avoid "skimming testing variance". Here, we choose more conventional data splitting [15] instead of using 10-fold cross validation. The idea is that more training data is a good thing because it makes the detection model better whilst more test data makes the error estimate more accurate.

3.3.2 Benchmark Methods for Comparisons

The proposed method is compared to six benchmark methods including discriminative regional feature integration (DRFI) [76], geodesic saliency using background priors (GS) [180], saliency detection via graph-based manifold ranking (GMR) [188], saliency detection using maximum symmetric surround (MSS) [3], saliency filters (SF) [138], and saliency optimization from robust background detection (RBD) [203] on the three benchmark datasets, i.e., SED1, ASD, and ECSSD.

3.3.3 Parameter Settings

PSO is run independently for 31 times with a different seed numbers to search for the optimal set of weights. The central limit theorem (CLT), one of the most important theorems in statistics, implies that under most distributions, normal or non-normal, the sampling distribution of the sample mean will approach normality, as the sample size (number of solutions in this thesis) increases [71]. This fact holds especially true for sample sizes over 30 [71]. This is to facilitate conducting a statistical significance test on the obtained results. To follow the mentioned fact and consider computational time, we run PSO 31 times. In this experiment, we choose 31 instead of 30 to have an odd number of solutions to return the median solution. As mean solution can be highly affected by outliers, we report the median which does not have the mentioned problem and it can give us the "*middle*" solution of the 31 runs. The population size and the maximum number of iterations are empirically chosen to be 100 and 50, respectively. In PSO, parameters are set as follows: $w = 0.7298$, $c_1 = c_2 = 1.49618$ following default values in [185] and fully connected topology is used.

As the employed benchmark datasets contain images with different sizes (described in Section 2.9 on page 71), re-sizing images would be helpful for image processing. In all datasets, the ground truth of images are grayscale. For each dataset, the raw images and respective ground truth are re-sized to 200×200 pixels to leverage computational efficiency. In all the datasets, the images are color images.

The number of pyramid levels L is set to 6 in computing $\boldsymbol{f}_1$. Following [138,203], $\sigma_{spa} = 0.25$ in $\boldsymbol{f}_7$, $\sigma_c = 20$ in $\boldsymbol{f}_9$, and $\sigma_p = 0.25$ in $\boldsymbol{f}_8$, which allows a balance between local and global effects. The number of superpixels for $\boldsymbol{f}_4$ is empirically set to 200 ([187]) and for $\boldsymbol{f}_7$, $\boldsymbol{f}_8$, and $\boldsymbol{f}_9$ is empirically set to 59 ([138]). The number of clusters for $\boldsymbol{f}_5$ and $\boldsymbol{f}_6$ is empirically set to 6 ([45].

3.3.4 Evaluation Metrics

In this study, three widely-used criteria including precision-recall (PR) curve, receiver operating characteristic (ROC) curve and F-measure are employed to evaluate the different SOD methods based on the quantitative results [27]. The first two evaluation criteria are based on the overlapping area between subjective annotation and saliency prediction. From these two criteria, F-measure is also computed, F-measure jointly considers recall and precision, and AUC, which is the area under the ROC curve.

Using the frequency-tuned (FT) benchmark [1], the PR curve is obtained by binarizing the saliency map using a number of thresholds ranging from 0 to 255, as in [138]. On each threshold, a pair of precision/recall scores are computed, by comparing the binarized mask and the corresponding ground truth. The average precision and recall for all images are then used to plot the PR curve. Precision corresponds to the fraction of the pixels correctly labeled against the total number of pixels assigned salient, whereas recall is the fraction of the pixels correctly labeled in relation to the number of ground truth salient pixels.

$$Precision = \frac{TP}{TP + FP} \tag{3.8}$$

$$Recall = \frac{TP}{TP + FN} \tag{3.9}$$

ROC curve and area under ROC (AUC) are popular methods used to display the trade-off between true positive and false positive rate [168]. ROC is a two-dimensional representation of a method's performance, the AUC represents this information into a single scalar. As the name implies, it is calculated as the area under the ROC curve. A perfect model will score an AUC of 1, while random guessing will score an AUC around 0.5. The ROC curve is generated based on the true positive rates (TPR) and false positive rates (FPR) obtained during the calculation of the PR curve. The ROC curve is the plot of TPR versus FPR by varying the threshold. TPR

which is the same as recall and FPR can be computed as

$$FPR = \frac{FP}{FP + TN} \tag{3.10}$$

Often, neither precision nor recall can fully evaluate the quality of a saliency map. To this end, the F-measure (F_β) presented in Equation (3.11), is utilized as the weighted harmonic mean of precision and recall with a non-negative weight β^2.

$$F_\beta = \frac{(1 + \beta^2) Precision \times Recall}{\beta^2 Precision + Recall} \tag{3.11}$$

As suggested in many salient object detection works [1,138], we set β^2 to 0.3, to weight precision more. The reason is because recall rate is not as important as precision [113]. For instance, 100% recall can be easily achieved by setting the whole map to be foreground.

An image dependent adaptive threshold T_a proposed by Achanta et al. [1] is used to binarize the saliency map ($\boldsymbol{S}$). T_a is computed as twice as the mean saliency of $\boldsymbol{S}$.

$$T_a = \frac{2}{W \times H} \sum_{x=1}^{W} \sum_{y=1}^{H} \boldsymbol{S}(x, y) \tag{3.12}$$

where W and H are the width and height of the saliency map $\boldsymbol{S}$, respectively, and $\boldsymbol{S}(x, y)$ is the saliency value of the pixel at coordinates (x, y).

Average precision, recall, and F-measure values are also used to evaluate SOD methods. Using Equation (3.12), an obtained saliency map of a given image is binarized. Then, the binary mask and ground truth of the image are employed to compute precision, recall, and F-measure. Finally, for each dataset, the average of precision, recall, and F-measure are taken over all the images in the dataset.

The area under the PR curve (AUCPR) is also used as a measure of performance for comparing methods [123]. As the methods are assessed by a precision-recall curve on the benchmarks for SOD, AUCPR score is the

most suitable ranking measure. The AUC measure is frequently used as a standard measure to evaluate the performance of the SOD methods [123]. For each saliency map, the AUCPR and AUC measures are computed as per the FT benchmark [1]. The saliency map is thresholded using multiple thresholds and compared with the binary ground truth to compute the precision-recall curve for AUCPR, the ROC curve for AUC. The area under the curves gives the corresponding metrics [123].

3.4 Results and Discussions

To show the efficacy of this study, we compare the qualitative and quantitative results of the wPSOSOD method and the six other SOD methods.

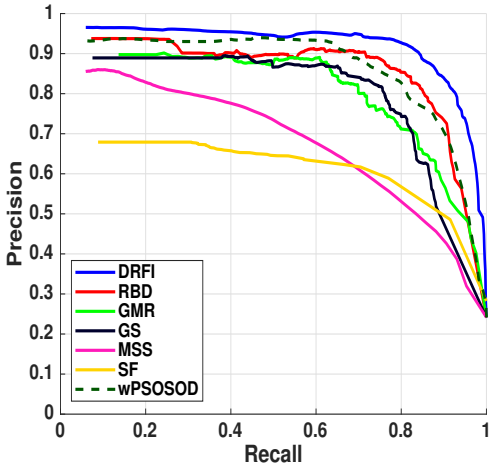
3.4.1 Quantitative Comparisons

In this section, we present the quantitative results of wPSOSOD and the six benchmark methods in Figures 3.3, 3.4, and 3.5 based on precision-recall curves, ROC curves, and F-measure criteria. Moreover, Table 3.1 gives the detailed quantitative results based on average precision, recall, and F-measure (discussed in Section 3.3.4). In this table, for the all the datasets, the median result from the 31 independent runs of wPSOSOD is reported. The Table 3.2 also provides the statistical significance test results based on t-test at the significance level 5% .

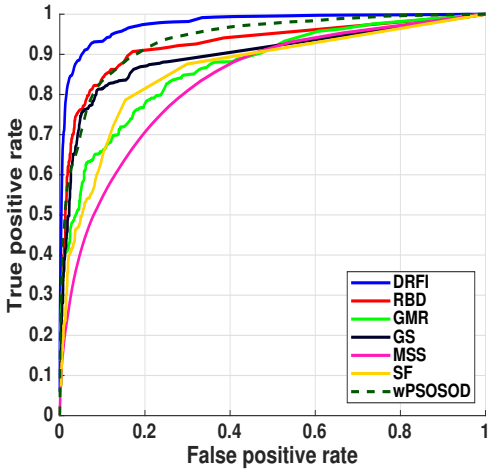
3.4.1.1 The SED1 Dataset

In Figure 3.3(a), wPSOSOD has worse performance than DRFI and comparable to RBD, but outperforms GS, GMR, SF, and MSS based on precision-recall curves on the SED1 dataset. As shown in Figure 3.3(b), wPSOSOD performs as the second best after DRFI in terms of AUC scores.

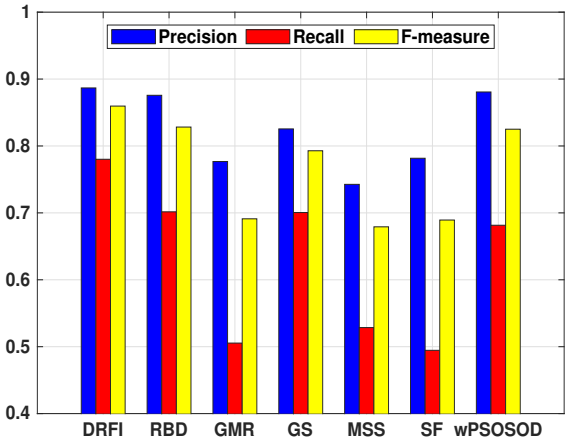
In Figure 3.3(c) and Table 3.1 (SED1), in terms of the average precision and F-measure values, DRFI and wPSOSOD have higher values of pre-



(a) Precision-recall curves



(b) ROC curves



(c) Average precision, recall, and F-measure

Figure 3.3: Quantitative results of wPSOSOD compared to the six other SOD methods based on the **SED1** dataset.

cision 0.8867 and 0.8807, and F-measure 0.7801 and 0.6815, respectively compared to the other SOD methods.

As the difference between average precision and recall values in RBD, GMR, MSS, SF, and wPSOSOD is not small, it can be concluded that these

Table 3.1: Quantitative results of wPSOSOD and the six other SOD methods based on average precision, recall, and F-measure values on the **SED1**, **ASD**, and **ECSSD** datastes. The abbreviations P, R, and F indicate precision, recall, and F-measure, respectively.

Dataset	SED1			ASD			ECSSD		
Method	P	R	F	P	R	F	P	R	F
DRFI	0.8867	0.7801	0.8596	0.8668	0.9079	0.8759	0.7622	0.6675	0.738
RBD	0.8757	0.7016	0.8283	0.8793	0.888	0.8813	0.7043	0.5969	0.6763
GMR	0.7768	0.5054	0.69111	0.8357	0.7486	0.8138	0.6286	0.3655	0.5391
GS	0.8255	0.7005	0.7928	0.8273	0.8967	0.8423	0.6499	0.6263	0.6443
MSS	0.7426	0.5286	0.6792	0.7146	0.6201	0.6904	0.5476	0.3911	0.5013
SF	0.7816	0.4946	0.6893	0.8512	0.7626	0.8187	0.6076	0.3731	0.5306
wPSOSOD	0.8807	0.6815	0.8250	0.848	0.8499	0.8485	0.7316	0.6187	0.7020

methods perform well on suppressing background with decreasing FP, but they do not have similar performance on highlighting the foreground object (decreasing FN).

The average AUCPR values of wPSOSOD and the different compared SOD methods are listed in Table 3.2, the last column is wPSOSOD which reports the average AUCPR value and standard deviation ($\bar{x} \pm s$). To assess the significance of the results, it is very important to use a suitable statistical test. Here, we perform t-test to investigate the statistical significance (if any) of the wPSOSOD method in comparison with the other six SOD methods. In Table 3.2, the symbol "↑" appears next to the method that has been significantly outperformed by wPSOSOD, and a "↓" is used to indicate that the corresponding method has significantly better performance than that of wPSOSOD. Based on the t-test at the significance level 5% in Table 3.2, apart from DRFI, wPSOSOD has significantly outperformed all the baseline methods on the SED1 dataset.

Table 3.2: The statistical comparison of wPSOSOD and the other seven SOD methods based on AUCPR on the **SED1**, **ASD**, and **ECSSD** datastes.

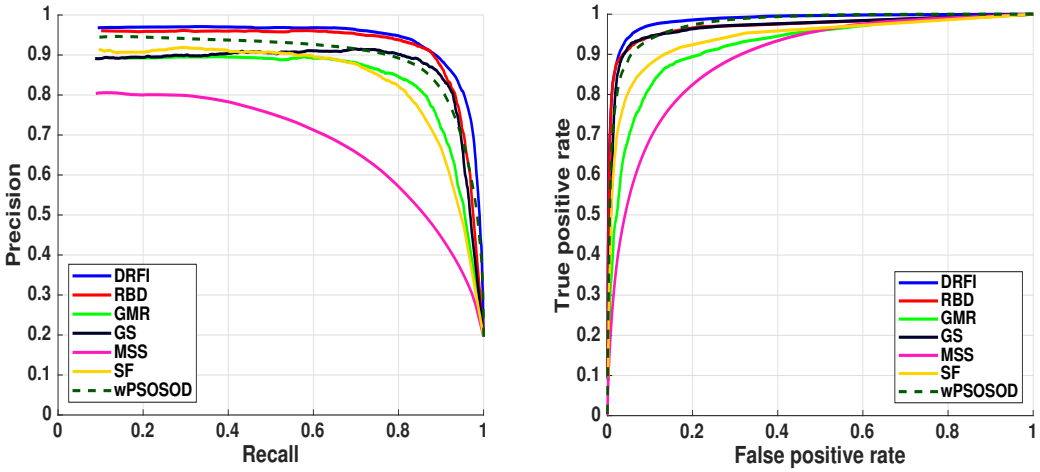
	DRFI	RBD	GMR	GS	MSS	SF	wPSOSOD
SED1	0.6959 ↓	0.6261 ↑	0.5615 ↑	0.5846 ↑	0.4606 ↑	0.5272 ↑	0.6178 ±0.0140
ASD	0.7520 ↓	0.7357 ↓	0.6617 ↑	0.7013 ↑	0.5330 ↑	0.6623 ↑	0.7088 ±0.0076
ECSSD	0.5721 ↓	0.4885 ↑	0.4119 ↑	0.4484 ↑	0.2794 ↑	0.3578 ↑	0.4922 ±0.0058

3.4.1.2 The ASD Dataset

Figure 3.4(a) shows that wPSOSOD has lower result compared to RBD and DRFI, but it has better performance than GMR, SF, and MSS methods based on precision-recall curves on ASD. Figure 3.4(b) shows that wPSOSOD has comparable performance to GS and RBD, and slightly lower than DRFI, and better performance than the GMR, SF, and MSS methods regarding ROC curves on ASD.

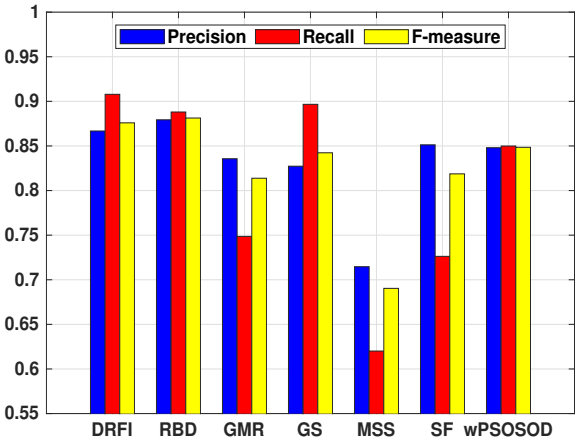
In Figure 3.4(c) and Table 3.1 (ASD), DRFI and RBD have the highest average precision, recall, and F-measure values compared to other SOD methods. Although SF and GMR with values of 0.8512 and 0.8537 have a slightly higher average precision than wPSOSOD with 0.848, wPSOSOD with values of 0.8499 and 0.8485 has higher average recall and F-measure than SF and GMR. The results show that SF and GMR mostly good at suppressing background (higher precision) and not that accurate on highlighting the salient object (lower recall), while wPSOSOD perform good on both sides on ASD.

Table 3.2 shows that DRFI and RBD have higher AUCPR compared to wPSOSOD, however, wPSOSOD significantly outperformed the other compared SOD methods, GMR, GS, SF, and MSS based on significant test on AUCPR values.



(a) Precision-recall curves

(b) ROC curves



(c) Average precision, recall, and F-measure

Figure 3.4: Quantitative results of wPSOSOD compared to the six other SOD methods based on the **ASD** dataset.

3.4.1.3 The ECSSD Dataset

In Figure 3.5(a), although wPSOSOD has lower performance than DRFI, it outperforms RBD, GS, GMR, SF, and MSS based on precision-recall curves on ECSSD. As shown in Figure 3.5(b), wPSOSOD performs as the second

best after DRFI in terms of AUC scores on ECSSD.

As shown in Figure 3.5(c) and Table 3.1 (ECSSD), after DRFI with values of 0.7622, 0.6675, and 0.738 for precision, recall, and F-measure, respectively, wPSOSOD have the better values of 0.7316, 0.6187, and 0.7020 for for precision,recall, and F-measure, respectively compared to the other SOD methods. As the ECSSD dataset has more complex images, most of the SOD methods do not achieve higher results compared to the other simple datasets such as ASD regarding precision, recall, and F-measure. However, wPSOSOD and DRFI show good performance on this complex dataset compared to other SOD methods.

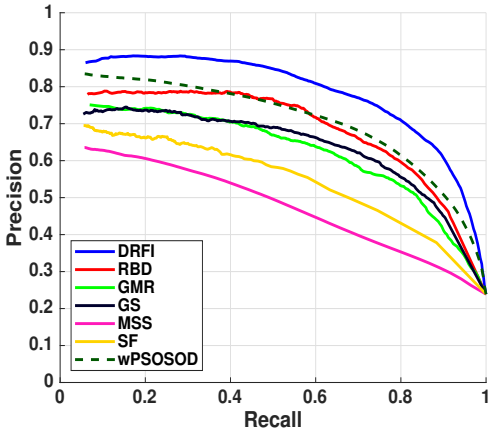
In Table 3.2, Although wPSOSOD has lower AUCPR value than DRFI, wPSOSOD with 0.4922 AUCPR value significantly outperform all other compared SOD methods.

It can be observed that wPSOSOD can tackle difficult datasets such as the SED1 and ECSSD datasets and perform well on those datasets. Although DRFI outperforms all the SOD methods on all the datasets, it employs 93 saliency features for saliency detection, which is much more than other methods.

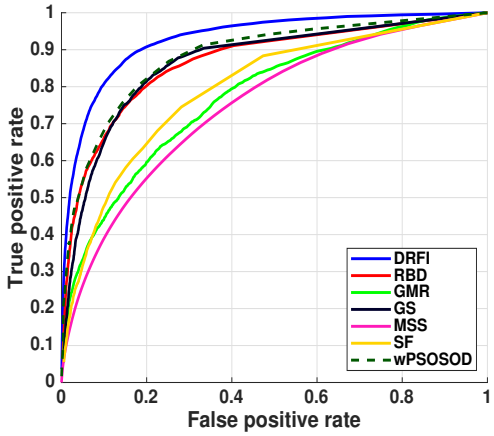
3.4.2 Qualitative Comparisons

For visual comparison, some saliency maps generated by wPSOSOD and the other six SOD methods for some samples of the three benchmark datasets, SED1, ASD, and ECSSD are presented in Figure 3.6(a)-(c).

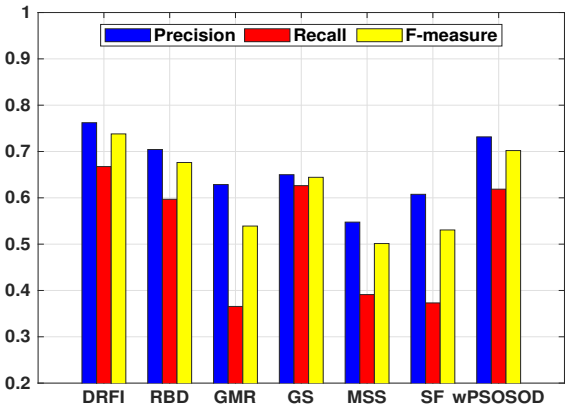
In Figure 3.6, the visual examples show that wPSOSOD has the ability to both detect salient object(s) and suppress background on all the three datasets. In Figures 3.6(a), 3.6(b), and 3.6(c), the images of the first row are challenging images, since they have complex background. Similar to RBD, wPSOSOD can successfully highlight salient objects and suppress background in those images. While the other methods such as DRFI, MSS, and SF have limitations in completely highlighting salient object; GS and



(a) Precision-recall curves



(b) ROC curves

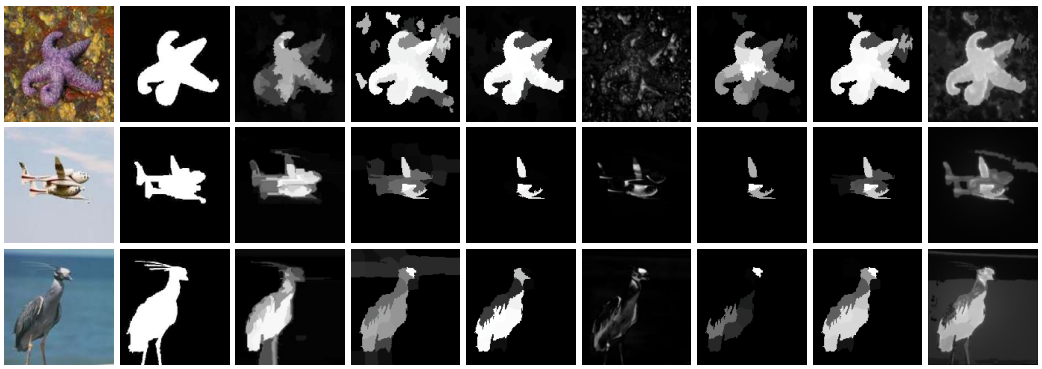


(c) Average precision, recall, and F-measure

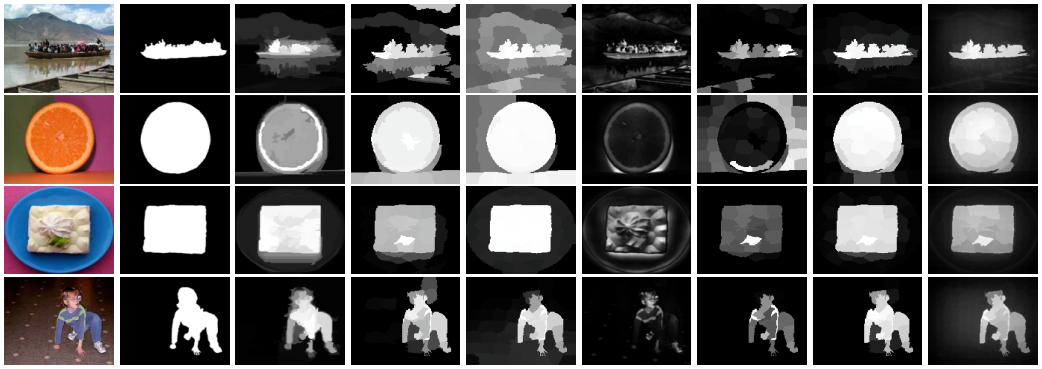
Figure 3.5: Quantitative results of wPSOSOD compared to the six other SOD methods based on based on the **ECSSD** dataset.

GMR lose their performance in suppressing background. The fourth row of Figure 3.6(c) shows an example for images having multiple salient objects, wPSOSOD can completely highlight both salient objects along suppressing background.

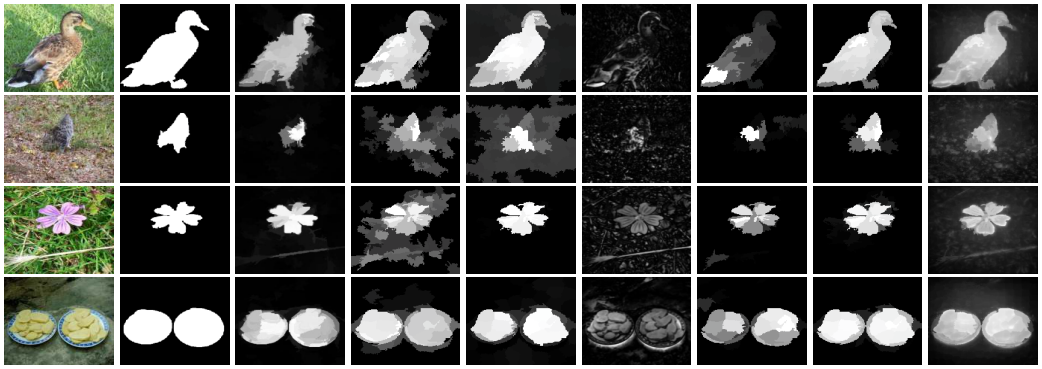
Figure 3.7 gives some examples of challenging and difficult images which selected from the SED1 dataset. Similar to the other SOD meth-



(a) **SED1**



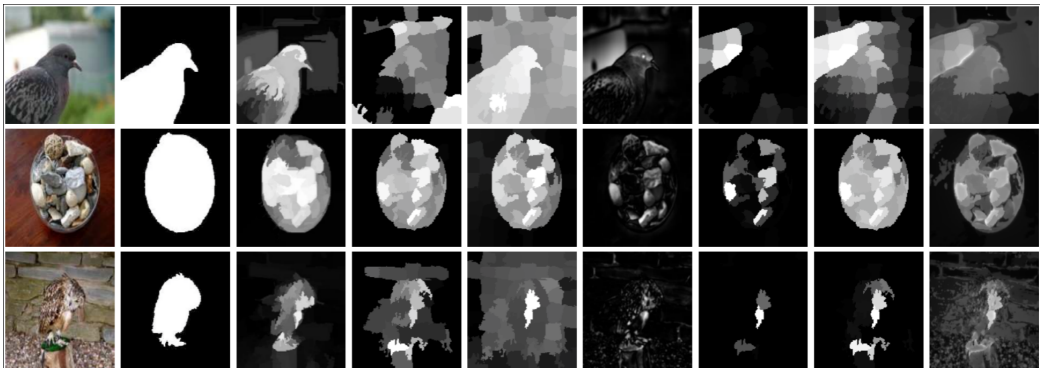
(b) **ASD**



(c) **ECSSD**

Original GT DRFI GS GMR MSS SF RBD wPSOSOD

Figure 3.6: Some visual examples of wPSOSOD and the six other SOD methods on the **SED1**, **ASD**, and **ECSSD** datasets.



Original GT DRFI GS GMR MSS SF RBD wPSOSOD

Figure 3.7: Some failure visual examples of wPSOSOD and the six other SOD methods on the **SED1** dataset.

ods, wPSOSOD has difficulties in completely detecting and highlighting salient objects and suppressing background. This problem can be caused by lack of some informative saliency features such as shape, texture or even high-level knowledge. In this study, most of the features are mainly based on the color features. However, the color information alone is not enough and some other features such as shape and texture features can be informative in the challenging cases.

3.4.3 Further Analysis

Table 3.3 shows the median evolved weight vectors by PSO for the nine features over the three different datasets. As shown in Table 3.3, the features have been assigned different weights based on the different datasets. In Table 3.3, weight w_7 for the feature $\boldsymbol{f}_7$ (background weighted contrast) has been found to be high for all the datasets consistently. This may be due to that $\boldsymbol{f}_7$ is an informative feature on background regions and it can effectively suppress background. $\boldsymbol{f}_3$ and $\boldsymbol{f}_7$ features have similar weights for the SED1 dataset and $\boldsymbol{f}_4$ and $\boldsymbol{f}_7$ features have similar weights for the ECSSD dataset, while $\boldsymbol{f}_7$ with the highest weight is the most important

Table 3.3: The evolved weight vectors for the nine saliency features.

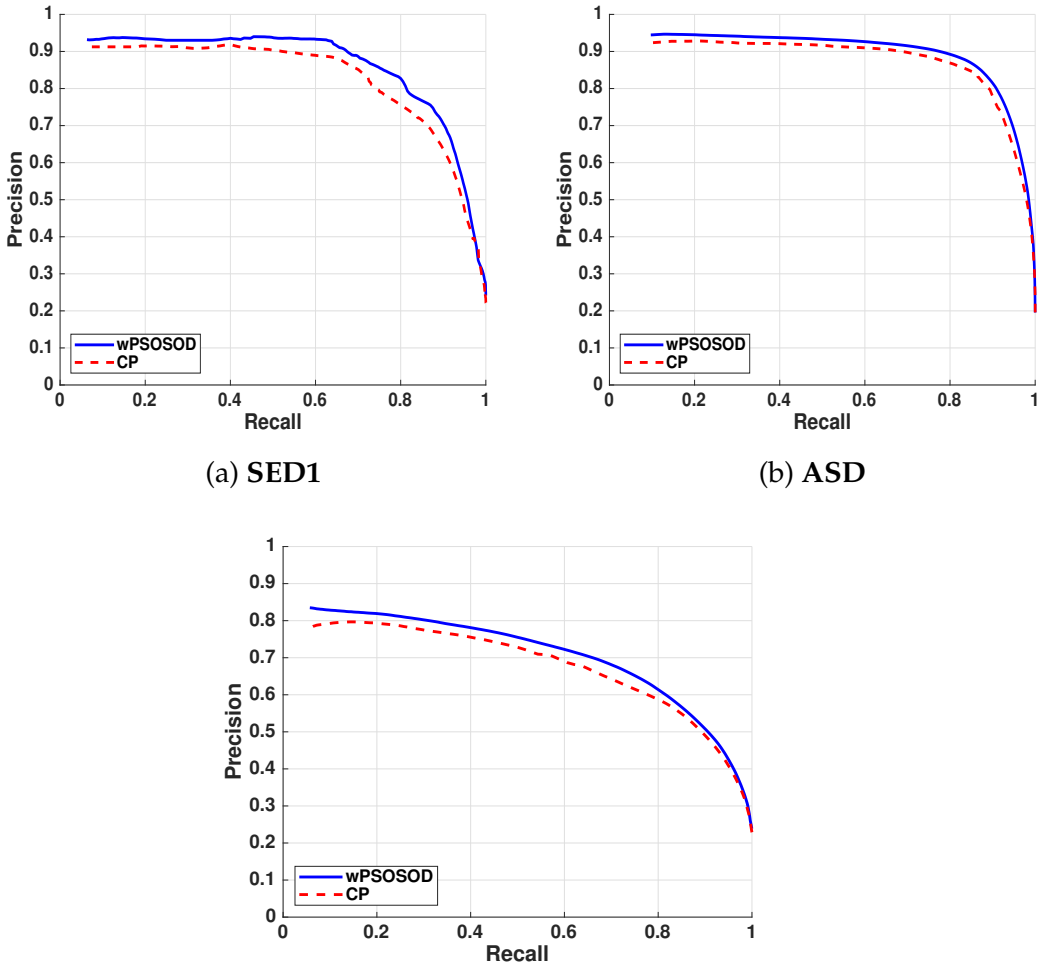
Dataset	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9
SED1	0.0125	0	0.3520	0	0.0379	0.2432	0.3520	0.0024	0
ASD	0.0292	0.0703	0.0863	0.1639	0	0	0.6503	0	0
ECSSD	0	0.0282	0.0196	0.4761	0	0	0.4761	0	0

feature for the ASD dataset.

In Table 3.3, $\boldsymbol{f}_9$ is weighted zero for all the datasets. This might be presented by the fact that the feature $\boldsymbol{f}_9$ is a redundant feature in the combination with the other features. Similar to the feature $\boldsymbol{f}_9$, $\boldsymbol{f}_5$ and $\boldsymbol{f}_8$ features are assigned zero or lower weights (close to zero), this probably caused by two reasons: 1) these features are redundant and do not add much information to the final result, or 2) these features do not complement with the other features and decrease the performance (fitness value) of the combination. The results reveal that not all the slaiency features from the feature set are required to be involved during the combination of features. Some features that are informative and can complement each other are given higher priority to contribute to the final result.

Table 3.3 shows different datastes favour different weights for the saliency features during the feature combination. One saliency feature may have higher weight for a dataset (e.g. w_4 for ECSSD), but lower weight for another one (e.g. w_4 for SED1).

Here, we also employ linearly combination of the nine features without weighting the features which is called "control performance" (shortly called CP) to compute saliency maps of images for the three datastes. By introducing CP, we can easily explore the effectiveness of weighted features compared to non-weighted features in a specific combination framework (linear combination). The reason of comparing the proposed weighting method with CP instead of other existing weighting methods is that CP provides a good baseline for a fair comparison using the same features and combination framework. Considering other feature weighting



(a) **SED1** (b) **ASD**

(c) **ECSSD**

Figure 3.8: Performance of wPSOSOD compared to CP based on precision-recall curves on the **SED1**, **ASD**, and **ECSSD** datasets.

methods in SOD, some SOD studies [45, 76, 119] have manually designed optimization frameworks for weighting features. However, it is not easy to manually investigate the priority of each feature in relation to the other features in the feature set, or the importance of each feature in the feature set based on the different image types. As shown in Figure 3.8, wPSOSOD has a better performance than CP based on precision-recall curves on three

different datasets, SED1, ASD, and ECSSD. Hence, it can be observed that assigning suitable weights to the features during the combination stage can improve the results. Meanwhile, the results show that wPSOSOD has the ability to find suitable weights for the features to reflect the importance degree of each feature in the combination stage.

3.5 Chapter Summary

This chapter investigated the capability of PSO for evolving suitable weight vectors for the nine saliency features on the different benchmark datasets. In this chapter, we studied nine types of features, covering different feature extraction levels (pixel, superpixel, and cluster) as an input feature set for PSO to evolve the weight vectors. The assigned weights to the different features on the three datasets were studied. Results show that each feature is given markedly different weight for the different datasets. Therefore, a feature may be informative for one dataset, but near useless for another one. However, the feature $\boldsymbol{f}_7$, which is a reliable background measure, is found to be the most important feature in all the three datasets. Generally, four features, color spatial distribution ($\boldsymbol{f}_3$), convex-hull-based center ($\boldsymbol{f}_4$), cluster-based spatial ($\boldsymbol{f}_6$), and background weighted contrast ($\boldsymbol{f}_7$) have been given higher importance by PSO compared to the other employed features.

For each benchmark dataset, wPSOSOD showed that it has the ability to find suitable weights for the features. Moreover, the results show that wPSOSOD considers complementary characteristic of the features during the weighting process.

In this chapter, the effectiveness of the evolved weights on the performance of SOD was studied. Evaluation of wPSOSOD performance based on precision-recall curves, ROC curves, and F-measure criteria demonstrates that wPSOSOD can consistently achieve a good performance across a wide range of images. Moreover, the average precision and recall results

show that wPSOSOD has the ability to perform well on both highlighting and suppress background. This indicates that assigning proper weights to the features help the SOD methods to achieve better results.

Although PSO automatically learns to find the importance degree of each feature in SOD, it does not give much information regarding which features are important in identifying foreground objects and/or background. Therefore, we aim to investigate the impact of different saliency features on foreground objects and background in Chapter 4. Based on the observations in this chapter, $\boldsymbol{f}_3$, $\boldsymbol{f}_4$, $\boldsymbol{f}_6$, and $\boldsymbol{f}_7$ features will be employed as informative features in the following chapter. In the next chapter, we will investigate the effectiveness of these important features on foreground and background regions. As wPSOSOD linearly combines all the nine saliency features and does not explore different ways for combining those features, we aim to investigate this in the next chapter.

Chapter 4

Manually Constructing and Combining Foreground and Background Features

4.1 Introduction

In Chapter 3, the wPSOSOD method generated a good weight vector for a set of existing saliency features to reflect the relative importance of each feature in the feature combination process. wPSOSOD improved the performance of the SOD method by giving different priorities to different features. However, wPSOSOD generally focused on exploring the importance degree of each feature on different images and it did not consider how different saliency features impact different regions of the image.

Most existing SOD methods focus on designing diverse features and combining them heuristically using simple approaches, such as multiplication [138] or weighted average [113]. Similar to the mentioned SOD methods, wPSOSOD simply combines the weighted features using linearly summation without considering how a suitable way of combination can be applied to enhance the complementary characteristic of features.

As SOD mostly concerns with two important parts of the image, fore-

ground objects and background, the combination process of features can benefit from the location information of salient and background pixels to accurately assign saliency values. Previous studies employed a center prior to compute the location of the foreground object and assign higher saliency to the regions near the image center [162, 163]. However, this principle becomes invalid when the objects are placed far from the image center. To avoid this problem here, the convex-hull of the foreground object is utilized to estimate the location of the foreground object [187].

wPSOSOD evaluates the goodness of each feature, but it does not provide information of which features are good at highlighting the salient objects and which features are good at suppressing background. In this chapter, we investigate the impact of using different features to determine saliency within foreground and background regions of the image.

4.1.1 Chapter Goals

In this chapter, a new bottom-up SOD method is developed by using the important saliency features found by wPSOSOD to investigate their effectiveness on representing the foreground and background regions. Here, our aim is to develop two informative features that are targeted at identifying foreground objects and background. Furthermore, we focus on designing a framework to assign saliency values to the foreground and background pixels. Specifically, this chapter aims to fulfill the following objectives:

- Construct two informative *foreground* ($\boldsymbol{FG}$) and *background* ($\boldsymbol{BG}$) features which preform well on highlighting foreground objects and suppressing background regions, respectively;
- Develop a new framework to assign saliency values to image pixels based on the convex-hull-based center prior;
- Investigate whether the new constructed features are better than the

individual (original) ones; and

- Investigate whether the constructed saliency features and the newly designed combination framework can improve the performance of the SOD method.

4.1.2 Chapter Organization

The remainder of the chapter is organized as follows. Section 4.2 details the proposed method. Section 4.3 provides the experiment design. Section 4.4 presents and discusses the results. Section 4.5 summarises this chapter.

4.2 New Bottom-up SOD Method

In this section, we develop a new bottom-up method for SOD, called FBC (foreground background features combination), including the overall structure, $\boldsymbol{FG}$ and $\boldsymbol{BG}$ saliency features constructions and convex-hull center prior are explained.

4.2.1 The Overall Algorithm

The proposed method is developed by constructing two informative features and designing a combination framework to effectively highlight foreground objects and suppress background. The idea is to exploit the complementary characteristic of the saliency features as well as the varying effectiveness of the features on foreground and background regions of the image. Thus, FBC contains two stages including: constructing $\boldsymbol{FG}$ and $\boldsymbol{BG}$ features, and developing a suitable way to combine the constructed features.

For the combination stage, we design a framework based on the foreground object's convex-hull to combine features to get the final saliency map. The idea behind using convex-hull is to assign each pixel the most

likely value of being foreground or background. Since the inner region of the convex-hull mostly covers the foreground regions, we employ the $\boldsymbol{FG}$ feature to assign saliency values to the inner pixels. In contrast, the outer region of the convex-hull is likely to have both foreground and background regions. The reason is that, although the convex-hull helps us to locate the foreground object, it does not cover or identify the whole foreground regions. As the outer region of the convex-hull may contain some foreground regions, we need to consider this fact in computing both $\boldsymbol{FG}$ and $\boldsymbol{BG}$ features.

4.2.2 Foreground and Background Feature Construction

The $\boldsymbol{FG}$ and $\boldsymbol{BG}$ features are manually constructed by integrating four features, pixel-based color spatial distribution $\boldsymbol{f}_3$, cluster-based color spatial distribution $\boldsymbol{f}_6$, and background weighted contrast $\boldsymbol{f}_7$ following the Chapter 3 and SUSAN edge feature [153].

The $\boldsymbol{FG}$ feature is formulated to perform well at identifying foreground regions, whereas the $\boldsymbol{BG}$ feature is formulated to identify the foreground in a way that respects the background. The result is that the $\boldsymbol{FG}$ feature will place marginal pixels into the foreground, while the $\boldsymbol{BG}$ feature will tend to classify them as background. In this chapter, we will use the aforementioned saliency features with the following definitions (Table 4.1). Figure 4.1 shows some sample images, their ground truth, and the ten saliency feature maps, where the first nine feature maps are the same as in Chapter 3 and $\boldsymbol{f}_{10}$ is explained as follows.

$\boldsymbol{f}_{10}$: A modified SUSAN technique by [153] is used as a edge detector to highlight boundaries and identify rapid color changes in the image [153].

The pixel-based color spatial distribution $\boldsymbol{f}_3$ in [113] and cluster-based color spatial distribution $\boldsymbol{f}_6$ in [45] have been shown to have good performance in highlighting foreground objects, and background weighted-contrast $\boldsymbol{f}_7$ in [138] has been proposed as an informative background fea-

Figure 4.1: Some image samples, ground truth, and the corresponding ten saliency feature maps.

Table 4.1: The input feature set.

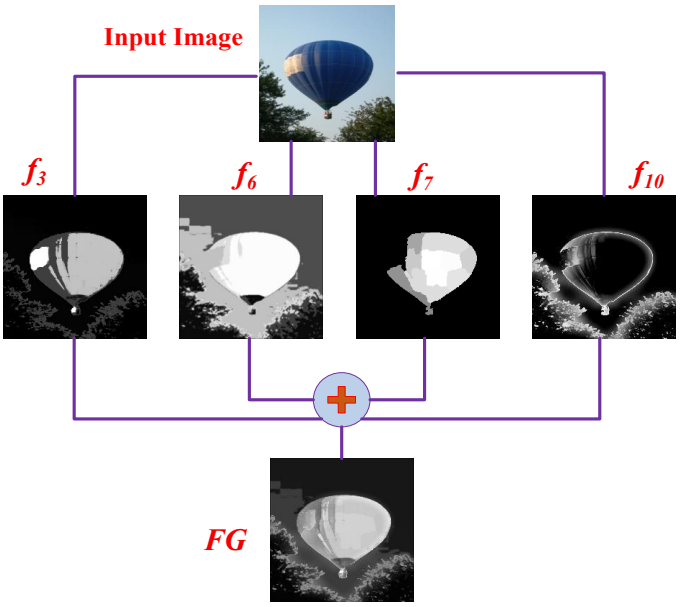
Feature	Definition
$\boldsymbol{f}_3$	Color spatial distribution
$\boldsymbol{f}_4$	Convex-hull-based center
$\boldsymbol{f}_6$	Cluster-based spatial
$\boldsymbol{f}_7$	Background weighted contrast
$\boldsymbol{f}_{10}$	SUSAN edge

ture. Furthermore, the Chapter 3 showed that $\boldsymbol{f}_6$ and $\boldsymbol{f}_7$ have been given high weights or importance by PSO compared to the other features on the three datasets in the experiment. Therefore, $\boldsymbol{f}_3$, $\boldsymbol{f}_6$, and $\boldsymbol{f}_7$ are used in this chapter for constructing $\boldsymbol{FG}$ nad $\boldsymbol{BG}$.

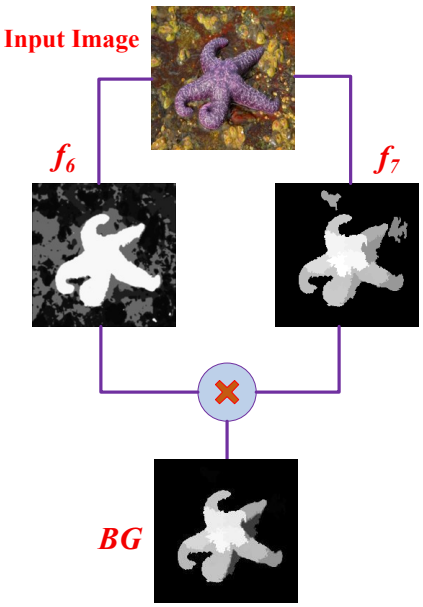
4.2.2.1 Foreground Feature ($\boldsymbol{FG}$)

The goal of computing $\boldsymbol{FG}$ is having a feature mainly targeting to highlight the foreground object. To achieve this goal, we need to increase the confidence of the foreground pixels. We therefore combine features that complement each other in an attempt to better capture the foreground object.

To compute $\boldsymbol{FG}$, $\boldsymbol{f}_3$, $\boldsymbol{f}_6$, $\boldsymbol{f}_7$, and $\boldsymbol{f}_{10}$ features are integrated as shown in Figure 4.2(a). Although $\boldsymbol{f}_3$ and $\boldsymbol{f}_6$ generally perform well in highlighting the foreground object, being color-based may cause them to incorrectly miss, i.e. give value below the threshold to, some foreground regions, e.g., when the foreground object is complex or non-homogeneous. As it can be seen in Figure 4.2(a), where the foreground object is not homogeneous, $\boldsymbol{f}_3$ does not highlight the blue region of the hot air balloon. $\boldsymbol{f}_6$ can detect the blue region, but it misses the wicker basket and skirt (bottom of the balloon) regions. Here, we use $\boldsymbol{f}_{10}$ to relieve the mentioned problem by identifying the boundary of the foreground object. However, $\boldsymbol{f}_3$, $\boldsymbol{f}_6$, and



(a)



(b)

Figure 4.2: An example demonstrates the process for computing (a) $\boldsymbol{FG}$, and (b) $\boldsymbol{BG}$.

$\boldsymbol{f}_{10}$ may falsely highlight some background regions, but $\boldsymbol{f}_7$ will be helpful to handle this problem. We add $\boldsymbol{f}_7$ to the other features, since the $\boldsymbol{FG}$ feature will be involved in computing saliency values of the outer regions in the combination framework, due to the convex-hull may not cover the whole salient object. Therefore, the $\boldsymbol{FG}$ feature is required to be informative towards background. Thus, $\boldsymbol{FG}$ is computed as the sum of the four features.

We use the add operator to combine the four features, since $\boldsymbol{FG}$ aims to increase the confidence of each pixel value being foreground. If all feature maps assign a high saliency value for a specified pixel, that pixel will be more likely a foreground pixel. If one feature map gives a high value to a pixel, while others give a low value, the pixel may will still be a foreground pixel, but the lower confidence of that is reflected in a relatively low saliency.

4.2.2.2 Background Feature ($\boldsymbol{BG}$)

To accurately suppress background, a background feature $\boldsymbol{BG}$ is required to focus mainly on suppressing background pixels. Thus, we integrate the features complementing each other to perform background suppression.

To compute $\boldsymbol{BG}$, $\boldsymbol{f}_6$ and $\boldsymbol{f}_7$ features are integrated as shown in Figure 4.2(b). Although $\boldsymbol{f}_7$ is mostly good at suppressing background, it may falsely highlight some background regions in some image types, e.g., images having cluttered and/or complex background (Figure 4.2(b)). Moreover, $\boldsymbol{f}_7$ often misses some foreground regions in some image types, and badly affect the overall performance. Therefore, we need to involve another feature like $\boldsymbol{f}_6$. We compute $\boldsymbol{BG}$ as

$$\boldsymbol{BG} = \boldsymbol{f}_6 \times \boldsymbol{f}_7 \tag{4.1}$$

Here, multiplying $\boldsymbol{f}_6$ by $\boldsymbol{f}_7$ helps $\boldsymbol{f}_7$ to increase the confidence of being background for the background pixels by decreasing their saliency values.

Therefore the saliency value of pixels which are wrongly highlighted in $\boldsymbol{f}_7$ are decreased.

We can see that the $\boldsymbol{FG}$ feature is, in a sense, an optimistic measure that a pixel is in the foreground, any feature indicating that the pixel is salient is considered to increase the final saliency. By contrast, $\boldsymbol{BG}$ is a pessimistic measure of saliency, because if either of constituent features judges the pixel not to be salient, then the final saliency of that pixel will be low.

4.2.3 Feature Combination Framework

4.2.3.1 Convex-hull Center Prior Design

We compute a convex-hull enclosing interesting points to estimate the location of the foreground salient object and then use the centroid of the convex-hull as the center to get the convex-hull-based center prior map. To achieve this, the color boosting Harris points [173] is adopted to find the corners or marginal points of the visual salient regions in the image. The corner points provide us a coarse location of the salient regions. Then, we remove any corner points near the boundary of the image, and employ convex-hull to circle the remaining salient points. As the color boosting Harris points usually cluster around the salient regions, the convex-hull would occupy the majority of the salient regions while including a small quantity of background.

The convex-hull of the foreground object roughly segments the image into two disjoint regions: the *inner* region (I) and the *outer* region (O). Pixels in the inner region tend to be foreground, while pixels in the outer region are more likely to be part of the background. We assume that the center of the convex-hull roughly denotes the center of the foreground object in the image. In our implementation, we use a centered anisotropic Gaussian distribution to model the center prior map. Convex-hull-biased

Gaussian model is

$$\boldsymbol{P}c(x_p, y_p) = \exp\left(-\frac{\|x_p - c_x\|^2}{2\sigma_x^2} - \frac{\|y_p - c_y\|^2}{2\sigma_y^2}\right) \tag{4.2}$$

where x_p and y_p denote the horizontal and vertical positions of the pixel p, (c_x, c_y) is the center of the foreground object, and σ_x^2 and σ_y^2 respectively indicate the horizontal and vertical variances and they are set $\sigma_x^2 = \sigma_y^2 = 0.15$ with pixel coordinates normalizes to [0,1] based on our empirical tuning and according to [187].

4.2.3.2 Assigning Saliency Values

$\boldsymbol{FG}$ and $\boldsymbol{BG}$ are fused by using the newly designed combination framework to compute the final saliency map as outlined in Figure 4.3. The convex-hull segments each image into inner I and outer O regions, a pixel located in I is assumed to be a foreground pixel (x_p^I, y_p^I) and a pixel located in O is assumed to be a background pixel (x_p^O, y_p^O). After dividing pixels into two groups, we compute the saliency value $\boldsymbol{S}(\cdot)$ pixelwise for each inner region pixel (x_p^I, y_p^I) as

$$\boldsymbol{S}(x_p^I, y_p^I) = \boldsymbol{P}c(x_p^I, y_p^I) \times \boldsymbol{FG}(x_p^I, y_p^I) \tag{4.3}$$

the saliency value $\boldsymbol{S}(\cdot)$ for each outer region pixel (x_p^O, y_p^O) is also computed pixelwise as

$$\begin{aligned}\boldsymbol{S}(x_p^O, y_p^O) = \boldsymbol{P}_c(x_p^O, y_p^O) \times \boldsymbol{FG}(x_p^O, y_p^O) + \\ (1 - \boldsymbol{P}_c(x_p^O, y_p^O)) \times \boldsymbol{BG}(x_p^O, y_p^O)\end{aligned} \tag{4.4}$$

In Equation (4.4) , the $\boldsymbol{FG}$ feature is also considered in computing the saliency values for pixels in the outer region. The point is that the convex-hull may not cover the whole foreground object in some image types, thus the outer region possibly contains some foreground regions. Therefore, we need to employ the $\boldsymbol{FG}$ feature in the outer region saliency computation to correctly highlight the foreground pixels in the outer region.

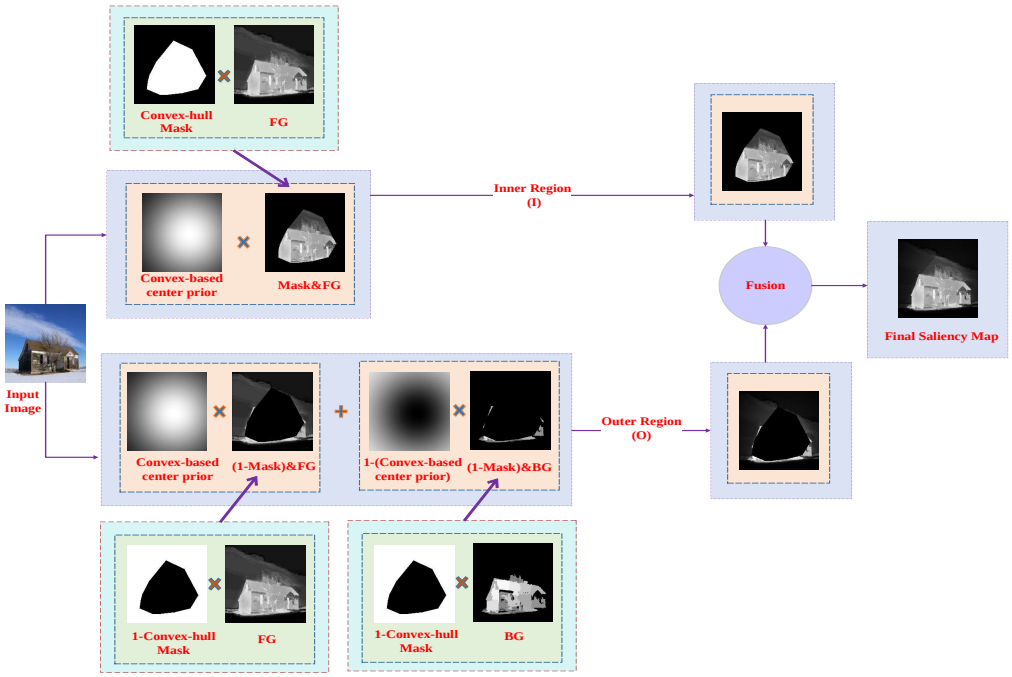


Figure 4.3: Scheme of the feature fusion strategy used to generate the final saliency map.

4.3 Experiment Design

4.3.1 Datasets

In this work, the performance of the proposed method is evaluated using three widely-used SOD datasets including SED1 [27], ECSSD [27], and ASD [113]. The details can be found in Section 2.9 (on page 71). In this experiment, similar to Chapter 3, we use 70% of the datasets to develop the FBC method. The FBC method does not need training, as it does not have a learning process. Similar to the Chapter 3, 30% of each dataset is used for testing the developed algorithm.

4.3.2 Benchmark Methods for Comparisons

The FBC method is compared to seven other SOD methods, four methods are selected from [27] including DRFI, GS, GMR, and SF, and three other methods, MSS [3], wPSOSOD (Chapter 3), and RBD [203].

4.3.3 Evaluation Metrics

The performance of the FBC method is evaluated using the evaluation criteria described in Section 3.3.4 (on page 91).

4.4 Results and Discussions

4.4.1 Quantitative Comparisons

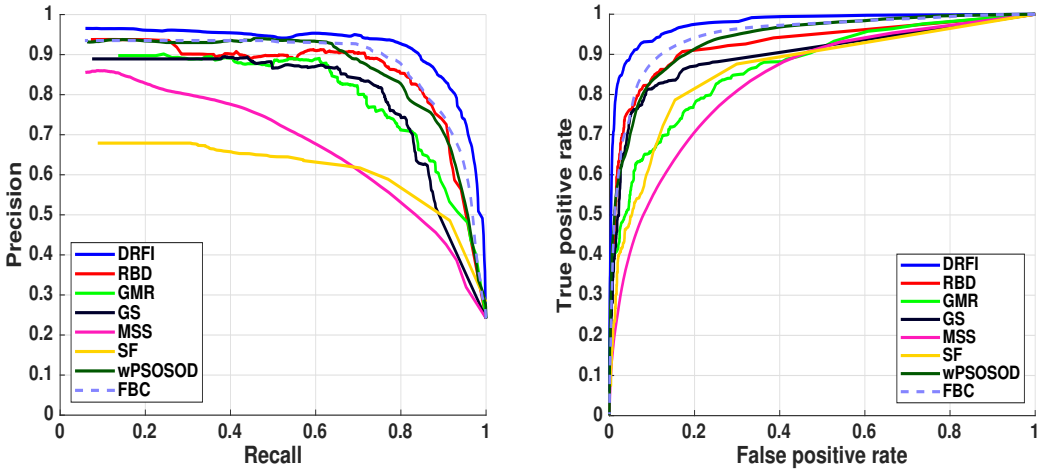
4.4.1.1 The SED1 Dataset

In Figure 4.4(a), although FBC has lower performance than DRFI, it outperforms SF, GMR, MSS, GS, and wPSOSOD on the SED dataset based on the precision-recall curves. Here, the precision-recall curve of wPSOSOD is slightly higher than RBD compared to wPSOSOD. Based on the ROC curves in Figure 4.4(b), FBC has the second best ROC curve on SED1.

Figure 4.4(c) and Table 4.2 (SED1) show that DRFI has the highest average precision, recall, and F-measure with the values of 0.8867 and 0.7801, and 0.8596 on the SED1 dataset. Although RBD and wPSOSOD have slightly higher precision than FBC, FBC has higher recall and F-measure values, 0.7555 and 0.8344, respectively.

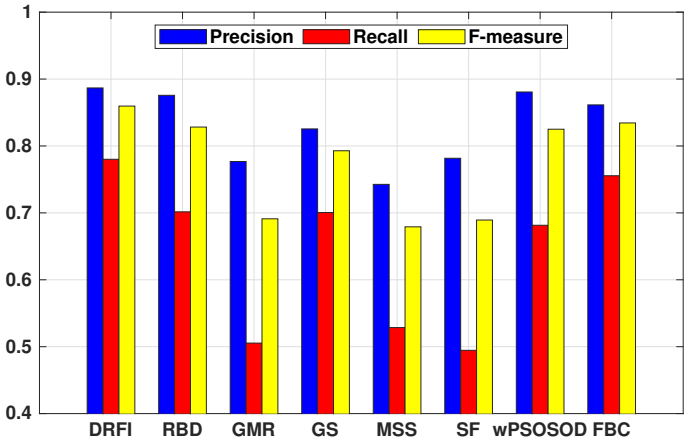
On the SED1 dataset, FBC performs better than wPSOSOD on highlighting the foreground object along with good performance on suppressing background.

Table 4.3 shows that FBC with 0.6308 AUCPR value significantly outperforms wPSOSOD and other benchmark methods except for DRFI based on the t-test at the significance level 5%.



(a) Precision-recall curves

(b) ROC curves



(c) Average precision, recall, and F-measure

Figure 4.4: Performance of FBC compared to the other seven SOD methods based on the **SED1** dataset.

4.4.1.2 The ASD Dataset

On the ASD dataset, as shown in Figure 4.5(a), DRFI and RBD have higher precision-recall curves than FBC and other SOD methods. Unlike the SED1 dataset, RBD performs better than the FBC method on the ASD

Table 4.2: Quantitative results of FBC and the seven other SOD methods based on average precision, recall, and F-measure values on the **SED1**, **ASD**, and **ECSSD** datastes. The abbreviations P, R, and F indicate precision, recall, and F-measure, respectively.

Dataset	SED1			ASD			ECSSD		
Method	P	R	F	P	R	F	P	R	F
DRFI	0.8867	0.7801	0.8596	0.8668	0.9079	0.8759	0.7622	0.6675	0.738
RBD	0.8757	0.7016	0.8283	0.8793	0.888	0.8813	0.7043	0.5969	0.6763
GMR	0.7768	0.5054	0.69111	0.8357	0.7486	0.8138	0.6286	0.3655	0.5391
GS	0.8255	0.7005	0.7928	0.8273	0.8967	0.8423	0.6499	0.6263	0.6443
MSS	0.7426	0.5286	0.6792	0.7146	0.6201	0.6904	0.5476	0.3911	0.5013
SF	0.7816	0.4946	0.6893	0.8512	0.7626	0.8187	0.6076	0.3731	0.5306
wPSOSOD	0.8807	0.6815	0.8250	0.848	0.8499	0.8485	0.7316	0.6187	0.7020
FBC	0.8614	0.7555	0.8344	0.8562	0.8724	0.8599	0.7585	0.6304	0.7245

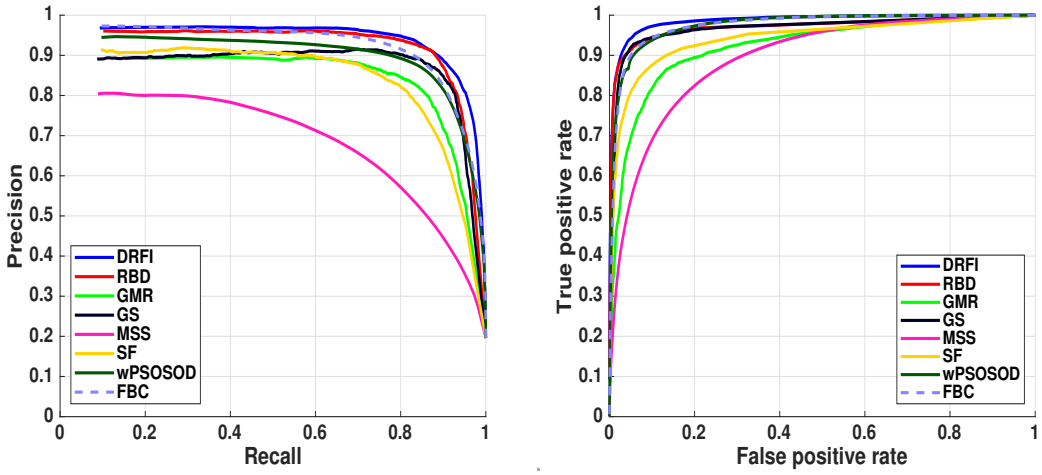
Table 4.3: The statistical comparison of FBC and the other seven SOD methods based on AUCPR on the **SED1**, **ASD**, and **ECSSD** datastes.

	DRFI	RBD	GMR	GS	MSS	SF	wPSOSOD	FBC
SED1	0.6959 $\downarrow$	0.6261 $\uparrow$	0.5615 $\uparrow$	0.5846 $\uparrow$	0.4606 $\uparrow$	0.5272 $\uparrow$	0.6178 $\pm$0.0140 $\uparrow$	0.6308
ASD	0.7520 $\downarrow$	0.7357 $\downarrow$	0.6617 $\uparrow$	0.7013 $\uparrow$	0.5330 $\uparrow$	0.6623 $\uparrow$	0.7088 $\pm$0.0076 $\uparrow$	0.7129
ECSSD	0.5721 $\downarrow$	0.4885 $\uparrow$	0.4119 $\uparrow$	0.4484 $\uparrow$	0.2794 $\uparrow$	0.3578 $\uparrow$	0.4922 $\pm$0.0058 $\uparrow$	0.5120

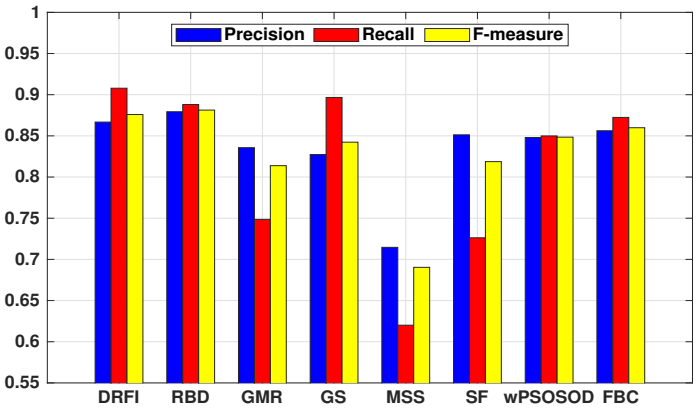
dataset. As ASD dataset generally has images with simple foreground object, RBD can increase its recall along precision. Based on the ROC curves in Figure 4.5(b), after DRFI, FBC, GS, RBD, and wPSOSOD have close performance.

In Figure 4.5(c) and Table 4.2 (ASD), after DRFI and RBD, FBC has high values of 0.8562 and 0.8599 for precision and F-measure, respectively. Similar to DRFI, RBD, and wPSOSOD, precision and recall values in the FBC method are close which means FBC performs on the foreground object as good as background.

Table 4.3 shows that FBC can significantly perform better than wPSOSOD and the four other competitive methods on ASD.



(a) Precision-recall curves (b) ROC curves

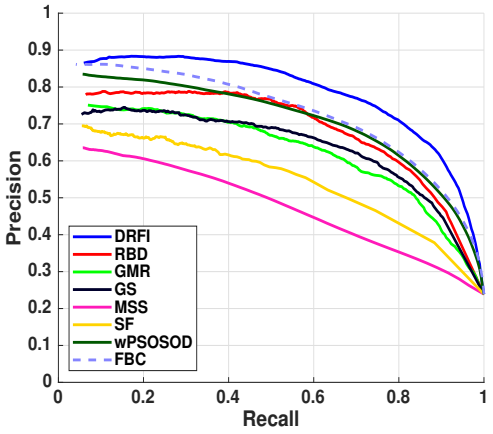


(c) Average precision, recall, and F-measure

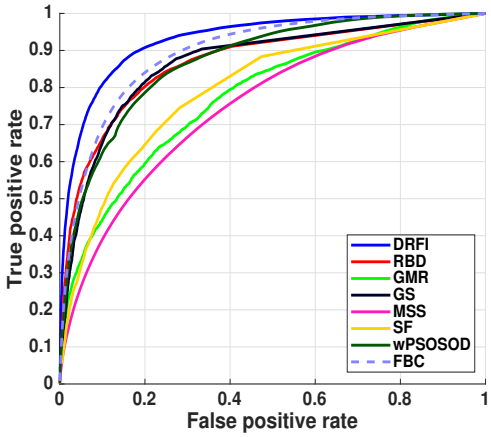
Figure 4.5: Performance of FBC compared to the seven other SOD methods based on the **ASD** dataset.

4.4.1.3 The ECSSD Dataset

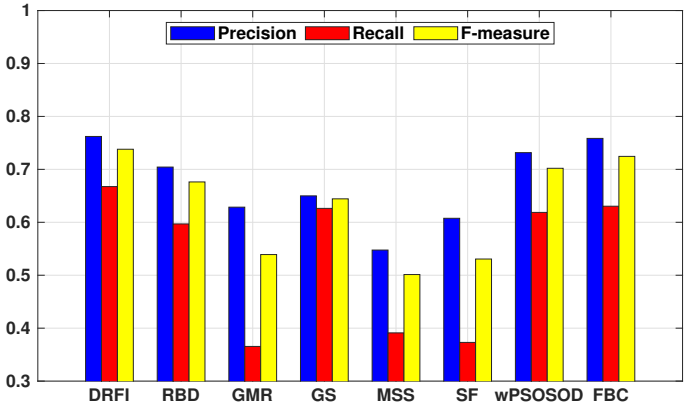
In Figures 4.6(a) and (b), FBC has similar and slightly better performance than wPSOSOD and RBD based on precision-recall and ROC curves on the ECSSD dataset.



(a) Precision-recall curves



(b) ROC curves



(c) Average precision, recall, and F-measure

Figure 4.6: Performance of FBC compared to the seven other SOD methods based on the **ECSSD** dataset.

In Figure 4.6(c) and Table 4.2 (ECSSD), after DRFI with 0.7622, 0.6675, and 0.738 values for average precision, recall, and F-measure, respectively, FBC has the second best values of 0.7585, 0.6304, and 0.7245 for average precision, recall, and F-measure, respectively.

Although designing a good feature combination framework and considering the effectiveness of the features on foreground and background

regions helped to improve the performance of the wPSOSOD method, the lack of enough informative features is still limitation in challenging datasets such as ECSSD and SED1.

Considering Table 4.3 for statistical results based on the t-test at the significance level 5%, after DRFI FBC with value of 0.5120 for AUCPR obtains a larger area under the precision-recall curve and significantly perform better than the other SOD methods.

4.4.2 Qualitative Comparisons

Figure 4.7 shows some sample saliency maps produced by the eight SOD methods. FBC successfully highlighted the foreground object and suppressed the background for the majority of the image types, e.g., simple and complex. It can be seen in Figure 4.7 that most of the image backgrounds are complex or cluttered, e.g., the 3rd, 5th, 6th, 7th, and 15th rows, apart from DRFI as it has probably better performance, FBC has the highest quality on suppressing background compared to the other SOD methods. For the image in the 14th row with multiple salient objects, FBC produces a saliency map covering all objects with uniform highlighting and suppressed background, unlike the other methods.

For the image in the 2nd row of Figure 4.7, although the foreground object is not homogeneous, FBC can mostly cover the salient object like DRFI. Moreover, it can be observed from the majority of the examples in Figure 4.7 that FBC can properly detect the boundaries of the salient objects.

Figures 4.8 shows some images containing cluttered/complex background, non-homogeneous foreground objects, and the saliency maps of the eight methods for these image types. As shown in Figure 4.8, although FBC misses some regions of the salient objects, it detects the precise location of salient object and it still completes some other regions of the salient object. For the 1st row, similar to the other methods and except for DRFI,

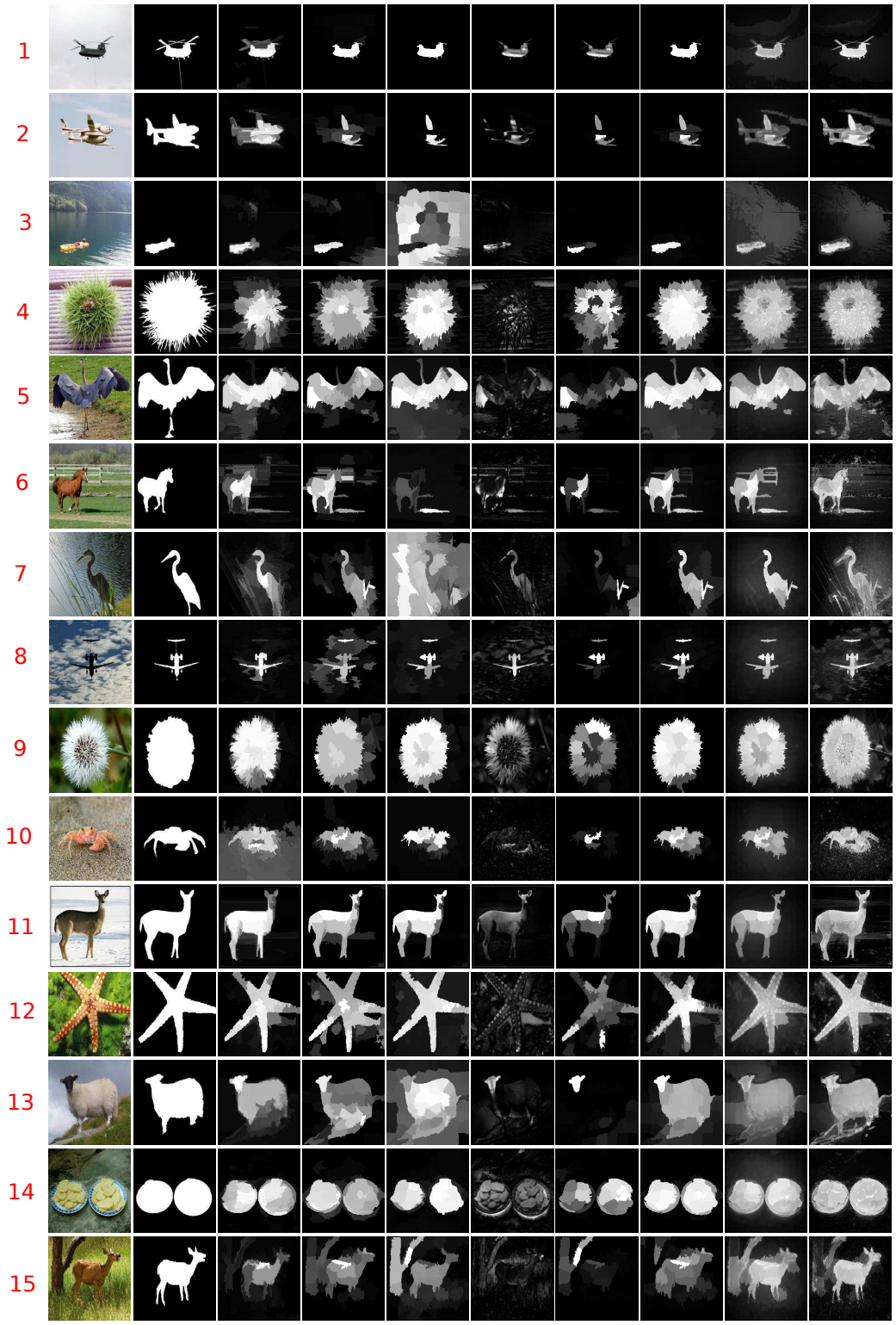
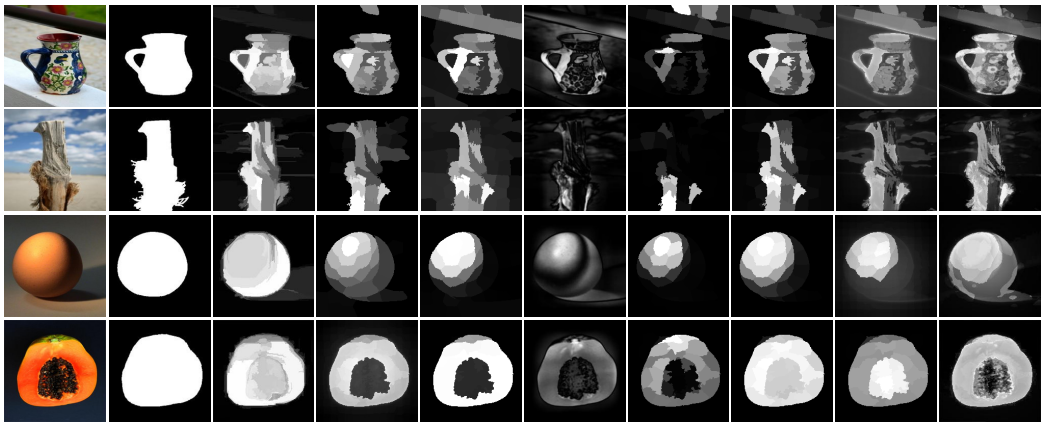


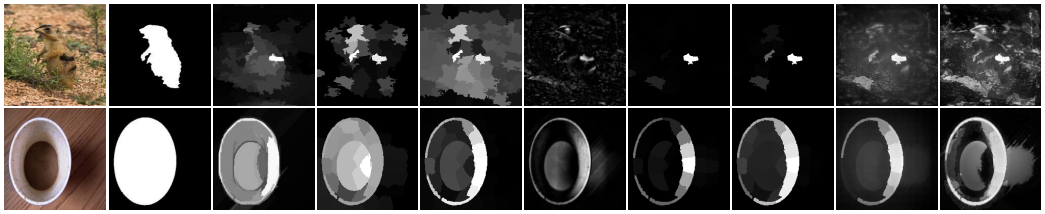
Figure 4.7: Some visual examples where the FBC method performs good on highlighting foreground object(s) and suppressing background on the images from the **SED1**, **ASD**, and **ECSSD** datastes.



Original GT DRFI GS GMR MSS SF RBD wPSOSOD FBC

Figure 4.8: Some challenging examples where the FBC method has slightly difficulties in returning accurate saliency maps. These sample images are taken from **SED1** and **ASD**.

the FBC method falsely highlights some regions of the background, since the background is complex.



Original GT DRFI GS GMR MSS SF RBD wPSOSOD FBC

Figure 4.9: Some failure visual examples of the FBC method and its comparison with saliency maps of the seven other SOD methods. These sample images are taken from **ECSSD** and **ASD**.

However, FBC has difficulty in detecting the correct salient object in some complex image types such as those presented in Figure 4.9. As it can be seen, the images are complex, i.e., the foreground object has similar color to the background (2nd row), or the foreground object is not homo-

Table 4.4: The average run time per image (seconds).

Method	GS	GMR	SF	RBD	DRFI	MSS	wPSOSOD	FBC
Time(s)	0.152	0.105	0.121	0.140	2.934	0.968	5.645	2.340

geneous (3rd row). However, the other methods also struggle with these challenging images.

4.4.3 Further Analysis

4.4.3.1 Analysis on Run Time

Table 4.4 shows the running times of the eight SOD methods on the benchmark datasets (e.g. SED1). Timings have been taken on an Intel Xeon E5-1620 3.50GHz with 8GB RAM. In Figure 4.10, although FBC is slower than MSS, GMR, SF, GS, and RBD, it shows higher precision than those methods. In this study, FBC speeds up the previous work (wPSOSOD), and improves the average precision. In Figure 4.10, DRFI shows the highest precision, while FBC is slightly faster than DRFI based on the average running time. Generally, it can be concluded that the new method makes a good balance (trade-off) between computational time and performance. If a task concerns having high performance (or high precision), DRFI would be a good choice. While, FBC is a reasonable choice when the task requires a method which is relatively fast and has good performance.

4.4.3.2 Constructed FG and BG Features Versus Individual Features

Figure 4.11 shows the comparison precision-recall curves among the four features, the foreground feature $\boldsymbol{FG}$, the background feature $\boldsymbol{BG}$, and FBC.

Figure 4.11(a) shows the $\boldsymbol{FG}$ feature has higher precision-recall curve than all the other features $(\boldsymbol{f}_3, \boldsymbol{f}_6, \boldsymbol{f}_7, \boldsymbol{f}_{10})$ and Figure 4.11(b) shows that $\boldsymbol{BG}$ has higher precision-recall curve than $\boldsymbol{f}_6$ and $\boldsymbol{f}_7$. In Figure 4.2(a), it

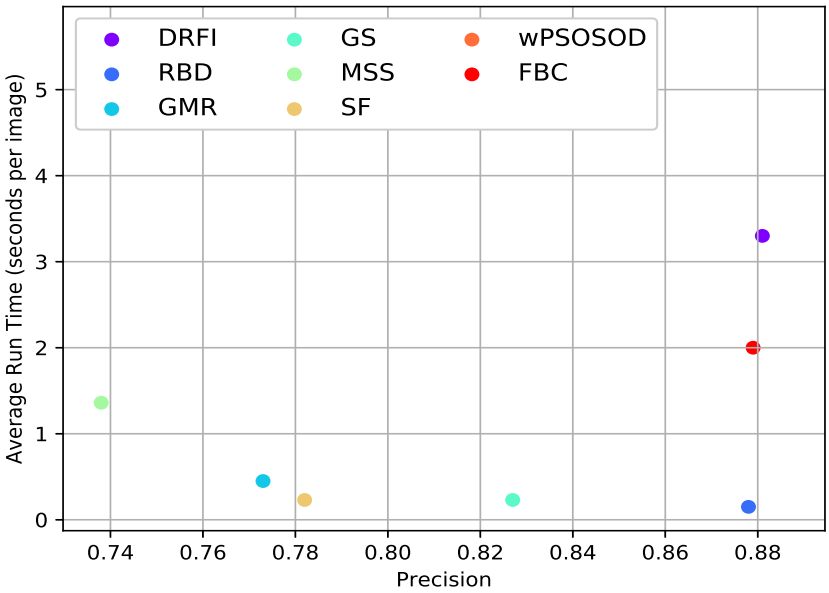


Figure 4.10: Good performing algorithms are supposed to take place in the lower right region of the graph.

can be seen that $\boldsymbol{FG}$ completely highlights the foreground object, while the other features ($\boldsymbol{f}_3, \boldsymbol{f}_6, \boldsymbol{f}_7, \boldsymbol{f}_{10}$) miss some regions of the foreground object. Similarly, $\boldsymbol{BG}$ performs better than $\boldsymbol{f}_6$ and $\boldsymbol{f}_7$ individually on suppressing the background regions as shown in Figure 4.2(b). Thus, the quantitative and qualitative results show that $\boldsymbol{FG}$ and $\boldsymbol{BG}$ are more informative than the individual features.

In Figure 4.11(c), $\boldsymbol{FG}$ has higher precision than $\boldsymbol{BG}$, when recall is within the range [0.7, 1]. While $\boldsymbol{BG}$ has higher precision than $\boldsymbol{FG}$, when the recall is in the range [0, 0.7]. Moreover, $\boldsymbol{FG}$ has higher TP than $\boldsymbol{BG}$, but $\boldsymbol{BG}$ has higher TN than $\boldsymbol{FG}$. The goal is to increase both TP and TN values to increase the precision value. Therefore, the precision will be increased by increasing TP and decreasing FP, and FP will be decreased by increasing TN. FBC combines the discussed strengths characteristics

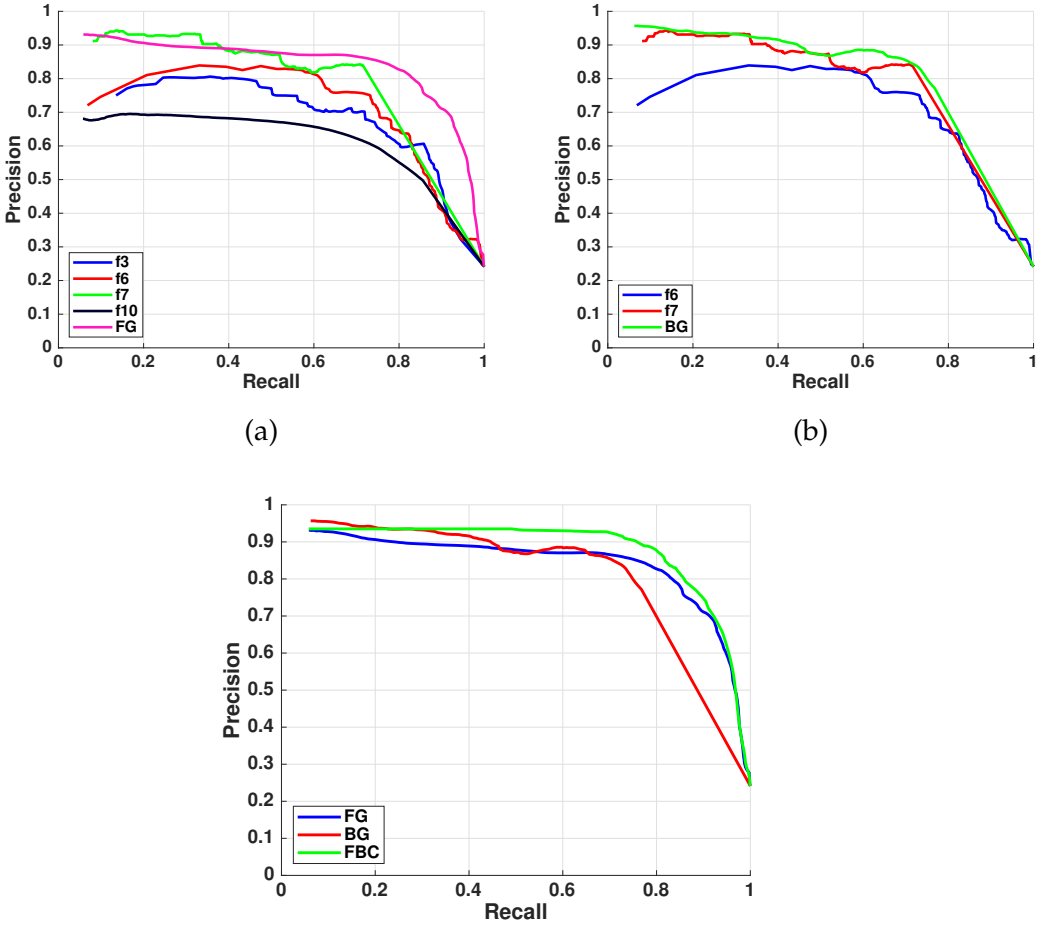


Figure 4.11: Plots show precision-recall curves for (a) $\boldsymbol{f}_3$, $\boldsymbol{f}_6$, $\boldsymbol{f}_7$, $\boldsymbol{f}_{10}$, and $\boldsymbol{FG}$, and (b) $\boldsymbol{f}_6$, $\boldsymbol{f}_7$, and $\boldsymbol{BG}$, and (c) $\boldsymbol{FG}$, $\boldsymbol{BG}$, and FBC.

of both $\boldsymbol{FG}$ and $\boldsymbol{BG}$ to increase the precision. Figure 4.11(c) demonstrates the precision-recall curve of the new method, which is infusion of $\boldsymbol{FG}$ and $\boldsymbol{BG}$, performs better than each individually.

4.5 Chapter Summary

In this chapter, we developed a bottom-up SOD method by constructing two new informative features and designing a combination framework. To focus on the foreground and background regions of the image and utilize their suitable corresponding features, we manually constructed a foreground ($\boldsymbol{FG}$) feature and a background ($\boldsymbol{BG}$) feature that can be used in combination to determine the final image saliency. The convex-hull-based center prior is employed to assign saliency values to the image pixels. This chapter investigated the importance of complementary characteristic of the saliency features and the way of combining those features.

The quantitative and qualitative results presented the effectiveness of the $\boldsymbol{FG}$ and $\boldsymbol{BG}$ features compared to the individual features. The performance of FBC is evaluated in terms of precision-recall, ROC, and F-measure using three benchmark datasets. The results showed that FBC has the potential to achieve a good performance across a wide range of images, which outperforms six benchmark methods.

However, the FBC method still has difficulties in accurately detecting salient objects in challenging images such as images with complex background or when a salient object is not homogeneous. Meanwhile, both feature construction and feature combination tasks have been manually designed, while ideally these tasks could be developed automatically. Hence, we will investigate automatically constructing foreground and background features in the next chapter.

Chapter 5

Automatically Constructing Foreground and Background Features using GP

5.1 Introduction

Over the last two decades, domain experts have designed various types of saliency features. Employing the existing primitive features and constructing new informative features would be helpful to enrich the existing feature set. In fact, powerful and informative constructed features can effectively contribute to the performance of the SOD method. For example, Liu et al. [113] developed three well-known saliency features including local, regional, and global features. However, their proposed method loses its performance in some challenging images due to the lack of enough informative features and a suitable combination method [50].

In the previous chapter, we *manually* constructed foreground and background saliency features, and designed a combination framework to produce the final saliency map. The newly constructed features were more informative than the individual features and have enhanced the performance of SOD. However, the feature construction process has been manu-

ally performed based on the domain knowledge and human intervention. Manually selecting features from the existing features and constructing new features is not an efficient approach nor scalable and is not guarantee to be optimal. Moreover, the manual feature construction process has difficulties, since the large search space makes it difficult to explore and find an optimal combination of the features.

The aforementioned issues motivate us to develop a method which can automatically explore the set of different features, select informative ones, consider their complementary characteristics and construct new features. We develop a GP-based automatic feature construction method to address the issues due to the approach's flexibility.

5.1.1 Chapter Goals

The overall goal of this chapter is to develop a GP-based method to automatically construct two informative features. For the two proposed GP methods, a suitable function set and two new fitness functions are designed to enable GP to effectively explore the search space. To the best of our knowledge, this study will be the first work using GP to automatically construct features in SOD. Specifically, we aim to fulfill the following objectives:

- Develop a GP-based method to automatically construct *foreground* feature that mainly targets detecting and highlighting the foreground object;
- Develop a GP-based method to automatically construct *background* feature which focuses on suppressing background;
- Investigate whether the automatically constructed features perform better than the manually constructed ones; and
- Compare performance of the proposed GP-based method and seven other SOD methods based on four benchmark datasets.

5.1.2 Chapter Organization

The remainder of the chapter is organized as follows. Section 5.2 details the proposed method. Section 5.3 provides the experiment design. Section 5.4 presents and discusses the results. Section 5.5 provides a summary of this chapter.

5.2 GP-based Feature Construction Method

5.2.1 The Overall Algorithm

In this chapter, the overall process of the proposed method is to develop two GP-based feature construction methods to automatically construct foreground ($\boldsymbol{GPFG}$) and background ($\boldsymbol{GPBG}$) saliency features and then combine these two features using the combination framework designed in Section 4.2.3 (on page 115). Despite the $\boldsymbol{FG}$ and $\boldsymbol{BG}$ features (manually constructed features), we employ GP to find a good combination of the predefined features to construct the $\boldsymbol{GPFG}$ and $\boldsymbol{GPBG}$ features. The overall process of the proposed method is depicted in Figure 5.1.

5.2.1.1 GP for Foreground and Background Feature Construction (GPFBC)

For the foreground feature, GP aims to formulate the $\boldsymbol{GPFG}$ feature to perform well at identifying foreground regions. Here, GP takes a predefined feature set as input (terminal) and constructs the $\boldsymbol{GPFG}$ feature as output which is a combination (mathematical expression) of different features. For this purpose, GP needs to employ a suitable fitness function to guide it to highlight the foreground objects.

For the background feature, GP formulates the $\boldsymbol{GPBG}$ feature to identify the foreground in a way that suppresses the background. This method

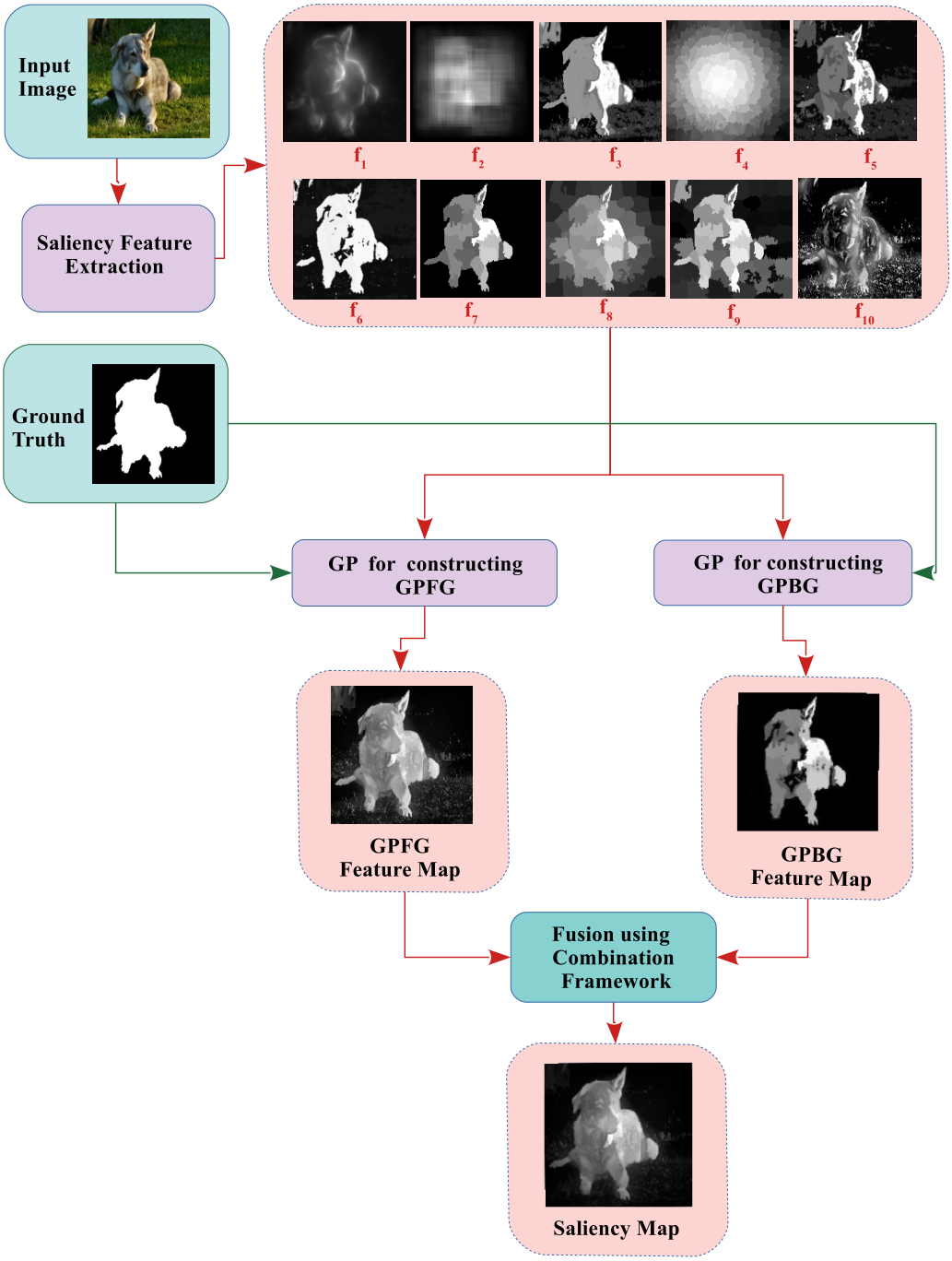


Figure 5.1: The overall scheme of the proposed system.

mainly targets the background regions to decrease the chance of being highlighted. Here, we design different fitness functions from the one used in constructing $\boldsymbol{GPFG}$, since $\boldsymbol{GPBG}$ operates as a complementary feature for the $\boldsymbol{GPFG}$ feature and focuses on the background. To construct $\boldsymbol{GPBG}$, GP may have a similar structure to the $\boldsymbol{GPFG}$ feature, but it has different evaluation strategy.

5.2.1.2 Feature Combination

The next step is to combine the constructed $\boldsymbol{GPFG}$ and $\boldsymbol{GPBG}$ features to produce the final saliency map. Here, we employ the previous feature combination framework which is designed in Chapter 4. This framework is used to assign each pixel the most likely value of being foreground or background. The saliency values for inner and outer region pixels are computed as

$$\boldsymbol{S}(x_p^I, y_p^I) = \boldsymbol{P}c(x_p^I, y_p^I) \times \boldsymbol{GPFG}(x_p^I, y_p^I) \tag{5.1}$$

$$\begin{aligned}\boldsymbol{S}(x_p^O, y_p^O) = \boldsymbol{P}_c(x_p^O, y_p^O) \times \boldsymbol{GPFG}(x_p^O, y_p^O) + \\ (1 - \boldsymbol{P}_c(x_p^O, y_p^O)) \times \boldsymbol{GPBG}(x_p^O, y_p^O)\end{aligned} \tag{5.2}$$

5.2.2 Function Set

Here, GP uses a simple set of arithmetic operations including addition, subtraction, and multiplication. In the function set $\{+, -, \times\}$, each one takes two feature maps as input in 2D-array and returns a feature map as output in 2D-array. We do not include $\div$ operator in the functions set, since $\div$ operator can behave opposite of the other operators and it does not help the system in producing the correct results. For example, if we assume that the saliency values in range [0,1], the division of the two non-salient pixels, 0.1 and 0.1 from two different inputs will be a salient pixel (1) in the output.

5.2.3 Terminal Set

The terminal set of the GP method includes ten feature maps that are summarized in Table 2.2 (on page 51). Each feature is a 2D-array map.

5.2.4 Fitness Functions

5.2.4.1 Fitness Function for $\boldsymbol{GPFG}$

The $\boldsymbol{GPFG}$ feature aims to perform well in highlighting the foreground objects. To achieve this goal, GP needs to increase the number of foreground pixels which are correctly detected as foreground. Here, we employ cross entropy as our fitness function to enhance precision of salient regions by decreasing the difference between the constructed feature and the ground truth. To compute the fitness value for each GP solution (individual), we take the average of the entropy values over all training images. The solutions with lower cross entropy values/lower fitness values are assumed to be better solutions compared to other solutions and they will have higher chance to contribute to the next generation.

$$Fitness_1 = \frac{1}{n}\sum_{i=1}^{n} H(\boldsymbol{G}_i, \boldsymbol{S}_i) \tag{5.3}$$

where n is the number of images in the training set, $\boldsymbol{S}_i$ is the output saliency feature map of the i^{th} image and G_i is the ground truth of the i^{th} image. The cross entropy is defined as

$$H(\boldsymbol{G}_i, \boldsymbol{S}_i) = -\sum_{p=1}^{pn} \boldsymbol{G}_p log \boldsymbol{S}_p \tag{5.4}$$

where pn is the number of pixels in the image. $\boldsymbol{G}_p$ is the ground truth value of the pixel p and $\boldsymbol{S}_p$ is the saliency value of the pixel p.

5.2.4.2 Fitness Function for $GPBG$

The $GPBG$ feature focus mostly on suppressing the background regions. Here, accuracy as a fitness function shows promising performance in guiding GP to construct the $GPBG$ feature.

In the majority of images in the SOD datasets such as PASCAL, foreground and background pixels are not equally distributed into two classes (foreground and background), and the number of background pixels are often larger than the foreground pixels [103]. As we aim $GPBG$ to be more accurate on the background regions, we use the biased characteristic of accuracy for binary classification in unbalanced data as an advantage for evaluating the evolved GP solutions. In the case of having a large number of background pixels from the total, increasing only TN will result a high accuracy. Hence, the function can guide the algorithm by giving high fitness values in the case of suppressing more background pixels.

To compute accuracy, we need to binarize the output of GP which is a saliency map $\boldsymbol{S}$ containing continuous values. The binarized mask $\boldsymbol{M}$ of the GP output is computed by employing the adaptive threshold described in Section 3.2.5.1 (on page 87). After obtaining the binarized mask, we compare the mask $\boldsymbol{M}$ with the corresponding ground truth $\boldsymbol{G}$ of the image. To compute the fitness value of each GP solution, we take the average of accuracy values over all training images which is computed as

$$Fitness_2 = \frac{1}{n}\sum_{i=1}^{n} Accuracy(\boldsymbol{G}_i, \boldsymbol{M}_i) \tag{5.5}$$

$$Accuracy(\boldsymbol{G}_i, \boldsymbol{M}_i) = \frac{TP + TN}{TP + TN + FP + FN} \tag{5.6}$$

5.3 Experiment Design

5.3.1 Datasets

In this work, we use the SOD datasets described in Section 2.9 (on page 71). In this chapter, we add the PASCAL dataset [101] to our previous three datasets. The reason to use the PASCAL dataset was to assess the performance of GPFBC over scenes with multiple objects with high background clutter. The details of the datasets can be found in Section 2.9 (on page 71). Each dataset is randomly split into three parts: a training set (60%), a validation set (20%) and a test set (20%). In Chapter 3 (on page 89), the datasets are randomly divided into two sets, training set and test set which contain 70% and 30% of the image dataset, respectively. However, in this chapter, we use a part of datasets as validation set to provide an unbiased evaluation of a model fit on the training dataset, therefore the test set is different.

5.3.2 Benchmark Methods for Comparisons

The proposed method is compared to the six SOD methods which have been presented in Section 4.3.2 (on page 118) and the FBC method from Chapter 4.

5.3.3 Parameter Settings

Similar to the other algorithms, GP has a number of parameters that required to be set. The GP parameters include, the initialization method, population size, minimum and maximum depth of GP individuals, terminating criteria, GP run numbers, mutation and crossover rates, and individual selection method. Table 5.1 gives a summary for the GP parameters. We use similar parameter values in constructing $\boldsymbol{GPFG}$ and $\boldsymbol{GPBG}$. The initial population of GP is generated by the ramped half-

Table 5.1: GP parameters.

Parameter	Value	Parameter	Value
Population Size	100	Generations	51
Minimum Depth	2	Maximum Depth	4
Mutation Rate	0.20	Crossover Rate	0.80
Elitism	Keep the single best	Selection Type	Tournament
Initial Population	Half-and-half	Size of Tournament	3

and-half method. The population size is set to 100 individuals. Increasing the population size beyond 100 individuals resulted in slowing down the convergence rate without any significant increase in performance. The minimum and maximum tree depth are set to 2 and 4, respectively. Since the input feature set of GP contains a small number of features (10 features), GP does not necessarily need a larger program size. The criterion for terminating the evolutionary process is the maximum number of generations which is set to 50. Running the evolutionary process for more than 50 generations did not improve the performance. Here, the mutation and crossover rates are set to 20% and 80%, respectively. The best evolved program is kept to prevent the performance of the subsequent generations from degrading. The tournament selection method is used for selecting individuals for the mating process and the tournament size is set to 3. In this chapter, the tournament size is small, since the population size is only 100.

In this work, on each dataset, GP is run 30 times with different random seeds. The best result is reported as the final result.

5.3.4 Evaluation Metrics

The performance of the GPFBC method is evaluated using the similar evaluation criteria in Section 3.3.4 (on page 91).

5.4 Results and Discussions

5.4.1 Quantitative Comparisons

5.4.1.1 The SED1 Dataset

As shown in Figures 5.2(a) and (b), after DRFI, GPFBC outperforms FBC, SF, MSS, and GS, and it has a comparable result with RBD on the SED1 dataset.

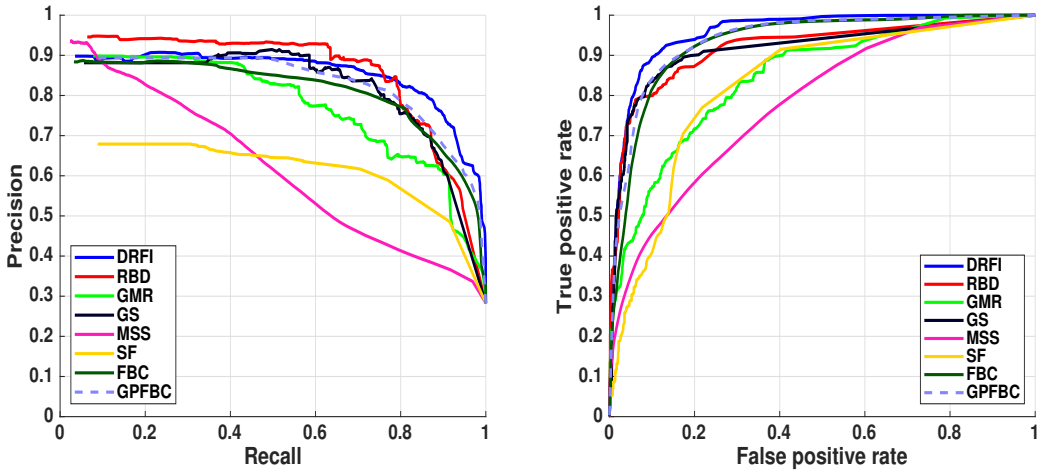
Based on Figure 5.2(c) and Table 5.2 (SED1), RBD has the highest average precision and F-measure values, 0.8884 and 0.8387, respectively. As discussed before, to compute F-measure, we weight precision more by following the literature [113], hence, F-measure would be high when the precision has high value. Unlike precision and F-measure, recall value of 0.707 for RBD is low, which means it is not performing good enough on identifying foreground regions on the SED1 dataset. Generally, after DRFI and RBD, GPFBC reported good performance on precision, recall, and F-measure, 0.8041, 0.7833, and 0.7992, respectively.

As shown in Table 5.3 GPFBC has lower AUCPR value than DRFI and RBD and comparable to GS, but it performs better than others.

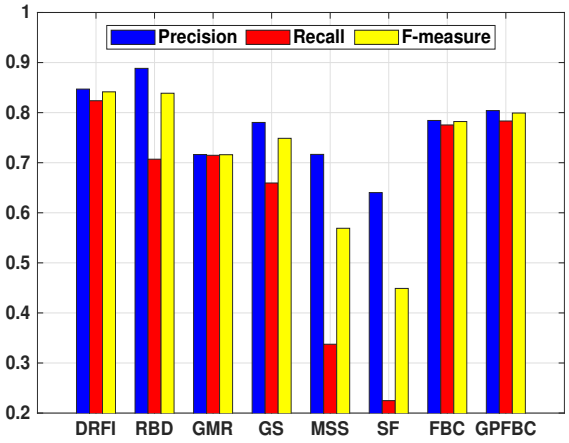
5.4.1.2 The ASD Dataset

In Figure 5.3(a), although GPFBC performs slightly lower than RBD and DRFI, it outperforms the SF, GMR, MSS, GS, and FBC methods on the ASD dataset based on the precision-recall curves. Based on the ROC curves in Figure 5.3(b), the ROC curve of the GPFBC method is the third best curve on ASD.

As shown in Figure 5.3(c) and Table 5.2 (ASD), DRFI has the highest average precision, recall and F-measure values, 0.9028, 0.9075, and 0.9039, respectively. After DRFI and RBD, GPFBC performs better than FBC with the results of 0.8563, 0.8623, and 0.8577 for average precision, recall and F-measure, respectively, while FBC reports 0.8323, 0.8359, and 0.8331, re-



(a) Precision-recall curves (b) ROC curves



(c) Average precision, recall, and F-measure

Figure 5.2: The performance of GPFBC compared to the seven other SOD methods based on the **SED1** dataset.

spectively. Figure 5.3(c) shows that DRFI, RBD, FBC, GS, and GPFBC have slightly higher recall than precision, this can be due to simple nature of the ASD dataset for foreground objects.

On the ASD dataset in Table 5.3, GPFBC can significantly outperforms

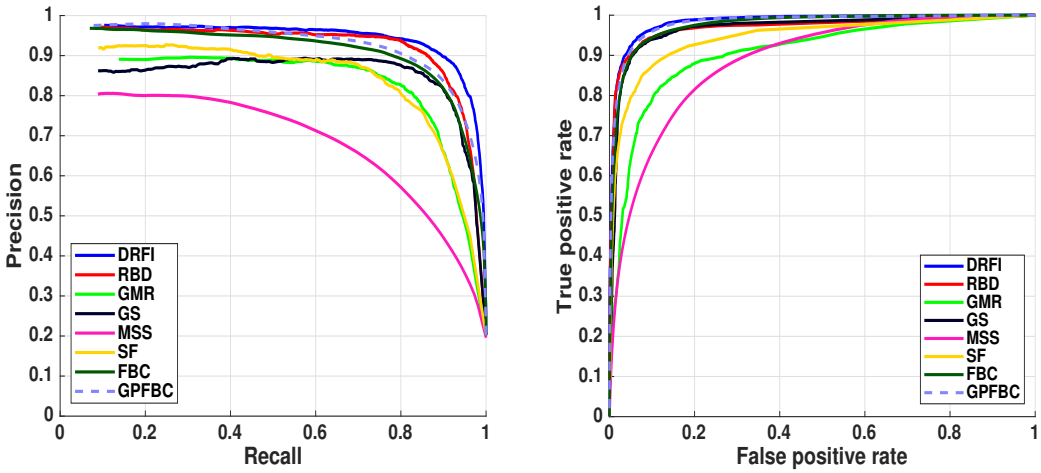
Table 5.2: Quantitative results of GPFBC and the seven other SOD methods based on average precision, recall, and F-measure values on the **SED1**, **ASD**, **ECSSD**, and **PASCAl** datastes. The abbreviations P, R, and F indicate precision, recall, and F-measure, respectively.

Dataset	Method	P	R	F	Dataset	P	R	F
SED1	DRFI	0.847	0 .8238	0.8415	ECSSD	0.7923	0.7161	0.7733
	RBD	0.8884	0.707	0.8387		0.7191	0.6522	0.7025
	GMR	0.7163	0.7148	0.7116		0.6611	0.4009	0.575
	GS	0.7804	0.6595	0.7487		0.6551	0.6721	0.6589
	SF	0.6403	0.2249	0.4489		0.6348	0.4112	0.564
	MSS	0.7166	0.3375	0.5691		0.5415	0.4021	0.5014
	FBC	0.7843	0.7754	0.7822		0.6792	0.7513	0.6946
	GPFBC	0.8041	0.7833	0.7992		0.7296	0.6843	0.7211
ASD	DRFI	0.9028	0.9075	0.9039	PASCAL	0.7514	0 .6736	0.7319
	RBD	0.8746	0.8803	0.8759		0.6634	0.557	0.6353
	GMR	0.8366	0.7286	0.8089		0.5504	0.3029	0.4631
	GS	0.8179	0.8808	0.8316		0.6063	0.5749	0.5987
	SF	0.86	0.7248	0.8245		0.5504	0.3417	0.4824
	MSS	0.6996	0.6029	0.6746		0.5424	0.4068	0.5037
	FBC	0.8323	0.8359	0.8331		0.6373	0.7022	0.6512
	GPFBC	0.8563	0.8623	0.8577		0.6849	0.6197	0.6686

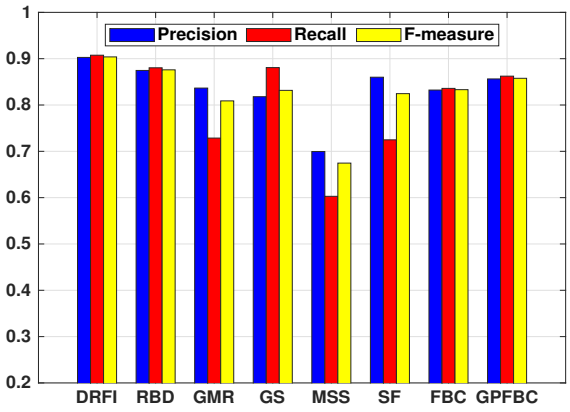
Table 5.3: The statistical comparison of GPFBC and the other seven SOD methods based on AUCPR on the **SED1**, **ASD**, **ECSSD** and **PASCAL** datastes.

	DRFI	RBD	GMR	GS	MSS	SF	FBC	GPFBC
SED1	0.5779 $\downarrow$	0.5630 $\downarrow$	0.4938 $\uparrow$	0.5477	0.3535 $\uparrow$	0.3593 $\uparrow$	0.5323 $\uparrow$	0.5480 $\pm$0.0573
ASD	0.7477 $\downarrow$	0.7339 $\downarrow$	0.6511 $\uparrow$	0.6840 $\uparrow$	0.5123 $\uparrow$	0.6566 $\uparrow$	0.7119 $\uparrow$	0.7256 $\pm$0.0112
ECSSD	0.5899 $\downarrow$	0.5229	0.4352 $\uparrow$	0.4766 $\uparrow$	0.2960 $\uparrow$	0.3875 $\uparrow$	0.4900 $\uparrow$	0.5265 $\pm$0.0213
PASCAL	0.5294 $\downarrow$	0.4388 $\uparrow$	0.3397 $\uparrow$	0.3901 $\uparrow$	0.2803 $\uparrow$	0.2953 $\uparrow$	0.4351 $\uparrow$	0.4480 $\pm$0.0193

some of benchmark methods such as FBC, GS, SF, MSS, GMR based on statistical t-test at the significance level 5%.



(a) Precision-recall curves (b) ROC curves

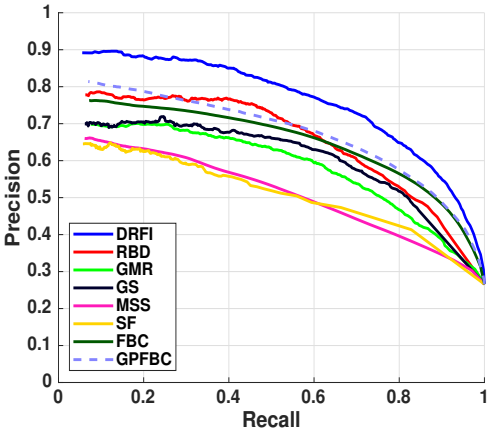


(c) Average precision, recall, and F-measure

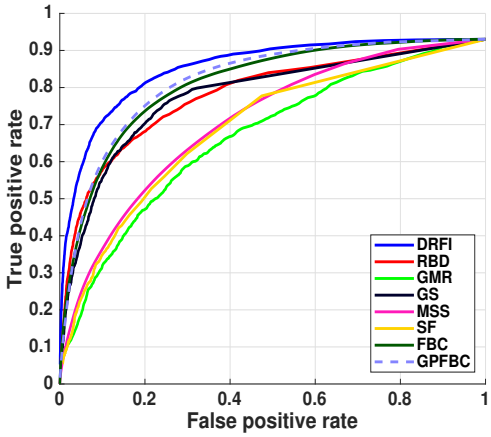
Figure 5.3: The performance of GPFBC compared to the seven other SOD methods based on the **ASD** dataset.

5.4.1.3 The ECSSD Dataset

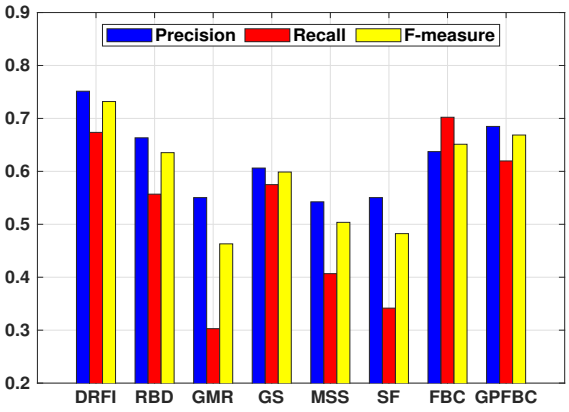
As shown in Figure 5.4(a), after DRFI, GPFBC performs better than SF, GMR, MSS, and GS, and FBC. Based on the ROC curves in Figure 5.4(b), GPFBC has the second best AUC value after DRFI.



(a) Precision-recall curves



(b) ROC curves



(c) Average precision, recall, and F-measure

Figure 5.4: The performance of GPFBC compared to the seven other SOD methods based on the **ECSSD** dataset.

On the ECSSD dataset in Figure 5.4(c) and Table 5.2, GPFBC performs as the second good performing method with values of 0.7296 and 0.7211 for average precision and F-measure. Although FBC with 0.7513 has the highest recall value, it has lower precision value compared to DRFI, RBD, and GPFBC. FBC loses its performance when the background is complex such as images in the ECSSD dataset.

In terms of statistical significance t-test on the ECSSD dataset in Table 5.3 (ECSSD), GPFBC has only lower AUCPR value than DRFI but better AUCPR value than the other SOD methods and it significantly outperforms the other methods.

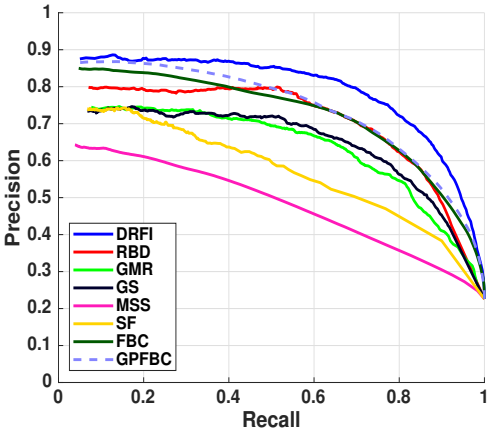
5.4.1.4 The PASCAL Dataset

On the PASCAL dataset in Figures 5.5(a) and (b), GPFBC has lower performance than DRFI, but better compared to SF, GMR, MSS, and GS, RBD, and FBC.

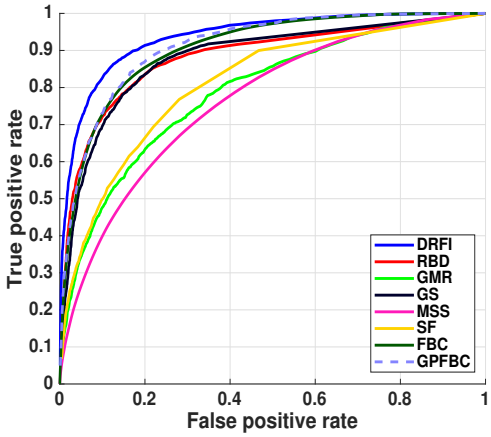
In Figure 5.5(c) and Table 5.2 (PASCAL), DRFI has the highest precision and F-measure values, 0.7514 and 0.7319, then GPFBC with 0.6849 and 0.6686 for precision and F-measure performs better than than other SOD methods. Similar to the ECSSD dataset, FBC has the highest recall and lower precision, as the PASCAl dataset also contains images with complex background.

Based on Table 5.2, the GPFBC method has slightly improved the results compared to the FBC method considering quantitative results. However, similar to FBC, it still has some difficulties on complex datasets such as ECSSD and PASCAL. Based on the observations, this is mostly caused by employing the same input saliency feature set and lack of enough informative saliency features in the feature set. Another potential reason is the constructed features may contain some limitations of the input features constructed from.

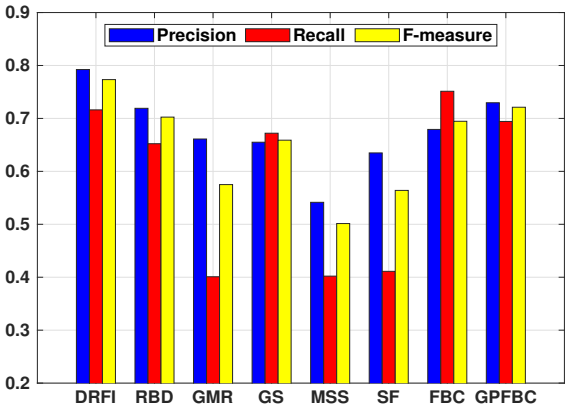
As it can be seen in Table 5.3 (PASCAL), the majority of the SOD methods have lower AUCPR values, since the PASCAL dataset is one of the challenging datasets. Hence, GPFBC reports lower AUCPR value for PASCAL compared to the other datasets.



(a) Precision-recall curves



(b) ROC curves



(c) Average precision, recall, and F-measure

Figure 5.5: The performance of GPFBC compared to the seven other SOD methods based on the **PASCAL** dataset.

5.4.2 Qualitative Comparisons

To show the qualitative performance of GPFBC and the other benchmark methods, some sample saliency maps are shown in Figures 5.6 and 5.7. We demonstrate the performance of GPFBC on the challenging and complex images, e.g., images having non-homogeneous foreground object(s), clut-

tered/complex background regions. Thus, it can be claimed that GPFBC method has a good performance on highlighting the foreground object(s) and suppressing background.

In Figure 5.6, the image in row 5 has two salient objects and the saliency map of GPFBC shows that it can successfully detect both objects on the contrary to RBD, SF, MSS, and GMR. In rows 3, 6, 10, and 13, where images have complex background, GPFBC can mostly suppress the background and properly cover the foreground objects. The images in rows 7 and 8 have non-homogeneous foreground objects, although GPFBC misses some small regions of the foreground objects, it detects the precise location of the salient objects. Although the images in rows 2, 5, and 12 have cluttered background, GPFBC performs good on both foreground and background regions. For the rows 13 and 14, GPFBC performs better at suppressing background compared to the wPSOSOD and FBC methods. In rows 1, 4, and 9, where images have challenging background (e.g. in the 4th row, there is low color contrast between the foreground object and background), GPFBC falsely highlights some small regions of background.

Figure 5.7 gives some examples for the images having complex background and non-homogeneous foreground objects. In the complex image types such as those presented in Figure 5.7, GPFBC has difficulties in completely highlighting the foreground objects and suppressing background specially in the first image.

Although the qualitative results of GPFBC are close to that of FBC and it may have similar false detection, the feature construction task of GPFBC is automatic and it can perform as good as the manual one. However, employing saliency feature containing texture, shape, and location information would be helpful and informative in some challenging image types where foreground object has lower contrast with background and background is cluttered.

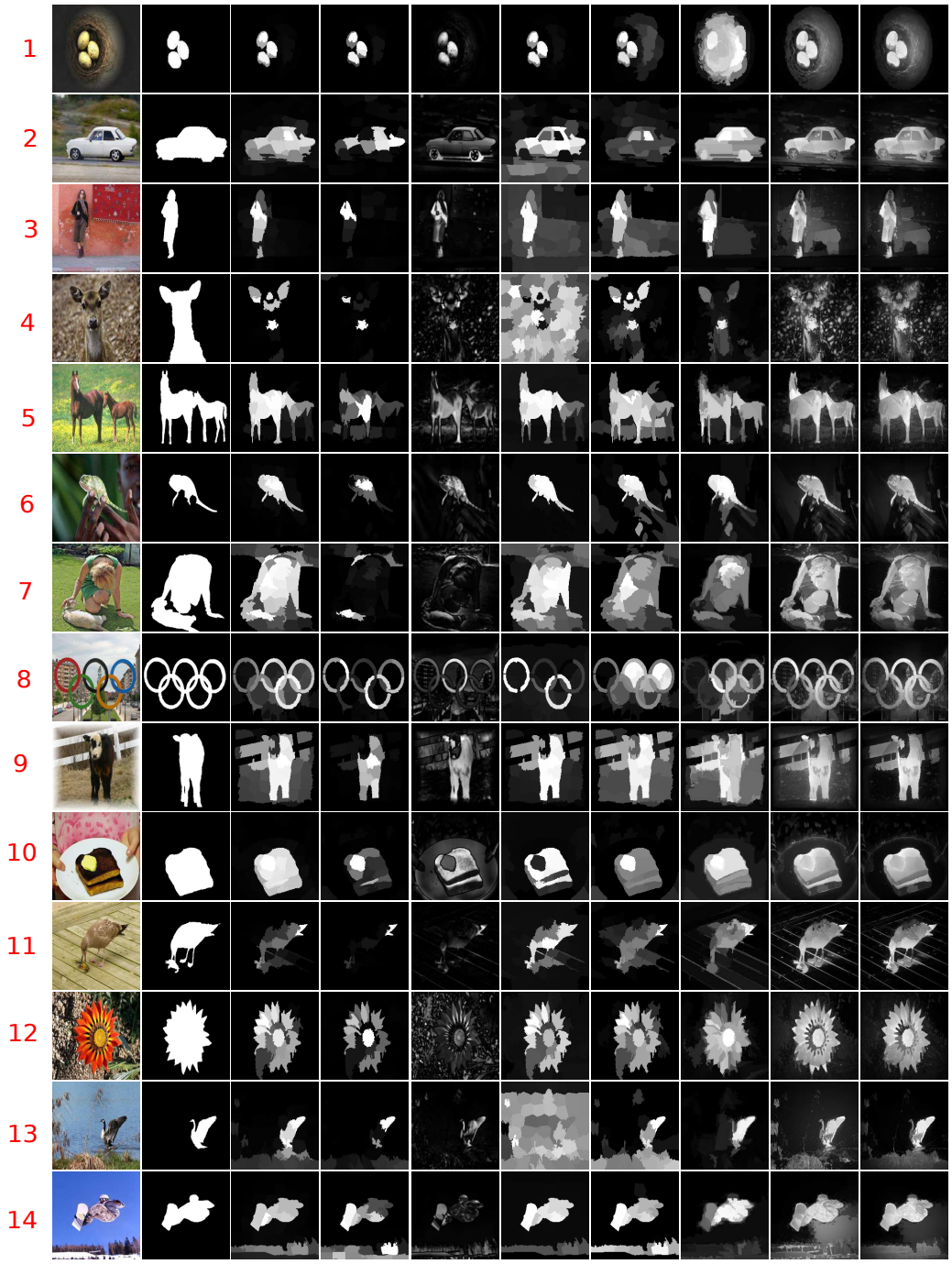
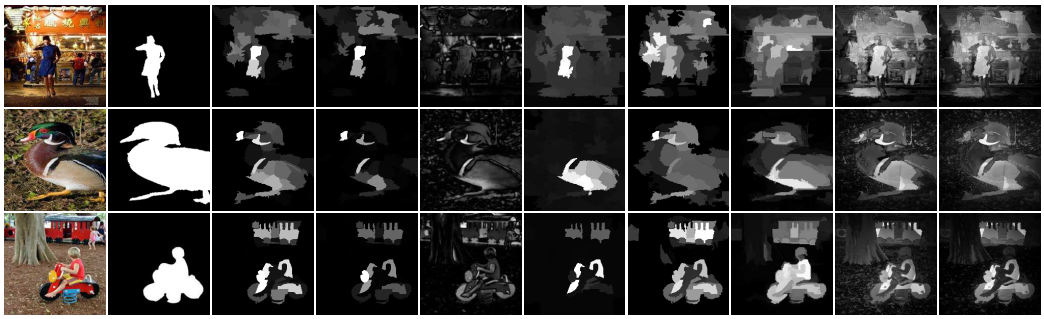


Figure 5.6: Some visual examples of GPFBC and the seven other SOD methods on **SED1**, **ASD**, **ECSSD**, and **PASCAl** datasets.



Original GT RBD SF MSS GMR GS DRFI FBC GPFBC

Figure 5.7: Some failure examples of GPFBC and the seven other SOD methods on **SED1**, **ECSSD** and **PASCAl** datasets.

5.4.3 Further Analysis

Figure 5.8 shows the comparison precision-recall curves between the manually constructed foreground feature $\boldsymbol{FG}$ of the FBC method (Chapter 4) and the automatically constructed foreground feature $\boldsymbol{GPFG}$ of the GPFBC method. The figure shows that $\boldsymbol{GPFG}$ has a higher performance than $\boldsymbol{FG}$ based on precision-recall curve. That indicates that GP has the capability to automatically construct more informative feature than the manually constructed one. The constructed background feature $\boldsymbol{GPBG}$ has similar performance to the manually constructed feature $\boldsymbol{BG}$.

Figure 5.9 shows two evolved solutions or constructed features, $\boldsymbol{GPFG}$ and $\boldsymbol{GPBG}$ by the GPFBC method on the ASD dataset. In Figure 5.9(a), the GP tree takes only two features, $\boldsymbol{f}_6$ and $\boldsymbol{f}_7$, as inputs (terminal) and chooses multiplication operator from the function set to construct the $\boldsymbol{GPBG}$ feature. Figure 5.9(a) is an example to show how the combination of $\boldsymbol{f}_6$ and $\boldsymbol{f}_7$ can decrease false positive (wrongly highlighted background region(s)) in the $\boldsymbol{GPBG}$ feature.

In Figure 5.9(b), the GP tree takes five features, $\boldsymbol{f}_1$, $\boldsymbol{f}_3$, $\boldsymbol{f}_6$, $\boldsymbol{f}_8$, and $\boldsymbol{f}_{10}$, and selects the add operator to construct the $\boldsymbol{GPFG}$ feature. In Figure 5.9(b), the input features are combined in a way to completely high-

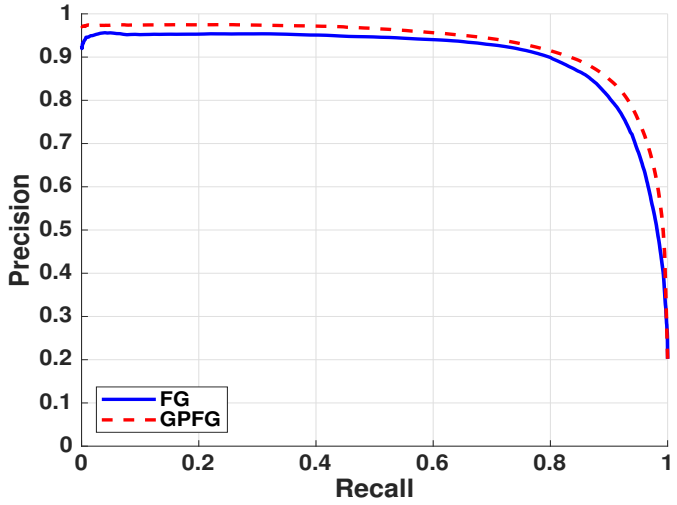
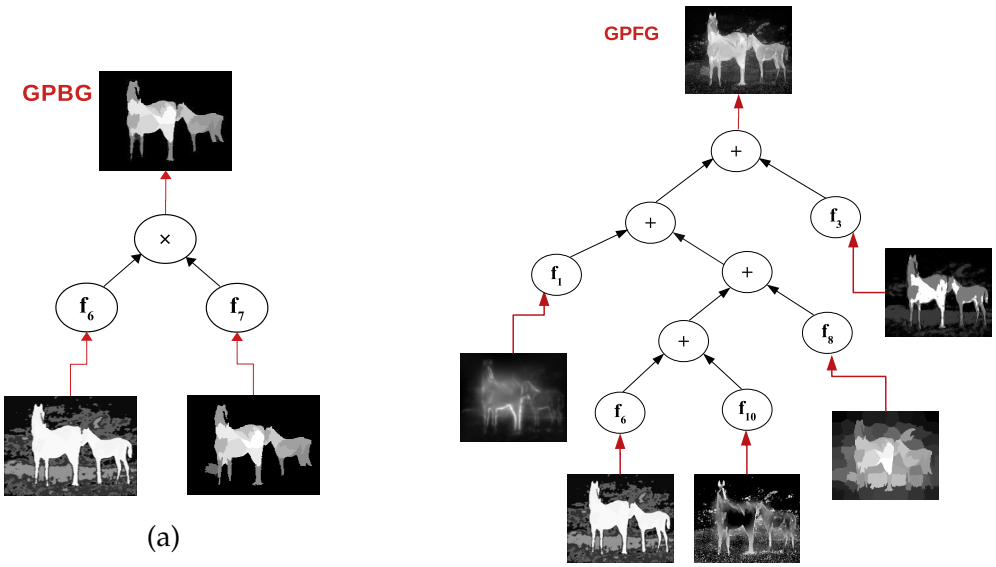


Figure 5.8: Precision-recall curves of $\boldsymbol{FG}$ and $\boldsymbol{GPFG}$ on the **ASD** dataset.



(a) (b)

Figure 5.9: An example of $\boldsymbol{GPBG}$ and $\boldsymbol{GPFG}$ evolved programs by GPFBC on the ASD dataset.

light the foreground objects. $\boldsymbol{f}_1$ and $\boldsymbol{f}_{10}$ help the other features in identifying boundaries and edges. Although $\boldsymbol{f}_8$ performs good on highlighting foreground object completely, it falsely highlights some background regions. As can be seen, by adding $\boldsymbol{f}_3$ to the combination, falsely highlighted background regions are mostly suppressed. As mentioned before, $\boldsymbol{GPFG}$ is required to be informative toward background, since it will help $\boldsymbol{GPBG}$ in highlighting some foreground regions placed in the outer region.

It can be concluded that GP has the ability to select the saliency features which complement each other well. In the $\boldsymbol{GPBG}$ feature, the two selected features complement each other, since they help each other to completely suppress background in $\boldsymbol{GPBG}$. For the $\boldsymbol{GPFG}$ feature, the five selected features tend to complement each other to highlight the foreground objects.

5.5 Chapter Summary

In this chapter, we developed a GP-based method to automatically construct foreground $\boldsymbol{GPFG}$ and background $\boldsymbol{GPBG}$ saliency features. We focused on addressing the limitations of manually constructing features in the FBC method from Chapter 4. The new proposed method automated the feature construction task by employing GP. Unlike the FBC method, GPFBC is robust towards the changes in the input feature set, since GP can be used with various saliency features. In contrast to FBC, GPFBC does not require domain knowledge and human intervention. Moreover, GPFBC improves the SOD performance by introducing more informative saliency features. It can be concluded that GP has a promising capability for exploring a search area of saliency features and finding a suitable way to combine those features.

The GPFBC method contains two phases, feature construction and feature combination. In this chapter, we fulfilled the goal of having an auto-

matic feature construction task, but the feature combination task must still been done manually. The next step is to automate the feature combination task as well.

As mentioned before, this chapter employed the same input feature set of the previous chapter to allow fair comparisons. However, employing a large feature set has some advantages. Therefore, we will investigate the following concepts in the next chapter. First, whether GP can cope with a large search space; second, whether involving diverse and informative features can improve the SOD performance.

Chapter 6

GP for Automatically Feature Selection and Combination in SOD

6.1 Introduction

In SOD, one challenge is to detect saliency in images with dramatic variations (e.g. containing a complex background, non-homogeneous foreground object), which requires effective feature sets to capture the distinctive information between the foreground object(s) and background. As mentioned in the previous chapter, employing more informative and various features might be expected to enhance the power of the SOD method to do more precise prediction for challenging images. Therefore, in this chapter, we will add different types of saliency features and enrich the employed feature set of the previous chapter. However, increasing the size of the feature space will cause some difficulties such as increasing the complexity of the feature space, leading to increased feature interaction and computational expense. Therefore, a suitable feature pre-selection method is required to simplify the algorithm. An exhaustive search for the best feature subset of a given dataset is practically infeasible in most

situations.

The feature combination stage is one of the fundamental stages in SOD for generating the final saliency map [25]. In this regard, a few studies attempt to address the feature combination problem by finding the optimal values for the weights in linear combination. For example, Liu et al. [113] employed the conditional random field (CRF) framework to learn the linear combination weights for different features. In Chapter 3, PSO is utilized to learn a suitable weight vector for the saliency features and linearly combine the weighted features. Due to the highly nonlinearity of the visual attention mechanism, the above linear mapping might not perfectly capture the characteristics of feature combination of human visual system. Consequently, nonlinear methods are required to fuse features to achieve higher performance on different image types. Moreover, some studies [113, 138, 186, 203] combined features heuristically to generate the final saliency map, and they do not perform well on challenging images.

In the majority of existing SOD methods [76,105,201], selecting features and designing a combination framework have been manually done by domain experts. In this scenario, the feature selection and combination tasks are highly dependent on domain knowledge and human intervention.

In this chapter, an embedded GP-based method is developed to do feature selection and combination tasks to produce the final saliency map.

6.1.1 Chapter Goals

This chapter aims to develop a GP-based method for SOD to automatically select features from different level and scales, and combine those selected features to produce the final saliency map.

- Develop a new automatic GP-based feature selection and combination method for SOD;
- Formulate an appropriate fitness function to measure the difference

between probability distribution of the GP-based produced saliency map and probability distribution of the ground truth;

- Evaluate the proposed GP method using datasets of varying difficulties to test the generalizability property of this method;
- explore whether the proposed method can select effective feature subsets for complex image types; and
- Compare the performance of the proposed method to that of six hand-crafted SOD methods to test whether those automatically evolved programs have the potential to achieve better or comparable performance to the domain-expert designed ones.

6.1.2 Chapter Organization

The remainder of the chapter is organized as follows. Section 6.2 details the proposed method. Section 6.3 provides the experiment design. Section 6.4 presents and discusses the results. Section 6.5 summarises this chapter.

6.2 GP-based SOD Method

6.2.1 The Overall Algorithm

This section describes the proposed GP-based method to automatically select and combine features (shortly called GPFSFC) to produce the final saliency map of the given image. In the GPFSFC method, GP takes a large set of primitive features from different categories as input (terminal) and produces the saliency map as output which is a combination (mathematical expression) of different features. The overall structure is depicted in Figure 6.1. For the training stage, first, different image segmentation-levels are computed for each image in the training set, then saliency features are extracted from the segmented images. Second, the feature set and ground

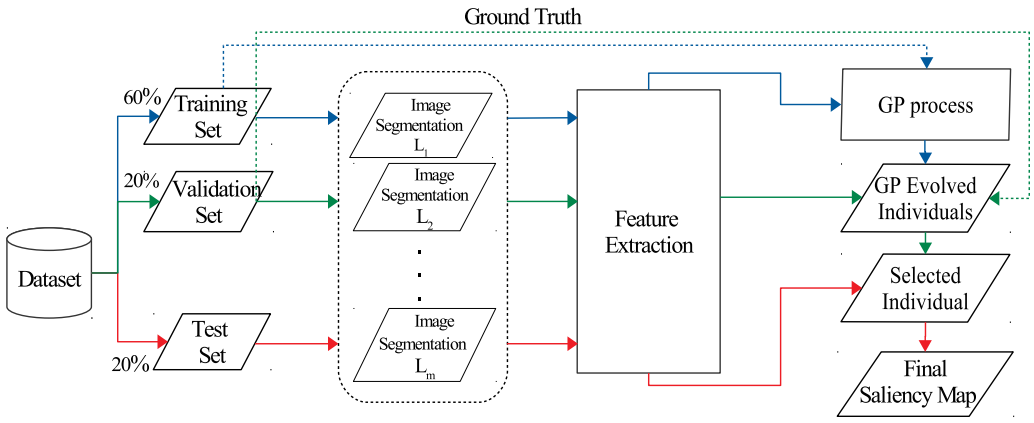


Figure 6.1: The overall algorithm of GPFSFC.

truth are fed into GP. Third, the GP process generates and evaluates the GP programs. Finally, the GP process results 50 best individuals each of which evolved in one generation (total 50 generations). For the validation stage, after completing the segmentation and feature extraction parts, the feature set and ground truth of the validation images are used to select the best individual from the evolved GP individuals. For the test stage, for a given test image, similar to the training and validation stages, multi-level image segmentation and feature extraction are computed. Then, saliency maps are produced by employing the selected GP individual.

6.2.2 Function Set

The function set is made up from three arithmetic operators, one trigonometric function and one conditional function, which are $\{+, -, \times, \sin, \mathit{if}\}$. The first three arithmetic operators and the trigonometric operator have their regular meaning, and the *if* operator takes three input arguments and returns the second argument if the first is less than zero; otherwise, it returns the third argument.

6.2.3 Terminal Set

To provide the terminal set for the GP process, the following preprocessing steps are employed. Firstly, for a given image i, a set of m-level segmentations $L = \{L_1, L_2, ..., L_m\}$ is computed, each segmentation is a decomposition of the image i as shown in Figure 6.2. Here, the graph-based image segmentation method [43] is employed to generate multiple segmentations using m groups of different parameters. In this study, m is set to 48 by following [76]. Secondly, each region of a segmentation level is represented by D-dimensional feature vector. D is set to 103 (10+93) by collecting 10 hand-crafted saliency features and 93 features from the DRFI method [76] (see Section 2.7.3.1 on page 52). Therefore, we will have four saliency feature groups, $\boldsymbol{fg} = \{\boldsymbol{fg}^1, \boldsymbol{fg}^2, \boldsymbol{fg}^3, \boldsymbol{fg}^4\}$. These feature groups are regional contrast ($\boldsymbol{fg}^1 = \{f_1^1, f_2^1, ..., f_{29}^1\}$), regional backgroundness ($\boldsymbol{fg}^2 = \{f_1^2, f_2^2, ..., f_{29}^2\}$), regional property ($\boldsymbol{fg}^3 = \{f_1^3, f_2^3, ..., f_{35}^3\}$), and hand-crafted ($\boldsymbol{fg}^4 = \{f_1^4, f_2^4, ..., f_{10}^4\}$). Table 2.2 in Section 2.2 (on page 51) gives more detail about the four groups of those saliency features.

The employed feature set in the previous chapters (Chapters 3, 4, and 5) is mostly based on the color features extracted from the different scales (such as pixel, superpixel, and cluster). Although the created SOD methods based on those features generally have good performance, they have limitations in the challenging image types. One potential reason for this problem is the employed feature set has lack of important information regarding different aspects of salient and background regions such as texture information of the regions, and generic properties of the regions including appearance and geometric.

To fulfill the mentioned requirements, we utilize saliency features from three different categories, regional contrast, regional backgroundness, and regional property, which are proposed in [76]. These three categories mostly cover the properties of regions for the purpose of identifying salient and background regions.

Since a large number of saliency features have been introduced in the

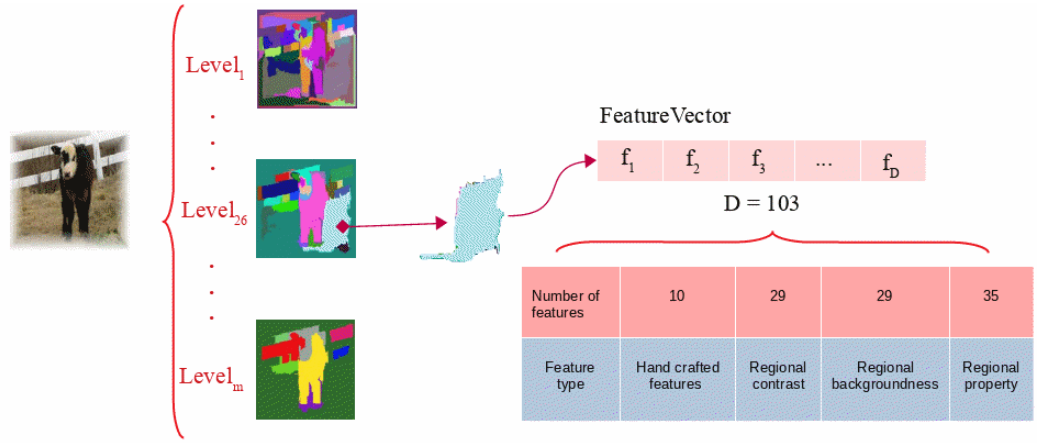


Figure 6.2: Different segmentation levels and feature groups.

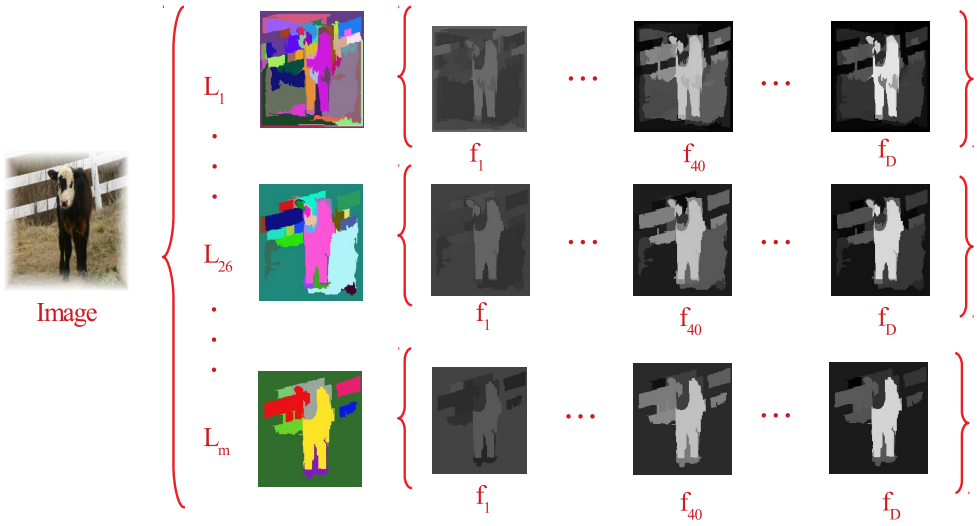


Figure 6.3: Feature extraction from different segmentation levels.

literature, it is worthwhile to develop an automatic method which can explore the large search space of different features. In this study, we provided a wide range of features to investigate how GP can cope with. Figure 6.3 demonstrates visualizations of different segmentation levels and saliency features belong to each level.

6.2.4 Fitness function

In this study, Kullback-Leibler (KL) divergence [91] is employed as a fitness function. KL is employed for measuring the difference between two probability distributions. In the context of saliency, KL is used to measure the distance between distributions of saliency values [73,74]. Borji et al. [30] provided detailed explanation regarding KL divergence metric and discussed that the saliency prediction methods which show good performance on saliency detection exhibit higher KL divergence.

We use KL divergence as the fitness function to measure the difference between the probability distribution of the GP-based produced saliency map and the probability distribution of the ground truth. This function is a minimization function. The average of KL values on the training images is used to compute the fitness value for each GP individual. To compute KL divergence, we apply softmax on both ground truth and GP output of each image to obtain the distribution probabilities. The fitness value is formulated as:

$$Fitness_s = \frac{1}{n}\sum_{i=1}^{n} D_{KL}(\boldsymbol{G}_i||\boldsymbol{S}_i) \tag{6.1}$$

where, n presents the number of the training images, $\boldsymbol{S}_i$ presents the output saliency map computed using GP on the i^{th} image and $\boldsymbol{G}_i$ presents the ground truth of the i^{th} image. Hence, the KL divergence is helpful in guiding GP to generate outputs similar to their ground truth. KL is computed as

$$D_{KL}(\boldsymbol{G}||\boldsymbol{S}) = \sum_{rg\in R} \boldsymbol{G}(rg)\frac{\ln \boldsymbol{G}(rg)}{\boldsymbol{S}(rg)} \tag{6.2}$$

where rg presents a region from the segmented image. $\boldsymbol{G}(rg)$ is the saliency distribution of the region rg in the ground truth and $\boldsymbol{S}(rg)$ denotes the saliency distribution of the region rg in the GP output/saliency map.

6.3 Experiment Design

6.3.1 Datasets

In this work, the performance of GPFSFC is evaluated using the same datasets and setting which employed in Section 5.3.1 (on page 138).

6.3.2 Benchmark Methods for Comparisons

The proposed method is compared to the similar methods from Section 5.3.2 (on page 138) and we also add DCNN [64] to the list of comparison methods. DCNN is a deep learning based state-of-the-art method which automatically learn from pixels and perform non-linear transformations for SOD. Thus, we can investigate how the GP-based method performs compared to the CNN-based SOD method.

6.3.3 Parameter Settings

GP has a number of parameters which can be altered for a given problem. In this study, the initial population is created by ramped half-and-half method. Here, the population size is 300 which is larger than the previous chapter (100 individual). The reason is that the size of the input feature set (103-D) of GP in this work is larger than the previous work (10-D). As the feature space has been increased, a larger number of individuals can be helpful for GP to find a good solution. Meanwhile, in this work, the maximum depth of GP tree (10) is larger compared to the previous work (4). This is another way to help GP to effectively explore a wide feature space and avoid code bloating [181].

The evolutionary process is terminated when the maximum number of 50 generations is reached. The evolutionary process is independently executed 30 times using different random seed values and the average performance and standard deviation are reported. Here, the mutation and

Table 6.1: GP parameters.

Parameter	Value	Parameter	Value
Population Size	300	Generations	50
Minimum Depth	2	Maximum Depth	10
Mutation Rate	0.40	Crossover Rate	0.60
Elitism	Keep the single best	Selection Type	Tournament
Population	Half-and-half	Size of Tournament	7

crossover rates are set to 40% and 60%, respectively, based on the fact that a higher mutation rate could produce better training performance by allowing a wider exploration of the search space [94]. The best evolved program (elitism) is kept to prevent the performance of the subsequent generation from degrading. The tournament selection method is used for selecting individuals for the mating process and the tournament size is set to 7. By increasing the population size, the tournament size is also increased in this chapter. Table 6.1 gives a summary of the GP parameters.

6.3.4 Evaluation Metrics

The performance of the GPFSFC method is evaluated using the evaluation criteria described in Section 3.3.4 (on page 91).

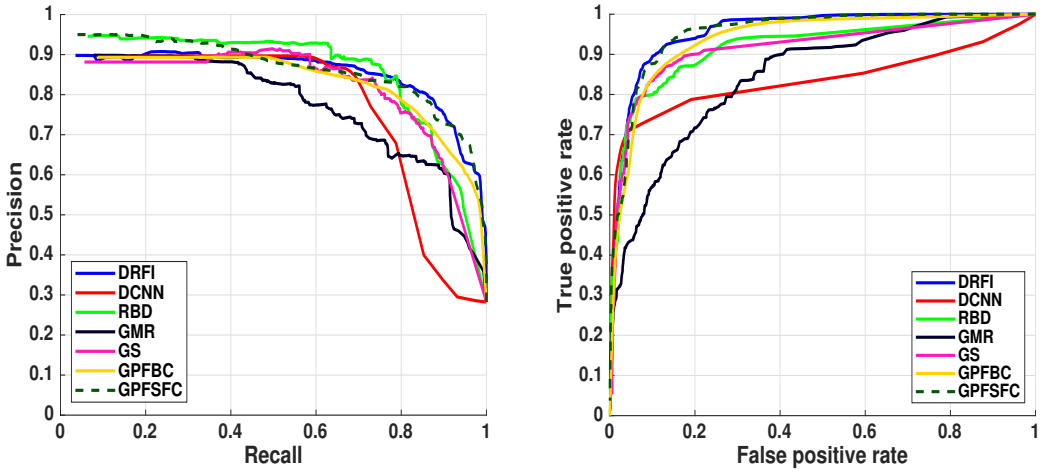
6.4 Results and Discussions

6.4.1 Quantitative Comparisons

6.4.1.1 The SED1 Dataset

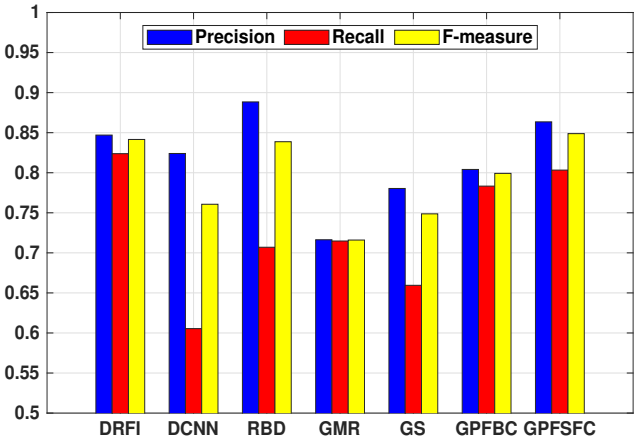
In Figures 6.4(a) and 6.4(b), precision-recall and ROC curves of GPFSFC is comparable with DRFI and outperforms other methods on the SED1 dataset.

Based on Figure 6.4(c) and Table 6.2 (SED1), although RBD has the highest precision 0.8884, recall and F-measure values of this method are



(a) Precision-recall curves

(b) ROC curves



(c) Average precision, recall, and F-measure

Figure 6.4: The performance of GPFSFC compared to the six other SOD methods based on the **SED1** dataset.

lower than DRFI and GPFSFC on the SED1 dataset. RBD mostly performs good on suppressing background and having high precision value, while it has limitations on completely highlighting foreground object. GPFSFC has the highest recall and F-measure, 0.8033 and 0.8488, respectively. Here,

Table 6.2: Quantitative results of GPFSFC and other SOD methods based on average precision, recall, and F-measure values on the **SED1**, **ASD**, **ECSSD**, and **PASCAl** datastes. The abbreviations P, R, and F indicate precision, recall, and F-measure, respectively.

Dataset	Method	P	R	F	Dataset	P	R	F
SED1	DRFI	0.847	0.8238	0.8415	ECSSD	0.7923	0.7161	0.7733
	DCNN	0.824	0.6054	0.7607		0.8823	0.8282	0.8692
	RBD	0.8884	0.707	0.8387		0.7191	0.6522	0.7025
	GMR	0.7163	0.7148	0.7116		0.6611	0.4009	0.575
	GS	0.7804	0.6595	0.7487		0.6551	0.6721	0.6589
	GPFBC	0.8041	0.7833	0.7992		0.7296	0.6843	0.7211
	GPFSFC	0.8635	0.8033	0.8488		0.7522	0.7105	0.7421
ASD	DRFI	0.9028	0.9075	0.9039	PASCAL	0.7514	0 .6736	0.7319
	DCNN	0.8702	0.8788	0.8722		0.7906	0.7913	0.7907
	RBD	0.8746	0.8803	0.8759		0.6634	0.557	0.6353
	GMR	0.8366	0.7286	0.8089		0.5504	0.3029	0.4631
	GS	0.8179	0.8808	0.8316		0.6063	0.5749	0.5987
	GPFBC	0.8563	0.8623	0.8577		0.6849	0.6197	0.6686
	GPFSFC	0.9142	0.882	0.9066		0.7187	0.6437	0.6999

Table 6.3: The statistical comparison of GPFSFC and the other six SOD methods based on AUCPR on the **SED1**, **ASD**, **ECSSD** and **PASCAL** datastes.

	DRFI	DCNN	RBD	GMR	GS	GPFBC	GPFSFC
SED1	0.5779	0.5128 ↑	0.5630 ↑	0.4938 ↑	0.5477 ↑	0.5480 ±0.0573 ↑	0.5778 ±0.0284
ASD	0.7477 ↓	0.6854 ↑	0.7339	0.6511 ↑	0.6840 ↑	0.7256 ±0.0112 ↑	0.7354 ±0.0080
ECSSD	0.5899 ↓	0.6558 ↓	0.5229 ↑	0.4352 ↑	0.4766 ↑	0.5265 ±0.0213 ↑	0.5347 ±0.0102
PASCAL	0.5294 ↓	0.5462 ↓	0.4388 ↑	0.3397 ↑	0.3901 ↑	0.4480 ±0.0193 ↑	0.4771 ±0.0144

GPFSFC has lower recall (0.8033) than DRFI with 0.8238, but GPFSFC has higher precision (0.8635) than DRFI with 0.847. Among compared methods, SF results the lowest performance with values of 0.6403, 0.2249, and 0.4489 for precision, recall, and F-measure, respectively.

GPFSFC improved GPFBC on more precisely detecting salient regions by increasing precision and also decreasing FN which causes to highlight

more foreground regions. As shown in Figure 6.4(c) and Table 6.2 (SED1), DCNN has lower performance compared to good performing methods such as GPFSFC and DRFI. On the SED1 dataset, DCNN does not perform well at highlighting foreground object completely, this can be caused by the limitation of the deep-learning based DCNN method when the dataset contains small number of samples like SED1.

In terms of statistical significance t-test on the SED1 dataset in Table 6.3, GPFSFC shows comparable AUCPR value 0.5578, to DRFI with 0.5779 and it outperforms the other five methods.

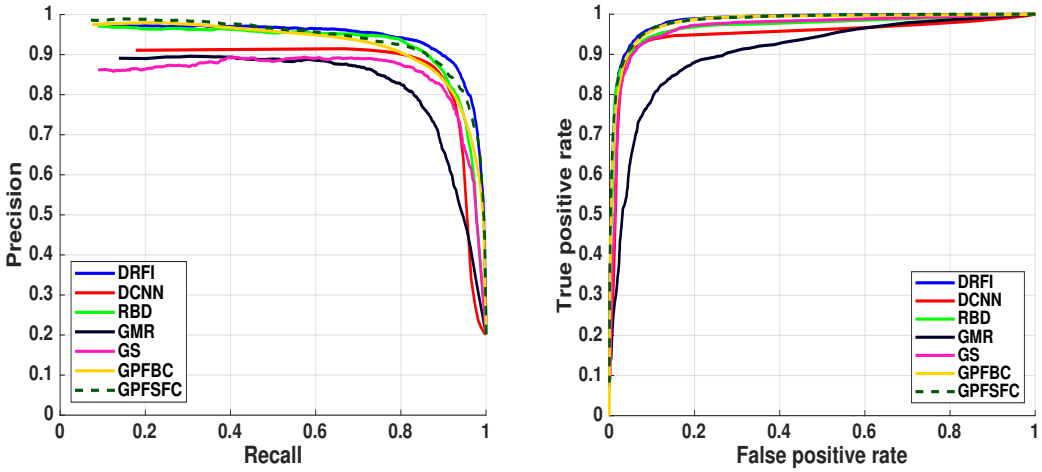
6.4.1.2 The ASD Dataset

As it can be seen in Figures 6.5(a) and 6.5(b), GPFSFC is performing as the second best method after DRFI regarding the precision-recall and ROC curves on the ASD dataset. GPFSFC performs nearly close to DRFI and better than RBD and DCNN on the ASD dataset.

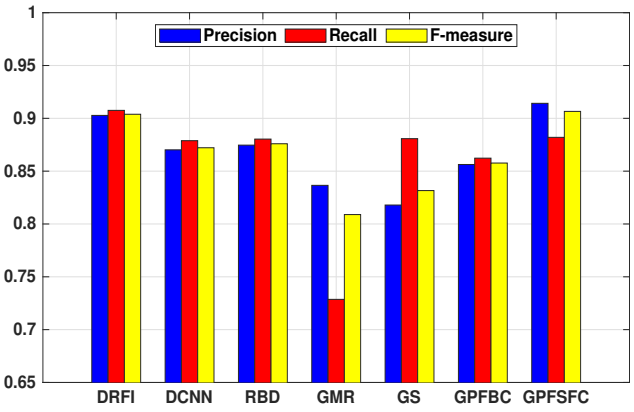
In Figure 6.5 (c) and Table 6.2 (ASD), although GPFSFC has slightly lower average recall 0.882 than DRFI with 0.9075, it results higher precision 0.9142 than DRFI with 0.9028. Thus, DRFI performs better at capturing more foreground regions and decreasing FN, while GPFSFC returns more accurate background regions by decreasing FP along with increasing TP.

To compare GPFSFC with DRFI, as DRFI employed an ensemble learning containing a large number of decision trees (200) to predict the saliency value of the regions, it can generally generate more accurate results than the one GP program (GP tree) which is returned as the final result.

On the ASD dataset in Table 6.3, GPFSFC with 0.7354 has a close AUCPR value to RBD (0.7339) and higher AUCPR value compared to other datastes. Table 6.3 shows that GPFSFC can increase the average area under the curve compared to GPFBC by employing more informative features on all the datastes.



(a) Precision-recall curves (b) ROC curves



(c) Average precision, recall, and F-measure

Figure 6.5: The performance of GPFSFC compared to the six other SOD methods based on the **ASD** dataset.

6.4.1.3 The ECSSD Dataset

Regarding precision-recall and ROC curves in Figures 6.6(a) and 6.6(b), GPFSFC is the third best performing method among the six SOD methods, where DCNN outperforms all the methods, although it could not perform

as good as GPFSFC and DRFI on the SED1 and ASD datasets.

In Figure 6.6 (c) and Table 6.2 (ECSSD), DCNN achieves the highest average precision, recall, and F-measure values, 0.8823, 0.8282, and 0.8692, respectively. DCNN shows good performance on both decreasing FP and FN along with increasing TP compared to the other six SOD methods. ECSSD is one of the challenging datasets, however, the quantitative results of DCNN show the capability of this method in handling complex images. GPFSFC is the third good performing method after DCNN and DRFI, and it slightly loses its performance in ECSSD compared to SED1 and ASD.

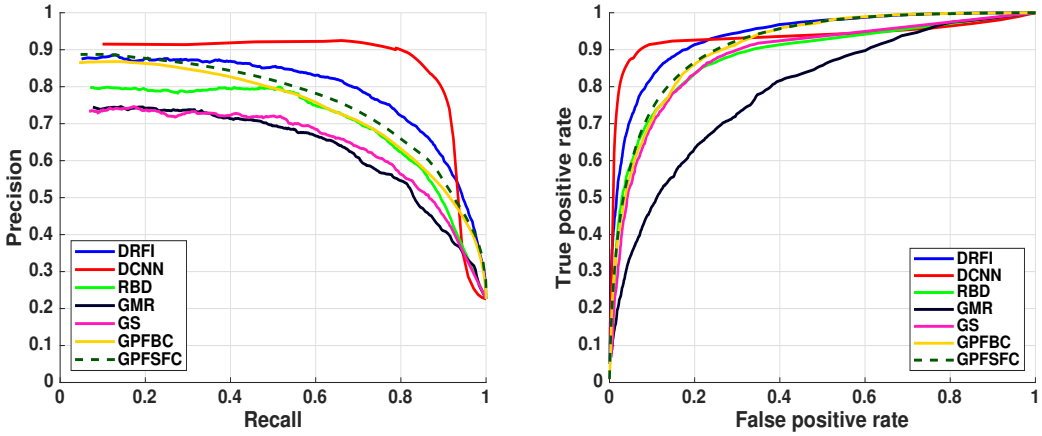
As the ECSSD dataset contains images with complex background and non-homogeneous foreground object, high-level features are required to capture foreground objects. High-level features can help low-level features in obtaining general concepts of foreground objects in complex images. One of the reasons for GPFSFC's performance degradation in challenging datasets is lack of high-level saliency features in the feature set of GPFSFC, while DCNN benefits of combining both low-level and high-level saliency features.

In Table 6.3 (ECSSD), DCNN with 0.6558 value significantly outperforms all the methods, however, GPFSFC with 0.5347 still shows good performance compared to the other SOD methods on ECSSD in terms of the statistical significance t-test at the significance level 5%.

6.4.1.4 The PASCAL Dataset

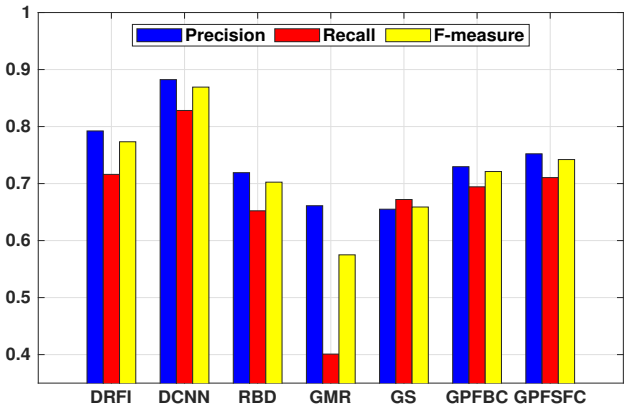
On the PASCAL dataset, GPFSFC shows good results after DCNN and DRFI based on precision-recall and ROC curves as shown in Figures 6.7 (a) and (b).

Figure 6.7(c) and Table 6.2 (PASCAL) show that DCNN achieves the highest precision, recall, and F-measure values, 0.7906, 0.7013, and 0.7319, respectively. PASCAL dataset is a challenging dataset similar to ECSSD, hence, DCNN which employs high-level features can obtain better performance than those methods which do not have high-level features in their



(a) Precision-recall curves

(b) ROC curves

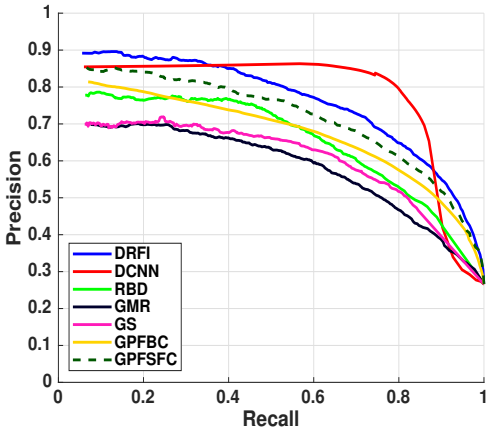


(c) Average precision, recall, and F-measure

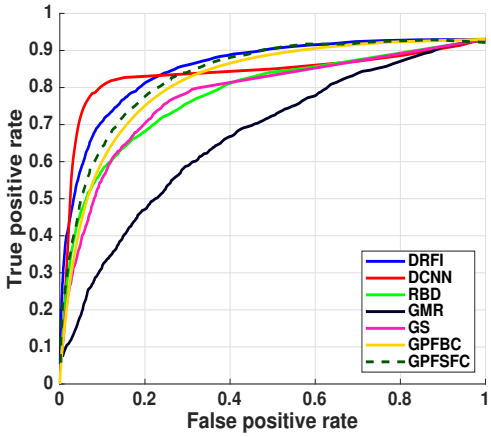
Figure 6.6: The performance of GPFSFC compared to the six other SOD methods based on the **ECSSD** dataset.

feature sets.

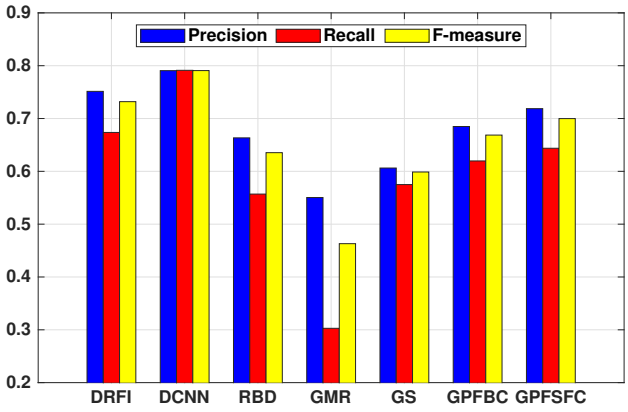
Table 6.3 (PASCAL), similar to ECSSD, DCNN with 0.5462 has the best AUCPR and then DRFI (0.5294), GPFSFC has lower average AUCPR to those methods on both ECSSD and PASCAL unlike SED1 and ASD.



(a) Precision-recall curves



(b) ROC curves



(c) Average precision, recall, and F-measure

Figure 6.7: The performance of GPFSFC compared to the six other SOD methods based on the **PASCAL** dataset.

6.4.2 Qualitative Comparisons

The qualitative comparisons of GPFSFC and six other benchmark methods are illustrated in Figures 6.8 and 6.9. For the qualitative comparisons, multiple representative images are selected from different datasets which incorporate a variety of difficult circumstances, including complex scenes,

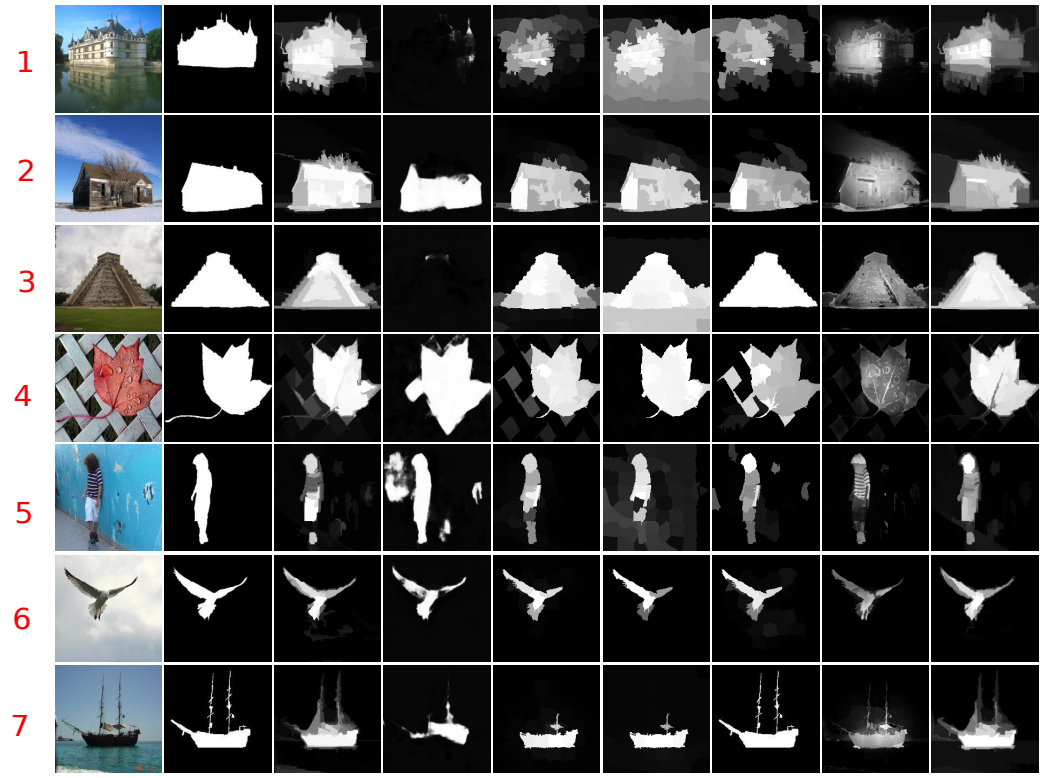
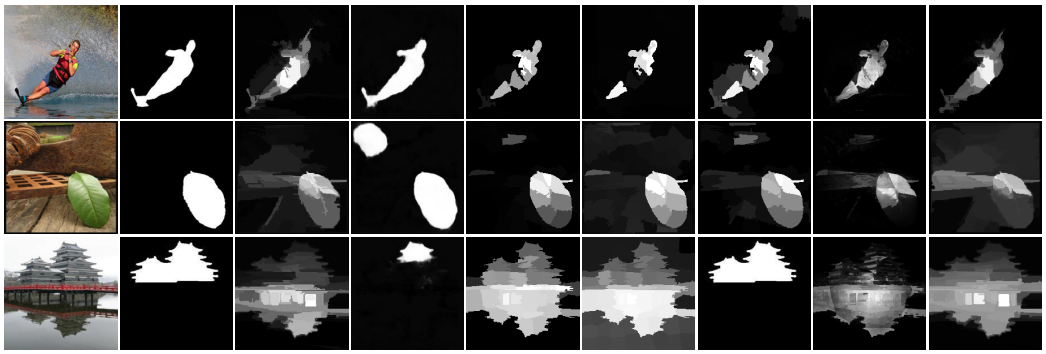


Figure 6.8: Qualitative results of GPFSFC and the six other SOD methods for sample images taken from the **SED1**, **ASD**, **ECSSD**, and **PASCAl** datasets.

salient objects with center bias, salient objects with different sizes, low contrast between foreground and background. Figure 6.8 presents some challenging cases where GPFSFC can successfully highlight salient objects and suppress background. For example, the image in the first row is a complex image where the building is not homogeneous, it has reflection on the water and complex background. However, GPFSFC and DRFI can deal with it, while the other methods including DCNN, RBD, GMR, and GS perform poorly to detect the salient object. DRFI is good in detecting the salient object, but it wrongly highlights some parts of the background regions.



Original GT DRFI DCNN RBD GMR GS GPFBC GPFSFC

Figure 6.9: Some failure examples of GPFSFC and the six other SOD methods on the **SED1**, **ECSSD** and **PASCAl** datasets.

In Figure 6.8 in row 4, GPFSFC can completely suppress the background where the background is cluttered, due to the advantage of selecting informative background features by the evolved GP program. As can be seen in the 5th image, it has non-homogeneous foreground object and complex background, GPFSFC shows good performance, while the other ten methods are struggling in both highlighting the object and completely suppressing the background. In the 6th row, GPFSFC properly covers the foreground object, although the color contrast between the object and the background is low. Choosing informative contrast, backgroundness, and property (appearance and geometric) features and combining them using suitable mathematical operations is the key point for having good performance on the aforementioned images.

Figure 6.8 shows that DCNN fails to completely detect the salient objects when the image has complex background and low contrast between the foreground object and background. One potential reason can be lack of segment-level information like prior knowledge on segment level.

Although both GPFSFC and DRFI show good performance in different scenarios, these methods suffer in some challenging cases such as images in Figure 6.9. GPFSFC fails to completely identify the foreground object

in all three images and wrongly highlights the background in the 2nd and 3rd images. This problem is due to the lack of the high-level knowledge and enough training samples to learn different scenarios and object types.

6.4.3 Further Analysis

Figure 6.10 shows an example of evolved GP program with high performance on the ASD dataset. Overall, there are 27 nodes in this program where 14 nodes are leaves and the other 13 are functions. The description of the selected features by this GP individual are represented in Table 6.4. As it can be seen in Figure 6.10, five regional backgroundness features $(f_3^2, f_5^2, f_6^2, f_7^2, f_{11}^2)$ are selected to suppress background regions. Three regional property features (f_7^3, f_8^3, f_{13}^3) are selected to consider the generic properties of regions. Finally, three contrast features (f_1^1, f_1^4, f_3^4) are chosen to capture the color differences space (changes), as a region is likely thought to be salient if it is different from the other regions. This GP program only chooses 11 features from 103 features and decrease the dimensionality nearly 89%. The GP process considers complementary characteristic of features in feature selection and combination stages using the fitness value of the evolved GP programs. Figure 6.10 demonstrates that the evolved GP program or mathematical expression is a non-linear function. Furthermore, it shows how the combination of different types of features such as color, backgroundness, appearance and geometric is important in properly detecting salient objects.

6.5 Chapter Summary

In this chapter, we developed a GP-based method which takes a set of saliency features and automatically produces saliency map. The proposed GPFSFC method can effectively incorporate any additional features and select the features complement each other. GPFSFC makes no assumption

Table 6.4: The description of selected features by the sample GP program on the **ASD** dataset.

Feature	Description	Feature type
f_1^1	average R value	regional contrast
f_3^2	average B value	regional backgroundness
f_5^2	average H value	regional backgroundness
f_6^2	average S value	regional backgroundness
f_7^2	average V value	regional backgroundness
f_{11}^2	average b^* value	regional backgroundness
f_7^3	normalized perimeter	regional property
f_8^3	aspect ratio of the bounding box	regional property
f_{13}^3	variances of the a^* value	regional property
f_1^4	multi-scale contrast	hand-crafted features
f_3^4	color spatial distribution	hand-crafted features

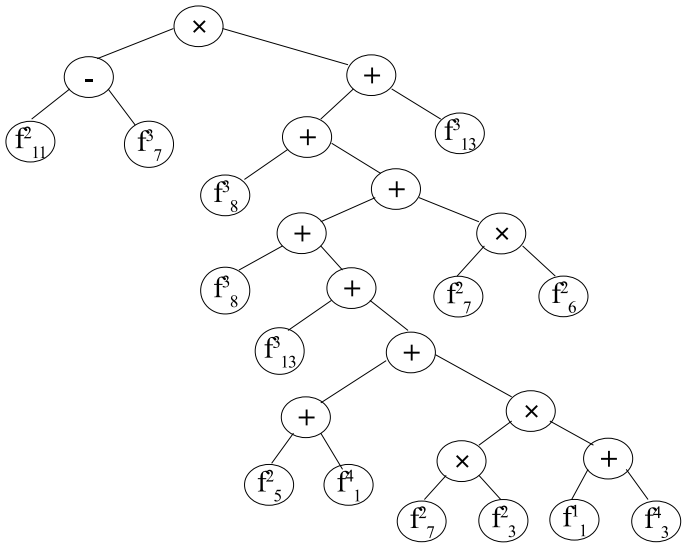


Figure 6.10: Sample program evolved by GPFSFC on the **ASD** dataset.

of linear superposition or equal weights of features and does not require domain-expert. GPFSFC has the ability to tackle a wide range of features from different segmentation levels and explore various mathematical expressions for the feature combination stage. The saliency features by themselves are not sufficient to properly detect the salient object and suppress background. Therefore, a good feature selection and combination method plays an important role in achieving high performance. The quantitative and qualitative results reveal that GPFSFC can effectively choose the features which complement each other, thus, the final combination of those features results in a good saliency map.

In this chapter, although GPFSFC was slightly worse on the ECSSD and PASCAL datasets, it showed promising results by outperforming one of the well-known and recent CNN-based methods (DCNN) on two datasets including SED and ASD.

Based on the achievements in this chapter, the large saliency feature set will be employed in the following chapter. Note that increasing the variety of available features causes GP to evolve complex solutions. The solutions having large sizes, are computationally expensive and difficult to interpret. Moreover, some of good features may lose the chance of contributing to the final solution, due to the difficulty in thoroughly exploring the very large search space. Therefore, the next chapter will investigate reducing the feature space by dividing features to different groups and applying GP on each group separately.

Chapter 7

GP for High-level Feature Construction

7.1 Introduction

SOD methods mainly rely on features that are extracted from different levels of information to compute a final saliency map or a binary mask. Therefore, many studies have developed a rich set of saliency features including heuristic features [27], hand-crafted local features [75], global features [76, 138], and both local and global features [28], and indicated the importance of powerful feature representations for SOD. A detailed review of these methods can be found in [25]. Although great progress has been achieved, there remain key problems to be addressed regarding designing powerful and robust saliency features.

Low-level and hand-crafted saliency features can handle simple scenarios, but they usually have difficulties in challenging cases. Moreover, the majority of those features have been manually designed/extracted to capture and focus on some aspects or parts of images. Local features that give information in pixel-level (or small region-level) can be effective in showing the boundaries or edges [171]. However, local features suffer from limited representation capability and robustness [194].

Global features are helpful in capturing information in large region-level or even image-level, while this type of features cannot well reflect the structural information of salient objects.

Heuristic features are generally designed/extracted by domain experts based on assumptions on color, location, spatial, shape of salient objects and background [44]. Since heuristic features are developed based on those assumptions, they often have good performance when images satisfy some or all of those assumptions; otherwise, these features may become unsuitable. Hence, the heuristic features have limitations in generalizability over different image types.

The aforementioned features have typically been designed to partially tackle the SOD problem. Only in special cases, a single simple feature would be sufficient to fully characterise saliency. Thus, each individual feature can not be expected to detect the whole salient objects and suppress background. This problem arises from the limitations in the definitions of the features and the wide variety of possible objects and background regions.

In order to tackle the drawbacks of the limited capability of the low-level and hand-crafted features, high-level features have been recently introduced [66]. A good high-level feature aims to capture the general concept along the details of salient objects. Recently, effort to develop deep convolutional neural networks (CNNs) for SOD have achieved good results; however, most CNN-based methods unavoidably drop the location information and low-level fine details (e.g. edges and corners) of salient objects, leading to unclear/blurry boundary predictions [66,98].

The majority of the reported saliency features have been manually developed/extracted by experts in the SOD domain [27]. However, manually exploring and extracting/designing powerful and high-level saliency features is a difficult task, especially in complex image types (e.g., low contrast between salient object and background, images with cluttered/complex background). The process has some difficulties including:

1) obtaining domain knowledge is difficult and time-consuming, 2) lack of availability for domain experts and the high cost of employing them, and 3) robustness of resulting high-level saliency feature. Therefore, an automatic method to generate high-level saliency features would have many advantages such as free from any prior knowledge or human intervention and saving time.

Since the late 1990s, GP has been employed to automatically extract, and construct features for images [192]. With the embedded feature selection capability, GP can select features which are relevant to build a new feature. GP-based constructed features have better interpretability than those constructed in the hidden layers of CNNs due to GPs tree structure representation.

In this chapter, we employ a large set of saliency features (taking from the previous chapter) for automatically constructing high-level saliency features. GP has shown the potential of handling a large feature space in the previous chapter for saliency detection. However, in the large search space, some of the good features may lose their chances to contribute to the final solution, since those features may be neglected by dominating features. While knowledge and information of those features may help to improve the performance. Moreover, GP often generates similar features by feeding one input feature set and feeding different input feature sets can help to generate diverse features.

7.1.1 Chapter Goals

This chapter targets to develop GP algorithms to automatically construct new high-level saliency features for SOD. The new method is called GPFC-SOD which is GP for feature construction in SOD. In this study, different high-level features will be constructed by feeding different input feature sets to GP. To produce the input feature sets for GP, a feature subset preparation system is developed. Precisely, the following objectives will be in-

vestigated :

- Develop a feature subset preparation method to choose features from different feature categories and create new feature subsets for the proposed GP method. The idea is to divide a large feature set to small feature subsets, therefore, all the features will have a good chance to contribute in the final result. This objective will decrease the search space of GP and help producing diverse saliency features;
- Develop a GP method to construct high-level saliency features to introduce new informative features to the SOD domain. These new automatically constructed high-level features are expected to augment to the existing features to handle complicated scenarios and generate more accurate saliency maps;
- Investigate whether the GP-based high-level saliency feature(s) can obtain better results than the hand-crafted saliency features; and
- Investigate the impact of the automatically constructed high-level features on detecting salient objects.

7.1.2 Chapter Organization

The remainder of the chapter is organized as follows. Section 7.2 details the proposed method. Section 7.3 provides the experiment design. Section 7.4 presents and discusses the results. Section 7.5 provides summary.

7.2 GP-based High-level Feature Construction

7.2.1 The Overall Algorithm

The overall algorithm of GPFCSOD is demonstrated in Figure 7.1. GPFCSOD consists of three phases including feature subset preparation, feature

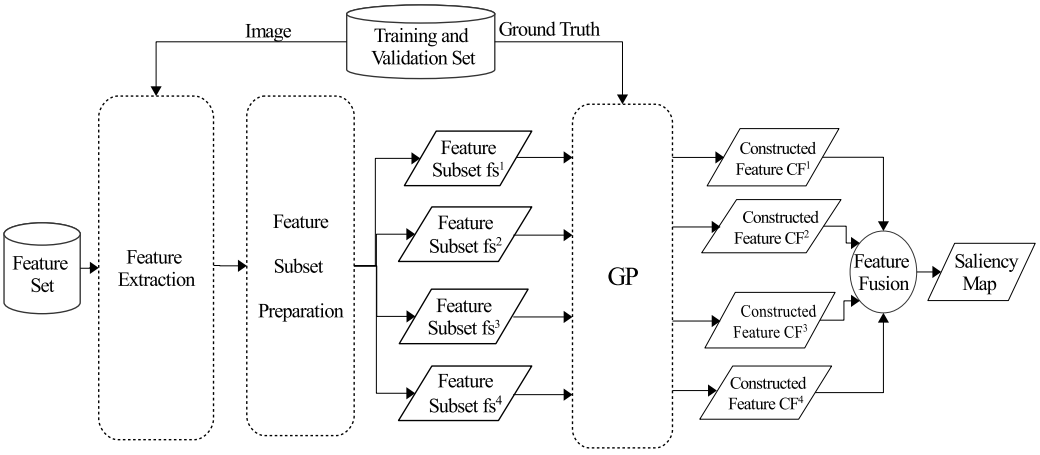


Figure 7.1: The overall structure of GPFCSOD.

construction, and feature fusion/combination. In the first phase, a large set of saliency features from different categories is divided to subset of features as demonstrated in Figure 7.1. The aim of this phase is to collect different types of features in small feature subsets. This will give a good chance for GP to widely explore the feature subset. Thus, GP can investigate the contribution of each feature during the feature construction process. Therefore, the algorithm can control dominating features by giving some opportunities to other features in the first generations of the evolutionary process. Another advantage of grouping features is ensuring the diversity of input features for GP, this is provided by appearing each feature only in one feature subset. In the second phase, GP is employed for automatically constructing a new high-level feature by giving a feature subset as input (Figure 7.1). By feeding different input feature subsets, GP can construct different features. Different constructed high-level features can play different roles in detecting various types of objects and background. In the third phase of feature fusion/combination, we simply linearly add the constructed features together to compute the final saliency map.

7.2.2 Feature Subset Preparation

We employ the feature set $\boldsymbol{f} = \{f_1, f_2, ..., f_D\}$ which has been used in Chapter 6 contains four groups $\boldsymbol{fg} = \{\boldsymbol{fg}^1, \boldsymbol{fg}^2, \boldsymbol{fg}^3, \boldsymbol{fg}^4\}$ of features. These feature groups are regional contrast ($\boldsymbol{fg}^1 = \{f_1^1, f_2^1, ..., f_{29}^1\}$), regional backgroundness ($\boldsymbol{fg}^2 = \{f_1^2, f_2^2, ..., f_{29}^2\}$), regional property ($\boldsymbol{fg}^3 = \{f_1^3, f_2^3, ..., f_{35}^3\}$), and hand-crafted ($\boldsymbol{fg}^4 = \{f_1^4, f_2^4, ..., f_{10}^4\}$).

Each feature group $\boldsymbol{fg}$ is randomly divided to four subsets $\boldsymbol{s} = \{\boldsymbol{s}_1, \boldsymbol{s}_2, \boldsymbol{s}_3, \boldsymbol{s}_4\}$. Here, randomly shuffling features in each category and then separating them to subsets will help to provide diverse subsets. $\boldsymbol{fg}^1$ (regional contrast) has 29 features, and it is randomly divided to $\{\boldsymbol{s}_1^1, \boldsymbol{s}_2^1, \boldsymbol{s}_3^1, \boldsymbol{s}_4^1\}$ including 8, 7, 7, and 7 features respectively. Similar to $\boldsymbol{fg}^1$, $\boldsymbol{fg}^2$ (regional backgroundness) has 29 features, and it is randomly divided to four feature subsets, $\{\boldsymbol{s}_1^2, \boldsymbol{s}_2^2, \boldsymbol{s}_3^2, \boldsymbol{s}_4^2\}$ including 8, 7, 7, and 7 features. $\boldsymbol{fg}^3$ (regional property) consists of 35 features and its feature subsets are $\{\boldsymbol{s}_1^3, \boldsymbol{s}_2^3, \boldsymbol{s}_3^3, \boldsymbol{s}_4^3\}$ including 9, 9, 9, and 8 features. Finally, $\boldsymbol{fg}^4$ (hand-crafted) with only 10 features formed $\{\boldsymbol{s}_1^4, \boldsymbol{s}_2^4, \boldsymbol{s}_3^4, \boldsymbol{s}_4^4\}$ feature subsets including 3, 2, 3, and 2 features.

In the next step, four feature sets $\boldsymbol{fs} = \{\boldsymbol{fs}^1, \boldsymbol{fs}^2, \boldsymbol{fs}^3, \boldsymbol{fs}^4\}$ are generated. To ensure diversity, each feature set $\boldsymbol{fs}$ is generated by randomly selecting features from each of the four feature groups. As shown in Figure 7.2, $\boldsymbol{fs}^2$ is obtained by randomly selecting feature subset $\{\boldsymbol{s}_3^1\}$ from feature group $\boldsymbol{fg}^1$, $\{\boldsymbol{s}_3^2\}$ from $\boldsymbol{fg}^2$, $\{\boldsymbol{s}_1^3\}$ from $\boldsymbol{fg}^3$, and $\{\boldsymbol{s}_2^4\}$ from $\boldsymbol{fg}^4$. As a result, each feature set $\boldsymbol{fs}$ contains features from all the four types.

Here, we don't use each feature group separately as a feature subset for GP, since it will not provide any chance for GP to explore different feature types together. In this case, GP will result four constructed features which are separately evolved and none of the features from different categories have had a chance to interact with each other. Therefore, it is a good idea to collect different features from different groups to make each constructed feature informative towards different feature categories. This causes the final saliency map which is generated by combining those fea-

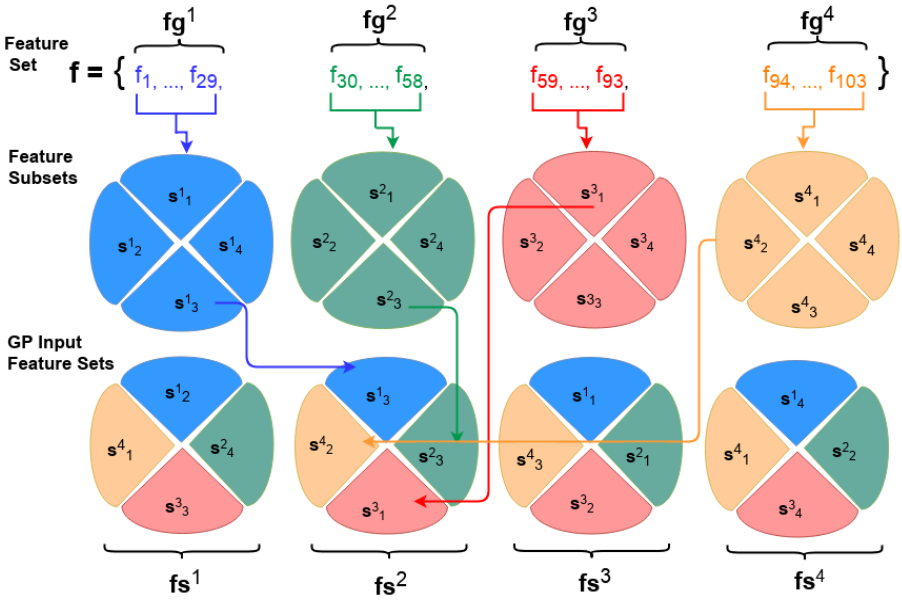


Figure 7.2: The feature subsets generation process.

tures will be comprehend. The reason is that each constructed feature has a contribution of different feature categories.

7.2.3 GPFCSOD

In this work, GP has been utilized and ran four times (each run employs one of the four feature subsets as show in Figure 7.2) with similar parameter settings, fitness function, input dataset (training and validation sets), but different input feature sets (terminal). Here, the idea is to use all the features of the feature set $\boldsymbol{f}$, but allow each GP to select different important and informative features. Each GP attempts to generate one high-level feature, hence a total of four high-level features $\boldsymbol{CF} = \{\boldsymbol{CF}^1, \boldsymbol{CF}^2, \boldsymbol{CF}^3, \boldsymbol{CF}^4\}$ are constructed (Figure 7.1). Finally, these constructed features are linearly fused to produce the final saliency map.

As shown in Figure 7.3, GP follows three phases including training, validation, and test to produce the final results. GP randomly generates

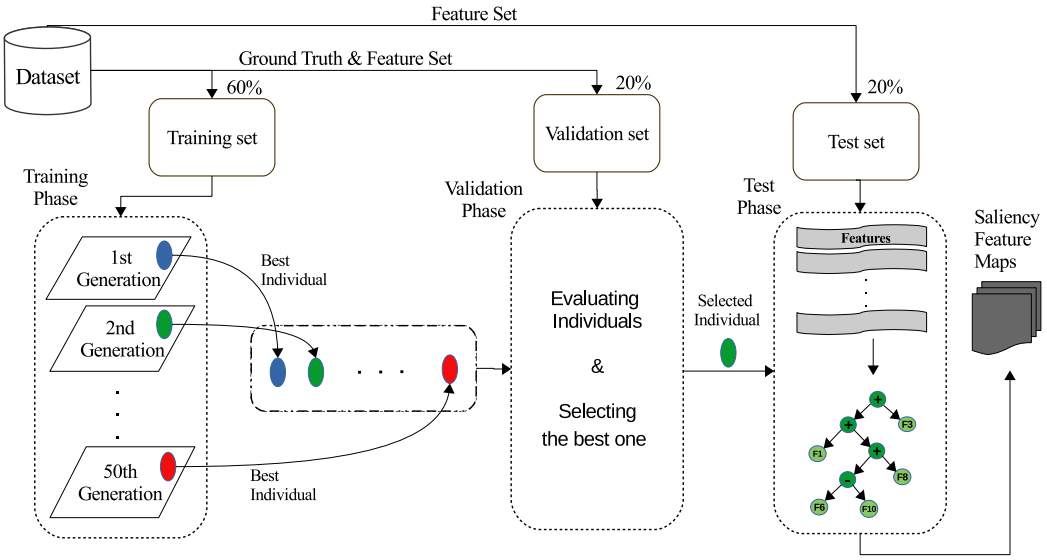


Figure 7.3: The GP evolution of the proposed method (GPFCSOD).

an initial population of individuals and each individual contains saliency features (automatically chosen from the corresponding feature set) and constants as terminals, and mathematical functions $\{+, -, \times, \sin\}$ as internal nodes. GP individuals are evaluated by employing a suitable fitness evaluation function (in Section 7.2.6) is employed. The population of next generations is produced by employing some genetic operators such as mutation, crossover, and elitism on individuals which are selected from the current population. Here, the algorithm uses a maximum number of generations as a stopping criterion. After completing the training process, the 50 best-of-generation programs are returned where each one is best program from one generation of the evolutionary training phase (Figure 7.3).

In the validation phase, we evaluate the 50 best-of-generation solutions taken from the training phase on the validation set and the solution which performs the best on the validation set is selected as the GP final solution.

In the test phase, after applying feature extraction on the test set, it is fed to the selected GP solution to compute their high-level feature maps. After computing the feature maps, they are fused to compute the final

saliency map.

7.2.4 Function Set

In this work, we use a similar function set to the previous chapter (see Section 6.2.2 on page 156).

7.2.5 Terminal Set

The terminal set for each GP run (totally four GP runs) is provided by taking constants in the range [-1,1] and a feature subset produced by the feature subset preparation phase (see Section 7.2.2).

7.2.6 Fitness Function

Since KL divergence [91] performed successfully as a fitness function for GP in Chapter 6, we also employ it to measure the difference between the probability distribution of the GP-based constructed high-level saliency feature and the probability distribution of the ground truth. The average of KL values on the training images is used to compute the fitness value for each GP individual. The fitness value is formulated as:

$$Fitness_{cf} = \frac{1}{n}\sum_{i=1}^{n} D_{KL}(\boldsymbol{G}_i||\boldsymbol{CFS}_i) \tag{7.1}$$

where, n presents the number of the training images, $\boldsymbol{CFS}_i$ presents the constructed high-level saliency feature map using GP on the i^{th} image and $\boldsymbol{G}_i$ presents the ground truth of the i^{th} image. KL is computed as

$$D_{KL}(\boldsymbol{G}||\boldsymbol{CFS}) = \sum_{rg\in R} \boldsymbol{G}(rg)\frac{\ln \boldsymbol{G}(rg)}{\boldsymbol{CFS}(rg)} \tag{7.2}$$

where rg presents a region from the segmented image. $\boldsymbol{G}(rg)$ is the saliency distribution of the region rg in the ground truth and $\boldsymbol{CFS}(rg)$ denotes the saliency distribution of the region rg in the GP output/constructed high-level saliency feature map.

Table 7.1: GP parameters.

Parameter	Value	Parameter	Value
Population	250	Generations	50
Minimum Depth	2	Maximum Depth	10
Mutation Rate	0.40	Crossover Rate	0.60
Elitism	Keep the single best	Selection Type	Tournament
Population	Half-and-half	Size of Tournament	5

7.3 Experiment Design

7.3.1 Datasets

In this work, the performance of GPFCSOD is evaluated using the same datasets and setting which employed in Section 5.3.1 (on page 138).

7.3.2 Benchmark Methods for Comparisons

The GPFCSOD method is compared to the similar methods from Section 6.3.2 (on page 160) .

7.3.3 Parameter Settings

The GP algorithm consists a set of parameters which are shown in Table 7.1. As feature subset selection process (described in 7.2.2) is stochastic, this process is run 10 times using different random seed values. For each run of feature subset selection phase, GP is separately run four times, feeding four different input feature sets. For a given input feature set, GP run 30 times using different random seed values. From the 30 solutions of the GP algorithm, the best solution is chosen. Finally, four solutions or constructed features are computed.

7.3.4 Evaluation Metrics

The quantitative performance of the GPFCSOD method and other SOD methods are evaluated using the evaluation criteria described in Section 3.3.4 on page 91.

7.4 Results and Discussions

7.4.1 Quantitative Comparisons

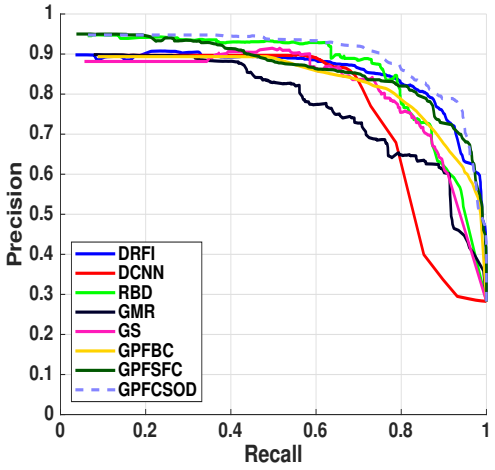
7.4.1.1 The SED1 Dataset

Figure 7.4(a) shows that GPFCSOD outperforms the seven other SOD methods in terms of precision-recall curves based on the SED1 dataset. Based on ROC curves of the eight SOD methods in Figure 7.4(b), GPFCSOD has a comparable result with GPFSFC and outperforms other SOD methods.

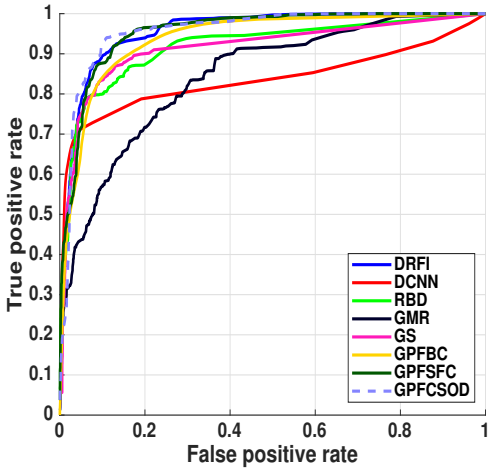
As shown in Figure 7.4(c) and Table 7.2 (SED1), GPFCSOD results the best performance with the values of 0.8832, 0.8442, and 0.8739 for average precision, recall, and F-measure, respectively. GPFCSOD performs better than GPFSFC on both accurately detecting foreground object and suppressing background. This can be due to the result of giving chance to GP to widely explore the feature space and involve informative features in the returned solutions.

Unlike deep learning CNN-based methods which have more failure cases for small datasets such as SED1, GPFCSOD can easily tackle a small number of samples for training.

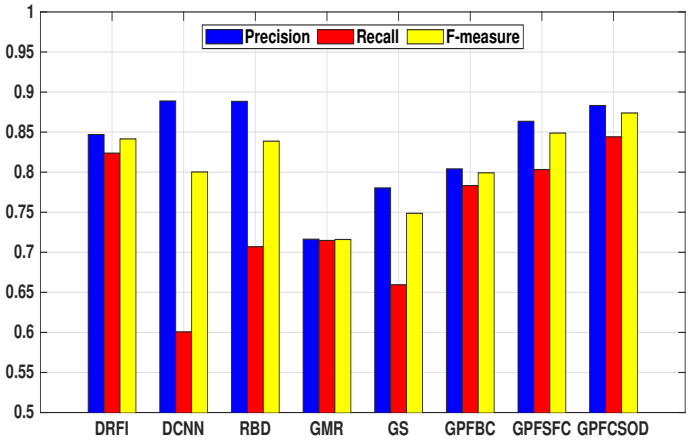
As shown in Table 7.3 (SED1), GPFCSOD with a large difference from the good performing methods, DRFI and DCNN, achieves the highest average AUCPR on the SED1 dataset.



(a) Precision-recall curves



(b) ROC curves



(c) Average precision, recall, and F-measure

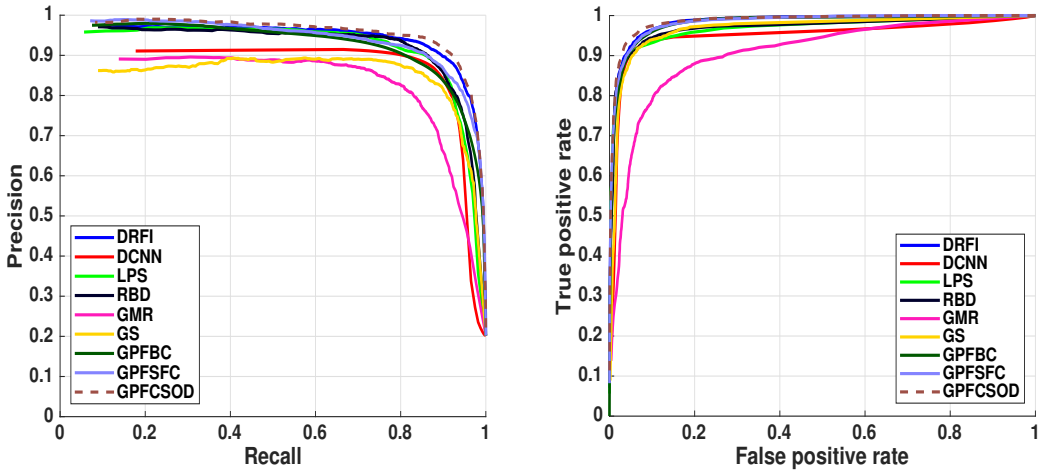
Figure 7.4: The performance of GPFCSOD compared to the seven other SOD methods based on the **SED1** dataset.

Table 7.2: Quantitative results of GPFCSOD and other SOD methods based on average precision, recall, and F-measure values on the **SED1**, **ASD**, **ECSSD**, and **PASCAl** datastes. The abbreviations P, R, and F indicate precision, recall, and F-measure, respectively.

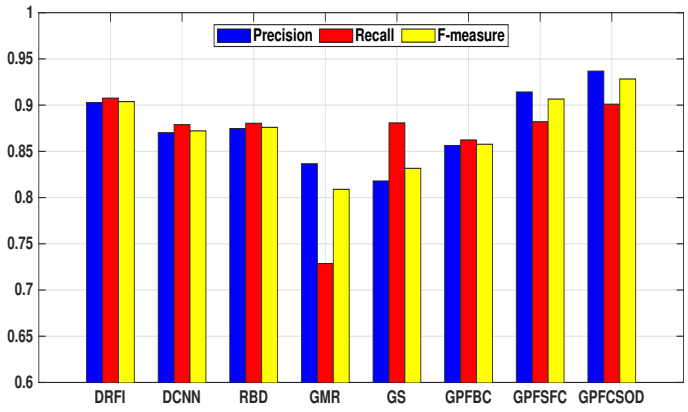
Dataset	Method	P	R	F	Dataset	P	R	F
SED1	DRFI	0.847	0.8238	0.8415	ECSSD	0.7923	0.7161	0.7733
	DCNN	0.824	0.6054	0.7607		**0.8823**	**0.8282**	**0.8692**
	RBD	0.8884	0.707	0.8387		0.7191	0.6522	0.7025
	GMR	0.7163	0.7148	0.7116		0.6611	0.4009	0.575
	GS	0.7804	0.6595	0.7487		0.6551	0.6721	0.6589
	GPFBC	0.8041	0.7833	0.7992		0.7296	0.6843	0.7211
	GPFSFC	0.8635	0.8033	0.8488		0.7522	0.7105	0.7421
	GPFCSOD	**0.8832**	**0.8442**	**0.8739**		0.8128	0.711	0.7868
ASD	DRFI	0.9028	0.9075	0.9039	PASCAL	0.7514	0 .6736	0.7319
	DCNN	0.8702	0.8788	0.8722		**0.7906**	**0.7913**	**0.7907**
	RBD	0.8746	0.8803	0.8759		0.6634	0.557	0.6353
	GMR	0.8366	0.7286	0.8089		0.5504	0.3029	0.4631
	GS	0.8179	0.8808	0.8316		0.6063	0.5749	0.5987
	GPFBC	0.8563	0.8623	0.8577		0.6849	0.6197	0.6686
	GPFSFC	0.9142	0.882	0.9066		0.7187	0.6437	0.6999
	GPFCSOD	**0.9368**	**0.901**	**0.9283**		0.7713	0.6535	0.7405

Table 7.3: The statistical comparison of GPFCSOD and the other seven SOD methods based on AUCPR on the **SED1**, **ASD**, **ECSSD** and **PASCAL** datastes.

Datasets	DRFI	DCNN	RBD	GMR	GS	GPFBC	GPFSFC	GPFCSOD
SED1	0.5779 ↑	0.5128 ↑	0.5630 ↑	0.4938 ↑	0.5477 ↑	0.5480 ±0.0573 ↑	0.5778 ±0.0284 ↑	0.6252 ±0.0075
ASD	0.7477	0.6854 ↑	0.7339 ↑	0.6511 ↑	0.6840 ↑	0.7256 ±0.0112 ↑	0.7354 ±0.0080 ↑	0.7490 ±0.0020
ECSSD	0.5899	0.6558 ↓	0.5229 ↑	0.4352 ↑	0.4766 ↑	0.5265 ±0.0213 ↑	0.5347 ±0.0102 ↑	0.5907 ±0.0063
PASCAL	0.5294	0.5462 ↓	0.4388 ↑	0.3397 ↑	0.3901 ↑	0.4180 ±0.0193 ↑	0.4771 ±0.0144 ↑	0.5269 ±0.0058



(a) Precision-recall curves (b) ROC curves



(c) Average precision, recall, and F-measure

Figure 7.5: The performance of GPFCSOD compared to the seven other SOD methods based on the **ASD** dataset.

7.4.1.2 The ASD Dataset

As shown in Figures 7.5(a) and (b) for the ASD dataset, GPFCSOD outperforming all the hand-crafted based baselines, and the deep learning approach (DCNN) acording to the precision-recall and ROC curves.

Based on Figure 7.5(c) and Table 7.2 (ASD), compared to the seven SOD methods, GPFCSOD achieves better performance on three evaluation criteria with values of o.9368, 0.901, 0.9283 for precision, recall, and F-measure, respectively.

As shown in Table 7.3 (ASD), GPFCSOD's average AUCPR is close to DRFI's best reported AUCPR, and lower than DCNN and better than the others. Generally, the average AUCPR of GPFCSOD is larger than GPFSFC which means GPFCSOD has better performance and it has a good generalizability over all the datasets.

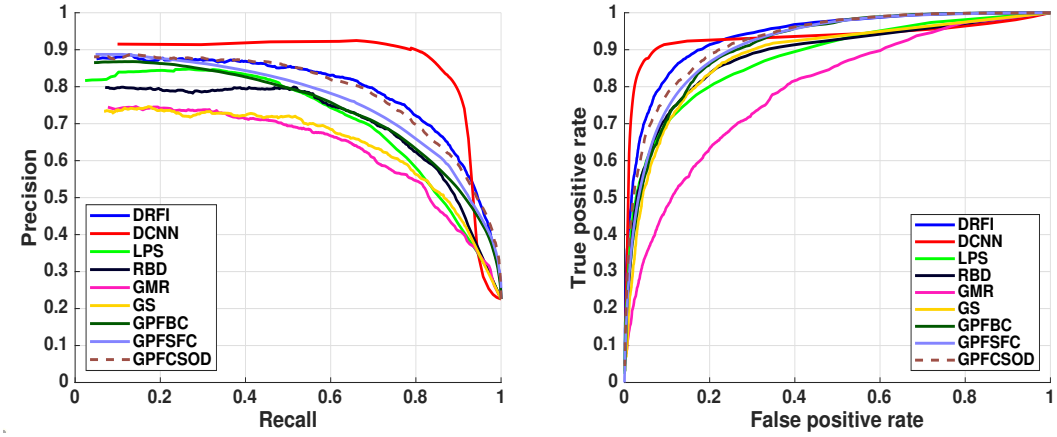
7.4.1.3 The ECSSD Dataset

Based on precision-recall and ROC curves in Figures 7.6(a) and (b), DCNN outperforms all the other SOD methods, and GPFCSOD has lower performance compared to DRFI and DCNN as the two good performing methods.

Figure 7.6(c) and Table 7.2 (ECSSD) provide more details regarding the performance of the compared methods. DCNN results 0.8823, 0.8282, and 0.8692 for precision, recall, and F-measure which shows that DCNN has a good capability on detecting foreground object and suppressing background on challenging images compared to the other methods. GPFCSOD results precision 0.8128 which is higher than the precision of DRFI, 0.7923, and both of them have similar recall value, 0.71. Based on the observed results GPFCSOD obtains better performance by employing high-level features in the saliency detection process.

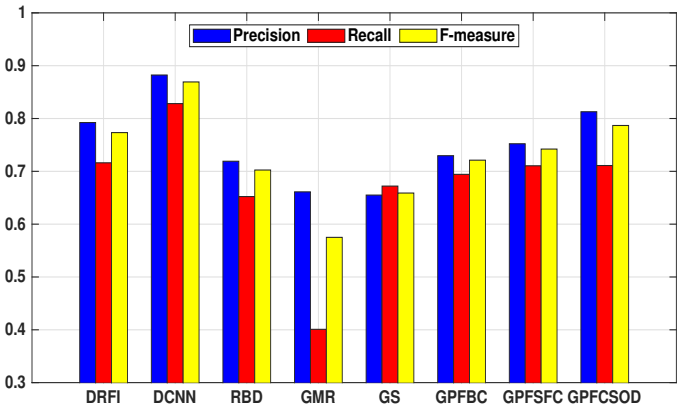
The ECSSD dataset contains more complicated images with multiple foreground objects and cluttered backgrounds. Therefore, most recently developed SOD methods still have limitations in correctly detecting salient objects in the challenging cases in those datasets.

The ECSSD dataset with its challenging images makes the saliency detection task harder for the SOD methods, however GPFCSOD reveals that it improves the GPFSFC's performance with employing high-level



(a) Precision-recall curves

(b) ROC curves



(c) Average precision, recall, and F-measure

Figure 7.6: The performance of GPFCSOD compared to the seven other SOD methods based on the **ECSSD** dataset.

features in the detection process (Table 7.3) based on the statistical t-test results at the significance level 5%.

7.4.1.4 The PASCAL Dataset

In Figures 7.7(a) and (b), Although GPFCSOD has inferior performance to DCNN, it has comparable performance to DRFI and better than other SOD methods.

As shown in Figure 7.7(c) and Table 7.2 (PASCAL), after DCNN with values of 0.7906, 0.7913, and 0.7907 for precision, recall, and F-measure, respectively, GPFCSOD shows better performance with values of 0.7713, 0.6535, and 0.7405.

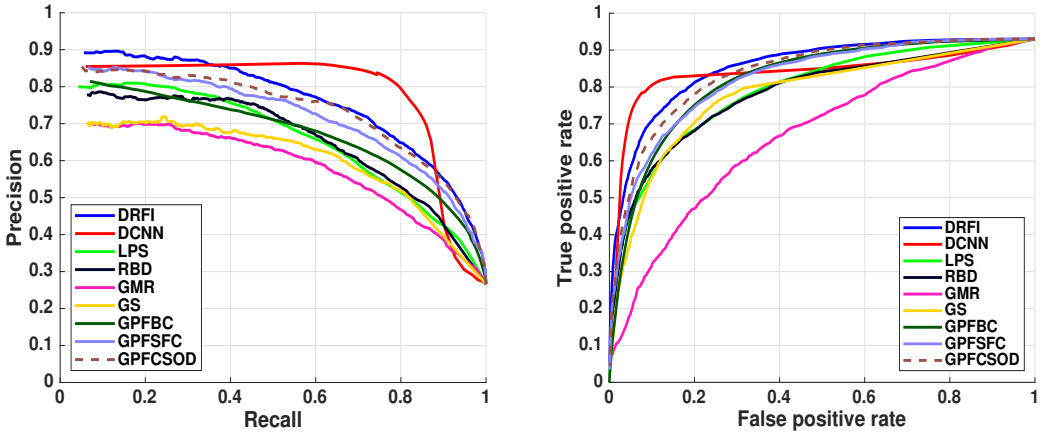
Comparing GPFCSOD with GPFSFC, GPFCSOD performs better on both challenging datasets, ECSSD and PASCAL, therefore, constructing high-level features help GPFCSOD to effectively tackle with complex images. Although GPFCSOD still has limitations on some of the images with complex backgrounds such as images with similar attractive colors in foreground and background which may cause an increase in FP, it shows good performance on other complex types of images such as images having low contrast between foreground object and background.

Here, we can highlight that GPFCSOD outperforms the compared GP-based SOD methods which mostly rely on the existing low-level and hand-crafted features. It can be concluded that constructing high-level features from the low-level and hand-crafted features improves their capability to accurately highlight salient objects and suppress background.

Table 7.3 (PASCAL), although GPFCSOD has slightly lower AUCPR than DCNN, and similar to DRFI, but higher than the other SOD methods on the PASCAL dataset.

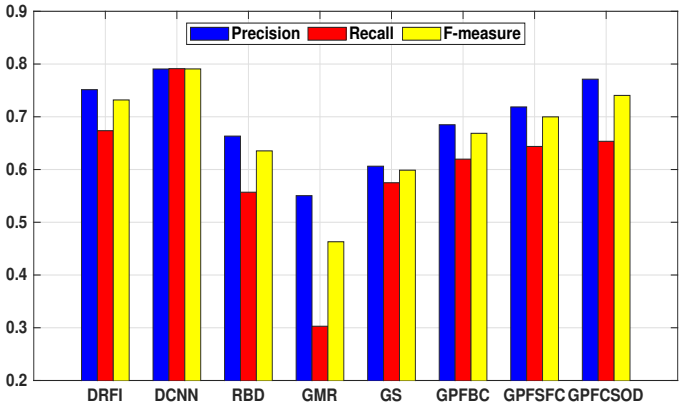
7.4.2 Qualitative Comparisons

Some visual results (saliency maps) of GPFCSOD and the eight SOD methods are demonstrated in Figures 7.8, 7.9, 7.10, and 7.11. For the qualitative comparisons, we choose visual examples from different datasets to consider different scenarios/cases such as images with complex back-



(a) Precision-recall curves

(b) ROC curves



(c) Average precision, recall, and F-measure

Figure 7.7: The performance of GPFCSOD compared to the seven other SOD methods based on the **PASCAL** dataset.

ground, non-homogeneous salient objects, and salient objects having low contrast with background. Figure 7.8 demonstrates some complex and challenging samples of saliency images, where GPFCSOD can successfully detect and highlight salient objects, and suppress background regions. Figure 7.8 shows visual examples of complex backgrounds in images of rows 1–4, where GPFCSOD can completely suppress the back-

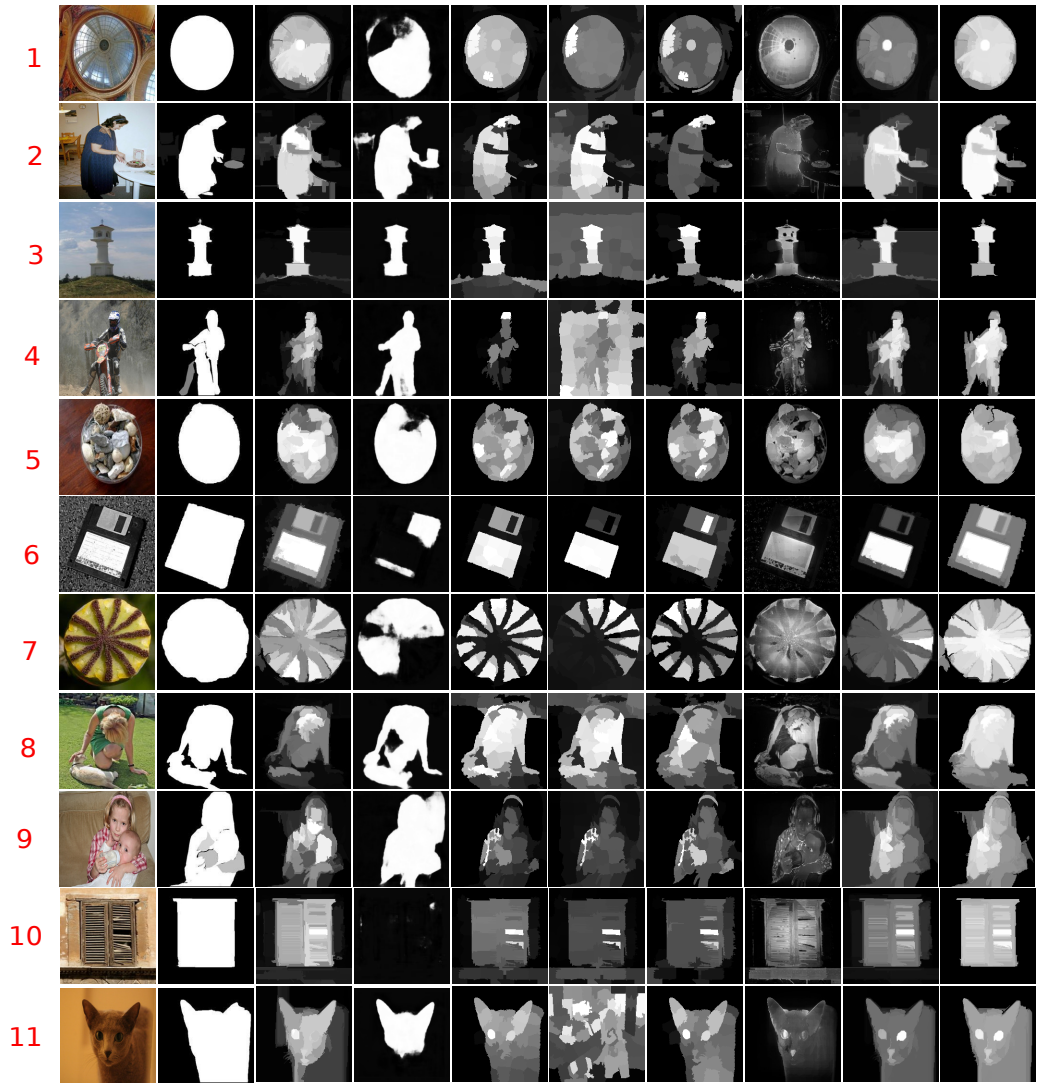
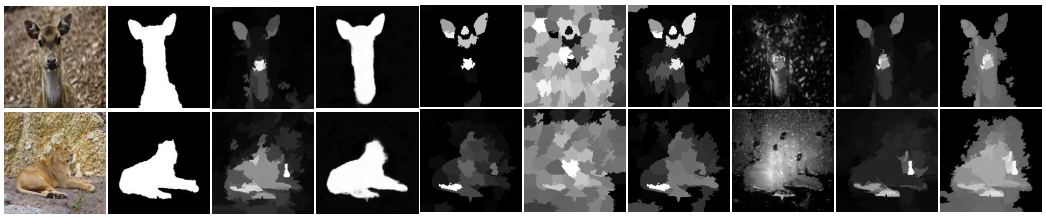


Figure 7.8: Qualitative results of GPFCSOD and compared SOD methods on some sample images from the **SED1**, **ASD**, **ECSSD**, and **PASCAL** datastes.

ground regions while covering the foreground object. Images in rows 5–9 contain non-homogeneous foreground object(s), GPFCSOD is capable of uniformly highlighting the foreground object(s) as well as suppressing the background, while DRFI, as an example of top performing SOD methods, it struggles to completely highlight the foreground object(s). In rows 8–9, where images having two foreground objects, GPFCSOD has potentially detected and highlighted both objects. The last two images are examples of challenging images, because of having low color contrast with background. Unlike the other eight SOD methods, GPFCSOD performs well in these cases.

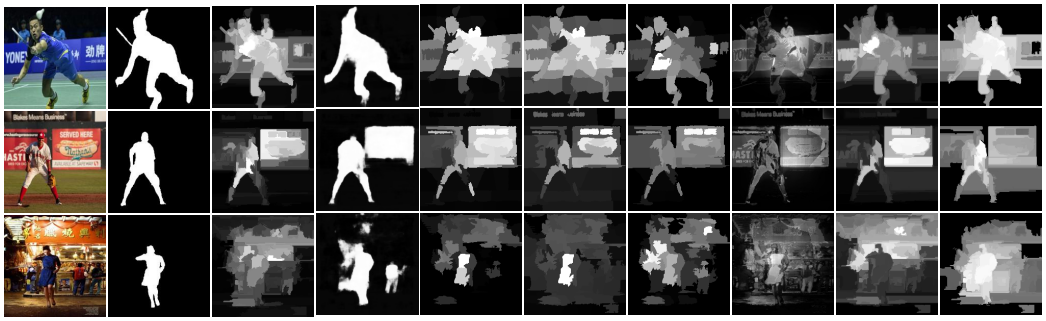
By visualizing different types of complicated examples in Figure 7.8, we show the power of the constructed high-level features in capturing the complex foreground object as a whole when the foreground object consists of different colors and shapes. The constructed high-level features provide an expression of different features learned on the different types of images during the training phase. Moreover, involving regional property features in constructing features by GP helps accurately localise the salient regions in some complicated scenarios. This can be contrasted with most of the CNN-based high-level features (e.g. DCNN), which return blurry edges/boundaries and weak detailed information. GP-based high-level features not only detect the general objects, but they are also good at capturing details and edges/boundaries.

The visual examples in Figure 7.9 illustrate that the foreground object and background have very low color contrast and the background is cluttered in both images. GPFCSOD performs well in locating the foreground object, which is a good progress from the previous GP-based SOD methods such as GPFBC and GPFSFC. However, the performance of GPFCSOD is poorer compared to the two similar images (last two images) in Figure 7.8, caused mainly by the cluttered background of the images in Figure 7.9. Although DCNN fails in Figure 7.8 for the last two images, it performs well in capturing the foreground object with cluttered back-



Original GT DRFI DCNN RBD GMR GS GPFBC GPFSFC GPFC

Figure 7.9: Visual examples where the foreground object and background have very low color contrast and the background is cluttered. Qualitative comparisons between GPFCSOD and the compared SOD methods on the **ECSSD** dataset.

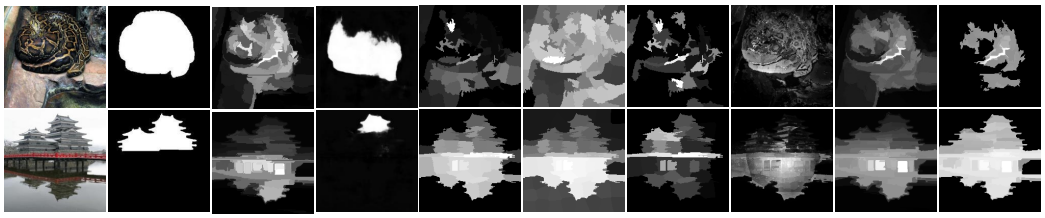


Original GT DRFI DCNN RBD GMR GS GPFBC GPFSFC GPFC

Figure 7.10: Visual examples where some part of background has similar attractive color with the foreground object. Qualitative comparisons between GPFCSOD and the compared SOD methods on the **ECSSD** dataset.

ground.

Figure 7.10 demonstrates a visual comparison of GPFCSOD with respect to the compared SOD methods on some complex background cases. As we observed in Figure 7.10, GPFCSOD struggles in completely suppressing the background regions, although it performs well on detecting and highlighting the foreground object compared to the other SOD methods. Based on observations, GPFCSOD may fail when background has similar color with the foreground object. It may be that the availability



Original GT DRFI DCNN RBD GMR GS GPFBC GPFSFC GPFC

Figure 7.11: Visual examples where the background is complex. Qualitative comparison of GPFCSOD and compared SOD methods on the **SED1** dataset.

of a good backgroundness feature in the input feature set for GPFCSOD would be helpful to correctly present the background.

Although some of the good SOD methods such as DRFI, DCNN, and GPFCSOD provide results close to ground truth, but these SOD methods still have difficulties in some complex cases such as images in Figure 7.11. For example, in the first image, GPFCSOD fails to completely detect and highlight the foreground object, while it performs well on background. In the second image, it wrongly highlights the reflection of the object in the water, but this is subjective. This problem is probably caused by the lack of having enough samples of these types of images in the training samples.

7.4.3 Further Analysis

7.4.3.1 Sample Program Evolved by GP

Here, we provide an example (Figure 7.12) of an evolved GP tree, which is one of the four best evolved programs (constructed features) on the SED1 dataset to demonstrate the structure of the GP-constructed high-level feature and the importance of the selected features. The evolved GP program comprises mathematical operations, $\{+, \sin\}$ as internal nodes and root node, and input features $\{t_1, t_{10}, t_{15}, t_{16}, t_{20}, t_{21}, t_{22}, t_{23}, t_{26}\}$ as terminal nodes (leaves). Table 7.4 gives detailed information about the GP pro-

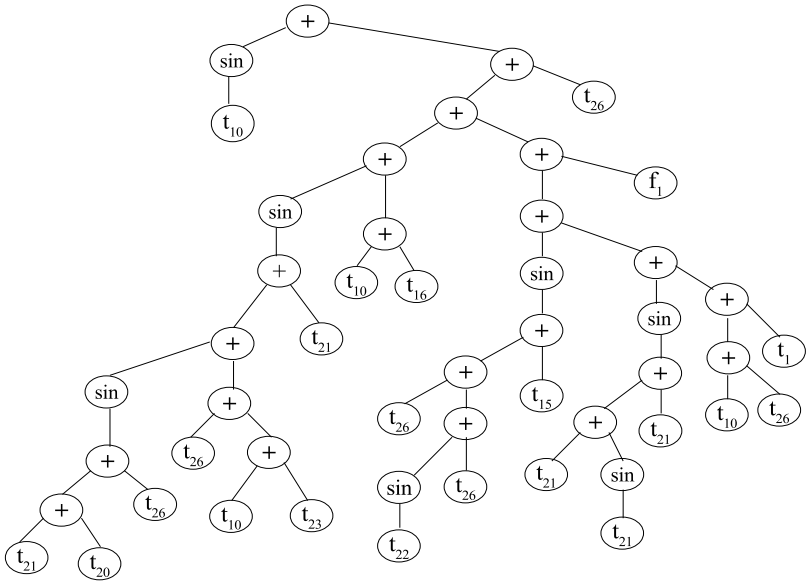


Figure 7.12: Sample program evolved by GP.

gram's leave nodes shown in Figure 7.12. In Figure 7.12, "Term" shows the terminal of the GP program, "Sel times" shows how many times a feature is being selected, "Feature" shows the corresponding feature notation, and "Definition" gives the definition of the selected feature. As can be seen in Table 7.4, features from all the four categories are selected by GP. Among the selected features, three features including $\{f_{11}^2, f_{31}^2, f_7^4\}$ have been selected more than the other features. This fact shows the importance of those three features selected by the evolved GP program. The background weighted contrast feature f_7^4 with 6 times selection in the GP program is an important feature which has been observed in Chapters 5 and 6. The geometric feature f_{31}^2, which captures the position of regions, is useful in describing spatial distribution of both foreground object and background. The average b^* feature, f_{11}^2 with four times selection has good contribution in constructing the new high-level feature. Selecting b^* color from the color space $L^*a^*b^*$ reveals the higher prevalence of the b^* color channel over the images.

Table 7.4: The selected features by sample GP program.

Term	Sel times	Feature	Definition
t_1	2	f_{20}^1	absolute response of LM filters
t_{10}	4	f_{11}^2	average b^* value
t_{15}	1	f_{25}^3	variances of the response of the LM filters
t_{16}	1	f_{28}^3	variances of the response of the LM filters
t_{20}	1	f_{22}^3	variances of the response of the LM filters
t_{21}	5	f_{31}^2	average norm y coordinates
t_{22}	1	f_{27}^3	variances of the response of the LM filters
t_{23}	1	f_{23}^3	variances of the response of the LM filters
t_{26}	6	f_7^4	background weighted contrast feature

7.4.3.2 Visual Example for a High-level Constructed Feature

Figure 7.13 gives a visual example for the constructed features and the produced saliency map. In the image, the foreground object is not homogeneous and has some color similarity with the background, which makes the saliency detection task more challenging. As shown in Figure 7.13, all the constructed features correctly detect and locate the foreground object, although they perform slightly different in highlighting the foreground object. For example, the first feature highlights all parts of the car, but it highlights the wheels with high confidence. The second feature performs very well on completely highlighting different parts of the car, although it does highlight a small portion of the background. The third feature gives higher saliency values to the upper regions of the foreground object, while the fourth feature assigns higher saliency values to the lower regions of the object. However, the saliency map computed by those features consistently highlight different regions of the object and also suppress the incorrectly highlighted background in the second feature. This example indicates that the high-level features constructed by GPFCSOD

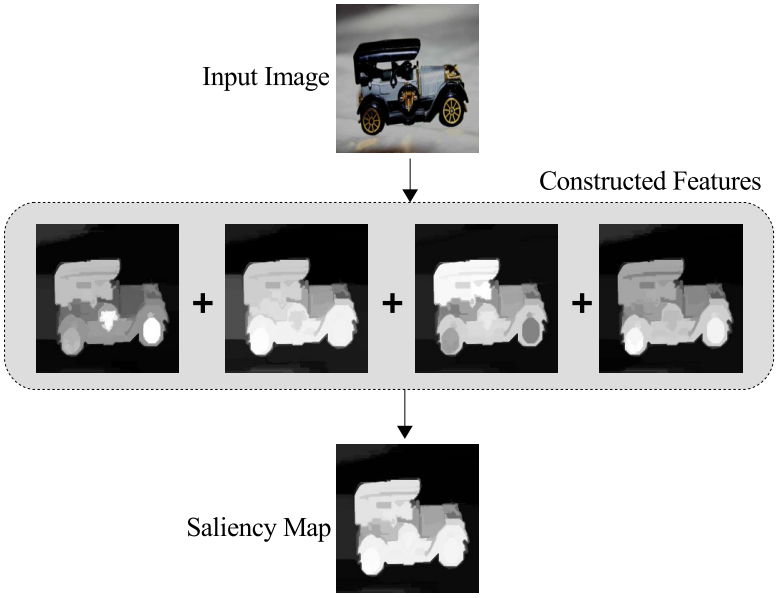


Figure 7.13: Example of a produced saliency map by four high-level constructed features.

from various types of features can effectively impact different regions of the image. Thus, the constructed features have the capability to complement each other to generate better overall saliency maps. In addition, the constructed high-level features provide interpretability, while most of the existing high-level saliency features are hard to interpret.

7.4.3.3 Selected Saliency Features by GP

Table 7.5 demonstrates the features employed in the four GP-based constructed features on the four benchmark datasets. Some features such as $\{f_0, f_{14}, f_{16}, f_{23}\}$ have not been selected by GP in any datasets. In Table 7.5, some features, $\{f_7, f_{29}, f_{35}, f_{36}, f_{37}, f_{40}, f_{67}, f_{68}, f_{72}, f_{83}, f_{85}, f_{93}, f_{96}, f_{98}, f_{100}\}$ have been used in constructing high-level features in all the four datasets. In fact, the mentioned features are important features which have been selected by GP during the evolutionary process and kept to the last gen-

Table 7.5: The employed saliency features in the constructed high-level features by GP on the four datasets.

Dataset	$\boldsymbol{CF}$	Selected Features
SED1	$\boldsymbol{CF^1}$	$f_2, f_4, f_6, f_{37}, f_{38}, f_{57}, f_{62}, f_{64}, f_{71}, f_{72}, f_{76}, f_{88}, f_{95}$
	$\boldsymbol{CF^2}$	$f_{20}, f_{40}, f_{60}, f_{80}, f_{81}, f_{83}, f_{86}, f_{85}, f_{100}$
	$\boldsymbol{CF^3}$	$f_1, f_{12}, f_{17}, f_{21}, f_{36}, f_{46}, f_{66}, f_{68}, f_{74}, f_{84}, f_{94}, f_{96}$
	$\boldsymbol{CF^4}$	$f_7, f_8, f_{30}, f_{35}, f_{63}, f_{67}, f_{82}, f_{89}, f_{90}, f_{93}, f_{98}$
ASD	$\boldsymbol{CF^1}$	$f_3, f_9, f_{11}, f_{15}, f_{20}, f_{29}, f_{31}, f_{35}, f_{65}, f_{66}, f_{68}, f_{75}, f_{79}, f_{84}, f_{93}, f_{98}$
	$\boldsymbol{CF^2}$	$f_{24}, f_{32}, f_{34}, f_{40}, f_{51}, f_{83}, f_{85}, f_{86}, f_{100}$
	$\boldsymbol{CF^3}$	$f_7, f_{25}, f_{37}, f_{38}, f_{39}, f_{50}, f_{67}, f_{73}, f_{92}, f_{94}, f_{96}, f_{97}$
	$\boldsymbol{CF^4}$	$f_5, f_6, f_{18}, f_{36}, f_{49}, f_{52}, f_{55}, f_{58}, f_{71}, f_{72}, f_{76}, f_{88}, f_{91}, f_{95}$
ECSSD	$\boldsymbol{CF^1}$	$f_3, f_{11}, f_{15}, f_{20}, f_{22}, f_{29}, f_{35}, f_{47}, f_{59}, f_{66}, f_{68}, f_{69}, f_{75}, f_{93}, f_{98}$
	$\boldsymbol{CF^2}$	$f_{12}, f_{28}, f_{34}, f_{40}, f_{60}, f_{80}, f_{83}, f_{85}, f_{100}$
	$\boldsymbol{CF^3}$	$f_7, f_{10}, f_{37}, f_{38}, f_{63}, f_{67}, f_{73}, f_{82}, f_{92}, f_{96}$
	$\boldsymbol{CF^4}$	$f_6, f_{19}, f_{36}, f_{46}, f_{49}, f_{58}, f_{62}, f_{64}, f_{71}, f_{72}, f_{95}$
PASCAL	$\boldsymbol{CF^1}$	$f_7, f_{32}, f_{33}, f_{40}, f_{41}, f_{56}, f_{58}, f_{61}, f_{62}, f_{64}, f_{88}, f_{71}, f_{72}, f_{93}, f_{98}$
	$\boldsymbol{CF^2}$	$f_{11}, f_{22}, f_{37}, f_{39}, f_{63}, f_{67}, f_{73}, f_{82}$
	$\boldsymbol{CF^3}$	$f_{13}, f_{17}, f_{21}, f_{24}, f_{28}, f_{36}, f_{44}, f_{54}, f_{55}, f_{60}, f_{77}, f_{78}, f_{81}, f_{83}, f_{85}, f_{86}, f_{99}, f_{100}$
	$\boldsymbol{CF^4}$	$f_4, f_5, f_{18}, f_{29}, f_{35}, f_{42}, f_{45}, f_{59}, f_{68}, f_{69}, f_{74}, f_{75}, f_{94}, f_{96}, f_{97}$

eration to contribute in the best returned individual/program. Table 7.5 shows that GP employs fewer features to get good results on the ECSSD dataset compared to the other datasets, while GP employs a large number of features to produce good solutions on the PASCAL dataset.

7.5 Chapter Summary

This study proposed a GP method to automatically construct new high-level features for SOD. The proposed method constructs features that are more informative than features designed by domain experts. The new high-level features can improve the saliency detection process. Compared to the low-level and hand-crafted features, the constructed high-level features have better generalizability on different image types, since the quantitative and qualitative results show that GPFCSOD can deal with both simple (e.g. ASD) and challenging datasets (e.g., SED1 and PASCAL) and provide consistent results. The existing low-level and hand-crafted features are not accurate enough and insufficient to appropriately detect the entirety of foreground objects and suppress background. The proposed GPFCSOD approach produces learned features that capture salient regions along with suppressing background regions over the whole image. GPFCSOD does not need any human intervention beyond the initial setting of GP parameters.

In this work, we limited the input feature set of GP to keep the search space small. Thus, the learning process will not be dominated by a few features. In addition, feeding different input feature sets to GP allows it to generate diverse saliency features. Moreover, the GP-based constructed features have better interpretability compared to CNN-based features. The quantitative and qualitative evaluation shows that GPFCSOD can effectively construct high-level features that have the capability to enhance the output of the SOD methods.

Based on the quantitative and qualitative results reported by GPFCSOD, this method can handle some challenging images when the foreground object is not homogeneous, the color contrast is low between the foreground object and background, or the background is complex. However, this method still has limitations on some other challenging image types such as image with attractive color in both foreground object and

background. Providing enough samples of complicated cases in the training dataset is one solution that is likely to address limitations of the current method. Although the proposed method has successfully constructed new features, it still requires the features to be manually combined. Hence, as a future work, the third stage can be replaced by an extra GP run that automatically combines those features. Another potential way is to design a feature fusion method to consider positive and negative aspects of different constructed high-level features in producing the final saliency map.

Chapter 8

Conclusions

This chapter provides the conclusions for the thesis, describes achieved objectives, and outlines possible directions for future work.

The overall goal of this thesis was to apply feature manipulation on saliency features by developing domain independent EC techniques including PSO and GP to evolve solutions that are capable of discovering the complex interactions among saliency features, selecting informative features, and constructing new high-level features that are robust to different image types. This goal has been achieved by developing EC techniques in three ways of feature manipulation: 1) PSO for feature weighting, 2) GP for both feature selection and feature combination, and 3) GP for feature construction. The developed methods were compared with state-of-the-art and benchmark methods based on different evaluation criteria on the benchmark datasets containing different types of images. The quantitative and qualitative results demonstrate that the proposed methods have achieved either comparable or better results than the state-of-the-art method and benchmark methods.

8.1 Achieved Objectives

This thesis has achieved the following research objectives:

- In Chapter 3, a PSO-based SOD method was proposed to evolve suitable weight vectors for features for the different benchmark datasets. The proposed method has the ability to consider their complementary characteristics during combination for producing saliency maps. The effectiveness of the evolved weights on the performance of SOD was studied.

- Chapter 4 developed a bottom-up SOD method that takes features and produce the final saliency map. To fulfill this objective, a new informative foreground feature and a new informative background feature have been manually constructed and a new feature combination framework has been designed to combine the constructed features to compute the final saliency map. This thesis investigated the importance of complementary characteristics of the saliency features and the way of combining those features. In this objective, the developed method was an unsupervised method which did not use any ground truth during saliency detection. The quantitative and qualitative results showed the effectiveness of the manually constructed foreground and background features compared to the individual features. The experimental results show that the proposed method achieved good performance regarding computational time and saliency detection.

- Chapter 5 developed a GP-based approach to automatically construct foreground and background saliency features. This objective mainly focuses on automating the feature construction task using the GP algorithm to relieve domain knowledge and human intervention. The proposed method improved the SOD performance by introducing more informative features. This work showed that GP has a

promising capability for exploring a search space of saliency features and finding a suitable way to combine different input saliency features. The results show that the GP-based constructed features have better performance than the manually designed ones, which lead to improve the final performance of the SOD method based on different datasets.

- Chapter 6 proposed a GP-based method to automatically select and combine features to produce the final saliency map. The proposed method can incorporate any additional features and select the complementary features from a large and complex saliency feature space without making any assumption or using domain knowledge. The quantitative and qualitative results reveal that the proposed method showed promising results by significantly outperforming one of the well-know and recent CNN methods (DCNN) and other benchmark methods on two SOD datasets out of four.

- Chapter 7 developed a GP method to automatically construct new high-level saliency features and designed a new feature subset preparation method to ensure the diversity of the constructed features. Unlike Chapter 6, in this chapter, we employ the feature subset preparation method to limit the input feature set of GP to keep the search space smaller. Thus, the learning process will not be dominated by a few features and good features can have a chance to contribute to the final solution. This objective also focused on addressing the limitations and difficulties of manually constructing saliency features. Compared with the low-level and hand-crafted features, the constructed high-level features have some advantages: more informative and accurate on the challenging images, better generalizability on different image types, better understanding of feature interaction, improving the saliency detection process, does not need any human intervention beyond initial setting of GP pa-

rameters. Moreover, the GP-based constructed features have better interpretability compared to CNN-based features. The quantitative and qualitative evaluations show that the proposed method has significantly better performance than both automatically designed (DCNN) and domain expert hand-crafted features (DRFI).

8.2 Main Conclusions

Overall, this thesis finds that PSO and GP can be used effectively for feature manipulation tasks, including feature weighting, feature selection, and feature construction, on saliency features to improve the performance of SOD.

This section discusses the major conclusions drawn from the five contribution chapters (Chapter 3 to Chapter 7)

8.2.1 PSO for Weighting Saliency Features

Chapter 3 investigates the capability of PSO for evolving suitable weight vectors for the features on the different benchmark datasets. It has been found that the performance of the SOD method has been improved after assigning weights to the features compared to the non-weighted one. PSO showed that it has the ability to consider complementary characteristics of the features during the combination process. From the experimental results of Chapter 3, it has been concluded that different datasets favour different weights for the features in the linear feature combination. One feature may have higher weight for one dataset, but lower weight for another one. It also has been found that PSO assigns zero or low weights (close to zero) when the features are redundant or do not complement other features. Based on the results of the PSO-based weighting method, it can be concluded that not all the employed features in this experiment are required to be involved for the feature combination.

8.2.2 Bottom-up SOD Method

Chapter 4 investigates *manually* constructing foreground and background features and designing a feature combination framework. From the experimental results of Chapter 4, it can be concluded that the proposed method improves the average precision and speeds up the previous work in Chapter 3. The proposed method is unsupervised and developed mainly based on domain knowledge in SOD. It has been found that the constructed foreground and background features are more informative and better than the individual features at representing the foreground object(s) and background. The proposed method in Chapter 4 makes a good balance (trade-off) between computational time and performance and it is a reasonable choice when a task requires a method which is relatively fast and has a good performance.

8.2.3 GP for Constructing Foreground and Background Saliency Features

Chapter 5 proposes a GP-based approach to *automatically* construct foreground and background features. The GP-based features are constructed in an automatic way and they can outperform manually constructed features in Chapter 4. It has been found that GP is robust towards the changes in the input feature set and it does not require domain knowledge and human intervention. Moreover, GP improves the SOD performance by introducing more informative features to the SOD domain. It has been also found that GP has a promising capability in exploring a wide search space of features and finding a suitable way to combine different input features.

8.2.4 GP for Feature Selection and Feature Combination

Chapter 6 proposes a new GP-based SOD method for automatically detecting salient objects. From Chapter 6, it has been found that GP can

effectively handle large and complex search space of features. It makes no assumption on linear superposition or equal weights of features and it does not require domain knowledge. Moreover, GP has the ability to tackle a wide range of saliency features from different segmentation levels and explore various mathematical expressions for the feature combination stage. From the experimental results in Chapter 6, it has been found that GP can effectively choose the features which are relevant and can complement each other, thus, the final combination of those features results in a good saliency map.

Considering the qualitative results in Chapter 6, it has been found that the combination of different types of features, such as color, backgroundness, appearance and geometric, is important in properly detecting salient objects. Therefore, adding the informative and different types of saliency feature, such such as texture information, generic properties including appearance and geometric features to the input feature set of Chapter 5 was helpful in producing more accurate results.

8.2.5 GP for Constructing High-level Saliency Features

Chapter 7 proposes the first GP-based approach to automatically construct high-level features for SOD, where human knowledge is not required to handle the design of the mathematical formula. It has been found from the qualitative results in Chapter 7 on the complicated examples that the constructed high-level features can successfully capture the complex foreground object(s) as a whole when the foreground object consists of different colors and shapes. The constructed high-level features provide an expression of different features learned on the different types of images during the training phase.

The GP-based constructed high-level features from various feature subsets can effectively impact different regions of the image. Thus, the constructed features have the capability to complement each other to gen-

erate better overall saliency maps.

In addition, the constructed high-level features provide interpretability, while most of the existing high-level features for SOD are hard to interpret. GP-based high-level features not only detect the general objects, but they are also good at capturing details and edges/boundaries unlike the CNN-based high-level features (e.g. DCNN), which return blurry edges/boundaries and weak detailed information.

Chapter 7 revealed that the proposed GP-based method can potentially cope with a small number of samples for training to obtain a good generalization as long as the given training data has enough information to represent the distribution of the data, unlike deep learning CNN-based methods which have more failure cases for small datasets.

8.3 Future Work

This section provides some possible research directions for further investigation.

8.3.1 Multi-tree GP for Multiple High-level Feature Construction

The Chapter 7 proposed a GP method for multiple high-level feature construction, however, the proposed method has been designed to construct one high-level feature at the time. However, it is possible to utilise the multi-tree representation of GP to construct multiple high-level features simultaneously. Although multi-tree GP might be more complex, it will make the feature construction process computationally more efficient.

8.3.2 GP for Automatic Feature Extraction

In this thesis, as GP showed promising results for feature manipulation tasks in SOD, it is worth considering the use of GP for automatically extracting saliency features from the raw images and investigate whether GP can successfully generate new features that are as informative as the hand-crafted features. The idea of studying feature extraction arises for several reasons. Different applications may need different new features to be designed, since features are not universal or expected to be perfect for all applications. Different types of image features are extracted by using different techniques, such as Fourier or wavelet transforms. Exploring a good feature set (or feature sets based on image types) for SOD is heavily depended on having specific knowledge about the domain and existing features [9]. Since obtaining this background knowledge is a difficult and time-consuming task, and it can not be guaranteed to be complete enough or correct in all cases, it will be favourable to have a method that can automatically extract informative feature(s) directly from the raw pixel values with no human intervention.

8.3.3 Unsupervised Feature Manipulation

In this thesis, we mostly focused on developing supervised feature manipulation algorithms. However, when only limited prior knowledge is available for supervising the algorithm during the training process, unsupervised approaches will be more helpful to address problems. In image analysis, when the ground truth of the dataset is not available, unsupervised methods have the potential to be applied to the training process. Moreover, manually annotating process or identifying the ground truth in images is an expensive and time-consuming task [170]. The process will cost time, money and human effort due to asking people to label salient object (e.g. draw a rectangle around the salient object) for a large number of images. Moreover, the process has the potential to have the problem of

labelling inconsistency (subjectivity) due to labelling by different people based on their understanding of salient object for a particular image. Figure 8.1 shows some examples for labelling inconsistency, in the first row, the first labelling only detects the signpost as a salient object, the second labelling identifies both the building and the signpost as the salient objects, and the third labelling only detects the building as the salient object. Finally, the potential of both supervised and unsupervised approaches still need to be investigated deeply.

8.3.4 Generalizability vs Particularizability

In this thesis, neither generalizability nor particularizability (image grouping) has been particularly investigated. In fact, this thesis attempt to generally cover both areas and not to be biased. Therefore, it has not provided different solutions for different image groups (e.g. images with large salient objects, images with small salient objects), however, it produces different solutions for different SOD datasets with images of different levels of complexity.

8.3.4.1 Particularizability (image grouping)

To investigate the SOD problem in a deep level, it is required to have comprehensive understanding regarding the nature of different image types. Moreover, the majority of SOD methods still have difficulties on some specific challenging cases, due to only focusing on solving the problem in general aspects. Saliency images can be divided to different image groups based on their characteristics. Therefore, it is desired to develop a method which can properly divide the saliency images to different groups. Studying and exploring each image group and finding the most related solution for each group will be a big step in further improvement of the SOD problem.

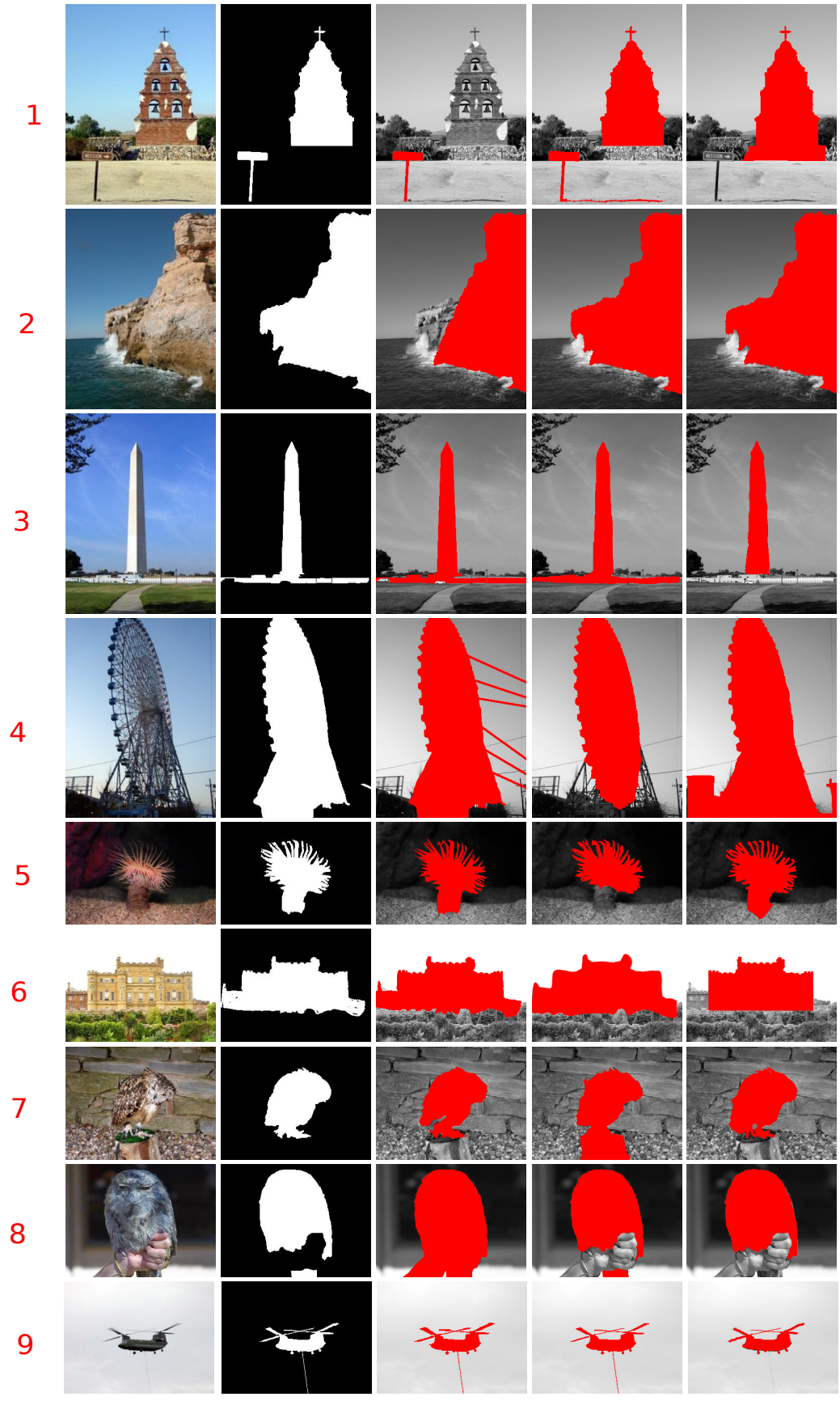


Figure 8.1: Some examples for labelling inconsistency (subjective) in SOD images.

8.3.4.2 Generalizability

The majority of heuristic methods exhibit low robustness to difficult stimuli and struggle to generalize to challenging cases of saliency detection, such as similar foreground and background, cluttered background and multiple salient objects. Hence, new methods are required to focus more on evolving automatic/artificial learning at various stages of the visual attention model in order to extend its generalizability to challenging cases in machine vision.

Generally finding a solution which works generally well on all different types of saliency images is difficult and complicated, however, it would be a robust and efficient solution which can provide generalizability.

8.3.5 Enrich the SOD Datasets with more Samples

In Chapter 7, the constructed high-level saliency features have been shown to be promising alternatives to those domain-expert designed features, but they still have some limitations in complex images. Providing enough samples of the complicated cases in SOD datasets can be helpful in the training stage of the algorithm.

For example, SOD has been used in a wide range of application scenarios, such as autonomous vehicles, video games, and medical image processing, since SOD is helpful at locating and identifying saliency object(s). Wang et al. [177] suggested to collect SOD datasets specific for the application domains, since domain-specific data can help build SOD methods that can better detect and segment the salient object(s) under specific task settings than generally trained SOD methods [26].

Bibliography

[1] ACHANTA, R., HEMAMI, S., ESTRADA, F., AND SUSSTRUNK, S. Frequency-tuned salient region detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2009), IEEE, pp. 1597–1604.

[2] ACHANTA, R., SHAJI, A., SMITH, K., LUCCHI, A., FUA, P., AND SÜSSTRUNK, S. Slic superpixels. Tech. rep., 2010.

[3] ACHANTA, R., AND SÜSSTRUNK, S. Saliency detection using maximum symmetric surround. In *Proceedings of the 17th IEEE International Conference on Image Processing* (2010), IEEE, pp. 2653–2656.

[4] ADAMS, A., BAEK, J., AND DAVIS, M. A. Fast high-dimensional filtering using the permutohedral lattice. In *Proceedings of the Computer Graphics Forum* (2010), vol. 29, Wiley, pp. 753–762.

[5] AHMED, S., ZHANG, M., PENG, L., AND XUE, B. Multiple feature construction for effective biomarker identification and classification using genetic programming. In *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation* (2014), ACM, pp. 249–256.

[6] AIN, Q. U., XUE, B., AL-SAHAF, H., AND ZHANG, M. Genetic programming for feature selection and feature construction in skin cancer image classification. In *Proceedings of the 15th Pacific Rim Interna-*

tional Conference on Artificial Intelligence (2018), vol. 11012, Springer, pp. 732–745.

[7] AL-SAHAF, H. Genetic programming for automatically synthesising robust image descriptors with a small number of instances.

[8] AL-SAHAF, H., SONG, A., NESHATIAN, K., AND ZHANG, M. Extracting image features for classification by two-tier genetic programming. In *Proceedings of the 2012 IEEE Congress on Evolutionary Computation* (2012), IEEE, pp. 1–8.

[9] AL-SAHAF, H., SONG, A., NESHATIAN, K., AND ZHANG, M. Two-tier genetic programming: Towards raw pixel-based image classification. *Expert Systems with Applications 39*, 16 (2012), 12291–12301.

[10] AL-SAHAF, H., XUE, B., AND ZHANG, M. A multitree genetic programming representation for automatically evolving texture image descriptors. In *Proceedings of the 11th International Conference on Simulated Evolution and Learning* (2017), Springer, pp. 499–511.

[11] AL-SAHAF, H., ZHANG, M., AL-SAHAF, A., AND JOHNSTON, M. Keypoints detection and feature extraction: A dynamic genetic programming approach for evolving rotation-invariant texture image descriptors. *IEEE Transactions on Evolutionary Computation 21*, 6 (2017), 825–844.

[12] AL-SAHAF, H., ZHANG, M., AND JOHNSTON, M. Genetic programming evolved filters from a small number of instances for multiclass texture classification. In *Proceedings of the 29th International Conference on Image and Vision Computing New Zealand* (2014), ACM, pp. 84–89.

[13] ALEXE, B., DESELAERS, T., AND FERRARI, V. What is an object? In *Proceedings of the 2010 IEEE Conference on Computer Vision and Pattern Recognition* (2010), IEEE, pp. 73–80.

[14] ALEXE, B., DESELAERS, T., AND FERRARI, V. Measuring the objectness of image windows. *IEEE Transactions on Pattern Analysis and Machine Intelligence 34*, 11 (2012), 2189–2202.

[15] ALPERT, S., GALUN, M., BRANDT, A., AND BASRI, R. Image segmentation by probabilistic bottom-up aggregation and cue integration. *IEEE Transactions on Pattern Analysis and Machine Intelligence 34*, 2 (2011), 315–327.

[16] AMIT, Y. *2D object detection and recognition: Models, algorithms, and networks*. MIT Press, 2002.

[17] ANDREOPOULOS, A., AND TSOTSOS, J. K. 50 years of object recognition: Directions forward. *Computer vision and image understanding 117*, 8 (2013), 827–891.

[18] AVIDAN, S., AND SHAMIR, A. Seam carving for content-aware image resizing. In *Proceedings of ACM Transactions on Graphics* (2007), vol. 26, ACM, pp. 1–10.

[19] AZEVEDO, G. L., CAVALCANTI, G. D., AND CARVALHO FILHO, E. C. An approach to feature selection for keystroke dynamics systems based on PSO and feature weighting. In *Proceedings of the 2007 IEEE Congress on Evolutionary Computation* (2007), IEEE, pp. 3577–3584.

[20] BÄCK, T., FOGEL, D. B., AND MICHALEWICZ, Z. *Handbook of evolutionary computation*. CRC Press, 1997.

[21] BÄCK, T., HAMMEL, U., AND SCHWEFEL, H.-P. Evolutionary computation: Comments on the history and current state. *IEEE Transactions on Evolutionary Computation 1*, 1 (1997), 3–17.

[22] BAI, Q. Analysis of particle swarm optimization algorithm. *Computer and Information Science 3*, 1 (2010).

[23] BANZHAF, W., NORDIN, P., KELLER, R. E., AND FRANCONE, F. D. *Genetic programming: an introduction*, vol. 1. Morgan Kaufmann, 1998.

[24] BONABEAU, E., DORIGO, M., AND THERAULAZ, G. *From natural to artificial swarm intelligence*. Oxford University Press, 1999.

[25] BORJI, A., , M.-M., JIANG, H., AND LI, J. Salient object detection: A benchmark. *IEEE Transactions on Image Processing 24*.

[26] BORJI, A. Saliency prediction in the deep learning era: An empirical investigation. *arXiv preprint arXiv:1810.03716* (2018).

[27] BORJI, A., CHENG, M., JIANG, H., AND LI, J. Salient object detection: A survey. *CoRR abs/1411.5878* (2014).

[28] BORJI, A., AND ITTI, L. Exploiting local and global patch rarities for saliency detection. In *Proceedings of 2012 IEEE Conference on Computer Vision and Pattern Recognition* (2012), IEEE, pp. 478–485.

[29] BORJI, A., AND ITTI, L. State-of-the-art in visual attention modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence 35*, 1 (2012), 185–207.

[30] BORJI, A., AND ITTI, L. State-of-the-art in visual attention modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence 35*, 1 (2013), 185–207.

[31] CARDIE, C. Using decision trees to improve case-based learning. In *Proceedings of the 10th International Conference on Machine Learning* (1993), pp. 25–32.

[32] CHAKRABORTY, B. Feature subset selection by particle swarm optimization with fuzzy fitness function. In *Proceedings of the 3rd International Conference on Intelligent System and Knowledge Engineering* (2008), vol. 1, IEEE, pp. 1038–1042.

[33] CHEN, Q. Improving the generalisation of genetic programming for symbolic regression.

[34] CHEN, Q., ZHANG, M., AND XUE, B. Genetic programming with embedded feature construction for high-dimensional symbolic regression. *Intelligent and Evolutionary Systems* (2017), 87.

[35] CHENG, M.-M., MITRA, N. J., HUANG, X., AND HU, S.-M. Salientshape: group saliency in image collections. *The Visual Computer 30*, 4 (2014), 443–453.

[36] CHENG, M.-M., MITRA, N. J., HUANG, X., TORR, P. H., AND HU, S.-M. Global contrast based salient region detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence 37*, 3 (2015), 569–582.

[37] DASH, M., AND LIU, H. Feature selection for classification. *Intelligent data analysis 1*, 1-4 (1997), 131–156.

[38] DASH, M., AND LIU, H. Consistency-based search in feature selection. *Artificial Intelligence 151*, 1-2 (2003), 155–176.

[39] DUAN, L., WU, C., MIAO, J., QING, L., AND FU, Y. Visual saliency detection by spatially weighted dissimilarity. In *Proceeding of 2011 IEEE Conference on Computer Vision and Pattern Recognition* (2011), IEEE, pp. 473–480.

[40] ENGELBRECHT, A. P. *Computational intelligence: An introduction*. John Wiley & Sons, 2007.

[41] ESPEJO, P. G., VENTURA, S., AND HERRERA, F. A survey on the application of genetic programming to classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part C 40*, 2 (2010), 121–144.

[42] FAN, Q., AND QI, C. Two-stage salient region detection by exploiting multiple priors. *Journal of Visual Communication and Image Representation 25*, 8 (2014), 1823–1834.

[43] FELZENSZWALB, P. F., AND HUTTENLOCHER, D. P. Efficient graph-based image segmentation. *International Journal of Computer Vision 59*, 2 (2004), 167–181.

[44] FILALI, I., ALLILI, M. S., AND BENBLIDIA, N. Multi-scale salient object detection using graph ranking and globallocal saliency refinement. *Signal Processing: Image Communication 47* (2016), 380 – 401.

[45] FU, H., CAO, X., AND TU, Z. Cluster-based co-saliency detection. *IEEE Transactions on Image Processing 22*, 10 (2013), 3766–3778.

[46] FU, H., XIAO, Z., DELLANDRÉA, E., DOU, W., AND CHEN, L. Image categorization using ESFS: a new embedded feature selection method based on SFS. In *International Conference on Advanced Concepts for Intelligent Vision Systems* (2009), Springer, pp. 288–299.

[47] FU, W., JOHNSTON, M., AND ZHANG, M. Automatic construction of invariant features using genetic programming for edge detection. In *Australasian Joint Conference on Artificial Intelligence* (2012), Springer, pp. 144–155.

[48] FU, W., JOHNSTON, M., AND ZHANG, M. Low-level feature extraction for edge detection using genetic programming. *IEEE Transactions on Cybernetics 44*, 8 (2014), 1459–1472.

[49] FU, W., JOHNSTON, M., AND ZHANG, M. Genetic programming for edge detection: a gaussian-based approach. *Soft Comput. 20*, 3 (2016), 1231–1248.

[50] GOFERMAN, S., ZELNIK-MANOR, L., AND TAL, A. Context-aware saliency detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence 34*, 10 (2011), 1915–1926.

[51] GOFERMAN, S., ZELNIK-MANOR, L., AND TAL, A. Context-aware saliency detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence 34*, 10 (2012), 1915–1926.

[52] GOLDBERG, D. E., AND HOLLAND, J. H. Genetic algorithms and machine learning. *Machine Learning 3*, 2 (1988), 95–99.

[53] GOPALAKRISHNAN, V., HU, Y., AND RAJAN, D. Random walks on graphs to model saliency in images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2009), IEEE, pp. 1698–1705.

[54] GOPALAKRISHNAN, V., HU, Y., AND RAJAN, D. Unsupervised feature selection for salient object detection. In *Proceedings of Asian Conference on Computer Vision* (2010), Springer, pp. 15–26.

[55] GUYON, I., AND ELISSEEFF, A. An introduction to variable and feature selection. *Journal of Machine Learning Research 3* (2003), 1157–1182.

[56] GUYON, I., GUNN, S., NIKRAVESH, M., AND ZADEH, L. A. *Feature extraction: Foundations and applications*, vol. 207. Springer, 2008.

[57] HAN, J., ZHANG, D., CHENG, G., LIU, N., AND XU, D. Advanced deep-learning techniques for salient and category-specific object detection: a survey. *IEEE Signal Processing Magazine 35*, 1 (2018), 84–100.

[58] HANCER, E., XUE, B., KARABOGA, D., AND ZHANG, M. A binary ABC algorithm based on advanced similarity scheme for feature selection. *Applied Soft Computing 36* (2015), 334–348.

[59] HASSAN, R., COHANIM, B., DE WECK, O., AND VENTER, G. A comparison of particle swarm optimization and the genetic algorithm. In *Proceedings of the 46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference* (2005).

[60] HAYHOE, M., AND BALLARD, D. Eye movements in natural behavior. *Trends in cognitive sciences 9*, 4 (2005), 188–194.

[61] HEIKKILÄ, M., PIETIKÄINEN, M., AND SCHMID, C. Description of interest regions with local binary patterns. *Pattern recognition 42*, 3 (2009), 425–436.

[62] HEISELE, B., SERRE, T., AND POGGIO, T. A component-based framework for face detection and identification. *International Journal of Computer Vision 74*, 2 (2007), 167–181.

[63] HOFFMAN, D. Object categorization: Computer and human perspectives, 2009.

[64] HOU, Q., CHENG, M.-M., HU, X., BORJI, A., TU, Z., AND TORR, P. Deeply supervised salient object detection with short connections. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017), IEEE, pp. 5300–5309.

[65] HOU, X., AND ZHANG, L. Saliency detection: A spectral residual approach. In *Proceedings of the 2007 IEEE Conference on Computer Vision and Pattern Recognition* (2007), IEEE, pp. 1–8.

[66] HU, P., SHUAI, B., LIU, J., AND WANG, G. Deep level sets for salient object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017), pp. 2300–2309.

[67] HUANG, T. Computer vision: Evolution and promise. *CERN EUROPEAN ORGANIZATION FOR NUCLEAR RESEARCH-REPORTS-CERN* (1996), 21–26.

[68] HUO, L., JIAO, L., WANG, S., AND YANG, S. Object-level saliency detection with color attributes. *Pattern Recognition 49* (2016), 162–173.

[69] IQBAL, M., NAQVI, S. S., BROWNE, W. N., HOLLITT, C. P., AND ZHANG, M. Salient object detection using learning classifier systems that compute action mappings. In *Proceedings of 2014 Genetic*

and Evolutionary Computation Conference (2014), ACM Press, pp. 525 – 532.

[70] IQBAL, M., XUE, B., AL-SAHAF, H., AND ZHANG, M. Cross-domain reuse of extracted knowledge in genetic programming for image classification. *IEEE Transactions on Evolutionary Computation 21*, 4 (2017), 569–587.

[71] ISLAM, M. R. Sample size and its role in central limit theorem (clt). *International journal of physics & mathematics 1*, 1 (2018), 37–47.

[72] ITTI, L. *Models of bottom-up and top-down visual attention*. PhD thesis, California Institute of Technology, 2000.

[73] ITTI, L., AND BALDI, P. F. Bayesian surprise attracts human attention. In *Advances in neural information processing systems* (2006), pp. 547–554.

[74] ITTI, L., AND KOCH, C. Computational modelling of visual attention. *Nature reviews neuroscience 2*, 3 (2001), 194.

[75] ITTI, L., KOCH, C., AND NIEBUR, E. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence 20*, 11 (1998), 1254–1259.

[76] JIANG, H., WANG, J., YUAN, Z., WU, Y., ZHENG, N., AND LI, S. Salient object detection: A discriminative regional feature integration approach. In *Proceedings of 2013 IEEE Conference on Computer Vision and Pattern Recognition* (2013), IEEE, pp. 2083–2090.

[77] JUDD, T., EHINGER, K., DURAND, F., AND TORRALBA, A. Learning to predict where humans look. In *Proceedings of the 2009 IEEE 12th International Conference on Computer Vision* (2009), IEEE, pp. 2106–2113.

[78] KENNEDY, J. Particle swarm optimization. In *Encyclopedia of Machine Learning*. Springer, 2011, pp. 760–766.

[79] KENNEDY, J., AND EBERHART, R. C. Particle swarm optimization. In *Proceedings of the IEEE International Conference on Neural Networks* (1995), pp. 1942–1948.

[80] KHAN, W. Image segmentation techniques: A survey. *Journal of Image and Graphics 1*, 4 (2013), 166–170.

[81] KOFFKA, K. *Principles of Gestalt psychology*, vol. 44. Routledge, 2013.

[82] KOZA, J. R. *Genetic programming: On the programming of computers by means of natural selection*, vol. 1. MIT press, 1992.

[83] KOZA, J. R. *Genetic programming III: Darwinian invention and problem solving*, vol. 3. Morgan Kaufmann, 1999.

[84] KOZA, J. R., KEANE, M. A., STREETER, M. J., MYDLOWEC, W., YU, J., AND LANZA, G. *Genetic programming IV: Routine human-competitive machine intelligence*, vol. 5. Springer Science & Business Media, 2006.

[85] KRAWIEC, K. Genetic programming-based construction of features for machine learning and knowledge discovery tasks. *Genetic Programming and Evolvable Machines 3*, 4 (2002), 329–343.

[86] KUMAR, N. Thresholding in salient object detection: a survey. *Multimedia Tools and Applications 77*, 15 (2018), 19139–19170.

[87] KUMAR, V., AND MINZ, S. Feature selection. *Smart Computing Review 4*, 3 (2014), 211–229.

[88] LANE, M. C., XUE, B., LIU, I., AND ZHANG, M. Gaussian based particle swarm optimisation and statistical clustering for feature selection. In *Proceedings of the European Conference on Evolutionary Computation in Combinatorial Optimization* (2014), Springer, pp. 133–144.

[89] LANGDON, W. B., POLI, R., MCPHEE, N. F., AND KOZA, J. R. Genetic programming: An introduction and tutorial, with a survey of techniques and applications. In *Computational Intelligence: A compendium*. Springer, 2008, pp. 927–1028.

[90] LANGLEY, P., ET AL. Selection of relevant features in machine learning. In *Proceedings of the AAAI Fall Symposium on Relevance* (1994), vol. 184, pp. 245–271.

[91] LAVRENKO, V., ALLAN, J., DEGUZMAN, E., LAFLAMME, D., POLLARD, V., AND THOMAS, S. Relevance models for topic detection and tracking. In *Proceedings of the 2nd International Conference on Human Language Technology Research* (2002), Morgan Kaufmann Publishers Inc., pp. 115–121.

[92] LEARNED-MILLER, E. G. Introduction to computer vision. *University of Massachusetts, Amherst* (2011).

[93] LEE, G., TAI, Y.-W., AND KIM, J. Deep saliency with encoded low level distance map and high level features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), pp. 660–668.

[94] LENSEN, A., AL-SAHAF, H., ZHANG, M., AND XUE, B. Genetic programming for region detection, feature extraction, feature construction and classification in image data. In *Proceeding of the European Conference on Genetic Programming*, vol. 9594. Springer, 2016, pp. 51–67.

[95] LEUNG, B. *Component-based car detection in street scene images*. PhD thesis, Massachusetts Institute of Technology, 2004.

[96] LÉVY, P. *Collective intelligence*. Plenum/Harper Collins, 1997.

[97] LI, G., AND YU, Y. Visual saliency based on multiscale deep features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2015), pp. 5455–5463.

[98] LI, G., AND YU, Y. Deep contrast learning for salient object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), pp. 478–487.

[99] LI, X., LU, H., ZHANG, L., RUAN, X., AND YANG, M.-H. Saliency detection via dense and sparse reconstruction. In *Proceedings of the IEEE International Conference on Computer Vision* (2013), pp. 2976–2983.

[100] LI, X., ZHAO, L., WEI, L., YANG, M.-H., WU, F., ZHUANG, Y., LING, H., AND WANG, J. Deepsaliency: Multi-task deep neural network model for salient object detection. *IEEE Transactions on Image Processing 25*, 8 (2016), 3919–3930.

[101] LI, Y., HOU, X., KOCH, C., REHG, J. M., AND YUILLE, A. L. The secrets of salient object segmentation. Georgia Institute of Technology.

[102] LIANG, M., AND HU, X. Feature selection in supervised saliency prediction. *IEEE Transactions on Cybernetics 45*, 5 (2015), 914–926.

[103] LIANG, Y. Genetic programming for supervised figure-ground image segmentation.

[104] LIANG, Y., ZHANG, M., AND BROWNE, W. N. A supervised figure-ground segmentation method using genetic programming. In *European Conference on the Applications of Evolutionary Computation* (2015), Springer, pp. 491–503.

[105] LIN, M., ZHANG, C., AND CHEN, Z. Predicting salient object via multi-level features. *Neurocomputing 205* (2016), 301–310.

[106] LIN, S.-W., YING, K.-C., CHEN, S.-C., AND LEE, Z.-J. Particle swarm optimization for parameter determination and feature selection of support vector machines. *Expert Systems with Applications 35*, 4 (2008), 1817–1824.

[107] LIN, Y., AND BHANU, B. Object detection via feature synthesis using MDL-based genetic programming. *IEEE Transactions on Systems, Man, and Cybernetics, Part B 35*, 3 (2005), 538–547.

[108] LIU, F., AND GLEICHER, M. Region enhanced scale-invariant saliency detection. In *Proceedings of the 2006 IEEE International Conference on Multimedia and Expo* (2006), IEEE, pp. 1477–1480.

[109] LIU, H., AND MOTODA, H. *Feature extraction, construction and selection: A data mining perspective*, vol. 453. Springer, 1998.

[110] LIU, N., AND HAN, J. Dhsnet: Deep hierarchical saliency network for salient object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), pp. 678–686.

[111] LIU, N., HAN, J., AND YANG, M.-H. Picanet: Learning pixel-wise contextual attention for saliency detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018), pp. 3089–3098.

[112] LIU, R., CAO, J., LIN, Z., AND SHAN, S. Adaptive partial differential equation learning for visual saliency detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2014), pp. 3866–3873.

[113] LIU, T., YUAN, Z., SUN, J., WANG, J., ZHENG, N., TANG, X., AND SHUM, H.-Y. Learning to detect a salient object. *IEEE Transactions on Pattern Analysis and Machine Intelligence 33*, 2 (2011), 353–367.

[114] LUO, Z., MISHRA, A. K., ACHKAR, A., EICHEL, J. A., LI, S., AND JODOIN, P.-M. Non-local deep features for salient object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2017), vol. 2, p. 7.

[115] MA, J., AND TENG, G. A hybrid multiple feature construction approach for classification using genetic programming. *Applied Soft Computing 80* (2019), 687–699.

[116] MAI, L., NIU, Y., AND LIU, F. Saliency aggregation: A data-driven approach. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2013), pp. 1131–1138.

[117] MEI, Y., NGUYEN, S., XUE, B., AND ZHANG, M. An efficient feature selection algorithm for evolving job shop scheduling rules with genetic programming. *IEEE Transactions on Emerging Topics in Computational Intelligence 1*, 5 (2017), 339–353.

[118] MILLER, A. *Subset selection in regression*. CRC Press, 2002.

[119] MING-MING CHENG, GUO-XIN ZHANG, N. J. M. X. H. S.-M. H. Global contrast based salient region detection. pp. 409–416.

[120] MOHEMMED, A. W., ZHANG, M., AND JOHNSTON, M. Particle swarm optimization based adaboost for face detection. In *Proceedings of the IEEE Congress on Evolutionary Computation, 2009* (2009), IEEE, pp. 2494–2501.

[121] MONTANA, D. J. Strongly typed genetic programming. *Evolutionary computation 3*, 2 (1995), 199–230.

[122] MOTODA, H., AND LIU, H. Feature selection, extraction and construction. *Communication of Institute of Information and Computing Machinery 5* (2002), 67–72.

[123] NAQVI, S. *Learning Feature Selection and Combination Strategies for Generic Salient Object Detection*. PhD thesis, Victoria University of Wellington, 2016.

[124] NAQVI, S. S., BROWNE, W. N., AND HOLLITT, C. Optimizing visual attention models for predicting human fixations using genetic algorithms. In *Proceedings of IEEE Congress on Evolutionary Computation* (2013), IEEE, pp. 1302–1309.

[125] NAQVI, S. S., BROWNE, W. N., AND HOLLITT, C. Evolutionary feature combination based seed learning for diffusion-based saliency. In *proceedings of the 10th International Conference on Simulated Evolution and Learning* (2014), pp. 822–834.

[126] NAQVI, S. S., BROWNE, W. N., AND HOLLITT, C. Genetic algorithms based feature combination for salient object detection, for autonomously identified image domain types. In *Proceedings of the IEEE Congress on Evolutionary Computation* (2014), pp. 109–116.

[127] NAQVI, S. S., BROWNE, W. N., AND HOLLITT, C. Genetic algorithms based feature combination for salient object detection, for autonomously identified image domain types. In *Proceedings of the 2014 IEEE Congress on Evolutionary Computation* (2014), IEEE, pp. 109–116.

[128] NESHATIAN, K. *Feature manipulation with genetic programming*. PhD thesis, Victoria University of Wellington, 2010.

[129] NESHATIAN, K., AND ZHANG, M. Genetic programming for performance improvement and dimensionality reduction of classification problems. In *Proceedings of the IEEE Congress on IEEE World Congress on Computational Intelligence* (2008), IEEE, pp. 2811–2818.

[130] NESHATIAN, K., ZHANG, M., AND ANDREAE, P. A filter approach to multiple feature construction for symbolic learning classifiers us-

ing genetic programming. *IEEE Transactions on Evolutionary Computation 16*, 5 (2012), 645–661.

[131] NESHATIAN, K., ZHANG, M., AND JOHNSTON, M. Feature construction and dimension reduction using genetic programming. In *Proceedings of the 20th Australian Joint Conference on Artificial Intelligence, Lecture Notes in Artificial Intelligence* (2007), vol. 4830, Springer, pp. 160–170.

[132] NGUYEN, T. V., NGUYEN, K., AND DO, T.-T. Semantic prior analysis for salient object detection. *IEEE Transactions on Image Processing 28*, 6 (2019), 3130–3141.

[133] NILSSON, N. J. Introduction to machine learning. an early draft of a proposed textbook.

[134] NIXON, M. S., AND AGUADO, A. S. *Feature extraction & image processing for computer vision*. Academic Press, 2012.

[135] OLIVEIRA, L. S., SABOURIN, R., BORTOLOZZI, F., AND SUEN, C. Y. Feature selection using multi-objective genetic algorithms for handwritten digit recognition. In *Proceedings of the 16th International Conference on Pattern Recognition* (2002), vol. 1, IEEE, pp. 568–571.

[136] O'NEILL, D., LENSEN, A., XUE, B., AND ZHANG, M. Particle swarm optimisation for feature selection and weighting in high-dimensional clustering. In *Proceedings of 2018 IEEE Congress on Evolutionary Computation* (2018), IEEE, pp. 1–8.

[137] PENG, Y., WU, Z., AND JIANG, J. A novel feature selection approach for biomedical data classification. *Journal of Biomedical Informatics 43*, 1 (2010), 15–23.

[138] PERAZZI, F., KRÄHENBÜHL, P., PRITCH, Y., AND HORNUNG, A. Saliency filters: Contrast based filtering for salient region detection.

In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2012), IEEE, pp. 733–740.

[139] POLI, R., LANGDON, W. B., MCPHEE, N. F., AND KOZA, J. R. *A field guide to genetic programming*. Lulu. com, 2008.

[140] PORIKLI, F. Integral histogram: A fast way to extract histograms in cartesian spaces. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* (2005), vol. 1, IEEE, pp. 829–836.

[141] PRICE, S. R., ANDERSON, D. T., AND PRICE, S. R. Goofed: Extracting advanced features for image classification via improved genetic programming. In *2019 IEEE Congress on Evolutionary Computation (CEC)* (2019), IEEE, pp. 1596–1603.

[142] QIN, Y., LU, H., XU, Y., AND WANG, H. Saliency detection via cellular automata. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2015), pp. 110–119.

[143] RAMANATHAN, S., KATTI, H., SEBE, N., KANKANHALLI, M., AND CHUA, T.-S. An eye fixation database for saliency detection in images. In *European Conference on Computer Vision* (2010), Springer, pp. 30–43.

[144] REN, Z., GAO, S., CHIA, L.-T., AND TSANG, I. W.-H. Region-based saliency detection and its application in object recognition. *IEEE Transactions on Circuits and Systems for Video Technology 24*, 5 (2014), 769–779.

[145] ROBERGE, V., TARBOUCHI, M., AND LABONTÉ, G. Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning. *IEEE Transactions on Industrial Informatics 9*, 1 (2013), 132–141.

[146] SÁNCHEZ-MAROÑO, N., ALONSO-BETANZOS, A., AND TOMBILLA-SANROMÁN, M. Filter methods for feature selection–a comparative study. In *Proceedings of the International Conference on Intelligent Data Engineering and Automated Learning* (2007), Springer, pp. 178–187.

[147] SETLUR, V., LECHNER, T., NIENHAUS, M., AND GOOCH, B. Retargeting images and video for preserving information saliency. *IEEE Comput. Graph. Appl. 27*, 5 (Sept. 2007), 80–88.

[148] SHAPIRO, L., AND STOCKMAN, G. *Computer Vision*. Prentice Hall, 2001.

[149] SHEN, X., AND WU, Y. A unified approach to salient object detection via low rank matrix recovery. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition* (2012), IEEE, pp. 853–860.

[150] SINGH, N., ARYA, R., AND AGRAWAL, R. A novel approach to combine features for salient object detection using constrained particle swarm optimization. *Pattern Recognition 47*, 4 (2014), 1731–1739.

[151] SINGH, N., ARYA, R., AND AGRAWAL, R. K. A novel approach to combine features for salient object detection using constrained particle swarm optimization. *Pattern Recognition 47*, 4 (2014), 1731–1739.

[152] SMART, W., AND ZHANG, M. Classification strategies for image classification in genetic programming. In *Proceedings of the 18th International Conference on Image and Vision Computing Conference* (2003), Massey University, pp. 402–407.

[153] SMITH, S. M., AND BRADY, J. M. SUSANA new approach to low level image processing. *International Journal of Computer Vision 23*, 1 (1997), 45–78.

[154] SONDHI, P. Feature construction methods: A survey. *sifaka. cs. uiuc. edu 69* (2009), 70–71.

[155] SONG, Z., CHEN, Q., HUANG, Z., HUA, Y., AND YAN, S. Contextualizing object detection and classification. In *CVPR 2011* (2011), IEEE, pp. 1585–1592.

[156] SONKA, M., HLAVAC, V., AND BOYLE, R. *Image processing, analysis, and machine vision*. Cengage Learning, 2014.

[157] SRIVATSA, R. S., AND BABU, R. V. Salient object detection via objectness measure. In *Proceedings of the 2015 IEEE International Conference on Image Processing* (2015), IEEE, pp. 4481–4485.

[158] TACKETT, W. A. Genetic programming for feature discovery and image discrimination. In *Proceedings of the 5th International Conference on Genetic Algorithms* (1993), pp. 303–311.

[159] TALAVERA, L. An evaluation of filter and wrapper methods for feature selection in categorical clustering. In *Proceeings of the International Symposium on Intelligent Data Analysis* (2005), Springer, pp. 440–451.

[160] TATLER, B. W. The central fixation bias in scene viewing: Selecting an optimal viewing position independently of motor biases and image feature distributions. *Journal of Vision 7*, 14 (2007), 4–4.

[161] TOET, A. Computational versus psychophysical bottom-up image saliency: A comparative evaluation study. *IEEE Transactions on Pattern Analysis and Machine Intelligence 33*, 11 (2011), 2131–2146.

[162] TONG, N., LU, H., RUAN, X., AND YANG, M.-H. Salient object detection via bootstrap learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2015), pp. 1884–1892.

[163] TONG, N., LU, H., ZHANG, Y., AND RUAN, X. Salient object detection via global and local cues. *Pattern Recognition 48*, 10 (2015), 3258–3267.

[164] TRAN, B., XUE, B., AND ZHANG, M. Genetic programming for feature construction and selection in classification on high-dimensional data. *Memetic Computing 8*, 1 (2015), 3–15.

[165] TRAN, B., XUE, B., AND ZHANG, M. Genetic programming for feature construction and selection in classification on high-dimensional data. *Memetic Computing 8*, 1 (2016), 3–15.

[166] TRAN, B., XUE, B., AND ZHANG, M. Genetic programming for multiple-feature construction on high-dimensional classification. *Pattern Recognition 93* (2019), 404–417.

[167] TRAN, B., XUE, B., ZHANG, M., AND NGUYEN, S. Investigation on particle swarm optimisation for feature selection on high-dimensional data: Local search and selection bias. *Connection Science* (2016), 1–25.

[168] TRAN, B. N. Evolutionary computation for feature manipulation in classification on high-dimensional data.

[169] TSOTSOS, J. K., CULHANE, S. M., WAI, W. Y. K., LAI, Y., DAVIS, N., AND NUFLO, F. Modeling visual attention via selective tuning. *Artificial Intelligence 78*, 1-2 (1995), 507–545.

[170] TUYTELAARS, T., LAMPERT, C. H., BLASCHKO, M. B., AND BUNTINE, W. Unsupervised object discovery: A comparison. *International Journal of Computer Vision 88*, 2 (2010), 284–302.

[171] TUYTELAARS, T., AND MIKOLAJCZYK, K. *Local invariant feature detectors: A survey*. Now Publishers Inc., 2008.

[172] ULUSOY, I., AND BISHOP, C. M. Comparison of generative and discriminative techniques for object detection and classification. In *Toward Category-Level Object Recognition*. Springer, 2006, pp. 173–195.

[173] VAN DE WEIJER, J., GEVERS, T., AND BAGDANOV, A. D. Boosting color saliency in image feature detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence 28*, 1 (2006), 150–156.

[174] WANG, L., LU, H., RUAN, X., AND YANG, M.-H. Deep networks for saliency detection via local estimation and global search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2015), pp. 3183–3192.

[175] WANG, L., WANG, L., LU, H., ZHANG, P., AND RUAN, X. Saliency detection with recurrent fully convolutional networks. In *European Conference on Computer Vision* (2016), Springer, pp. 825–841.

[176] WANG, T., BORJI, A., ZHANG, L., ZHANG, P., AND LU, H. A stagewise refinement model for detecting salient objects in images. In *Proceedings of the IEEE International Conference on Computer Vision* (2017), pp. 4019–4028.

[177] WANG, W., LAI, Q., FU, H., SHEN, J., AND LING, H. Salient object detection in the deep learning era: An in-depth survey. *arXiv preprint arXiv:1904.09146* (2019).

[178] WANG, W., SHEN, J., DONG, X., AND BORJI, A. Salient object detection driven by fixation prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018), pp. 1711–1720.

[179] WANG, W., SHEN, J., AND SHAO, L. Consistent video saliency using local gradient flow optimization and global refinement. *IEEE Transactions on Image Processing 24*, 11 (2015), 4185–4196.

[180] WEI, Y., WEN, F., ZHU, W., AND SUN, J. Geodesic saliency using background priors. In *Proceedings of the European Conference on Computer Vision*, Springer, pp. 29–42.

[181] WHIGHAM, P. A., AND DICK, G. Implicitly controlling bloat in genetic programming. *IEEE Transactions on Evolutionary Computation 14*, 2 (2009), 173–190.

[182] WYSE, N., DUBES, R., AND JAIN, A. K. A critical evaluation of intrinsic dimensionality algorithms. *Pattern Recognition in Practice* (1980), 415–425.

[183] XUE, B., AND ZHANG, M. Evolutionary computation for feature manipulation: Key challenges and future directions. In *Proceedings of 2016 IEEE Congress on Evolutionary Computation* (2016), IEEE, pp. 3061–3067.

[184] XUE, B., ZHANG, M., BROWNE, W. N., AND YAO, X. A survey on evolutionary computation approaches to feature selection. *IEEE Transactions on Evolutionary Computation 20*, 4 (2015), 606–626.

[185] XUE, B., ZHANG, M., BROWNE, W. N., AND YAO, X. A survey on evolutionary computation approaches to feature selection. *IEEE Transactions on Evolutionary Computation 20*, 4 (2016), 606–626.

[186] YAN, Q., XU, L., SHI, J., AND JIA, J. Hierarchical saliency detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2013), pp. 1155–1162.

[187] YANG, C., ZHANG, L., AND LU, H. Graph-regularized saliency detection with convex-hull-based center prior. *IEEE Signal Processing Letters 20*, 7 (2013), 637–640.

[188] YANG, C., ZHANG, L., LU, H., RUAN, X., AND YANG, M.-H. Saliency detection via graph-based manifold ranking. In *Proceed-*

ings of the IEEE Conference on Computer Vision and Pattern Recognition (2013), IEEE, pp. 3166–3173.

[189] YANG, J., AND YANG, M.-H. Top-down visual saliency via joint CRF and dictionary learning. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition* (2012), IEEE, pp. 2296–2303.

[190] ZHANG, L., DAI, J., LU, H., HE, Y., AND WANG, G. A bi-directional message passing model for salient object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018), pp. 1741–1750.

[191] ZHANG, L., GU, Z., AND LI, H. SDSP: A novel saliency detection method by combining simple priors. In *Proceeding of the 20th IEEE International Conference on Image Processing* (2013), IEEE, pp. 171–175.

[192] ZHANG, M., CIESIELSKI, V. B., AND ANDREAE, P. A domain-independent window approach to multiclass object detection using genetic programming. *EURASIP Journal on Advances in Signal Processing 2003*, 8 (2003), 1–19.

[193] ZHANG, M., AND LETT, M. Genetic programming for object detection: Improving fitness functions and optimising training data. *IEEE Intelligent Informatics Bulletin 7*, 1 (2006), 12–21.

[194] ZHANG, P., LIU, W., LU, H., AND SHEN, C. Salient object detection with lossless feature reflection and weighted structural loss. *IEEE Transactions on Image Processing* (2019).

[195] ZHANG, P., WANG, D., LU, H., WANG, H., AND RUAN, X. Amulet: Aggregating multi-level convolutional features for salient object detection. In *Proceedings of the IEEE International Conference on Computer Vision* (2017), pp. 202–211.

[196] ZHANG, P., WANG, D., LU, H., WANG, H., AND YIN, B. Learning uncertain convolutional features for accurate saliency detection. In *Proceedings of the IEEE International Conference on Computer Vision* (2017), pp. 212–221.

[197] ZHANG, W., WU, Q. J., WANG, G., AND YIN, H. An adaptive computational model for salient object detection. *IEEE Transactions on Multimedia 12*, 4 (2010), 300–316.

[198] ZHANG, X., WANG, T., QI, J., LU, H., AND WANG, G. Progressive attention guided recurrent network for salient object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2018), pp. 714–722.

[199] ZHAO, H., SINHA, A. P., AND GE, W. Effects of feature construction on classification performance: An empirical study in bank failure prediction. *Expert Systems with Applications 36*, 2 (2009), 2633–2644.

[200] ZHAO, R., OUYANG, W., LI, H., AND WANG, X. Saliency detection by multi-context deep learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2015), pp. 1265–1274.

[201] ZHOU, L., YANG, Z., YUAN, Q., ZHOU, Z., AND HU, D. Salient region detection via integrating diffusion-based compactness and local contrast. *IEEE Transactions on Image Processing 24*, 11 (2015), 3308–3320.

[202] ZHU, M.-Q., WANG, Z.-L., AND CHEN, Z.-H. Human visual intelligence and particle filter based robust object tracking algorithm. *Control and Decision 27*, 11 (2012), 1720–1724.

[203] ZHU, W., LIANG, S., WEI, Y., AND SUN, J. Saliency optimization from robust background detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2014), IEEE, pp. 2814–2821.

[204] ZHUGE, Y., ZENG, Y., AND LU, H. Deep embedding features for salient object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2019), vol. 33, pp. 9340–9347.

[205] ZOU, W., KPALMA, K., LIU, Z., AND RONSIN, J. Segmentation driven low-rank matrix recovery for saliency detection. In *Proceedings of the 24th British Machine Vision Conference* (2013), pp. 1–13.